Biplot methodology for analysing and evaluating missing multivariate nominal scaled data

Submitted by

JOHANÉ NIENKEMPER-SWANEPOEL

Dissertation presented for the degree of Doctor of Philosophy in the Department of Statistics and Actuarial Science in the Faculty of Economic and Management Sciences at Stellenbosch University.

Supervisor:	Professor NJ le Roux
Co-supervisor:	Professor S Gardner-Lubbe

December 2019

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Johané Nienkemper-Swanepoel

Date: December 2019

Signature: (Declaration with signature in possession of candidate and supervisors.)

Abstract

This research aims at developing exploratory techniques that are specifically suitable for missing data applications. Categorical data analysis, missing data analysis and biplot visualisation are the three core methodologies that are combined to develop novel techniques. Variants of multiple correspondence analysis (MCA) biplots are used for all visualisations.

The first study objective addresses exploratory analysis after multiple imputation (MI). Multiple plausible values are imputed for each missing observation to construct multiple completed data sets for standard analyses. Biplot visualisations are constructed for each completed data set after MI which require individual exploration to obtain final inference. The number of MIs will greatly affect the accuracy and consistency of the interpretations obtained from several plots. This predicament led to the development of GPAbin, to optimally combine configurations from MIs to obtain a single configuration for final inference. The GPAbin approach advances from two statistical techniques: generalised orthogonal Procrustes analysis (GPA) and the combining rules used to combine estimates obtained from MIs, Rubin's rules.

Albeit a superior missing data handling approach, MI could be daunting for the non-technical practitioner. Therefore, an adequate alternative approach could be appealing and contribute to the variety of available methods for the handling of incomplete multivariate categorical data. The second objective aims at confirming whether visualisations obtained from non-imputed data sets are a suitable alternative to visualisations obtained from MIs. Subset MCA (sMCA) distinguishes between observed and missing subsets of a multivariate categorical data set by creating an additional response category level (CL) for missing responses in the indicator matrix. Missing and observed responses can be visualised separately by only considering the subset of interest in the recoded indicator matrix. The visualisation of the observed responses utilises all available information which would have been forfeited by deletion methods.

The third study objective explores the possibility of predicting a complete multivariate categorical data set from MI visualisations obtained from the first study objective. The distances between the coordinates of a biplot in the full space are used to predict plausible responses. Since the aim of this research is to advance missing data visualisations, the visualisations obtained from predicted completed data sets are compared to visualisations of simulated complete data sets. The emphasis is on preserving inference and not recreating the original data.

Missing data techniques are typically developed to address a specific missing data problem. It is therefore crucial to understand the cause of missingness in order to apply suitable missing data techniques. The fourth study objective investigates the sMCA biplot of the missing subset of the recoded indicator matrix. Configurations of the incomplete subsets enable the recognition of non-response patterns which could provide insight into the particular missing data mechanism (MDM). The missing at random (MAR) MDM refers to missing responses that are dependent on the observed information and is expected to be identified by patterns and groupings occurring in the incomplete sMCA biplot. The missing completely at random (MCAR) MDM states that all observations have the same probability of not being captured which could be identified by a random cloud of points in the incomplete sMCA biplot. Cluster analysis is applied to confirm distinguishable groupings in the incomplete sMCA biplot which could be used as a guideline to identify the MDM.

The proposed methodologies to address the different study objectives are evaluated by means of an extensive simulation study comprising of various sample sizes, variables and varying number of CLs which are simulated from three different distributions. The findings of the simulation study are applied to a real data set to aid as a guide for the analysis.

Functions have been developed for `R` statistical software to perform all methodology presented in this research. It is included as a tool pack provided as an appendix to assist in the correct handling and unbiased visualisation of multivariate categorical data with missing observations.

Keywords: biplots; categorical data; missing data; multiple correspondence analysis; multiple imputation; Procrustes analysis.

Opsomming

Die doel van hierdie navorsing is om verkennende tegnieke te ontwikkel wat spesifiek vir ontbrekende data geskik is. Kategoriese data-analise, ontbrekende data-analise en bi-stipping visualisering is die drie kern metodologieë wat gekombineer word om nuwe tegnieke te ontwikkel. Variante van meervoudige ooreenkomsanalise bi-stippings word gebruik vir alle visualiserings.

Die eerste doelstelling fokus op die verkennende analise van datastelle nadat meervoudige imputasie uitgevoer is. Meervoudige realistiese waardes word vir elke ontbrekende waarde ingevul om sodoende meervoudige voltooide datastelle te konstrueer vir verdere standaard analises. Bi-stipping visualiserings word vir elke voltooide datastel na ‘n meervoudige imputasie gekonstrueer. Aparte verkenning van die individuele visualiserings word vereis om ‘n finale inferensie te verkry. Die aantal meervoudige imputasies sal die akkuraatheid en konsekwentheid van die interpretasies van verskeie stippings beïnvloed. Hierdie probleem het tot die ontwikkeling van die GPAbin metode gelei om die meervoudige visualiserings van meervoudige imputasies optimaal in een figuur vir ‘n finale inferensie te kombineer. Die GPAbin metode vloei uit twee statistiese tegnieke voort: veralgemeende ortogonale Procrustes analise en Rubin se reëls vir die samevoeging van beramings.

Alhoewel meervoudige imputasie bo ander tegnieke vir die hantering van ontbrekende data verkies word, kan meervoudige imputasie uitdagend vir die nie-tegniese gebruiker wees. ‘n Voldoende alternatiewe tegniek kan aanloklik wees en tot die verskeidenheid van beskikbare metodes vir die hantering van ontbrekende data bydra. Die tweede doelstelling poog dan juis om vas te stel of visualiserings van nie-geïmputeerde datastelle ‘n geskikte alternatief vir visualiserings van meervoudige imputasies is. Sub-meervoudige ooreenkomsanalise onderskei tussen waargenome en ontbrekende deelversamelings van ‘n meerveranderlike kategoriese datastel deur ekstra respons kategorievlakke vir ontbrekende waarnemings in die indikatormatriks te skep. Ontbrekende en waargenome response kan apart gevisualiseer word deur spesifieke deelversamelings in die indikatormatriks in ag te neem. Die visualisering van waargenome response benut alle beskikbare inligting, dus word geen inligting verbeur soos in die geval van skrappingsmetodes nie.

Die derde doelstelling ondersoek die moontlikheid om ‘n meerveranderlike kategoriese datastel te voorspel vanaf meervoudige imputasie visualiserings wat in die eerste doelstelling verkry is. Die afstand tussen die koördinate van ‘n bi-stipping in die volle ruimte word gebruik om realistiese responswaardes te voorspel. Aangesien die doel van hierdie navorsing is om visualiserings vir ontbrekende data te bevorder, sal die visualiserings wat van ‘n voorspelde datastel verkry word met die visualiserings van die oorspronklike gesimuleerde datastelle vergelyk word. Die behoud van die oorspronklike inferensie is van belang en nie die herskepping van die volledige oorspronklike data nie.

Tegnieke vir ontbrekende data word vir spesifieke ontbrekende data probleme ontwikkel. Dit is dus noodsaaklik om die oorsaak van die ontbrekenheid te verstaan om sodoende toepaslike ontbrekende data tegnieke toe te pas. Die vierde doelstelling fokus op die ontbrekende deelversameling van die sub-meervoudige ooreenkomsanalise bi-stipping deur die gekodeerde indikatormatriks te gebruik. Visualiserings van die onvolledige deelversamelings maak die herkenning van nie-respons patrone moontlik wat insig rakende die spesifieke ontbrekende data meganisme verskaf. Die ewekansig ontbrekende meganisme verwys na ontbrekende waarnemings wat afhanklik is van die waargenome responswaardes. Dit word verwag dat hierdie meganisme sal lei tot patrone en groeperings in die sub-meervoudige ooreenkomsanalise bi-stipping van die ontbrekende deelversameling. Wanneer alle waarnemings dieselfde waarskynlikheid het om te ontbreek of waargeneem te word, word dié meganisme as die algeheel ewekansig ontbrekende meganismse geklassifiseer. Aangesien ontbrekende waardes onafhanklik van die waargenome waardes is, word dit verwag dat hierdie meganisme geen merkbare patrone sal voortbring in die sub-meervoudige ooreenkomsanalise bi-stipping nie. Trosanalise word toegepas om vas te stel of die visuele groeperings betekenisvol van mekaar geskei kan word in die deelversameling sub-ooreenkomsanalise bi-stipping geskei. Die graad van skeiding in die visualisering kan as ‘n riglyn gebruik word om die ontbrekende data meganisme te identifiseer.

Die voorgestelde metodologieë om die verskillende doelwitte van hierdie studie aan te spreek, word deur middel van ‘n omvangryke simulasie studie geëvalueer. Die simulasie studie bevat datastelle met ‘n verskeidenheid van steekproefgroottes, aantal veranderlikes en wisselende aantal kategorievlakke wat uit drie verskillende verdelings gesimuleer word.

Die bevindings van die simulasie studie word toegepas op ‘n bestaande datastel en dien as ‘n gids vir die analise daarvan.

Funksies vir R statistiese sagteware is ontwikkel om alle metodes in hierdie navorsing te kan uitvoer. Dit word as ‘n gereedskappakket in die bylae gegee om bystand te bied vir die korrekte hantering en onsydige visualisering van meerveranderlike kategoriese data met ontbrekende waardes.

Sleutelwoorde: bi-stippings; kategoriese data; meervoudige imputasie; meervoudige ooreenkomsanalise; ontbrekende data; Procrustes analise.

Acknowledgements

I wish to express my sincere gratitude and appreciation to the following persons and institutions:

- My Creator for blessing me with talents and equipping me for any and all obstacles on my path. Thank you, Holy Spirit, for Your continuous guidance in everything that I pursue. In You and through You I am able.

  Psalm 37:3-4 Afrikaanse Bybel 1983 Vertaling

- My supervisors, Prof. Niël le Roux and Prof. Sugnet Lubbe.

  I will always be indebted to you for the time you have invested in my future. Thank you for sharing your knowledge and for always being available, your support have no bounds. Thank you for your holistic approach to supervision and also taking my well-being into consideration. Thank you for creating opportunities to include me in the exciting and evolving applications of multivariate data visualisations.

  Prof. le Roux, thank you for noticing my potential since our first encounter in 2013 and taking me under your 'wing'.

- My husband, Franré. Thank you for being my partner in every challenge and celebration. Thank you for being invested in my research and always believing that the completion of this dissertation was within reach. Thank you for your immeasurable love and support.

- My support system:

  - Dorothy (my mother), Johan (my father) and Marisan (my sister).

  Thank you for our daily conversations, your endless encouragement, moral support, unconditional love and for **always** being proud of me.

  A special thank you to my mother for imparting her love of research to me. Thank you for taking the time to read through this dissertation.

- My extended family and friends for your ongoing support and prayers.

- Dr. Michael von Maltitz for introducing me to multiple imputation and sparking my interest to join the 'missing data movement'.

- Computations for the simulation study were performed using the University of Stellenbosch's HPC1 (Rhasatsha) and the University of Stellenbosch Central Analytical Facilities' HPC2 (CAF-HPC1): http://www.sun.ac.za/hpc. A special thank you to Gerhard Van Wageningen for his tireless technical assistance.
- Mr. Johan van Rooyen and Mr. Chris Bosman for setting up a virtual machine for computationally intensive tasks which I could access remotely.
- The Teaching Development Grant National Collaborative Project for securing funding for a study leave opportunity of six months in 2018.
- Dr. Rachelle Bester for her assistance to utilise the HPC clusters.
- Dr. Marietjie Vosloo for taking responsibility of my modules and presenting lectures during my study leave.
- Mr. Morney Engelbrecht for alleviating my administrative responsibilities in the final two years of my PhD and for always showing interest in my progress.
- The department of Genetics and the Faculty of AgriScience for granting me study leave opportunities during my PhD and accommodating my research responsibilities.
- To the examiners for their time and positive feedback.

Table of Contents

Declaration ii

Abstract iii

Opsomming v

Acknowledgements viii

Table of Contents x

List of Figures xvi

List of Tables xxii

List of Abbreviations xxiv

Chapter 1 Rationale 1

1.1 Introduction 1

1.2 Data 2

1.3 Visualisations 2

1.4 Aim and study objectives 3

1.4.1 The GPAbin objective 4

1.4.2 The subset multiple correspondence analysis objective 6

1.4.3 The prediction objective 7

1.4.4 The missing data mechanism objective 7

1.5 Layout of the dissertation 9

Chapter 2 Multivariate categorical data: analysis and visualisation 10

2.1 Introduction 10

2.2 Historical overview of categorical data analysis 11

2.3 Indicator matrices 13

2.4 Dimension reduction 13

2.4.1 Square matrices 14

2.4.2 Rectangular matrices 15

2.5 Visualisation 17

2.5.1 Modest visualisation 17

2.5.2 Biplots 19

2.6 Correspondence analysis 21

2.6.1 Historical overview 22

2.6.2 Computations 23

2.6.3 Correspondence analysis visualisations 28

2.7 Multiple Correspondence Analysis 29
2.7.1 Multiple correspondence analysis biplots 30
2.8 Principal component analysis 31
2.9 Orthogonal Procrustes analysis 32
2.9.1 Generalised orthogonal Procrustes analysis 33
2.10 Conclusion 35
Chapter 3 Missing data 36
3.1 Introduction 36
3.2 Missing data mechanisms 38
3.3 Missing data patterns 42
3.4 Handling techniques for missing data 43
3.4.1 Deletion 44
3.4.2 Data reconstruction 46
3.4.2.1 Missing passive modified margin 46
3.4.2.2 Active handling 46
3.4.2.3 Fuzzy coding 47
3.4.3 Weighting 49
3.4.4 Imputation 49
3.4.4.1 Single imputation 50
3.4.4.2 Multiple imputation 51
3.4.4.3 Rubin's rules 54
3.4.4.4 Number of multiple imputations 55
3.5 Compositional data 57
3.6 Conclusion 58
Chapter 4 Methodology 59
4.1 Introduction 59
4.2 Imputation methods 63
4.2.1 Regularised iterative multiple correspondence analysis 63
4.2.1.1 Algorithm: Regularised iterative multiple correspondence analysis 64
4.2.1.2 Code: Regularised iterative multiple correspondence analysis 66
4.2.2 Multiple imputation using multiple correspondence analysis 69
4.2.2.1 Algorithm: Multiple imputation using multiple correspondence analysis 69
4.2.2.2 Code: Multiple imputation using multiple correspondence analysis algorithm 70
4.2.2.3 Multiple correspondence analysis biplots of multiple imputed data sets 71

4.3 GPAbin 73
4.3.1 Optimally aligning multiple imputed configurations 73
4.3.2 Visual inspection 74
4.3.3 Combining the aligned multiple imputed configurations 78
4.4 Prediction methods 80
4.4.1 Majority rule prediction 80
4.4.2 GPAbin prediction 83
4.5 Subset multiple correspondence analysis 83
4.5.1 Code: Subset multiple correspondence analysis 85
4.6 Comparison with simulated data 86
4.6.1 Measures of comparison 89
4.6.2 Further visual comparison 92
4.6.3 Similarity percentage 94
4.7 Clustering 95
4.8 Conclusion 98
Chapter 5 Simulation Protocol 99
5.1 Introduction 99
5.2 Complete data sets: continuous 99
5.2.1 Uniform distribution 99
5.2.2 Skewed distribution 99
5.2.3 Symmetrical distribution 100
5.3 Complete data sets: categorical 100
5.4 Generating missingness 103
5.4.1 True percentages of missing values for missing at random simulations 105
5.4.2 Reproducibility of results 108
5.5 High performance computing 109
5.6 Conclusion 113
Chapter 6 Results: Missing data approaches 114
6.1 Introduction 114
6.2 Similarity percentages 115
6.3 Measures of comparison 125
6.3.1 Measures of bias 130
6.3.2 Measures of fit 136
6.4 Comparison in maximum available dimensions 145

6.5 Conclusion 148
Chapter 7 Results: Prediction of categorical data sets 149
7.1 Introduction 149
7.2 Measures of comparison 150
7.3 Measures of bias 151
7.4 Measures of fit 155
7.5 Conclusion 161
Chapter 8 Results: Identification of the missing data mechanism 162
8.1 Introduction 162
8.2 Presentation of results 164
8.3 Subset multiple correspondence analysis biplots: missing subsets 166
8.4 Visualisations of silhouette coefficients 176
8.4.1 Scatterplots and stacked bar plots (10% missingness) 177
8.4.2 Scatterplots and stacked bar plots (30% missingness) 183
8.4.3 Scatterplots and stacked bar plots (50% missingness) 188
8.4.4 Heat maps of silhouette coefficients 191
8.5 Limited multiple active results 197
8.6 Conclusion 197
Chapter 9 Real data application 199
9.1 Introduction 199
9.2 Variable information 200
9.3 MDM identification 203
9.4 Missing data approach for real application 205
9.4.1 Multiple imputation and generalised orthogonal Procrustes analysis 205
9.4.2 GPAbin and subset multiple correspondence analysis biplots 208
9.4.3 Comparison of GPAbin and subset multiple correspondence analysis biplots 212
9.5 Conclusion 213
Chapter 10 Concluding remarks 214
10.1 Review 214
10.2 Future work 216
10.3 Impact of this research 217
Reference list 218
Appendix 231
Appendix A Simulation Code 231

A.1 Creating complete categorical data sets 231
A.2 Creating a MAR MDM 232
A.3 Creating an MCAR MDM 234
Appendix B OPA function 234
Appendix C Fit measures 235
C.1 Measures of comparison 235
C.2 Similarity percentage 236
Appendix D GPA function 237
Appendix E GPAbin 239
Appendix F Functions related to RIMCA 239
F.1 Single imputation call functions 239
F.2 Estimating the number of dimensions 240
F.3 Adapted RIMCA function 243
Appendix G Functions related to MIMCA 245
G.1 Multiple imputation call functions 245
G.2 Adapted MIMCA function 246
Appendix H Majority rule prediction 248
H.1 Majority rule prediction call functions 248
H.2 Category predictions from multiple imputed configurations 249
H.3 Assigning the final category levels of multiple predicted data sets 251
Appendix I GPAbin prediction 252
I.1 GPAbin prediction call functions 252
I.2 GPAbin prediction function 253
Appendix J sMCA 254
J.1 sMCA call functions 254
J.2 Adapted mjca() function 256
Appendix K Cluster analysis 257
Appendix L Auxiliary functions 257
L.1 Removing empty factor levels 257
L.2 Removing variables with one category level 258
L.3 Creating missing category levels 258
L.4 Formatting of a dataframe to a factor 259
L.5 Detecting zero length integers 260
L.6 Constructing an indicator matrix 260

L.7 Determining the number of elements in the upper triangle of a matrix 261
L.8 Formatting data 261
Appendix M Example of script file submitted to PBS 263
Appendix N Plotting functions 265
N.1 MCA biplot 265
N.2 Stacked barplots 265
Appendix O Measures of comparison: missing data approaches 266
O.1 Mean bias: GPAbin and RIMCA (MAR MDM) 266
O.2 Mean bias: GPAbin and RIMCA (MCAR MDM) 267
Appendix P Measures of comparison: prediction of categorical data sets 269
P.1 Heat map of MB values for prediction methods 269
P.2 Selection of density plots of prediction methods compared to GPAbin 270
Appendix Q sMCA biplots: missing subsets 278

List of Figures

Figure 2.1 Bigraph of five samples (circles) and two categorical variables (squares: two CLs and triangles: three CLs). 18

Figure 2.2 Attraction graph of professions (circles) and qualities (triangles). 18

Figure 3.1 Missing data patterns. Left panel: Univariate pattern. Middle panel: Monotone pattern. Right panel: Arbitrary pattern. 42

Figure 3.2 Illustration of the steps of MI 51

Figure 4.1 Schematic representation of the GPAbin objective (cf. 1.4.1). 60

Figure 4.2 Schematic representation of the sMCA objective (cf. 1.4.2). 61

Figure 4.3 Schematic representation of the prediction objective (cf. 1.4.3). 62

Figure 4.4 Schematic representation of the MDM objective (cf. 1.4.4). 63

Figure 4.5 Illustration of MCA biplots constructed from ten MIs using MIMCA. Each panel represents an MCA biplot for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 72

Figure 4.6 Illustration of CLPs before and after GPA for ten MIs. Each panel represents the CLPs for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 75

Figure 4.7 Illustration of transformed GPA MCA biplots for ten MIs. Each panel represents the GPA MCA biplot for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 77

Figure 4.8 Superimposed CLPs from Figure 4.7 for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 78

Figure 4.9 The GPAbin biplot for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 79

Figure 4.10 Illustration of predictions from MCA biplots after MIMCA (cf. Figure 4.5) for the third sample of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 82

Figure 4.11 Illustration of predictions from GPAbin biplot (cf. Figure 4.9) for the third sample of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 83

Figure 4.12 The sMCA biplot (observed subset) for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution). 86

Figure 4.13 Comparison of visualisation approaches for missing data compared to the complete MCA biplot. Left panel: complete MCA biplot (five variables and 1000 samples simulated from a uniform distribution). Middle panel: GPAbin biplot (cf. Figure 4.9). Right panel: sMCA biplot (complete subset) (cf. Figure 4.12). Middle- and right panels: 10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution. 87

Figure 4.14 The CLP plot for the target configuration (CLPs from complete MCA) and testee (CLPs from GPAbin biplot) configuration before OPA. 88

Figure 4.15 The OPA steps. Left panel: Reflection and rotation of testee CLPs (GPAbin CLPs). Middle panel: Scaling of updated CLPs (reflected and rotated). Right panel: Target CLPs compared to updated testee CLPs.......... 89

Figure 4.16 Manually constructed convex hulls to mimic groupings in complete biplots. Left panel: complete MCA biplot (five variables and 100 samples simulated from a uniform distribution). Middle panel: GPAbin biplot. Right panel: sMCA biplot (complete subset). Middle- and right panels: 50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution.......... 93

Figure 4.17 The sMCA biplot (missing subset) of the additional example data set (50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution). 96

Figure 4.18 The pam clustering of CLPs in an sMCA biplot (missing subset) of the additional example data set (50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution). Left panel: $k = 2 \rightarrow si = 0.6800$. Right panel: $k = 3 \rightarrow si = 0.1848$. 98

Figure 5.1 MCA biplots for Dirichlet simulations Left panel: $n = 100, p = 5$. Middle panel: $n = 100, p = 10$. Right panel: $n = 3000, p = 15$. Sample points are depicted by green open circles and CLPs by black triangles. 101

Figure 5.2 MCA biplots for uniform simulations. Left panel: $n = 100, p = 5$. Middle panel: $n = 100, p = 10$. Right panel: $n = 3000, p = 15$. Sample points are depicted by green open circles and CLPs by black triangles. 102

Figure 5.3 MCA biplots for normal simulations. Left panel: $n = 100, p = 5$. Middle panel: $n = 100, p = 10$. Right panel: $n = 3000, p = 15$. Sample points are depicted by green open circles and CLPs by black triangles. 102

Figure 5.4 MCA biplot with CLP labels of Figure 5.3 (left panel).......... 103

Figure 5.5 True percentage of missing values for Dirichlet simulations. Left panels: MAR 10% (mean missing values: 9.97%, minimum missing value percentage: 9.2%). Middle panels: MAR 30% (mean missing values: 29.4%, minimum missing value percentage: 20.1%). Right panels: MAR 50% (mean missing values: 38.8%, minimum missing value percentage: 20.1%). The simulations are sorted in the upper panels.......... 106

Figure 5.6 True percentage of missing values for uniform simulations. Left panels: MAR 10% (mean missing values: 9.96%, minimum missing value percentage: 9.3%). Middle panels: MAR 30% (mean missing values: 29.8%, minimum missing value percentage: 21.1%). Right panels: MAR 50% (mean missing values: 42.1%, minimum missing value percentage: 21.1%). The simulations are sorted in the upper panels.......... 107

Figure 5.7 True percentage of missing values for normal simulations. Left panels: MAR 10% (mean missing values: 9.96%, minimum missing value percentage: 9.2%). Middle panels: MAR 30% (mean missing values: 29.8%, minimum missing value percentage: 15.3%). Right panels: MAR 50% (mean missing values: 46.0%, minimum missing value percentage: 15.3%). The simulations are sorted in the upper panels. 108

Figure 6.1 Similarity percentages of all simulation scenarios for GPAbin, sMCA and RIMCA. 116

Figure 6.2 Density ridge plots to illustrate skewness of measures of comparison. A selection of simulation scenarios are illustrated. 127

Figure 6.3 AMB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median AMB values. Right panel: Mean AMB values. 130

Figure 6.4 sMCA approach: 50% missing values with a MAR MDM for the three simulation distributions. 131

Figure 6.5 MB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median MB values. Right panel: Mean MB values. 132

Figure 6.6 Scatterplots of the MB values for the sMCA method (MAR MDM). Left panels: 10% missing values. Middle vertical panels: 30% missing values. Right panels: 50% missing values. Upper panels: Dirichlet distribution. Middle horizontal panels: uniform distribution. Lower panel: normal distribution. 133

Figure 6.7 Scatterplots of the MB values for the sMCA method (MCAR MDM). Left panels: 10% missing values. Middle vertical panels: 30% missing values. Right panels: 50% missing values. Upper panels: Dirichlet distribution. Middle horizontal panels: uniform distribution. Lower panel: normal distribution. 134

Figure 6.8 RMSB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median RMSB values. Right panel: Mean RMSB values. 135

Figure 6.9 PS values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median PS values. Right panel: Mean PS values. 137

Figure 6.10 Scatterplots of PS values for the GPAbin approach across all simulation scenarios. 140

Figure 6.11 Scatterplots of PS values for the RIMCA approach across all simulation scenarios. 141

Figure 6.12 Scatterplots of PS values for the sMCA approach across all simulation scenarios. 142

Figure 6.13 CC values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median CC values. Right panel: Mean CC values. 143

Figure 6.14 Scatterplots of normal distribution simulations: 50% missing values and MAR MDMs. Left panel: GPAbin. Middle panel: sMCA. Right panel: RIMCA. 144

Figure 6.15 Measures of comparison values per simulation scenario over 1000 repetitions (comparison in maximum dimensions). Left panel: Mean RMSB values. Right panel: Mean AMB values. 146

Figure 6.16 Mean PS values per simulation scenario over 1000 repetitions (comparison in maximum dimensions). 147

Figure 7.1 Measures of comparison values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Mean AMB values. Right panel: Mean RMSB values. 152

Figure 7.2 Kernel density estimates of AMB values for prediction methods (MCN 30% missing values). 153

Figure 7.3 Kernel density estimates of AMB values for prediction methods (MC 50% missing values). 154

Figure 7.4 Measures of fit values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Mean PS values. Right panel: Mean CC values. 156

Figure 7.5 PS values: Dirichlet distribution (GPAbin prediction, Majority rule prediction and GPAbin) 158

Figure 7.6 PS values: Uniform distribution (GPAbin prediction, Majority rule prediction and GPAbin) 159

Figure 7.7 PS values: Normal distribution (GPAbin prediction, Majority rule prediction and GPAbin) 160

Figure 8.1 sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 5$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 167

Figure 8.2 sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 168

Figure 8.3 sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 169

Figure 8.4 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 5$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 170

Figure 8.5 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 171

Figure 8.6 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 172

Figure 8.7 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 5$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 173

Figure 8.8 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 174

Figure 8.9 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4). 175

Figure 8.10 Silhouette coefficients of the Dirichlet distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 177

Figure 8.11 Silhouette coefficients of the uniform distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 180

Figure 8.12 Silhouette coefficients of the normal distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 182

Figure 8.13 Silhouette coefficients of the Dirichlet distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 184

Figure 8.14 Silhouette coefficients of the uniform distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 185

Figure 8.15 Silhouette coefficients of the normal distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 186

Figure 8.16 Silhouette coefficients of the Dirichlet distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 188

Figure 8.17 Silhouette coefficients of the uniform distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 189

Figure 8.18 Silhouette coefficients of the normal distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots. 190

Figure 8.19 Heat map of the percentage of silhouette coefficients above 0.5 for 30% missingness. 192

Figure 8.20 Heat map of the percentage of silhouette coefficients above 0.5. 193

Figure 8.21 Heat map of the percentage of silhouette coefficients above 0.6................................ 195

Figure 9.1 Bar graphs of each variable in the adapted survey: 1 – agree, 2 – neutral, 3 – disagree, NA – missing 202

Figure 9.2 sMCA biplot: missing subset of real data application. Left panel: standard sMCA biplot display. Right panel: sMCA biplot with samples colour coded according to educational level and represented as convex hulls. 203

Figure 9.3 Illustrating allocated clusters for the sMCA biplot (missing subset) of the real data application. Left panel: $k = 2 \rightarrow si = 0.7832$. Middle panel: $k = 3 \rightarrow si = 0.5016$. Right panel: $k = 4 \rightarrow si = 0.5805$. 204

Figure 9.4 Illustration of CLPs before and after GPA for ten MIs. Each panel represents the CLPs for a particular MI for the real data application. 206

Figure 9.5 Superimposed transformed CLPs from Figure 9.4 for the real data application. 207

Figure 9.6 Left panel: Convex hulls of superimposed CLPs after GPA. Right panel: Colour coded convex hulls with increasing colour intensity according to the percentage of missing values.. 208

Figure 9.7 Left panel: GPAbin biplot. Right panel: CLP descriptions of GPAbin biplot. 208

Figure 9.8 Left panel: sMCA biplot. Right panel: CLP descriptions of sMCA biplot........................ 209

Figure 9.9 Colour coded CLPs according to the response level. Left panel: GPAbin biplot. Right panel: sMCA biplot. 210

Figure 9.10 GPAbin biplots with convex hulls for sample separation based on demographic information. Top panels: Gender (left), Region (middle) and Marital status (right). Lower panels: Education (left) and Age (right). 211

Figure 9.11 sMCA biplots with convex hulls for sample separation based on demographic information. Top panels: Gender (left), Region (middle) and Marital status (right). Lower panels: Education (left) and Age (right) 212

List of Tables

Table 2.1 Toy data set 13

Table 3.1 Toy data set with missing values 44

Table 3.2 Adapted toy data set with missing values 48

Table 4.1 Measures of comparison for the GPAbin CLPs of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution) compared to the CLPs of the complete MCA biplot (five variables and 1000 samples simulated from a uniform distribution). 92

Table 4.2 Similarity percentages obtained from GPAbin and sMCA when compared to the complete MCA biplot in two dimensions, respectively. Example data set: 10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution 95

Table 5.1 Allocating CLs from continuous values 101

Table 5.2 The number of cores used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM. 111

Table 5.3 The RAM (in GB) used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM. 112

Table 5.4 The walltime (in minutes) used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM. 112

Table 6.1 Abbreviations for simulation distributions and MDMs 114

Table 6.2 Coefficients of variation of similarity percentages (GPAbin, sMCA and RIMCA) for all simulation scenarios. Values above 3% are indicated in **bold** font. 118

Table 6.3 Similarity percentages of simulation scenarios with 30% missing values with a coefficient of variation greater than 3% between approaches (**bold** values of Table 6.2). 119

Table 6.4 Similarity percentages of simulation scenarios with 50% missing values with a coefficient of variation greater than 3% between approaches (**bold** values of Table 6.2). 120

Table 6.5 Similarity percentages below 75% for simulation scenarios with 10% missing values. Investigating the percentages of Figure 6.1. 121

Table 6.6 Similarity percentages below 75% for simulation scenarios with 30% missing values. Investigating the percentages of Figure 6.1. 122

Table 6.7 Similarity percentages below 75% for simulation scenarios with 50% missing values. Investigating the percentages of Figure 6.1. 123

Table 6.8 Ranges for measures of comparison: missing data approaches 128

Table 6.9 Top 10% ranges for measures of comparison: missing data approaches 129

Table 7.1 Ranges for measures of comparison: prediction methods. 150

Table 7.2 Top 10% ranges for measures of comparison: prediction methods 150

Table 9.1 Variable information for survey questions of ISSP 1994. 200

Table 9.2 Variable information for demographic questions of ISSP 1994. 201

Table 9.3 Measures of comparison: GPAbin and sMCA real application 213

List of Abbreviations

Acronyms

CART	Classification and regression trees
MAR	Missing at random
OPA	Orthogonal Procrustes analysis
MIMCA	Multiple imputation using multiple correspondence analysis
pam	partitioning around medoids
RAM	Random-access memory
RIMCA	Regularised iterative multiple correspondence analysis

Initialisms

AMB	Absolute mean bias
CA	Correspondence analysis
CC	Congruence coefficient
CL	Category level
CLs	Category levels
CLP	Category level point
CLPs	Category level points
EM	Expectation-Maximisation
FCS	Fully conditional specification
GB	Gigabyte
GPA	Generalised orthogonal Procrustes analysis
GSVD	Generalised singular value decomposition
HPC	High performance computing
JM	Joint modelling

MA	Data simulated from a uniform distribution with a missing at random missing data mechanism
MAD	Data simulated from a Dirichlet distribution with a missing at random missing data mechanism
MAN	Data simulated from a normal distribution with a missing at random missing data mechanism
MB	Mean bias
MC	Data simulated from a uniform distribution with a missing completely at random missing data mechanism
MCA	Multiple Correspondence analysis
MCD	Data simulated from a Dirichlet distribution with a missing completely at random missing data mechanism
MCN	Data simulated from a normal distribution with a missing completely at random missing data mechanism
MDM	Missing data mechanism
MDS	Multidimensional scaling
MI	Multiple imputation
MIs	Multiple imputations
MSEP	Mean square error of prediction
PS	Procrustes Statistic
RMSB	Root mean squared bias
sCA	Subset correspondence analysis
SI	Single imputation
sMCA	Subset multiple correspondence analysis
SVD	Singular value decomposition

Combination of acronyms and intialisms

GPAbin	Generalised Procrustes analysis and Rubin’s rules
	‘GPA’ – initialism and ‘bin’ – acronym
MCAR	Missing completely at random
	‘M’ – initialism and ‘CAR’ – acronym
MNAR	Missing not at random
	‘M’ – initialism and ‘NAR’ – acronym

Chapter 1
Rationale

1.1 Introduction

Missing data have become a prevalent problem which is in a sense inseparable from data collection. Over the past decades, proper missing data handling techniques have been developed and applied for the imputation of missing data sets (Van Buuren, 2012). Various imputation techniques are available to substitute missing observations with plausible response values. Single imputation (SI) is the substitution of a single plausible response for each missing observation, which is easy to apply but generally results in biased estimation, as the standard errors are underestimated (Rubin, 1987). Multiple imputation (MI) is the preferred method that generates several plausible values for each missing value until a predetermined number of data sets are imputed. This results in unbiased representation of the missing observations, as it provides a distribution of plausible responses to capture the uncertainty involved in the task of imputation. Standard analyses for complete data are then applied to the multiple data sets and the estimates of interest can be combined to formulate a final inferential statement (Rubin, 1987; Schafer, 1997; Van Buuren, 2012). The study of MI might be considered somewhat controversial, because the imputations are not applied with the aim of obtaining a final completed data set, but rather to preserve the population variation and, most importantly, the relationships between variables (Wayman, 2003).

The importance of both SI and MI is acknowledged and the development of MI should not be regarded as a replacement for SI, but rather an improvement to handle missing data in more complex scenarios. In some cases, however, the aim might be to only obtain one completed data set to be used by field specialists, and therefore SI is still relevant. Nevertheless, from an analyst's perspective, there is much to be gained from the unbiased estimation of the variation when using MI.

The study of data imputation remains a growing topic of interest, but visualisations for missing data and completed data have not yet been placed on the foreground of statistical analysis (Eaton, Plaisant & Drizd, 2005; Fernstad, 2019; Templ, Alfons & Filzmoser, 2012). Exploratory analysis of multiple plausible responses becomes a cumbersome task that in itself could lead to biased inference. Therefore, this study focusses on the development of appropriate exploratory techniques for multivariate categorical missing data and multiple imputed data sets. In a recent review on missing data protocols the remark was made that "new research on diagnostics and visualisation may inform analyses with missing values" (Josse & Reiter, 2018: 141). This is motivation that the developments in this research will address current missing data issues and contribute to the science of handling missing data.

In this study, three core methodologies will be extended and combined, namely multivariate categorical data analysis, missing data analysis and biplot visualisation.

1.2 Data

Only nominal scaled multivariate categorical data are considered in this study. Consider a set of individuals where measurements are made for each individual (referred to as a sample) on a set of categorical questions (referred to as variables). A nominal scaled measurement on a categorical variable can only be one of a finite unordered number of qualities, for example hair colour: red, blonde or brunette. There is no specific order of importance of the qualitative response options (Agresti, 2007). These qualities are referred to as the category levels (CLs) of a categorical variable. Typically, the categorical variables are represented as the columns of a matrix and the samples as its rows. The proposed methodology will be applied to a variety of simulated data scenarios (cf. Chapter 5) consisting of combinations from different distributions, dimensions, percentages of missing values and missing data mechanisms (MDMs). The results obtained from a real application are presented in Chapter 9.

1.3 Visualisations

Multiple correspondence analysis (MCA) is a multivariate categorical technique that enables the simultaneous exploration of samples and their different qualitative responses measured for all the categorical variables. The focus of MCA is to obtain an understanding of how the samples are associated based on their responses to variables. Samples are regarded as similar if they have a majority of the same responses to variables (i.e. CLs). The main interest is the interpretation of the responses, as these capture information of both the samples and the variables (Blasius & Thiessen, 2012; Greenacre, 2010; Husson, Lê & Pagès, 2011; Josse &

Husson, 2012). Even though quantities of association can be calculated, it is only truly understood when it is visualised (Beh & Lombardo, 2014). Biplots are constructed from MCA solutions for visual inspections of the categorical data. Biplots are regarded as multivariate scatterplots approximated in lower dimension, in which multiple variables and samples are represented in a single configuration. The biplot display enables the immediate grasp of response patterns and associations between samples based on the distances between coordinates in the display space. Points in close proximity are regarded as being highly associated and reflect individuals with similar response profiles. The biplot consists of sample coordinates, one for each sample, and category level points (CLPs), one for each CL of a particular variable. Each sample is positioned such that the response CL will be in close proximity in the approximated two-dimensional space. Biplots are typically displayed in two dimensions. However, the prefix, 'bi', does not refer to the dimension of the display space, but refers to the simultaneous representation of both samples and CLPs (Gower & Hand, 1996; Husson *et al.*, 2011). The term 'configurations' will refer to MCA biplots throughout the dissertation.

Computational power and sufficient software have greatly influenced the use and popularity of exploratory analysis (Unwin, Chen & Härdle, 2008). Because multiple visualisations can be produced within seconds, the adequate interpretation of the resulting large number of visualisations can become overwhelming. This is an additional branch of data science and big data.

There is a need for methodology to provide unbiased combined visual representations of different variations of the same data. This will allow the evaluation of the subtle differences between multiple visualisations that could be lost when examining a large number of individual graphical displays.

1.4 Aim and study objectives

The aim of this research is to develop unbiased visualisations for multivariate categorical missing data. This will be achieved by means of the following four study objectives:

- Obtaining an unbiased single visualisation after MI
- Determining the applicability of visualisations without prior imputation
- Obtaining a single completed categorical data set using predictions from visualisations
- Identifying the MDM using visualisation.

The four study objectives can be achieved by formulating novel methodologies by unifying the concepts of categorical data analysis, missing data analysis and data visualisation.

Simulated data play a vital role in the evaluation of missing data techniques. Simulation enables the comparison of inferences obtained from complete data and completed data of the same initial simulated data set. Therefore, an extensive simulation study is called for to evaluate the methodology of the four study objectives when applied to various missing data scenarios.

1.4.1 The GPAbin objective

The first objective is to optimally combine the configurations obtained from MIs of multivariate categorical data with missing data entries into a single biplot display.

A single visualisation for MIs can enhance the exploratory analysis of incomplete data. The visualisation of combined MIs has not yet been explored to aid in the understanding of the variation between the imputations. Visualisations of MIs have been projected as supplementary points onto a reference framework for continuous data using principal component analysis (PCA) by Josse and Husson (2012). Procrustes analysis was used to align the imputation visualisations to a reference set individually in order to evaluate the variation between the imputations. Apart from using confidence ellipses to illustrate the uncertainty between MIs, no concrete measure was used to evaluate the success of the projection methods.

Since MCA is a suitable technique to evaluate the relationships between variables and consistencies among samples for multivariate categorical data, MCA biplots are a fitting choice for the visualisation (Blasius & Thiessen, 2012; Greenacre, 2010; Josse & Husson, 2012). An MCA biplot is constructed for each multiple imputed (or completed) data set. This results in the separate inspection of multiple configurations to infer information. In order to address the methodology to achieve this study objective, a few known techniques have to be considered. Two configurations with the same dimensions (number of samples and variables) can be compared using orthogonal Procrustes analysis (OPA). One configuration is set as a target to which the testee configuration is transformed using admissible transformations (Borg & Groenen, 2005; Gower & Dijksterhuis, 2004; Ten Berge, 1977). As we are interested in the visualisation of MIs, a technique that enables the comparison of more than two configurations is required. Generalised orthogonal Procrustes analysis (GPA) allows the comparison of multiple configurations to determine the associations or dissimilarities among multiple configurations when compared to a chosen target configuration (Gower & Dijksterhuis, 2004). After the application of GPA, the multiple configurations are optimally aligned, which improves the visual interpretation thereof. The visualisation of multiple plausible final data sets allows the interpretation of MIs from a new perspective, unlocking additional information only available utilising visual aids. The differences between each imputation can be explored to determine the robustness of the chosen imputation technique. As MI successfully incorporates variation and uncertainty, it is to be expected that there will be differences between the MI MCA solutions. However, if substantial differences are observed between MI visualisations, the validity of the imputation technique should be investigated for the particular data and missing data problem.

Even though GPA eases visual inspection by aligning the configurations, inspecting a large number of separate visualisations (one figure for each imputation) could become tedious and as the number of MIs increases, it becomes impossible to draw accurate inferences from the separate configurations. A single combined display could allow the instantaneous interpretations of associations between variables and samples, which would not have been possible with separate configurations for each imputation. Therefore, a centroid configuration containing the mean coordinates of the optimally aligned configurations is proposed to represent the visualisation of the MIs. The centroid configuration resonates with the application of Rubin’s rules (Rubin, 1987) to combine estimates obtained from MIs.

Now, the aim is not to obtain a final inferential statement, but rather a final descriptive visualisation for the exploratory analysis as opposed to a confirmatory one. The combined visualisation technique is defined by the term, ‘GPAbin’, which pays tribute to the amalgamation of **G**eneralised orthogonal **P**rocrustes **A**nalysis and Ru**bin**’s rules.

The application of GPA has shown to be successful in obtaining a final loading matrix from PCA loadings of MIs (Van Ginkel & Kroonenberg, 2014) and has also been proposed to visually evaluate the between-imputation variation of MIs in PCA (Josse & Husson, 2012). However, GPA has not yet been explored to aid as a combination technique for MI visualisations in incomplete categorical data analysis, which confirms the novelty of this development. All final configurations are compared to the MCA biplot of the simulated complete data using measures of comparison within the Procrustes framework. An SI technique is also applied in order to determine the success of constructing a GPAbin configuration in comparison to a single configuration from single imputed data.

1.4.2 The subset multiple correspondence analysis objective

The second objective is to determine whether non-imputed data can be successfully visualised to preserve the associations between samples and their responses.

The visualisation of incomplete multivariate data without prior imputation is an intriguing possibility for the non-technical practitioner. Subset correspondence analysis (sCA) has been used to explore incomplete categorical data consisting of two-way contingency tables (Greenacre, 2017; Hendry, North, Zewotir & Naidoo, 2014). The complete correspondence analysis (CA) can be restricted using a chosen subset of a data matrix while maintaining the original column and row masses for the calculation of the distances. Therefore, the total variation (inertia) is partitioned into components associated with the various subsets and no interpretable information is lost. This idea is extended to MCA, referred to as subset MCA (sMCA). The data matrix of a multivariate categorical data set is commonly coded as an indicator matrix of zeros and ones. The columns of the indicator matrix represent the CLs and the rows correspond to the samples. A particular response will be represented by a one in the column corresponding to the chosen CL and zero elsewhere. In the case of multi-way contingency tables, sMCA can be applied for the visualisation of the missing observations by recoding the indicator matrix. New CLs are created for the missing responses, which is an active handling approach to missing data (cf. 3.4.2.2). The missing CLs can then be separated from the observed CLs, which allows a focused analysis on either missing or observed subsets. In order to establish whether the non-imputation technique is a competitive alternative to MI techniques, the GPAbin configurations should be compared to the sMCA configurations.

1.4.3 The prediction objective

The third study objective advances naturally from the MI- and GPAbin visualisations. Even though MI techniques focus on overall inference obtained from multiple plausible completed data sets, it is an intriguing idea to determine whether a single data set can be successfully predicted. The distances between coordinates in the visualisations of MIs and the GPAbin procedure (cf. 1.4.1) can be used to predict possible responses. Two approaches to predict a final categorical data set are proposed. The distances in the full space (all available dimensions) between the sample coordinates and CLPs are used to identify the CLP in closest proximity to a sample coordinate for each variable. The use of distances to predict a response is similar to nearest-neighbour imputation (Biemer & Lyberg, 2003; Ohmann, Gregory, Henderson & Roberts, 2011).

As the focus of this research is visualisation, MCA biplots are also constructed for the predicted data sets. The success of the prediction methods is determined by comparing the visualisations of the predicted data sets to the visualisations of the simulated complete data sets.

1.4.4 The missing data mechanism objective

The fourth study objective is to identify the MDM using the missing CLs obtained from the sMCA procedure discussed in the second study objective (cf. 1.4.2). It is crucial to understand the cause of missingness before selecting a missing data handling technique (Buhi, Goodson & Neilands, 2008; Kowarik & Templ, 2016). Exploring visualisations of missing values expose structures and patterns that are not perceptible in a data table (Templ *et al.*, 2012). The occurrence of missing values can be explained as the result of a random process referred to as the MDM (Van Buuren, 2012). Three mechanisms are defined: missing at random (MAR), missing completely at random (MCAR) and missing not at random (MNAR). Non-responses in categorical data sets commonly occur in questionnaires, which could be due to the deliberate omission of sensitive questions that in some cases are related to answered questions in the questionnaire. This is an example of observations that are classified as being MAR, as the missing values are dependent on the observed values in the data set. Respondents may also decide to omit questions due to loss of interest, which is an example of MCAR observations, as missing observations are independent of observed and missing observations (García-Laencina, Sancho-Gómez & Figueiras-Vidal, 2010). Missing observations that are unobserved due to the MNAR mechanism are dependent on observations that are not captured by the questionnaire, therefore dependent on other missing values. This occurs when questions in the questionnaire could be related to information that is not captured or considered by the particular questionnaire (Schafer & Graham, 2002). There are three assumptions that have to be satisfied for a proper MI procedure (cf. 3.4.4.2). Only the first assumption is considered for this study objective, which is the uncertainty of identifying the MDM before selecting an imputation procedure (Rubin, 2003). Apart from the three MDM descriptions, there are two main classifications: ignorable and non-ignorable non-response. The MNAR MDM is categorised as non-ignorable (informative), as the missing data entries are dependent on information that is not available. This means that standard statistical analyses will not capture the uncertainty of these particular missing values (Buhi *et al.*, 2008). However, the MAR MDM and MCAR MDM are in some way related to the missing or observed observations and are regarded as ignorable (non-informative) missingness, which allows the application of standard missing data techniques (Buhi *et al.*, 2008; Schafer & Olsen, 1998).

Only the ignorable MDMs will be explored in this objective. The missing CLPs from the sMCA solution are configured in a biplot. The dependency of the missing data entries on the observed entries in the MAR MDM is expected to result in distinguishable patterns and clusters / groupings. It is expected that no particular pattern will be identified in the sMCA biplot of the missing CLPs of a MCAR MDM, as all observations have the same probability of being missing.

The partitioning around medoids (pam) clustering technique (Maechler, Rousseeuw, Struyf, Hubert & Hornik, 2017) will be used to determine the number of clusters that can be successfully separated in the incomplete sMCA biplots.

1.5 Layout of the dissertation

Two review chapters are presented on the core statistical areas as identified above:

- Chapter 2: Multivariate categorical data: Analysis and visualisation. This chapter provides an overview of categorical data techniques and visualisations. Multivariate techniques for categorical data are presented by focusing on dimension reduction techniques and their biplot visualisations.
- Chapter 3: Missing data. This chapter defines missing data and discusses the causes of missingness and handling techniques in more detail.

The complete methodology to address the study objectives is presented in Chapter 4. The simulation protocol is presented in Chapter 5. In order to establish the difference between imputation and non-imputation visualisations, the results of GPAbin and sMCA will be compared to the simulated complete configurations in Chapter 6. The results obtained from predicted multivariate categorical data sets from visualisations are presented in Chapter 7 by again comparing the results to the simulated complete configurations. The results for the detection of the MDMs from sMCA biplots of the missing subsets of the data are presented and discussed in Chapter 8.

Finally, all proposed techniques are applied on a real data set, which is presented and discussed in Chapter 9. All concluding remarks are presented in Chapter 10.

Functions have been programmed for the newly developed methodology using `R` statistical software (R Core Team, 2017). These functions are presented as a tool pack provided in the Appendix.

Chapter 2
Multivariate categorical data: analysis and visualisation

2.1 Introduction

This chapter will provide a literature review of selected concepts and techniques relevant to achieve the study objectives (cf. 1.4) of this research. First, a general discussion on categorical data analysis will be presented, followed by sections on dimension reduction techniques and biplots. Thereafter, specific multivariate analyses will be presented in conjunction with their visualisation techniques. A discussion on Procrustes analysis will conclude this chapter.

Categorical data consists of measurements that are regarded as counts or classifications. Categorical variables can be defined by the possibility of categorising sample responses to mutually exclusive classes (Bishop, Fienberg & Holland, 1975), also referred to in this research as CLs. Since the recorded responses for categorical variables will never overlap between CLs, all responses are discrete. Discrete variables are however not exclusively used for categorical data, and can be further classified into qualitative and quantitative measurements. Numerical responses such as counts are considered to be both discrete and quantitative (Agresti, 2013). The data considered for all applications in this manuscript are categorical and qualitative in which the numerical discrete values of CLs will be of no computational interest.

Binary variables only measure two possible outcomes, for example a ‘yes’ or ‘no’ response, whereas polytomous variables consist of multiple possible CLs (Tang, He & Tu, 2012). Polytomous response variables that are dependent on a natural order of the possible CLs are measured on an ordinal scale. An example of an ordinal variable is the hierarchy of academic degrees ordered from the lowest to the highest level of education: bachelors-, honours-, masters- and doctorate degree. A variable consisting of CLs that do not differ with respect to a fixed order, is classified as a qualitative variable measured on the nominal scale. This could refer to faculties at a specific university: Natural Sciences, Engineering, Agricultural Sciences, etc. (Agresti, 1990). It is crucial to understand the nature of recorded measurements in order to apply suitable statistical methods. The techniques proposed and illustrated in this research, focus on the application of statistical techniques for nominal scaled data, which could be extended for ordinal scaled data and continuous scaled data in the future.

Multivariate analysis is the study of multiple characteristics of a single subject that are measured simultaneously. Cross-classification refers to the measurement of more than two categorical characteristics on multiple subjects. Further on, these characteristics will be referred to as variables consisting of a finite set of CLs and the subjects will be referred to as the samples. Samples with the same responses to variables are regarded as similar and samples with varying responses to variables are considered to be dissimilar (Van de Geer, 1993). Exploratory analysis of multivariate categorical data sets is focused on the possible associations between samples regarding responses to variables.

Visualisations are typically displayed in lower dimension (one, two or three); the visualisations in this research will be displayed in two dimensions. Therefore, all multivariate data sets have to be approximated in lower dimension with the aim of capturing as much of the initial variation as possible. This will be further addressed in the dimension reduction section (cf. 2.4) of this chapter.

According to Unwin *et al.* (2008) the literature of the science and art of data visualisation is not as well documented as other computational focus areas, since emphasis is placed upon the applications. This is in contradiction to computational publications, which are focused on the theoretical development with little or no real application. A metaphor of a garden is depicted by Friendly (2008) for the development of data visualisation. Friendly (2008) relates the current data visualisation techniques as the fruit that are visible in the garden and the unknown history and theoretical underpinnings of the trees from which the fruit germinated are found beneath the surface, in the roots. The roots lay a strong foundation for the trees in the garden which result in different branches carrying fruit with varying characteristics. This garden of data visualisation imparts the possibility of future growth and development in this field of data science. This research aims at developing novel branches for the unbiased visualisations of trees devoted to missing data techniques.

2.2 Historical overview of categorical data analysis

Categorical data analysis was not initially at the foreground of statistical applications as its continuous data counterpart. The development of appropriate techniques for categorical data was initiated by Karl Pearson *circa* 1900 with contributions starting a decade thereafter.

Pearson's publication in (1901) with the title, "On lines and planes of closest fit to systems of points in space", is considered as the precursor of the derivation of CA.

An extensive list of contributing authors to the development of categorical data analysis is presented by Agresti (2002, 2013). Only contributions relevant to the methodology of this research will be mentioned.

Pearson (1904) developed the Chi-squared (χ^2) statistic and he was the first to use the term, 'contingency', which defined the deviation from the assumption of independence. Between 1900 and 1912, Yule focused on categorical data analysis and developed the odds ratio and measurements for the association between categorical variables, e.g. (Yule, 1900, 1906, 1912). Fisher (1922) elaborated on Pearson's χ^2-statistic with the addition of degrees of freedom to portray its distribution. He understood the relevance of appropriate methods for small samples, which led to the development of Fisher's exact test for 2×2 contingency tables, originally published in 1925 (Fisher, 1934). At the end of these two decades Hotelling (1936) introduced methods for canonical correlation, which were later found to be linked to the ideas of Benzécri (1973). A pioneering paper on cross-classified data was published by Bartlett (1935). The need for technological advances in computer and available software delayed further development of similar analyses (Bishop *et al.*, 1975). Techniques on multi-way contingency tables were not regarded as important at this point in time, but Bartlett's (1935) ideas were later extended and applied to multi-way table analyses by e.g. (Darroch, 1962; Plackett, 1962; Roy & Kastenbaum, 1956). Cochran (1954) was interested in a variety of topics concerning categorical data analysis, a highlight was his publication on choosing sample sizes for χ^2 approximations and separating χ^2-statistics into components. A test for the conditional independence in 2×2 contingency tables was also proposed by Cochran (1954), which was similar to the Mantel and Haenszel test (Mantel & Haenszel, 1959). Goodman and Kruskal (1979) continued the research of Pearson and Yule on the association of variables in contingency tables. Goodman's vast series of publications, e.g. (Goodman, 1964, 1969, 1971, 1985, 1996, 2000), contributed greatly to the development of categorical data analysis.

2.3 Indicator matrices

The data matrix of a multivariate categorical data set is commonly expressed as a coded dummy matrix consisting of zeros and ones, referred to as the indicator matrix, $\mathbf{G}$. Each variable has a set of mutually exclusive characteristics, regarded as the CLs. A column for each CL will be constructed in the indicator matrix and a one will be recorded for samples with a particular CL response. If a specific CL was not selected, a zero will be coded in the indicator matrix. The row totals of a complete indicator matrix should therefore add up to the total number of variables (Van de Geer, 1993).

Consider the following toy data set which is an example of a categorical data set with two variables with three possible CLs each:

Table 2.1 Toy data set

	Variable 1	Variable 2
Sample 1	1	2
Sample 2	3	1
Sample 3	1	3
Sample 4	2	2

The indicator matrix, $\mathbf{G}$, is presented as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{V1:1} & \mathbf{V1:2} & \mathbf{V1:3} & \mathbf{V2:1} & \mathbf{V2:2} & \mathbf{V2:3} \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

2.4 Dimension reduction

In general, the term decomposition refers to the separation of an object into smaller parts. Therefore, the decomposition in terms of matrix algebra, is the restructuring of a data matrix into the product of smaller matrices while preserving the initial information. The decomposition relies on the eigenvectors and eigenvalues of the particular matrix of interest.

The following authors contributed to the development of the well-known singular value decomposition (SVD): The first application was for square real matrices, independently developed and published by Beltrami in 1873 and Jordan in 1874, see (Eckart & Young, 1939; Klema & Laub, 1980; Stewart, 1993). The second extension was for complex square matrices, published in French by Autonne (1902) and further developed by Autonne (1913) and Browne (1930). Eckart and Young (1936) developed the theory to obtain the best approximation of a rectangular matrix in lower rank, regarding it as a problem of least squares.

2.4.1 Square matrices

If an eigenvalue, λ, and an eigenvector, $\mathbf{x}$, exist for a square matrix, $\mathbf{A}_{n\times n}$, the following expression holds:

$$\mathbf{Ax} = \lambda\mathbf{x}.$$

The eigenvectors and eigenvalues can be obtained from:

$$(\mathbf{A} - \lambda\mathbb{I})\mathbf{x} = \mathbf{0},$$

where $\mathbb{I}_{n\times n}$ is the identity matrix. The eigenvalues are calculated by solving the so called, characteristic equation:

$$|\mathbf{A} - \lambda\mathbb{I}| = \mathbf{0}.$$

The matrix, $\mathbf{A}$, will be associated with n eigenvalues, which will only be unique and non-zero for real non-singular data matrices (Rencher, 2002).

The eigendecomposition can be obtained for symmetric matrices and most square matrices. A square symmetric matrix, $\mathbf{A}_{n\times n}$, can be decomposed into the product of orthogonal matrices ,$\mathbf{U}$ and $\mathbf{U}'$, and a diagonal matrix ,$\mathbf{\Lambda}$, in the following way:

$$\mathbf{A} = \mathbf{U\Lambda U}', \quad \text{orthogonal: } \mathbf{UU}' = \mathbf{U}'\mathbf{U} = \mathbb{I}.$$

The eigendecomposition can also be written as:

$$\mathbf{AU} = \mathbf{U\Lambda},$$

where the eigenvalues are the diagonal elements of $\mathbf{\Lambda}_{n\times n}$ and the corresponding eigenvectors are the column vectors of $\mathbf{U}_{n\times n}$. It is the convention to order the eigenvalues, along with the eigenvectors, decreasingly (Borg & Groenen, 2005).

Now, the eigendecomposition of symmetric matrices can be rewritten as the product of two vectors ($\mathbf{U\Lambda}$ and $\mathbf{U}$) to obtain the spectral decomposition:

$$\begin{aligned}\mathbf{A} =&\ \mathbf{U\Lambda U}' \\ &= [\lambda_1\mathbf{u}_1 \quad \lambda_2\mathbf{u}_2 \quad \dots \quad \lambda_n\mathbf{u}_n]\begin{bmatrix}\mathbf{u}_1' \\ \mathbf{u}_2' \\ \vdots \\ \mathbf{u}_n'\end{bmatrix} \\ &= \lambda_1\mathbf{u}_1\mathbf{u}_1' + \lambda_2\mathbf{u}_2\mathbf{u}_2' + \dots + \lambda_n\mathbf{u}_n\mathbf{u}_n'.\end{aligned}$$

2.4.2 Rectangular matrices

The SVD enables the decomposition of any rectangular matrix into the product of three matrices, similar to the eigendecomposition of square matrices. A rectangular matrix, $\mathbf{A}_{n\times p}$, can be decomposed as follows:

$$\mathbf{A} = \mathbf{U}_{n\times p}\mathbf{\Lambda}_{p\times p}\mathbf{V'}_{p\times p},$$

where $\mathbf{U}$ and $\mathbf{V}$ are respectively the left and right singular vectors of $\mathbf{A}$. The diagonal matrix, $\mathbf{\Lambda}$, contains the singular values in decreasing order . The left and right singular vectors are orthonormal ($\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbb{I}_p$) and the singular values are positive (Borg & Groenen, 2005; Greenacre, 2017).

The SVD can be linked to the spectral decomposition (cf. 2.4.1) of symmetric square matrices by evaluating the decomposition of the product of $\mathbf{A}$ with its transpose, $\mathbf{A}'$:

$$\begin{aligned}\mathbf{A}'\mathbf{A} &= (\mathbf{U\Lambda V}')'\mathbf{U\Lambda V}' & \text{or} \qquad \mathbf{AA}' &= \mathbf{U\Lambda V}'(\mathbf{U\Lambda V}')' \\ &= \mathbf{V\Lambda U}'\mathbf{U\Lambda V}' & &= \mathbf{U\Lambda V}'\mathbf{V\Lambda U}' \\ &= \mathbf{V\Lambda}^2\mathbf{V}' & &= \mathbf{U\Lambda}^2\mathbf{U}',\end{aligned}$$

which results in the eigendecomposition of $\mathbf{A}'\mathbf{A}$ and $\mathbf{AA}'$. Therefore, the left singular vectors of $\mathbf{A}$ are the eigenvectors of $\mathbf{AA}'$ and the right singular vectors of $\mathbf{A}$ are the eigenvectors of $\mathbf{A}'\mathbf{A}$ (Borg & Groenen, 2005; Greenacre, 2017). This also shows that the SVD of a positive semi-definite matrix (all eigenvalues ≥ 0) results in the eigendecomposition (Abdi, 2007; Borg & Groenen, 2005).

Both the SVD of $\mathbf{A}$ and $\mathbf{A}'\mathbf{A}$ are regarded as the rank deficient cases, since zero singular values might be included in the last entries on the diagonal, which result in redundant singular vectors. The full rank SVD reduces the left and right singular vectors according to the rank of the matrix that is to be decomposed. The number of non-zero singular values is equal to the rank of a matrix (Green & Carroll, 1976; Ientilucci, 2003). Suppose the matrix, $\mathbf{A}_{n\times p}$, is of rank k with $n \geq p$ and $k < p$, the full rank SVD (or complete SVD) can be expressed by the following:

$$\mathbf{A} = \mathbf{U}_{n\times n}\mathbf{\Lambda}_{n\times p}\mathbf{V}'_{p\times p},$$

with orthogonal matrices, $\mathbf{U}$ and $\mathbf{V}$, such that $\mathbf{UU}' = \mathbf{U}'\mathbf{U} = \mathbb{I}_n$ and $\mathbf{VV}' = \mathbf{V}'\mathbf{V} = \mathbb{I}_p$.

The diagonal matrix of singular values is expressed as follows:

$$\mathbf{\Lambda}_{n\times p} = \begin{bmatrix} \mathbf{\Lambda}_{k\times k} & \mathbf{0}_{k\times(p-k)} \\ \mathbf{0}_{(n-k)\times k} & \mathbf{0}_{(n-k)\times(p-k)} \end{bmatrix}$$

It now follows that $\mathbf{A}$ can be approximated in lower dimension, k:

$$\mathbf{A} = \mathbf{U}_{n\times k}\mathbf{\Lambda}_{k\times k}\mathbf{V}'_{k\times p},$$

with orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$ (Borg & Groenen, 2005; Gower, Lubbe & Le Roux, 2011). Although the singular vectors are mutually orthogonal, the matrices of singular vectors are not orthogonal. It is only possible when $p = k$ ($\mathbf{VV}' = \mathbb{I}_{p\times p}$) and $p = n = k$ ($\mathbf{UU}' = \mathbb{I}_{n\times n}$) (Green & Carroll, 1976).

When using the SVD to approximate the best, in a least squares sense, lower rank representation of a rectangular matrix; the method is also referred to as the Eckart-Young decomposition (Eckart & Young, 1936). The Eckart-Young theorem relies on a least squares approximation; suppose $\mathbf{X}^*$ is the approximated lower rank matrix of $\mathbf{X}$, with the approximated rank being equal to k. The following sum of squares is minimised:

$$||\mathbf{X} - \mathbf{X}^*||^2 = tr\{(\mathbf{X} - \mathbf{X}^*)(\mathbf{X} - \mathbf{X}^*)'\}$$

for all possible matrices $\mathbf{X}^*$ of a rank not exceeding k (Cox & Cox, 2001; Gower *et al.*, 2011).

The result of $\mathbf{X}^*$ can conveniently be obtained by introducing the following matrix, $\mathbf{J}$:

$$\mathbf{J}_{p\times p} = \begin{bmatrix} \mathbb{I}_{k\times k} & \mathbf{0}_{k\times(p-k)} \\ \mathbf{0}_{(p-k)\times k} & \mathbf{0}_{(p-k)\times(p-k)} \end{bmatrix}$$

Now, it follows that the lower rank approximation of $\mathbf{X}$, $\mathbf{X}^*$, is obtained from the following decomposition:

$$\mathbf{X}^*_{n\times k} = \mathbf{U\Lambda JV}' = \mathbf{UJ\Lambda V}' = \mathbf{UJ\Lambda JV}'.$$

The final columns, $p - k$, of the matrices $\mathbf{UJ}_{p \times p}$ and $\mathbf{VJ}_{p \times p}$ will fall away.

The lower rank approximation can therefore be expressed by:

$$\mathbf{X}^*_{n \times k} = \mathbf{U}_{n \times k} \mathbf{\Lambda}_{k \times k} \mathbf{V}'_{k \times k}.$$

2.5 Visualisation

According to Yau (2013: 44), visualisation is not a tool, but a medium. In his own words, visualisation is "a way to explore, present and express meaning in data."

The term data visualisation aims to direct the focus to grasping underlying structures and trends in the data, rather than simply representing data in a table or insufficient graphical display (Unwin *et al.*, 2008). Visualisation exposes information and when used correctly can ease the understanding of complex data problems. In a sense, visualisations remove the barriers of scientific language and complex mathematics to make data approachable to a wider community (Keim, 2002). However, incorrect use of visualisations could lead to more confusion and biased interpretations. Data visualisation is a science of its own and care should be taken with the construction and publication of suitable visual representations (Unwin *et al.*, 2008). The subsequent sections on visualisation present a short discussion on simplistic visualisations for categorical data, followed by an overview of biplots.

2.5.1 Modest visualisation

A bipartite graph / bigraph can be useful in the visualisation of small multivariate categorical data sets. A bigraph visualises samples and CLs as two disjoint vertices and connects the responses of samples to specific CLs, similar to affiliation network analysis (Dramalidis & Markos, 2016). It is a simple display which aids in the discovery of possible associations between samples and responses to variables (Michailidis & De Leeuw, 1998). Figure 2.1 is adapted from Michailidis and De Leeuw (1998) and illustrates a bigraph of the responses of five samples to two variables.

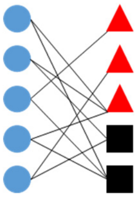


Figure 2.1 Bigraph of five samples (circles) and two categorical variables (squares: two CLs and triangles: three CLs).

Attraction graphs can be used to display the associations between two categorical variables in CA. The graph of attractions is similar to the display of Figure 2.1, but now each variable is represented in a separate vertex. The lines that connect the CLs of the two categorical variables are drawn based on the rate of attraction between the specific CLs. The attraction rates between two CLs are calculated using the proximities of a two-way contingency table. Consider two categorical variables, J and K, with the following possible CLs , p_j, where $j = 1, \dots, J$ and p_k, where $k = 1, \dots K$. The marginal frequencies are indicated by f_j and f_k. The association rate (t^{jk}) can be calculated as follows (Le Roux & Rouanet, 2004):

$$t^{jk} = \frac{\left(f_{jk} - f_j f_k\right)}{f_j f_k}.$$

Figure 2.2 illustrates an attraction graph for a profession variable with eight CLs and a work ethic variable with nine CLs. The original data were published by Maisonneuve in 1987 and is presented in Le Roux and Rouanet (2004). Individuals from different professions had to select three qualities that they associated with a ‘nice person’. The following attractions were deemed significant by using an attraction rate of 0.35:

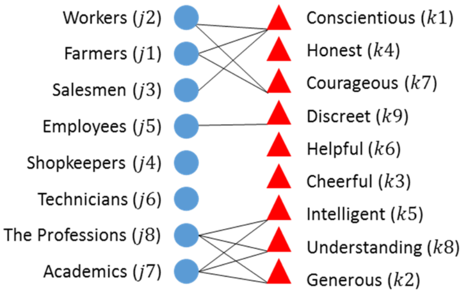


Figure 2.2 Attraction graph of professions (circles) and qualities (triangles).

The rate of significant attraction is determined *a priori*, but in large data sets all CLs with positive attraction rates will be connected in the graph of attractions. Even in this simple illustration (cf. Figure 2.2) two groupings can be identified. Conscientiousness, honesty and courageousness are qualities that Workers, Farmers and Salesmen associate with a ‘nice person’. Also, intelligence, understanding and generosity are qualities that The Professions and Academics regard as qualities of a ‘nice person’.

2.5.2 Biplots

First, we should reflect upon the well-known scatterplot for the illustration of the relationship between two continuous variables. The scatterplot is regarded as one of the modern graphical displays that first appeared before the ‘Golden Age’ of data visualisation between 1800 and 1850 (Friendly, 2008). The first publication of a modern scatterplot, as we know it today, was published by William Herschel in 1833 (Friendly & Denis, 2005). However, the first application was referred to as the scatter diagram by the French military to investigate the patterns of rifle shots fired at a target (Beh & Lombardo, 2014; Friendly & Denis, 2005). This particular application of a scatter diagram is still used as a popular example to explain the concepts of precision and bias in elementary statistics courses (Biemer & Lyberg, 2003). Credit is also given to Kurtz and Edgerton (1939) for their mentioning of the term scatterplot in the *Statistical dictionary of terms and symbols*. Other terms to describe a scatterplot are: scatter diagram, dot diagram and when displaying multiple continuous variables, a scatterplot matrix.

When displaying the responses of two continuous variables, the range of possible responses to each variable is annotated on calibrated orthogonal axes, one for each variable. The responses are depicted as points in the visualisation, also known as the samples. The response of a specific observation can be determined by projecting the point perpendicularly to one of the axes and reading the corresponding response from the calibrated axis (Beh & Lombardo, 2014; Friendly & Denis, 2005; Gower & Hand, 1996).

Gabriel (1971) was the first to introduce the concept of a biplot display. The ‘bi’ in ‘biplot’ refers to the display of two modes, samples and variables, and not the dimension of the configuration. It is considered to be a generalisation of a scatterplot (Greenacre, 2010) and presents each row and column of a matrix as unique vectors, resulting in as many axes as columns. The rows refer to the samples of a data matrix and the columns refer to the variables. The vectors are obtained such that any element of the matrix will be equal to the inner-product of the corresponding row and column in the data matrix (Gabriel, 1971). Multidimensional scaling (MDS) is the technique of obtaining a lower-dimensional representation of high-dimensional data by preserving the distances between the observations (Cox & Cox, 2001). Therefore, biplots can be constructed for MDS techniques, since samples and responses are displayed according to the interpoint distances between them (Greenacre, 2010). The classic biplot can be constructed by first obtaining the SVD of a data matrix, $\mathbf{X}$, expressed as:

$$\mathbf{X} = \mathbf{U\Lambda V}',$$

which is equivalent to:

$$\mathbf{X} = (\mathbf{U\Lambda}^{\alpha})(\mathbf{V\Lambda}^{1-\alpha})';\ 0 \leq \alpha \leq 1.$$

If the biplot is presented in two dimensions, plotting the first two columns of $\mathbf{U\Lambda}^{\alpha}$ will represent the rows of the data matrix, $\mathbf{X}$, and plotting the first two columns of $\mathbf{V\Lambda}^{1-\alpha}$ will represent the columns of the data matrix, $\mathbf{X}$. By changing the value of α, the contribution of the singular values for the row vectors, $\mathbf{U\Lambda}^{\alpha}$, and column vectors ,$\mathbf{V\Lambda}^{1-\alpha}$, can be altered which will result in different biplots displays (Cox & Cox, 2001). Symmetric biplots occur when $\alpha = 0.5$, since an equal contribution of the singular values are made to both the row and column vectors (Greenacre, 2010).

The classic biplot of Gabriel was the precursor of the popular PCA biplots. The PCA biplot is an example of an asymmetric biplot, since $\alpha = 1$. The principal components are represented by the right singular vectors, $\mathbf{V}$, and the component loadings are represented by $\mathbf{U\Lambda}$, which is equivalent to $\mathbf{XV}$ (Cox & Cox, 2001).

Biplots are not limited to certain variable types, but the displays differ for continuous and categorical variables. Continuous variables are represented as calibrated axes, an axis for each continuous variable. The multiple axes are not orthogonal, but similar to two-dimensional scatterplots, sample points are projected perpendicularly to a calibrated axis to obtain the response value. The CLs of categorical variables are illustrated as a simplex of points referred to as the CLPs, one point for each CL (Gower & Hand, 1996). The display of the variables, whether it is an axis or CLP simplex, is considered to be the 'framework' or 'scaffolding' of the display (Cox & Cox, 2001). The proximity of the sample points and CLPs reveal associations. Dissimilarities are illustrated through points that are not situated in close proximity, whereas closely positioned points reflect high association and similarity (Gower *et al.*, 2011). The sample points are positioned at the vector sum of its observed CLPs (Gower & Hand, 1996).

Since the biplot is an approximation in lower dimension, it is possible that some observed CLPs will not be in closest proximity to the specific sample as indicated in the data matrix. Biplots are low-dimensional representations of high-dimensional data sets in Euclidean space. The low-dimensionality eases visual interpretation and the properties of the Euclidean space enables the use of geometrical properties (Michailidis & De Leeuw, 1998). The biplot display will always be an approximated representation, since the data points are presented in a reduced dimensional space (Greenacre, 2010). The goal is to utilise a dimension reduction technique that minimises the amount of information that is lost. Different distance measures can be used in the construction of biplots. The choice of the distance measure is based on the criterion for minimising the loss of information (Gardner, 2001). Biplots reveal the main structures in the data, highlighting correlated variables and possible similarities between observations (Greenacre, 2010). Since biplots convey information regarding data in an approximated dimensional space, two approaches of interpretations from continuous data biplot displays are used: interpolation and prediction. Interpolation is used to determine the position of samples in the display, whereas prediction is used to determine the values of the sample from the display using orthogonal projections onto the axes (Gower & Hand, 1996). In a standard scatterplot, axes with different calibrations are not required for prediction and interpolation, since this visualisation does not approximate high-dimensional data in two dimensions.

The classic biplot by Gabriel (1971) was based on PCA and displayed the row and column vectors using Euclidean distances. This idea was extended by Gower and Harding (1988) by allowing the use of different distance measures to express the points in the display.

2.6 Correspondence analysis

The CA technique has been independently developed by a number of researchers and therefore has a variety of names aside from the well-known French translation of "*analyse des correspondances*". Two main streams of development can be identified for CA: the algebraic numerical interpretation and the geometric graphical interpretation. The form adopted in this research is the graphical interpretation, which in its simplest form can be described as a technique to uncover the association between two categorical variables (rows and columns of a contingency table) in a lower-dimensional visual representation. The CA display contains both the lower-dimensional vector spaces for the rows and columns in the same configuration (Greenacre, 1984).

2.6.1 Historical overview

Here follows a brief chronological summary of the development of CA and MCA, summarised from various sources: Beh and Lombardo (2012), Di Franco (2016), Greenacre (1984), Le Roux and Rouanet (2004) and Tenenhaus and Young (1985). This summary only provides an overview and not a comprehensive reflection of the development of these methods.

The first publication related to CA was that of Hirschfeld (1935) describing the correlation between the two variables (rows and columns) in a contingency table. Non-mathematical approaches similar to Hirschfeld's work were published by Richardson, Kuder (1933) and Horst (1935) with psychometric applications. The term, 'reciprocal averaging', was initiated by Horst (1935). Fisher (1940) focused on discriminant analysis and published the famous hair- and eye colour data set. Scales for categorical variables, referred to as dual scaling, were developed by Guttman (1941) with the same underlying method as his predecessors. He also generalised his methods to more than two categorical variables which led to MCA as we know it today, also referred to as optimal scaling. The well-known MCA application of the Burt matrix was published by Burt (1950). The Burt matrix , $\mathbf{G}'\mathbf{G}$, is a block-diagonal matrix with each diagonal block containing the cross-tabulations of the indicator matrix of a particular categorical variable. The off-diagonal block elements contain the pairwise two-way contingency tables of variables.

The mathematical development of CA was further studied between 1940 and 1950 following on Guttman's dual scaling. Japanese scientists under the leadership of Hayashi published numerous papers on the quantification of categorical data in the 1950s, e.g. (Hayashi, 1950, 1951, 1953). Most applications of CA related techniques before 1970 are on biometric and psychometric data problems. A French research group led by Benzécri started studying data tables related to linguistics and developed the geometric form, known as CA, in the 1960s. Unfortunately, the French developments of CA were not available in English and therefore did not receive immediate global attention. An English translation of a paper by Benzécri (1969) appeared in the late 1960s, which is regarded as the source of further development and application of this method. The following authors are frequently mentioned for the development of CA: Green and Carroll (1976), Gifi (Michailidis & De Leeuw, 1998), Greenacre (1984) and Nishisato (1980).

2.6.2 Computations

Since visualisation is a core element of this research, the geometrical approach of CA is applied. In the context of multivariate visualisation techniques, the French statisticians, refer to a data triplet containing information of a data set. The three elements of the data triplet consist of: (1) observations represented in multidimensional space, (2) their weights and (3) distances between observations (Greenacre, 2010). When working with qualitative data, especially frequencies, the weights of certain objects might be significantly different from others which could distort the visual representation in a lower dimension. Euclidean distances are therefore not suitable to determine the distances between rows and columns in a contingency table, since the marginal sums will influence and bias the distances (Cox & Cox, 2001). The χ^2-distance, also referred to as the weighted Euclidean distance, is used to measure distances in CA. This follows intuitively from the χ^2 -statistic to test the independence between rows and columns of a contingency table (Greenacre, 2017). The data triplet for CA will be discussed in the following section and is defined as follows: (1) profiles, (2) masses and (3) χ^2-distances (Greenacre, 2010).

The notation proposed by Greenacre (2017) is adopted; suppose that $\mathbf{N}_{I\times J}$ represents a data matrix with positive column and row totals, where the columns and rows represent two categorical variables with multiple CLs each. The well documented Pearson's χ^2-statistic is first reviewed in which the deviations from the independence assumption of the rows and columns of a two-way contingency table are investigated (Blasius & Greenacre, 2006). The cell frequencies are denoted by n_{ij} and the grand total of the data matrix is given by $n = \sum_{i=1}^{I}\sum_{j=1}^{J} n_{ij} = \mathbf{1}'\mathbf{N}\mathbf{1}$. The expected frequencies, $\hat{n}_{ij}$, are calculated by the product of the row and column marginal sums which are divided by the grand total of the data matrix (e.g. $\hat{n}_{12} = \frac{n_{1\bullet}\times n_{\bullet 2}}{n}$). The columns and rows of a data matrix are considered to be independent if the expected frequencies and observed frequencies are equal.

The χ^2-statistic is calculated as follows:

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{\left(n_{ij} - \hat{n}_{ij}\right)^2}{\hat{n}_{ij}}.$$

Reference will be made to the calculation of the χ^2-statistic later in this discussion. Now, the data matrix is transformed into a correspondence matrix of proportions, $\mathbf{P}$, by dividing all elements of $\mathbf{N}$ by the grand total of the data matrix, $n = \sum_{i=1}^{I} \sum_{j=1}^{J} n_{ij} = \mathbf{1}'\mathbf{N1}$. Therefore, the correspondence matrix is obtained from: $\mathbf{P} = \frac{\mathbf{N}}{n}$ and now has the property that all row and column marginal sums will be equal to one, as well as the grand total of the correspondence matrix. In this context, the relative frequencies are referred to as profiles. We distinguish between row- and column profiles, where each element in a two-way contingency table will comprise of both profiles, the row profile is obtained by dividing the frequency for a particular cell by its row total and the column profile is obtained by dividing the cell frequency by its column total. The row profiles are denoted by the vector, $\mathbf{a}_i$, with element a_{ij} referring to the j^{th} element of the i^{th} row profile. Similarly, the column profiles are denoted by the vector, $\mathbf{b}_j$, with element b_{ij} referring to the i^{th} element of the j^{th} column profile. The average row profile is the profile of the column marginal sums in the original contingency table and the average column profile is the profile of the row marginal sums in the original contingency table.

The elements of the average row profile provide the column masses and those of the average column profile, the row masses. These masses are used as weights to correctly reflect the importance of CLs within a variable in comparison to the other recorded frequencies. The row masses are denoted by $r_i = \sum_{j=1}^{J} p_{ij}$ or $\mathbf{r} = \mathbf{P1}$, which is the mass of the i^{th} row and the column masses are denoted by $c_j = \sum_{i=1}^{I} p_{ij}$ or $\mathbf{c} = \mathbf{P}'\mathbf{1}$, which is the mass of the j^{th} column.

The χ^2-distances is calculated using the profiles and masses. The profiles incorporate the specific contribution of each cell in relation to the others and are scaled by the average profiles (masses) by using the inverse of the masses as the weighting factor.

The χ^2-distance between row i and i' is then calculated as follows:

$$d_{ii'}^2 = \sum_{j=1}^{J} \frac{\left(a_{ij} - a_{i'j}\right)^2}{c_j}.$$

The χ^2-distance between column j and j' is similarly calculated by:

$$d_{jj'}^2 = \sum_{i=1}^{I} \frac{\left(b_{ij} - b_{ij'}\right)^2}{r_i}.$$

The χ^2-distance equalises the contributions of the categories by dividing by the average profile. This does not mean that the CLs with lower frequency will be unrealistically inflated, since the weighting is applied in accordance with its mass (Greenacre, 2017). The weighted distance measure also satisfies the property of distributional equivalence. This means that if rows (or columns) with the same profiles are combined, the χ^2-distance will not be affected (Gower & Hand, 1996; Greenacre, 1984). Profiles that are the same will not provide additional discriminating information of the data and therefore do not benefit the analysis when considered separately. Since the profiles are weighted by the same masses, there will be no difference between equal profiles, this will however not be the case when using the Euclidean distance measure (Greenacre, 2017). It has been debated whether the χ^2-distance is suitable, especially in the case of outlier categories with either low or high frequencies. The contribution of a CL should not be confused by its position in the biplot. It is true that CLs with lower or higher frequency will be regarded as outliers when compared to frequently occurring CLs. These outlier CLs will be positioned with a larger distance between the commonly occurring CLs. The CLs are however proportionally weighted by their masses and hence the position in the map does not relate to the influence the CL has on the CA analysis. Therefore, the low frequency categories do not result in inflated representation (Greenacre, 2013).

The algebraic understanding of the rank of a matrix is synonymous with the geometric understanding of dimensionality. The previous sections on dimension reduction (cf. 2.4) focused on the lower rank approximation of a matrix, whereas CA aims to display a multidimensional data set in a lower-dimensional visualisation. The coordinates of CA visualisations are directly obtained from the SVD solution (Greenacre, 2017). However, the classic SVD approach as presented in Section 2.4 will not suffice, since certain constraints are now imposed on the rows and columns of the correspondence matrix. Generalised SVD (GSVD) allows the decomposition of a rectangular matrix when there are constraints placed on the rows and columns of the data matrix. The lower rank approximation will now be a weighted least squares estimate of the full rank matrix (Abdi, 2007).

Before presenting the computational algorithm of CA, the masses have to be expressed as diagonal matrices; row masses: $\mathbf{D}_r = diag(\mathbf{r})$ and column masses: $\mathbf{D}_c = diag(\mathbf{c})$. It follows that the profiles are obtained through the following matrix multiplications; row profiles: $\mathbf{R} = \mathbf{D}_r^{-1}\mathbf{P}$ and column profiles: $\mathbf{C} = \mathbf{D}_c^{-1}\mathbf{P}'$ (Greenacre, 1984).

The matrix of standardised residuals, $\mathbf{S}$, is derived from the independence assumption of the χ^2-statistic. If the rows and columns are independent, using the correspondence matrix notation, the following will hold (Beh & Lombardo, 2014):

$$\mathbf{P} = \mathbf{rc}'.$$

The standardised residuals are obtained by expressing the χ^2-statistic in terms of the correspondence matrix notation:

$$\mathbf{S} = \mathbf{D}_r^{-1/2}(\mathbf{P} - \mathbf{rc}')\mathbf{D}_c^{-1/2}.$$

The row profiles are centred in accordance with their column masses, $\mathbf{rc}' = \mathbf{r1}'\mathbf{P}$, the profiles are weighted by their masses by pre-multiplication of the diagonal row masses, $\mathbf{D}_r^{1/2}$, and the post-multiplication of the diagonal column masses, $\mathbf{D}_c^{1/2}$. This results in acquiring the χ^2-distance between the row profiles (Greenacre, 2013).

Now, the classic SVD approach can be applied to the standardised residuals to obtain the lower dimension representation of $\mathbf{S}$:

$$\mathbf{S} = \mathbf{U\Lambda V}',$$

where the right- and left singular vectors are orthonormal, $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbb{I}$.

The decomposition of the centred correspondence matrix, $\mathbf{P}$, can be obtained by equating the solutions of the standardised residuals (Rencher, 2002):

$$\begin{aligned}\mathbf{D}_r^{-1/2}(\mathbf{P} - \mathbf{rc}')\mathbf{D}_c^{-1/2} &= \mathbf{U\Lambda V}' \\ \mathbf{P} - \mathbf{rc}' &= \mathbf{D}_r^{1/2}(\mathbf{U\Lambda V}')\mathbf{D}_c^{1/2} \\ &= \left(\mathbf{D}_r^{1/2}\mathbf{U}\right)\mathbf{\Lambda}\left(\mathbf{V}'\mathbf{D}_c^{1/2}\right) \\ &= \mathbf{A\Lambda B}',\end{aligned}$$

where $\mathbf{U}'\mathbf{D}_r^{-1}\mathbf{U} = \mathbf{V}'\mathbf{D}_c^{-1}\mathbf{V} = \mathbb{I}$. The generalised left singular vectors are given in $\mathbf{A} = \mathbf{D}_r^{1/2}\mathbf{U}$ and the generalised right singular vectors are given in $\mathbf{B} = \mathbf{D}_c^{1/2}\mathbf{V}$.

It is however common practice to obtain the GSVD by applying the SVD to the transformed data as was shown in the first instance above (Greenacre, 1984).

Principal axes can be understood in regression terminology as the best fitting line that approximates the observations. However, in MDS a principal axis refers to a direction (or spread) in multidimensional space that captures maximal variation in a lower-dimensional approximation (Greenacre, 2017). Usually the most variation is represented in the first principal axes.

There are two sets of coordinates for CA maps; standard and principal. The weighted sum of squares of the principal coordinates of a certain dimension, say k, will be equal to the associated eigenvalue, λ_k^2, also referred to as the principal inertia. The sum of squares of the standard coordinates of any dimension will be equal to one (Greenacre, 2017; Greenacre & Pardo, 2006a). The standard coordinates will tend to frame a visualisation, since the coordinates will occur on the extremities of the visualisation space due to the standardisation.

The standard coordinates:

Rows: $\mathbf{\Phi} = \mathbf{D}_r^{-1/2}\mathbf{U}$; Columns: $\mathbf{\Gamma} = \mathbf{D}_c^{-1/2}\mathbf{V}$

These coordinates are scaled as follows: $\mathbf{\Phi}\mathbf{D}_r\mathbf{\Phi}' = \mathbf{\Gamma}\mathbf{D}_c\mathbf{\Gamma}' = \mathbb{I}$.

The principal coordinates:

Rows: $\mathbf{F} = \mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{\Lambda} = \mathbf{\Phi}\mathbf{\Lambda}$; Columns: $\mathbf{G} = \mathbf{D}_c^{-1/2}\mathbf{V}\mathbf{\Lambda} = \mathbf{\Gamma}\mathbf{\Lambda}$

These coordinates are scaled as follows: $\mathbf{F}\mathbf{D}_r\mathbf{F}' = \mathbf{G}\mathbf{D}_c\mathbf{G}' = \mathbf{\Lambda}^2$.

The relationship between the rows and columns can be expressed by the following formulae (Blasius & Greenacre, 2006):

$$\mathbf{G} = \mathbf{D}_c^{-1}\mathbf{P}'\mathbf{F}\mathbf{\Lambda}^{-1}, \qquad \mathbf{F} = \mathbf{D}_r^{-1}\mathbf{P}\mathbf{G}\mathbf{\Lambda}^{-1}$$

where the standard coordinates of the rows are given by $\mathbf{F}\mathbf{\Lambda}^{-1}$ and the standard coordinates of the columns are given by $\mathbf{G}\mathbf{\Lambda}^{-1}$. Since $\mathbf{F}\mathbf{\Lambda}^{-1} = \mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{\Lambda}\mathbf{\Lambda}^{-1} = \mathbf{D}_r^{-1/2}\mathbf{U} = \mathbf{\Phi}$ and $\mathbf{G}\mathbf{\Lambda}^{-1} = \mathbf{D}_c^{-1/2}\mathbf{V}\mathbf{\Lambda}\mathbf{\Lambda}^{-1} = \mathbf{D}_c^{-1/2}\mathbf{V} = \mathbf{\Gamma}$. The column profiles are given in $\mathbf{D}_c^{-1}\mathbf{P}'$ and the row profiles in $\mathbf{D}_r^{-1}\mathbf{P}$. This shows that the principal coordinates, $\mathbf{F}$ and $\mathbf{G}$, are weighted according to their standard coordinates ,$\mathbf{\Phi}$ and $\mathbf{\Gamma}$.

A final remark on CA is the measure of variance, which is referred to as the inertia. The inertia is related to the calculation of the χ^2-statistic and measures the total variation across all profiles in the two-way contingency table (Greenacre, 2010, 2017):

$$inertia = n^{-1}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\frac{\left(n_{ij}-\hat{n}_{ij}\right)^2}{\hat{n}_{ij}}\right) = \frac{\chi^2}{n},$$

which is also equivalent to the sum of squares of the standardised residuals, $tr(\mathbf{SS}')$, also equivalent to the sum of squares of the squared singular values ($\lambda_1^2 + \lambda_2^2 \ldots + \lambda_n^2$).

2.6.3 Correspondence analysis visualisations

If coordinates of the same scale are used for the visualisation of the rows and columns, the display is referred to as a symmetric CA map. Since there is no distance defined between row and column profiles, the distances presented in the symmetric map cannot be directly interpreted as if measured on the same scale. The CA map simultaneously visualises two different vector spaces. Fortunately, this predicament is solved by using asymmetric CA maps, referred to as CA biplots which simultaneously display rows and columns in one visualisation. The rows are presented using principal coordinates and the columns using standard coordinates (Greenacre, 2017).

Using the `ca()` function in the `R` package, `ca` (Nenadić & Greenacre, 2007; R Core Team, 2017) the coordinates for the construction of the CA biplot can be obtained.

The default parameters are given in the argument list of the `ca()` function:

```
ca(obj, nd = NA, suprow = NA, supcol = NA, subsetrow = NA,
subsetcol = NA, ...)
```

The '`obj`' parameter is the categorical data set and the number of dimensions to be used for the output of the results are specified in the '`nd`' parameter. The remaining parameters are not of particular interest for this research study and will not be addressed.

The principal coordinates of the rows are obtained from the '`rowpcoord`' object and the standard coordinates of the columns are obtained from the '`colcoord`' object, which are available as the output of the `ca()` function.

2.7 Multiple Correspondence Analysis

The MCA approach is an extension of CA when more than two categorical variables are considered. The MCA technique enables the investigation of the interrelationships between CLs within a variable (Greenacre & Pardo, 2006a). The MCA method successfully removes unnecessary information by approximating the samples and responses in lower dimension while maximising the variation to express the associations between the variables and samples (Blasius & Thiessen, 2012; Iodice D'Enza & Markos, 2015).

As with the history of CA, MCA has been independently discovered by a number of researchers from different countries and therefore also has a variety of names aside from MCA: homogeneity analysis, dual scaling, optimal scaling and quantification methods.

Tenenhaus and Young (1985) show that the different approaches to MCA are superficially different and result in similar outcomes. A more recent paper by Di Franco (2016) highlights the similarities between the approaches to MCA that were independently developed by the Dutch- and French schools. This paper concludes by questioning whether the different approaches should not rather be regarded as "variations of the same technique" than labelled as independent methods (Di Franco, 2016: 1313).

The MCA approach followed in this research is the application of CA on the indicator matrix. Consider a multivariate categorical data set, $\mathbf{X}$, with n samples and p categorical variables. Similar to the notation followed in Section 2.6, the indicator matrix is now weighted by the rows and columns:

$$\mathbf{S} = p^{-1/2}\mathbf{G}\mathbf{C}^{-1/2},$$

where $\mathbf{G}$ is the indicator matrix of $\mathbf{X}$, $\mathbf{C}$ is a diagonal matrix containing the column marginal sums of the indicator matrix, i.e. the total occurrences per CL in the indicator matrix. The weighting of the rows, $p^{-1/2}$, is not necessary, but is enforced to relate to the CA approach. The row weighting is expressed as a scalar, since all row weights are equal when there are no missing responses.

The SVD of the weighted indicator matrix is expressed by the familiar result of CA:

$$\mathbf{S} = p^{-1/2}\mathbf{G}\mathbf{C}^{-1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}'.$$

2.7.1 Multiple correspondence analysis biplots

The MCA biplot is constructed using the principal coordinates of the rows to display the sample coordinates and the standard coordinates of the columns to display the CLPs. The coordinates are obtained from the solution of the SVD expressed in Section 2.7.

The sample coordinates, in terms of principal coordinates, are obtained from (Gower *et al.*, 2011):

$$\mathbf{Z} = p^{-1/2}\mathbf{U\Lambda}.$$

The coordinate matrix for the CLPs, in terms of standard coordinates, is obtained from:

$$\mathbf{CLP} = \mathbf{C}^{-1/2}\mathbf{V}.$$

Each sample coordinate will be located at the vector sum of the CLPs associated to the responses of the particular sample (Gower *et al.*, 2011).

Non-zero distances result only from differing responses, if two samples are in agreement with respect to a certain variable, the distances between the identical responses for the two samples will be zero. Short distances therefore reflect strong associations (or similarities). Therefore, the MCA solution is focussed on emphasizing differences in order to differentiate between individuals (Le Roux & Rouanet, 2004).

The MCA biplots in this research are constructed using the coordinates obtained from the `mjca()` function in the `R` package, `ca` (Nenadić & Greenacre, 2007; R Core Team, 2017).

The default parameters are given in the argument list of the `mjca()` function:

```
mjca(obj, nd = 2, lambda = c("adjusted", "indicator", "Burt",
"JCA"), supcol = NA, subsetcat = NA, ps = ":", maxit = 50,
epsilon = 0.0001, reti = FALSE, ...)
```

Again, the '`obj`' parameter refers to the categorical data set and '`nd`' refers to the number of dimensions of the MCA solution. In order to perform MCA on the indicator matrix, the '`lambda`' parameter should be set to "`indicator`". The parameters of this function will be further discussed in Chapter 4 (cf. 4.5.1).

The sample coordinates are obtained from the '`rowpcoord`' output and the CLPs are obtained form the '`colcoord`' output from the `mjca()` function.

2.8 Principal component analysis

A slight detour from categorical data analysis has to be made in order to introduce the theory of PCA for multivariate continuous data which preceded the development of CA and MCA. In short, PCA is a technique to reduce the dimension of a continuous multivariate data set whilst optimising the amount of variation captured in lower dimension. Principal components are obtained from linear combinations of the weighted original variables, also referred to as the component loadings, which maximise the correlation between the principal component and the component loadings. The consecutive principal components are obtained in a similar manner with the constraint that all principal components have to be uncorrelated. The first principal component will capture most of the variation in the first dimension, therefore illustrating maximal separation between observations, the second principal component will be orthogonal to the first principal component, which will capture maximal variation in the second dimension. The last dimension will capture the least variation and will result in the smallest correlation between the last principal component and the component loadings (Rencher, 2002; Van de Geer, 1993).

Consider a continuous data matrix, $\mathbf{X}$, with I samples and J variables. First, the matrix is centred at the origin (Gower *et al.*, 2011; Greenacre, 2010):

$$\mathbf{X}^* = \left(\mathbb{I} - \frac{1}{I}\mathbf{11}'\right)\mathbf{X}.$$

Variables that are not measured on the same scale should be standardised (Le Roux & Rouanet, 2004). A popular approach to scaling each variable of the centred data matrix is to normalise the columns by dividing the observations of each column by the standard deviation of the particular column (Gower, 2006).

Similar to the CA and MCA solutions (cf. 2.6 and 2.7), the PCA solution can be obtained from the SVD of the centred and standardised data matrix, $\mathbf{X}^*$, with principal coordinates obtained from $\mathbf{U\Lambda}$ and standard coordinates obtained from $\mathbf{V}$.

In order to draw a closer comparison to MCA, weights can also be enforced on the rows and the columns of the data matrix before performing PCA, which is then referred to as the weighted PCA approach. Suppose that the rows and columns are equally weighted by the number of samples and variables, respectively. Therefore, the SVD of the weighted centred data matrix is to be determined as follows:

$$\left(1/\sqrt{I}\right)\mathbf{X}^{*}\left(1/\sqrt{J}\right) = \left(1/\sqrt{IJ}\right)\mathbf{X}^{*} = \mathbf{U\Lambda V}'.$$

The principal coordinates are obtained from $(1/I)\mathbf{U\Lambda}$ and the standard coordinates for the columns are obtained from $(1/J)\mathbf{V}$.

A categorical alternative to PCA, Categorical PCA, is suitable for the analysis of ordered categorical variables, typically measured on the Likert scale (Blasius & Thiessen, 2012). This approach is however beyond the scope of this research.

2.9 Orthogonal Procrustes analysis

In general, Procrustes analysis consists of different applications of configurative matching (Poor & Wherry, 1976). In the simplest form, orthogonal Procrustes analysis (OPA) is an MDS technique in which one configuration (testee) is matched to another configuration (target) of the same dimension. The two configurations are optimally aligned by applying admissible transformations in order to maintain the ratio of the distances between the plotted points. The admissible transformations consist of rotation, dilation, reflection and translation of the testee configuration until the sum of squared errors is minimised (Borg & Groenen, 2005; Gower & Dijksterhuis, 2004; Ten Berge, 1977).

Suppose the target configuration is referred to as the matrix, $\mathbf{Y}$, and the testee configuration is referred to as the matrix, $\mathbf{X}$. The aim is to minimise the sum of the squared distances between the transformed (or updated) testee configuration, $\mathbf{X}^{*}$, and the target (Van Ginkel & Kroonenberg, 2014):

$$SS = \|s\mathbf{XQ} - \mathbf{Y}\|^{2},$$

where	s	→	dilation factor,
	$\mathbf{X}$	→	testee configuration,
	$\mathbf{Q}$	→	orthogonal rotation matrix and
	$\mathbf{Y}$	→	target configuration.

The following steps can be used to optimally align the testee configuration with the target configuration (Borg & Groenen, 2005; Cox & Cox, 2001; Gower & Dijksterhuis, 2004; Sibson, 1978; Ten Berge, 1977):

a) Obtain the SVD of $\mathbf{Y}'\mathbf{X}$, where $\mathbf{Y}'\mathbf{X} = \mathbf{U\Lambda V}'$.

b) Obtain the orthogonal rotation matrix, $\mathbf{Q}$, from the SVD in (a): $\mathbf{Q} = \mathbf{VU}'$.

c) Obtain the dilation factor, s, by $s = \frac{tr(\mathbf{Y}'\mathbf{XQ})}{tr(\mathbf{Q}'\mathbf{X}'\mathbf{XQ})} = \frac{tr(\mathbf{Y}'\mathbf{XQ})}{tr(\mathbf{X}'\mathbf{X})} = \frac{tr(\mathbf{XQY}')}{tr(\mathbf{X}'\mathbf{X})}$.

d) Obtain the translation factor, $\mathbf{b}$, by $\mathbf{b} = \frac{1}{n}(\mathbf{Y} - s\mathbf{XQ})'\mathbf{1}$.

The translation step may be disregarded by mean centring the columns of the matrices prior to the Procrustes analysis.

The updated testee configuration is obtained from:

$$\mathbf{X}^* = \mathbf{b} + s\mathbf{XQ}.$$

Biplots are not confined to a specific orientation and can therefore be transformed by Procrustes analysis to ease the comparison of multiple displays (Blasius, Eilers & Gower, 2009).

2.9.1 Generalised orthogonal Procrustes analysis

Generalised orthogonal Procrustes analysis (GPA) allows the comparison of multiple configurations by iteratively comparing each configuration to a target configuration. The target configuration is updated on each iteration until the distances between the multiple configurations and the current target are minimised (Gower & Dijksterhuis, 2004).

The initial target configuration is typically set to a multidimensional average that represents each configuration. The target configuration is referred to as the centroid configuration in this research.

The notation of Section 2.9 will be adopted with the exception of the target configuration (centroid configuration) now expressed as the coordinate matrix, $\mathbf{C}$.

The iterative GPA procedure minimizes the following sum of squares (SS):

$$SS = \sum_{k=1}^{K} \|\mathbf{s}_k \mathbf{X}_k \mathbf{Q}_k - \mathbf{C}\|^2,$$

where k → number of configurations

$\mathbf{s}_k$ → isotropic dilation factor for the k^{th} configuration,

$\mathbf{X}_k$ → coordinate matrix of the k^{th} configuration,

$\mathbf{Q}_k$ → optimal orthogonal rotation matrix for the k^{th} configuration and

$\mathbf{C}$ → coordinate matrix for the current centroid configuration.

The centroid configuration is obtained from the current setting of the coordinate matrices of the k configurations:

$$\mathbf{C} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{s}_k \mathbf{X}_k \mathbf{Q}_k.$$

The initial centroid configuration, $\mathbf{C}^0$, will be the mean coordinates of the original configurations prior to transformations:

$$\mathbf{C}^0 = \frac{1}{K} \sum_{k=1}^{K} \mathbf{X}_k.$$

Again, the translation transformation is incorporated by centring the configurations before the Procrustes analysis.

The algorithm for GPA with isotropic scaling as presented by Gower and Dijksterhuis (2004) is applied and briefly summarised in this section.

A constraint is imposed on the overall size before and after scaling to avoid a trivial solution of the scaling factor, $\mathbf{s}_k = 0$:

$$\sum_{k=1}^{K} \|\mathbf{X}_k\|^2 = \sum_{k=1}^{K} \|\mathbf{s}_k \mathbf{X}_k \mathbf{Q}_k\|^2.$$

The multiple configurations are mean centred before applying the GPA algorithm, therefore the translation transformation is negligible. All configurations are aligned to the current centroid configuration using OPA (cf. 2.9 (a) and (b)). The final step of the current iteration is to obtain the isotropic scaling factor from the following calculations:

a) Obtain the symmetric matrix, $\mathbf{S}$, of the sum of squares of $\mathbf{XQ}$, where $\mathbf{S} = tr(\mathbf{Q}'\mathbf{X}'\mathbf{XQ})$.

b) The eigenvector associated with the largest eigenvalue, the principal eigenvector, of the normalised matrix $\mathbf{S}$ is used to estimate the isotropic scaling factor, $\mathbf{s}_k$.

The GPA algorithm converges when the difference between the current and previous sum of squares ($\sum_{k=1}^{K} \|\mathbf{s}_k \mathbf{X}_k \mathbf{Q}_k - \mathbf{C}\|^2$) reaches a predetermined threshold of at most 0.001, for example.

2.10 Conclusion

This concludes the literature review on two of the core methodologies for this research: categorical data analysis and visualisation. The discussions on MCA (cf. 2.7) and MCA biplots (cf. 2.7.1) are of importance to achieve the aim of this research.

The first and third study objectives (cf. 1.4.1 and cf. 1.4.3) also rely on OPA (cf. 2.9) and its generalisation, GPA (cf. 2.9.1). Discussions on MI will be presented in the subsequent chapter (cf. Chapter 3). Detail on the specific MI technique applied in this research is given in Chapter 4 (cf. 4.2.2).

The second and fourth study objectives (cf. 1.4.2 and cf. 1.4.4) rely on a variant of MCA, sMCA. The methodology on sMCA will be discussed in Chapter 4 (cf. 4.5). The fourth study objective (cf. 1.4.4) is addressed by the addition of clustering techniques, which will be presented in Chapter 4 (cf. 4.7).

In order to compare complete and completed configurations of the first three study objectives (cf. 1.4), OPA (cf. 2.9) is applied. Measures of comparison (cf. 4.6.1) within the Procrustes framework will be presented in the methodology chapter of this research (cf. Chapter 4).

A review on missing data terminology and techniques is presented in the following Chapter 3.

Chapter 3
Missing data

3.1 Introduction

It is true that missing data could lead to much frustration for an analyst, but not only is the burden due to a loss of information, but also the use of ambiguous terminology. Commonly used terms will be defined to clarify the conventions used in this research. Subtle differences can be defined for the terms missing data, incomplete data and non-response. Missing data is regarded as the group name that encapsulates the terms incomplete data and non-response, which refer to types of missing data.

Since missing data techniques were introduced in survey applications, the different types of non-response will first be defined using survey analysis terminology. Non-responses are also known as errors of non-observation in survey analysis. There are three types of non-response errors: (1) unit non-response, (2) item non-response and (3) incomplete responses. The survey analysis terminology might cause additional confusion when used in the context of biplot methodology. The terminology used for biplot methodology will be enforced throughout the dissertation, therefore the term, unit (survey analysis), is referred to as a sample and the term, item (survey analysis), is referred to as an observation (element of a data matrix).

Thus, if no measurements are recorded for a sample, it is referred to as unit non-response. In the case where only some measurements are unobserved, it is referred to as item non-response (Biemer & Lyberg, 2003; Schafer & Graham, 2002). Item non-response can also occur due to deficient data, which refers to the removal of unrealistic or incorrectly processed observations before analysis (Rubin, 1987). Incomplete responses refer to measurements that are recorded, but do not provide sufficient information. This is a common non-response error for open questions in which individuals have to answer questions without set parameters (Biemer & Lyberg, 2003; Schafer & Graham, 2002).

It is common to refer to either an incomplete sample or an incomplete variable when some missing responses occur in a data set. This suits the definition of incomplete non-response in the sense that the information per sample or variable is not adequate. Furthermore, missing data are frequently referred to as incomplete data, again fulfilling the requirement of the given definition that inadequate information is available. In general, a non-response is used to refer to a missing value when missingness occurs due to an unknown reason. It does not necessarily refer to the physical disregarding of a question by a respondent.

Other counterintuitive terms are the referral to the words 'random' and 'ignorable'. This has been noted by numerous authors (Collins, Schafer & Kam, 2001; Schafer & Graham, 2002; Van Buuren, 2012). It should be emphasised, that when referring to randomness in an MDM, it does not necessarily imply that the missingness is due to an unrelated cause. Also, ignorable MDMs do not imply that the cause of missingness can be completely disregarded. This chapter addresses these terms to clarify common misconceptions.

The methodology presented in this research aims to develop solutions to item non-responses not only in survey data, but categorical data in general. Item non-response could be due to the type and content of questions, as well as the questionnaire type (electronic, paper, interview). According to Biemer and Lyberg (2003) the following reasons could cause item non-response:

- sensitive or difficult questions which are commonly ignored by respondents,
- lengthy questionnaires with complicated questions could also discourage respondents to complete all questions,
- open questions frequently result in non-response and
- if a questionnaire is completed by an interviewer on behalf of a respondent, human error could result in responses that have to be removed during data editing.

Rubin (1987) mentions three general problems experienced with non-response, especially in the context of surveys: (1) less efficient estimates due to the decreased sample size, (2) standard techniques for complete data cannot be used for analysis and (3) it is intuitive to consider that there is a systematic difference between respondents and non-respondents which causes bias to exist. The problem of bias cannot be easily resolved, since the cause of missingness (cf. 3.2) is not always evident.

Furthermore, a distinction can be made between intentional and unintentional missing data. Examples of intentional non-responses include sampling from finite populations and randomisation of experiments (Rubin, 1976). Intentional non-response implies that the unobserved information was planned, as opposed to unintentional non-responses that are expected but uncontrolled (Van Buuren, 2012).

Two goals for the handling of missing data are set by Rubin (1987): reconstruct the data to be able to perform standard complete data techniques and in order to obtain unbiased inferences, the handling technique should successfully adjust for the differences between the samples with observed and samples with unobserved responses.

This chapter will firstly address the structure of missing data which will be followed by different approaches to handle missing data.

3.2 Missing data mechanisms

The type of missing data, as discussed in the introduction (cf. 3.1), differs from the cause of missingness. The cause of missingness can be explained as the result of a random process referred to as the MDM (Van Buuren, 2012). Rubin (1976) was the first to classify the MDMs by exploring when correct inferences could be obtained by ignoring the cause of missingness. Before his pioneering article (Rubin, 1976) it was common practice to ignore the cause of missing data where unintentional non-responses were observed. The general assumption was made that all measurements in a data set had the same probability of being missing. Rubin based the classification of the MDMs upon the probability of a data entry to be unobserved.

Consider a hypothetical example in which a complete data set is generated, $\mathbf{X}_{complete}$, with n samples and p variables, where $i = 1, \dots, n$ and $j = 1, \dots, p$. Missing values are inserted in $\mathbf{X}_{complete}$ which leads to two subsets: observed responses, $\mathbf{X}_{observed}$, and missing responses, $\mathbf{X}_{missing}$.

In order to mathematically define the missingness, a matrix, $\mathbf{R}$, is constructed with the same dimensions ($n \times p$) as the data matrix, $\mathbf{X}$, of interest. The $\mathbf{R}$ matrix is coded with zeros and ones to indicate whether observations are missing (1) or observed (0) in the data matrix (Schafer, 2003; Schafer & Graham, 2002; Van Buuren, 2012; Zhang, 2003):

$$r_{ij} = \begin{cases} 0, & when\ x_{ij}\ is\ observed \\ 1, & when\ x_{ij}\ is\ missing. \end{cases}$$

The pattern of $\mathbf{R}$ provides insight to the cause of missingness and the relationship between the observed ($\mathbf{X}_{observed}$) and missing ($\mathbf{X}_{missing}$) subsets in the data matrix. The observations of $\mathbf{R}$ are conditional on the complete matrix which is denoted by: $\mathrm{P}\left(\mathbf{R}\middle|\mathbf{X}_{complete}\right) = \mathrm{P}(\mathbf{R}|\mathbf{X}_{observed}, \mathbf{X}_{missing}, \phi)$, where ϕ represents the parameters of the MDM.

The distribution of missingness, the MDM, is defined by the probability of missingness conditional on the observed and missing components of the data matrix: $\mathrm{P}\left(\mathbf{R} = 1\middle|\mathbf{X}_{observed}, \mathbf{X}_{missing}, \phi\right)$.

The MDM is MCAR if the following condition holds:

$$\mathrm{P}\left(\mathbf{R} = 1\middle|\mathbf{X}_{observed}, \mathbf{X}_{missing}, \phi\right) = \mathrm{P}(\mathbf{R} = 1|\phi).$$

The probability of missingness is thus independent of the observed and / or missing observations. This can also be explained as the missing values that are not conditional on observed values in the data. Data entries have the same probability to be missing or observed. Thus, the distributions of missingness and observed responses are expected to be the same. The MCAR MDM can be regarded as a simple random sample of the complete data set (Van Buuren, 2012; Zhang, 2003). This is a convenient classification since inference obtained from this MDM is regarded as valid and unbiased, since the observed subset of the data is expected to be a representative sample of the population of interest (Hron, Templ & Filzmoser, 2010; Little & Rubin, 2002).

As an example of MCAR, suppose that the eating habits of people are investigated and information is collected in the form of a survey. Some questions in the survey refer to food products that are not familiar to the respondent (e.g. Kimchi, Blatjang, Miso, etc.) and leads to the omission of questions. The missingness is not related to other questions in the survey and also not dependent on information that is not captured by the survey. The MCAR MDM is also easily understood by missing values due to a loss of interest of participants or the loss of information due to a computer / storage problem.

The MDM is MAR if the following condition holds:

$$\mathrm{P}\left(\mathbf{R} = 1\middle|\mathbf{X}_{observed}, \mathbf{X}_{missing}, \phi\right) = \mathrm{P}(\mathbf{R} = 1|\mathbf{X}_{observed}, \phi).$$

Now, the probability of missingness is independent of the missing observations, but dependent on known responses. Here, the terminology of missing data analysis can lead to confusion. A further explanation of the MAR MDM is that the missing data are conditional on certain observed responses. It is proposed by Graham (2009) to rather refer to MAR as conditionally missing at random, this is however not in common use since it is easily confused with the existing MCAR MDM. The MAR MDM is not as restrictive as the MCAR MDM and the missing responses can be viewed as a simple random sample taken from specific subgroups defined by the observed information (Zhang, 2003). These subgroups are based on specific responses and therefore result in a dependency between observed and missing data. The MAR MDM can be explained by the deliberate omission of sensitive questions that might be related to answered questions in the same questionnaire (García-Laencina *et al.*, 2010; Schafer & Olsen, 1998). The MAR MDM provides a general explanation for the cause of missingness and is assumed for most missing data conundrums. Assuming a MAR MDM is plausible, since it implies that the missing responses per sample (individual) will vary from one sample to the next, which means that missingness depends on the observed information within each sample (Schafer & Graham, 2002).

Consider the eating habit survey example again, one of the variables allows options for typical food allergies. An example of MAR is if a participant indicated that he / she has fish allergies and has a missing response for a subsequent question regarding preferred fish meals. Thus, this missing value is conditional on a known response in the survey.

The MNAR MDM refers to non-ignorable missingness and can be expressed by the following probability:

$$\mathrm{P}\left(\mathbf{R} = 1 \middle| \mathbf{X}_{observed}, \mathbf{X}_{missing}, \phi\right).$$

This condition does not simplify as the previous cases and therefore implies that the probability of missingness depends on both the observed- and missing data.

Missing observations that are unobserved due to the MNAR mechanism are dependent on the information that was not captured by the data, therefore dependent on unknown and unmeasured causes. This occurs when questions in the questionnaire could be related to questions that are not asked, thus not captured or considered by the particular questionnaire.

Again, consider the eating habit survey example, suppose that the survey does not include a question to indicate dietary requirements, such as vegetarian or vegan. Vegetarians will for example leave out the questions related to meat products, but will still answer questions related to animal by-products, such as dairy products, eggs, etc. Vegan participants will however leave out all questions related to animal products. Missing responses from these individuals are missing due to different reasons which are not captured by the survey, which is an example of MNAR.

A further distinction can be made between ignorable and non-ignorable MDMs. The MAR and MCAR mechanisms are labelled as ignorable MDMs, since the distributions of missingness do not depend on the missing information. Hence, the observed information is adequate to handle the effect of the missingness in the data. However, the MNAR MDM relies on unobserved information and therefore the available information will not be sufficient for the handling of missing values and continued analysis. The MNAR MDM is classified as non-ignorable or informative (Collins *et al.*, 2001; Van Buuren, 2012). Ignorable MDMs are therefore plausible when the available information provided by the observed data succeeds in explaining the tendency of missing responses (Schafer, 1997).

A few lesser known classifications of missing data are also recorded in the literature, which are briefly discussed here. Missing not categorisable is a classification for missing values that are not 'really' missing, which occurs when responses are omitted due to the respondents not knowing the answer. Consequently, the reason for missingness is unknown and cannot be estimated by observed information in the data. It is proposed to apply specific coding techniques, such as active handling (cf. 3.4.2.2), to missing values that are not categorisable. This definition is found to be confounded with the definition of MNAR, since observations that are missing due to the MNAR MDM also depend on information that is not captured by the survey. It can be argued that there are subtle differences, since the MNAR MDM defines missingness due to questions that are not stated, whereas the non-categorisable missing values are due to the participant not knowing or understanding the question. From an analyst's perspective the true difference will not be intuitive and since this research aims at developing methodology for ignorable non-responses, this will not be investigated further. Another classification is missing values that are deliberately created, referred to as created missing. This is the deletion of certain responses that are not of interest or variables with low frequency that could result in outliers.

The aim of this research is to obtain unbiased representations of missing data and this cannot be achieved by the deletion of certain responses (Van der Heijden & Escofier, 2003).

3.3 Missing data patterns

Investigating the pattern of non-response in the form of a missingness map enables the quick identification of samples and variables burdened with non-response. It could also be insightful and a first step to ponder on the possible cause of missingness.

Three patterns are described in the literature (Schafer & Graham, 2002):

- Univariate: Non-response only occurs in one of the observed variables.
- Monotone: Variables can be ordered in increasing order of non-response. Fully observed variables or variables consisting of a smaller proportion of missing observations are considered first and variables with increasing proportions of missing observations are considered thereafter. This pattern is typically observed in longitudinal studies in which patients fall out towards the end of the study.
- Arbitrary: No specific pattern can be deduced from the non-response.

Figure 3.1 illustrates the missing data patterns for ten variables and 100 samples using the `missmap()` function in the `R` package, `Amelia` (Honaker, King & Blackwell, 2011; R Core Team, 2017):

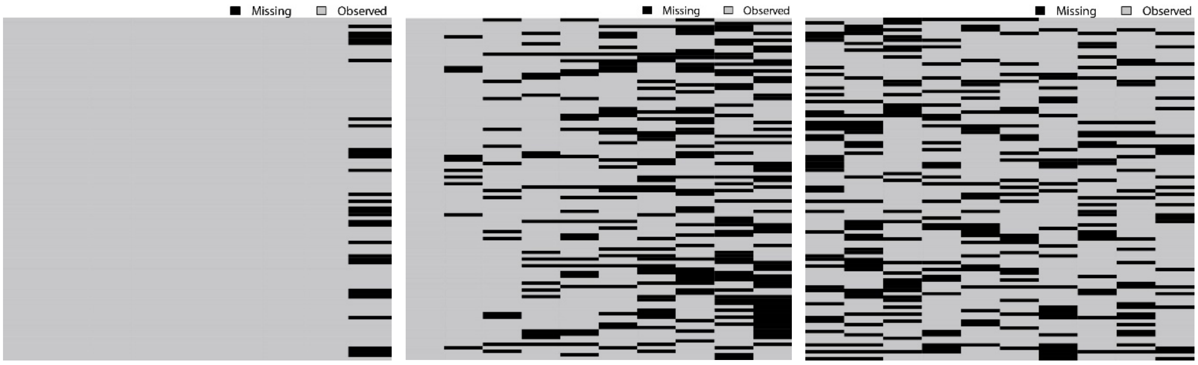


Figure 3.1 Missing data patterns. Left panel: Univariate pattern. Middle panel: Monotone pattern. Right panel: Arbitrary pattern.

Additional missing data patterns have been defined more recently which focus on the distribution of the missing data. Wang and Wang (2007) define three missingness patterns: (1) missing at random, (2) uneven symmetric missing and (3) uneven asymmetric missing. The name of the first pattern is unfortunate and confusing, since it does not refer to the well documented MAR MDM. Wang and Wang's (2007) missing at random pattern refers to a random occurrence of missing values which is rather associated to the MCAR MDM. The second pattern, uneven symmetric missing, refers to missing values that occur more frequently in certain variables and it is expected that variables with missing values are correlated. The third pattern, uneven asymmetric missing, identifies missing values that occur in numerous variables, but could be due to a specific class (or grouping) of samples in the data.

Fernstad (2019) developed three new patterns to better address classification of missing data in multivariate data: (1) amount missing, (2) joint missingness and (3) conditional missingness. The first, amount missing, records the number of missing values per sample or variable. If the missing values occur relatively evenly across variables, it can be an indication of the MCAR MDM. The second, joint missingness, refers to missing values that occur simultaneously for the same sample. This could be due to questions in a survey that are directly related, which could reflect highly correlated variables. This pattern will provide evidence against the MCAR MDM and perhaps provide reason to believe that the missingness is due to the MNAR MDM. The latter suggestion is plausible if variables are correlated with information that is not captured by the survey. The third, conditional missingness, again refers to the relationship between two or more variables, but now the missingness in one variable will be due to the response in another variable. This is related to the interpretation of the MAR MDM, since missing values are dependent on certain responses and therefore dependent on the observed information in the data.

The missing data patterns presented in the missingness maps in Figure 3.1 do not enable the investigation of the interrelationships of variables which could be indicative of the MDM. Therefore, the use of clustering techniques for multivariate configurations are investigated for the identification of MDMs in Chapter 8.

3.4 Handling techniques for missing data

Missing data are regarded as an obstacle, since most analysis techniques are developed for complete data sets. It is however crucial to understand the type and cause of non-response before attempting to impute and analyse data. Each data set measures unique characteristics, therefore care must be taken to find suitable techniques to handle each missing data problem properly. Handling missing values is a two-fold process in which the missing data entries are first attended to before a standard analysis can be applied. Also, there are two different agendas when handling missing values. If emphasis is placed on extracting information from the missingness itself, which means it is of importance to follow and understand the missing observations in the analysis, coding techniques are applied. Whereas, if the focus is on obtaining unbiased inference from the completed data, the exploratory analysis is not aimed at understanding the missingness and therefore no information is extracted from the missing data (Van der Heijden & Escofier, 2003). Both these agendas are followed in this research, but used to address different missing data problems. The first, second and third study objectives focus on the final inference obtained from completed or handled missing data. Thus, emphasis is not placed on the extraction of the missingness itself. The fourth study objective focusses on the structure of missing data in order to identify the cause of missingness, which is achieved by extracting information form the missing observations.

The following sections will discuss typical handling protocols for missing data.

3.4.1 Deletion

Deletion is the most straightforward method of handling missing data. The missing values are ignored without any reconstruction or updating of the data matrix. In the R programming environment these missing responses will be indicated by 'NA' (R Core Team, 2017). Since the missing values are not attended to and cannot contribute to the analysis, deletion techniques are applied. Consider the toy data set (cf. Table 2.1), now containing missing values in Table 3.1:

Table 3.1 Toy data set with missing values

	Variable 1	Variable 2
Sample 1	1	2
Sample 2	3	1
Sample 3	1	NA
Sample 4	NA	NA

Should the missing responses from the toy data set in Table 3.1 be handled as passive, the indicator matrix, $\mathbf{G}$, will be presented by the following (missing observations indicated in **bold** font):

$$\mathbf{G} = \begin{bmatrix} \mathbf{V1{:}1} & \mathbf{V1{:}2} & \mathbf{V1{:}3} & \mathbf{V2{:}1} & \mathbf{V2{:}2} \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Deletion methods disregard samples comprising of non-response, which is an easy solution to the handling of missing values, but it is not encouraged if the distribution of the unobserved data differs from the distribution of the observed data (Penn, 2007). The only instances in which deletion methods may result in unbiased inference are when the percentage of missing values is small and for MCAR scenarios where all samples have the same probability of being observed (or unobserved) (Raghunathan, Lepkowski, Van Hoewyk & Solenberger, 2001; Schafer, 1997; Van Buuren, 2012).

Complete-case analysis, also referred to as listwise deletion or case deletion, is the default approach to handle non-response in most statistical software. This approach merely deletes samples containing non-response. Case deletion is considered to be valid for MCAR MDMs and also when the MAR MDM follows a univariate non-response pattern (Schafer & Graham, 2002). When the complete data set is available, for example in a simulation study, the MCAR MDM can be confirmed by comparing the complete-case analysis results with the simulated results.

Available-case analysis, also referred to as pairwise deletion, uses sets of available information within samples. Available-case analysis preserves more observed information, but statistics across variables are not directly comparable due to the difference in available samples per variable. It is impossible to know whether the remainder of samples after deletion is representative of the population. Deletion methods are therefore expected to result in biased and inefficient inference, especially when the percentage of missing values is high (Josse & Reiter, 2018; Schafer & Graham, 2002).

3.4.2 Data reconstruction

3.4.2.1 Missing passive modified margin

The deletion methods (cf. 3.4.1) can also be referred to as passive missing data handling, since the available observed information plays an active role in the analysis and not the missing values (Van der Heijden & Escofier, 2003). When applying, for example, available-case analysis prior to MCA, the problem could occur that the row totals of the indicator matrix will not all be equal to the number of variables. This problem is resolved by the missing passive modified margin method, which fixes the row margins of all samples to the number of variables in order to satisfy the requirements for MCA. The non-responses are still indicated by zeros in the indicator matrix and therefore the missing observations are skipped and play a passive role in the analysis (Josse, Chavent, Liquet & Husson, 2012).

3.4.2.2 Active handling

Active data handling, in the context of multivariate categorical data analysis, consists of adding a CL for missing responses before standard multivariate data analysis procedures are applied. The missing data entries are therefore regarded as responses to an unknown CL (Josse *et al.*, 2012). The addition of missing CLs ensures that the row margins are equal to the number of variables (Van de Geer, 1993). This is a popular approach, since standard complete data techniques, such as MCA, can be applied with the addition of extra CLs for missing observations which enables missing values to play an active role in the analysis.

There are two approaches to apply active handling to non-responses: single- and multiple active data handling. It is regarded as single active handling when a missing CL per variable is created, since all missing entries irrespective of the particular sample are pooled together. A drawback is that when allocating the same missing CL for samples it is assumed that the samples with non-responses are similar, which could be misrepresentative of the population in question. It does however allow easy separation of samples with missing values and those without. Caution should be taken when using the missing active categories for further analysis, since it is to be expected that these CLs will have low frequency, which might dominate the solution in the form of outliers (Nishisato, 1980; Van der Heijden & Escofier, 2003).

Multiple active handling addresses the problem of single active handling, by creating a unique missing CL for each individual with a non-response per variable. The additional number of CLs will be equal to the number of missing values in the data set (Michailidis & De Leeuw, 1998). This technique has the disadvantage of creating a large number of new CLs with low frequency causing possible outliers, but it has the advantage of not assuming similarities between samples comprising of non-response. The decision of single- and multiple active data handling is based upon the data and the research aim (Van de Geer, 1993; Van der Heijden & Escofier, 2003).

Consider again the toy data set with missing values (cf. Table 3.1). Single active handling of the missing responses will be reflected in an additional missing CL (e.g. missing responses for the first variable, 'V1:?') for each variable with missing responses (missing observations indicated in **bold** font):

$$\mathbf{G} = \left[\begin{array}{ccccccc} \mathbf{V1:1} & \mathbf{V1:2} & \mathbf{V1:3} & \mathbf{V1:?} & \mathbf{V2:1} & \mathbf{V2:2} & \mathbf{V2:?} \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \end{array}\right]$$

Multiple active handling of the unobserved responses will be reflected in an additional CL for each sample (e.g. missing response for the first variable of the fourth sample, 'V1:s4?') with a missing response (missing observations indicated in **bold** font):

$$\mathbf{G} = \left[\begin{array}{ccccccccc} \mathbf{V1:1} & \mathbf{V1:2} & \mathbf{V1:3} & \mathbf{V1:s4?} & \mathbf{V2:1} & \mathbf{V2:2} & \mathbf{V2:3} & \mathbf{V2:s3?} & \mathbf{V2:s4?} \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \end{array}\right]$$

When the indicator matrix is coded with dummy variables, responses of zeros and ones, this is referred to as crisp coding. Another approach is to assign continuous values between zero and one that expresses the probability of a CL to be chosen, this is referred to as fuzzy coding and will be further discussed in Section 3.4.2.3 (Michailidis & De Leeuw, 1998).

3.4.2.3 Fuzzy coding

The dimensions of the data matrix can be preserved by inserting probabilities for the CLs of missing responses. Even though this is regarded as an SI method which will be discussed in a subsequent section (cf. 3.4.4.1), it is presented as part of the data reconstruction approaches. Here the term, fuzzy, refers to a continuous value between zero and one that is an element of an indicator matrix. The row margins for the CLs per variable have to add up to one to satisfy the requirements for multivariate techniques, such as MCA. In the case where no response is recorded for a variable, fuzzy values are allocated to each CL for this specific variable which will result in a row margin of one for the particular variable.

The missing fuzzy average method utilises the observed information for each variable and allocates the fuzzy values based on the proportions of the CLs that are observed across all samples (Van der Heijden & Escofier, 2003). Again, consider the responses of the toy data set (cf. Table 3.1). Variable 1: two responses (CL 1), one response (CL 3), one missing response and Variable 2: one response (CL 1), one response (CL 2), two missing responses.

The proportions of the CLs across the observed samples are used for the substitution:

$$\mathbf{G} = \left[\begin{array}{ccccc} \mathbf{V1:1} & \mathbf{V1:2} & \mathbf{V1:3} & \mathbf{V2:1} & \mathbf{V2:2} \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & \mathbf{1/2} & \mathbf{1/2} \\ \mathbf{2/3} & \mathbf{0/3} & \mathbf{1/3} & \mathbf{1/2} & \mathbf{1/2} \end{array}\right]$$

The missing fuzzy subgroup method relies on additional information of the samples. The fuzzy values are determined within a variable according to a certain class (or subgroup) of samples (Van der Heijden & Escofier, 2003). To illustrate this, the toy data set with missing values (cf. Table 3.1) is now adapted to include a sex variable in Table 3.2:

Table 3.2 Adapted toy data set with missing values

	Sex	Variable 1	Variable 2
Sample 1	Male	1	2
Sample 2	Male	3	1
Sample 3	Female	1	NA
Sample 4	Male	NA	NA

The indicator matrix after application of the missing fuzzy subgroup method is as follows:

$$\mathbf{G} = \left[\begin{array}{cccccc} \mathbf{V1:1} & \mathbf{V1:2} & \mathbf{V1:3} & \mathbf{V2:1} & \mathbf{V2:2} & \mathbf{V2:3} \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & \mathbf{1/3} & \mathbf{1/3} & \mathbf{1/3} \\ \mathbf{1/2} & \mathbf{0/2} & \mathbf{1/2} & \mathbf{1/2} & \mathbf{1/2} & \mathbf{0/2} \end{array}\right]$$

Since no responses are recorded for female respondents for the second variable in the adapted toy data set (cf. Table 3.2), the probability of the three CLs (cf. Table 2.1) are the same. The two observed responses for the first variable for male participants differ (CL 1 and CL 3). Therefore, the probability of the missing CLs are the same for the two observed CLs, as is the case with the fuzzy coding of the second variable for male respondents.

In the presence of additional observed descriptive information, the missing fuzzy subgroup coding method is the preferred choice as it presents unbiased plausible response probabilities in accordance with the available information.

3.4.3 Weighting

Weighting methods are commonly used to handle unit non-response (Van Buuren, 2012). The technique of weighting is popular in survey methodology and was also proposed for the handling of non-response in regression applications in the 1990s (Schafer & Graham, 2002). The goal in weighting is to emulate the distribution of the full population by assigning weights to the observed samples that relate to the response probabilities after deletion methods have been applied. Weighting is accordingly regarded as a correcting method to restore the distribution after deletion. If the underlying assumptions of the weighting scheme holds, it could lead to a decrease in the non-response bias (Biemer & Lyberg, 2003).

3.4.4 Imputation

Imputation is the process of completing a data set to its original dimensions by replacing missing observations with plausible values (Little & Rubin, 2002). Imputation techniques are typically used to handle item non-response (Van Buuren, 2012). Once the missing observations have been attended to and the data set has been reconstructed to its original dimensions, the updated data set is referred to as the completed data.

A variety of imputation techniques have been established with new additions emerging for specific applications. Sections 3.4.4.1 (SI) and 3.4.4.2 (MI) will elaborate on the two main streams of imputation.

3.4.4.1 Single imputation

A single value is imputed per missing observation which results in a completed data set that can be used for a standard analysis. The SI approach assumes that the data are fully observed and discards the fact that the imputed values are in a sense a sample from what could have been observed in the population. This leads to underestimation of the variance and biased estimates, since the uncertainty is not incorporated in the final estimates (Rubin & Schenker, 1986; Tang *et al.*, 2012).

The most popular SI method is mean substitution for continuous data (Ambler, Omar & Royston, 2007). The mean of each variable is calculated using the observed responses and is used to impute the missing data entries per variable. This technique could provide unbiased estimates for the mean under the MCAR MDM, but would underestimate the variance and not reflect the true correlations between the variables (Schafer & Graham, 2002; Van Buuren, 2012). In the case of categorical variables, the mode can be used instead of the mean as described in mean substitution. An alternative to mean substitution that attempts to preserve the distribution of the observed samples, is hot-deck imputation. This technique substitutes missing responses with randomly selected observed responses from samples with similar response patterns. The disadvantage of this approach is that it is assumed that there is no difference between respondents and non-respondents in the questionnaire (Buhi *et al.*, 2008). Similar to hot-deck imputation, nearest-neighbour imputation makes use of distance measures between samples and observed responses to determine the similarity between samples. The distances between samples according to observed responses are used to identify similar samples. Samples that are in close proximity are considered to be similar. Missing responses in one sample are then imputed by using observed responses from a sample that is most alike (Biemer & Lyberg, 2003).

3.4.4.2 Multiple imputation

The MI approach is the process of repeatedly replacing missing responses with a plausible value until a predetermined number of data sets (m) have been completed. It can be regarded as a three step process: (1) impute multiple plausible response values to create multiple completed data sets, (2) perform separate standard complete data analysis on the completed data sets and (3) combine the results obtained from the standard complete data analysis into a single inferential statement (Tang *et al.*, 2012; Van Buuren, 2012; Zhang, 2003).

The steps of the MI procedure for three MIs are illustrated by means of the flowchart in Figure 3.2, adapted from Van Buuren (2012):

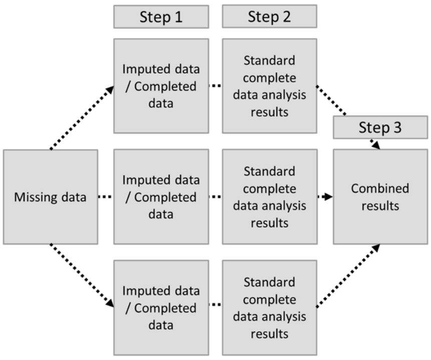

Step 1
Step 2
Step 3
Missing data
Imputed data / Completed data
Imputed data / Completed data
Imputed data / Completed data
Standard complete data analysis results
Standard complete data analysis results
Standard complete data analysis results
Combined results

Figure 3.2 Illustration of the steps of MI

The method of MI was initially developed to assist in the analysis of large public-use survey data. The database constructers and data users were independent. The data preparation consisting of imputation was performed by the database constructers and then made available to the users. Fortunately, MI has been further developed and is widely applied as a preferred handling technique for missing data (Rubin, 1996; Van Buuren, 2012).

Multivariate data are typically imputed according to two approaches: joint modelling (JM) or fully conditional specification (FCS) (Van Buuren, Brand, Groothuis-Oudshoorn & Rubin, 2006). A JM approach assumes that a multivariate distribution can describe the behaviour of the data. A joint model is used for all variables with the same response patterns. Samples are imputed by drawing imputations from a suitable distribution according to the missing data pattern of the particular samples (Van Buuren, 2012). The multivariate normal distribution is typically used for the JM imputation approach and assumes that all variables follow a joint multivariate normal distribution. The multivariate normal model can also be used for categorical data by rounding the imputed values to the nearest CL (Van Buuren, 2012). The loglinear model assumes a joint multinomial distribution for all variables and is the preferred approach for small categorical data sets (Audigier, Husson & Josse, 2017). Another option for categorical data is the Dirichlet process mixture of products of multinomial distributions using a latent class model (Akande, Li & Reiter, 2017). The MI procedure that is applied in this research is a JM approach for the MI of the indicator matrix of a multivariate categorical data set (cf. 4.2.2).

The FCS approach is also known as sequential regressions and chained equations methods (Van Buuren, 2012). These methods impute the missing values variable-by-variable while conditioning on all remaining variables. The imputed variable is therefore regarded as the response variable while the remaining variables are treated as the explanatory variables (Kowarik & Templ, 2016). The variables are imputed sequentially by utilising different multiple regression models (linear or logistic) depending on the type of variables considered. Thus, the FCS approach is more flexible and enables the use of varying imputation models (Raghunathan *et al.*, 2001). Random forests and classification and regression trees (CART) can also be applied as FCS methods. If the response variable is categorical, the CART approach refers to classification trees and when the response variable is continuous, it refers to regression trees (Doove, Van Buuren & Dusseldorp, 2014).

It is important to be mindful of the fact that the imputations are not applied with the aim of obtaining one final completed data set, but rather to preserve the characteristics of the population while incorporating the uncertainty of imputation (Lall, 2016; Schafer & Graham, 2002; Wayman, 2003). The MI techniques can be regarded as multiple sampling procedures in which the multiple plausible imputed values form a distribution of possible responses. This incorporates the uncertainty of the actual guesswork involved when imputing missing observations (Rubin, 1987; Rubin & Schenker, 1986; Tang *et al.*, 2012).

The drawback of SI in comparison to MI, is that the uncertainty of the imputation procedure is not reflected in the completed data set. When combining the results from the standard complete data analysis after MI, the combined variance includes both the between- and within-imputation variance. This results in a realistic variation which captures the uncertainty, whereas the standard errors obtained from SI contain no between-imputation variation and lead to underestimation of the variance (Zhang, 2003).

In order to successfully incorporate the uncertainty in the completed data, MI procedures should capture the following uncertainty (Rubin, 2003; Zhang, 2003):

(1) Uncertainty in selecting the correct MDM which can also be explained as the uncertainty of identifying the correct joint distribution for missing and observed data.

(2) Uncertainty in selecting the correct parameters for the chosen imputation model, therefore general uncertainty in the imputation model.

(3) Residual uncertainty when drawing the final imputed values from the distribution of possibilities.

According to Rubin (2003), the focus should be on constructing an MI procedure that preserves the associations among samples across variables under the assumption of an ignorable MDM. Thereafter, the MI procedure can be extended to non-ignorable non-responses, if necessary. There are no tests to confirm the MDM with certainty, since information regarding the missing values are typically not available (Schafer & Olsen, 1998). Schafer (1997) remarks that a majority of missing data handling techniques rely on some assumption of ignorability. Approaches that make use of the observed information for the handling of missing data entries are referred to as general ignorable procedures. The MNAR MDM is a plausible assumption for real data, since missingness could be dependent on both the observed and missing components of the data. Therefore, MI will result in biased inference when the missing data are strongly dependent on the information that cannot be retrieved. Hypothetically, if the proportion of missing values that is related to the observed data is larger than the proportion of missing values related to the unobserved data, MI could still be an appropriate handling technique for MNAR missingness (Lall, 2016). Graham (2009) suggests that since it is impossible to determine the MDM, the focus should rather be on the magnitude of the violation against a certain MDM and the sensitivity of the analysis due to these violations.

It has been shown that MI procedures are robust when inappropriate imputation models are applied. Even if a particular imputation from a set of MIs has shortfalls, it only affects the defected part and not all inference. Therefore, using any suitable form of MI as opposed to SI is preferred and deemed to provide reliable inference (Rubin, 2003). However, it is important to choose an imputation model that is relatable to the planned analyses on the completed data. Ideally a generalised imputation model should be selected that will preserve the relationships captured by different variables in the data (Schafer & Graham, 2002; Schafer & Olsen, 1998).

3.4.4.3 Rubin's rules

After standard complete data analyses are applied to the multiple imputed data sets, the estimates of interest (mean, variance, factor loadings, regression coefficients, etc.) can be combined to formulate a final inferential statement which incorporates the within-imputation and between-imputation variance by using Rubin's rules (Graham, Olchowski & Gilreath, 2007; Rubin, 1987; Rubin & Schenker, 1986; Van Buuren, 2012).

Suppose that there are no missing values and a quantity of interest, Q, is estimated by $\hat{Q}$. This quantity can represent any population estimate (e.g. means, coefficients). The inference of the data set is based on $(Q - \hat{Q}) \sim N(0, U)$, where U is the variance of $Q - \hat{Q}$.

Now, suppose that some samples consist of missing observations. After applying m MIs and subsequent complete data analyses. There are m estimates of the quantity / parameter of interest available, $\hat{Q}^*_m$ and U^*_m. The inference is now based on $(Q - \hat{Q}^*_{m\bullet}) \sim N(0, T^*)$, where the estimate of Q is dependent on the number of imputations and calculated as:

$$\hat{Q}^*_{m\bullet} = \sum_{l=1}^{m} \frac{\hat{Q}^*_l}{m}.$$

The variance, T^*, is calculated by incorporating the average within-imputation variance, $\widehat{W}$, and between-imputation variance, $\hat{B}$:

$$T^* = \widehat{W} + (1 + m^{-1})\hat{B},$$

where

$$\widehat{W} = \sum_{l=1}^{m} \frac{U^*_l}{m} \qquad \text{and} \qquad \hat{B} = \sum_{l=1}^{m} \frac{\left(\hat{Q}^*_l - \hat{Q}^*_\bullet\right)^2}{m - 1}.$$

The magnitude of information that is unobserved is expressed in the size of the between-imputation variance relative to the estimated variance (Schafer & Olsen, 1998). The total standard error of the estimate, $\hat{Q}$, is obtained from $\sqrt{T^*}$. If the data are fully observed, there will only be one set of estimates obtained from the complete data analysis. This means that the between-imputation variance will be equal to zero and consequently, $T^* = \widehat{W}$.

Inferences can be based on the t distribution:

$$\hat{Q}^*_{m\bullet} \pm t_{\left(v;1-\frac{\alpha}{2}\right)}\sqrt{T^*},$$

where the degrees of freedom, v, depends on the between- and within-imputation variation as well as the number of imputations:

$$v = (m-1)\left[1 + \frac{\widehat{W}}{(1+m^{-1})\hat{B}}\right]^2.$$

The degrees of freedom will increase as the number of imputations increases, resulting in the approximation of the normal distribution (Schafer, 1997).

The literature is vague with regard to the application of Rubin's rules to estimates obtained from categorical data analysis. It is accepted to assume approximate normality for ordinal scaled variables and rounding the final estimates to integer values to assign responses to a CL (Schafer, 1997). However, there are no guidelines for nominal scaled data. It is advised to transform the estimates with severe deviations from normality to achieve approximate normality before Rubin's rules are applied and then back-transform the combined estimates for the final inference (Van Buuren, 2012).

Rubin's rules will not be applied in its entirety as presented in this section. The handling of MIs in this research will be discussed in Chapter 4 (cf. 4.3).

3.4.4.4 Number of multiple imputations

According to Rubin (1987) the number of imputations should range between two and ten, when the amount of missing information is small. Rubin and Schenker (1986) show that by only generating two MIs, the coverage of the parameter estimates improve in comparison to SI.

The number of imputations has been further investigated by numerous authors, to name a few: Royston (2004), Graham, Olchowski and Gilreath (2007), Bodner (2008), White, Royston and Wood (2011) and Von Hippel (2009, 2018).

Initially, the use of the fraction of missing values or missing information was recommended to be indicative of the number of imputations that is chosen (Rubin, 1987). In a multivariate data set the number of cases with non-responses will not be an indication of the amount of missing information, since the percentage of non-response is not synonymous with the magnitude of missing information. The loss of information will be different for each parameter of interest and must be determined accordingly (Allison, 2002; Bodner, 2008). Rubin (1987) proposed a measure of relative efficiency, λ, which incorporates the fraction of missing information, γ:

$$\lambda = \left(1 + \frac{\gamma}{m}\right)^{-1}.$$

The fraction of missing information, γ, expresses the increase of precision of the estimate of interest for the case where no observations are missing (Schafer & Olsen, 1998). First, the relative increase in the variance caused by missing values is calculated:

$$r_m = \frac{(1 + m^{-1})\hat{B}}{\widehat{W}},$$

which simplifies the degrees of freedom equation to:

$$\nu = (m - 1)(1 + r_m^{-1})^2.$$

The fraction of missing information can now be calculated by the following (Rubin, 1987; Schafer, 1997):

$$\gamma = \frac{\left(r_m + \frac{2}{\nu + 3}\right)}{(r_m + 1)}.$$

Schafer and Olsen (1998) evaluated the effect of the fraction of missing information and the number of imputations on the relative efficiency. The results confirmed that not much is gained by increasing the number of imputations beyond five.

However, the use of the relative efficiency measure was questioned, which lead to investigations proposing that a minimum of 20 imputations are advisable. Graham *et al.* (2007) focused the choice of the number of imputations on a power analysis. It was then proposed that the percentage of samples containing missing values should be used as the number of imputations that is required (Von Hippel, 2009). Bodner (2008) based the decision on the coverage of confidence intervals by varying the number of imputations, the p-value and the estimated proportion of variation caused by the missing observations. Lastly, it was suggested that five imputations should be sufficient to determine whether the imputation procedure is appropriate. Once the imputation procedure is chosen, depending on computational and storage capacity, any number between 20 and 100 imputations can be generated (Van Buuren, 2012).

Abovementioned approaches are based on the imputation of continuous data and guidelines for the number of imputations required in categorical data analysis are not easily acquired. The focus of this research is not to compare different MI techniques and determine the most suitable number of imputations, but to develop unbiased exploratory techniques to be applied after MI. Therefore, the choice is made to use ten imputations for all MI applications in this research.

3.5 Compositional data

In some cases, the response profile of a sample add up to a certain constant; proportions add up to one and the sum of percentages is equal to 100. Each response is referred to as an additive term or composition (Greenacre, 2017). The true response values of the variables are not of importance for the imputation, but rather the relative information. The total of the additive terms is commonly standardised to one or can be expressed as a percentage, since the responses can be viewed as a probability of occurrence (Hron *et al.*, 2010).

Consequently, the non-response of one variable per sample could be easily obtained by calculating the difference from the known total. This method of handling missing values is known as deductive imputation, since unknown information is deduced from observed information (Van Buuren, 2012). Deductive imputation commonly occurs in panel surveys since certain questions are repeatedly recorded. If non-responses occur in some of the records, the information can be obtained from previous records of the same response. This imputation method can be regarded as part of the initial data editing procedure and not a core imputation method, since no uncertainty is captured by the imputation procedure (Nordholt, 1998).

Compositional data analysis is beyond the scope of this research and will be considered for future research projects. The developed methodology (cf. Chapter 4) offers the possibility of extending the principles to compositional data applications.

3.6 Conclusion

This chapter provided a general overview of the relevant missing data terminology and commonly used techniques. The investigation of the MDM (cf. 3.2) is of particular interest and will be addressed in the fourth study objective of this research (cf. 1.4.4 and Chapter 8). Different imputation techniques were not compared, since the aim of this research is to develop unbiased visualisation techniques after missing data techniques are applied. The following Chapter 4 presents the methodology to achieve the study objectives for this research.

Chapter 4
Methodology

4.1 Introduction

This chapter contains the complete methodology of this research with the exception of the simulation protocol which is presented in Chapter 5. The subsequent chapter presents the methods for firstly generating complete categorical data sets from three different distributions. Artificial missingness is then created according to two MDMs and different percentages of missingness.

The methodology that will be presented in this chapter will be applied to all simulated data sets as will be discussed in Chapter 5. The methodology for each of the four study objectives cannot be separated completely, since the objectives emanate from each other. The following schematic representations and short discussions of the four study objectives aim at providing an orientation before the technical details will follow in the sections of this chapter.

The first objective is summarised in Figure 4.1. This objective is concerned with obtaining a final configuration from MIs by using the novel GPAbin procedure. Data sets containing missing observations are created according to the simulation protocol (cf. Chapter 5). There are a 1000 repetitions of each simulation scenario. The success of the missing data techniques are determined by comparing MCA biplots of the simulated complete data sets to the configurations obtained from the proposed methodology.

Multiple imputation using multiple correspondence analysis (MIMCA) is applied to simulated data sets which result in ten completed data sets for each simulation. An MCA biplot is constructed for each of the ten completed data sets from which a centroid configuration is calculated. The centroid configuration is used as the initial target configuration to which the configurations of the MIs are aligned using GPA. Once the multiple configurations are aligned, the means of the aligned coordinates are calculated to obtain the GPAbin configuration which is then compared to the MCA biplot of the original complete simulated data set.

A regularised iterative MCA (RIMCA) SI technique is also applied to the data sets which results in a single completed data set for each simulation. An MCA biplot is constructed for the completed data set and compared to the MCA biplot of the original complete data set, as was done for the GPAbin approach. The SI technique is included in this study objective to determine the success of the GPAbin method, which is essentially an MI technique, compared to SI. The broad comparison will therefore be between visualisations from MI and SI.

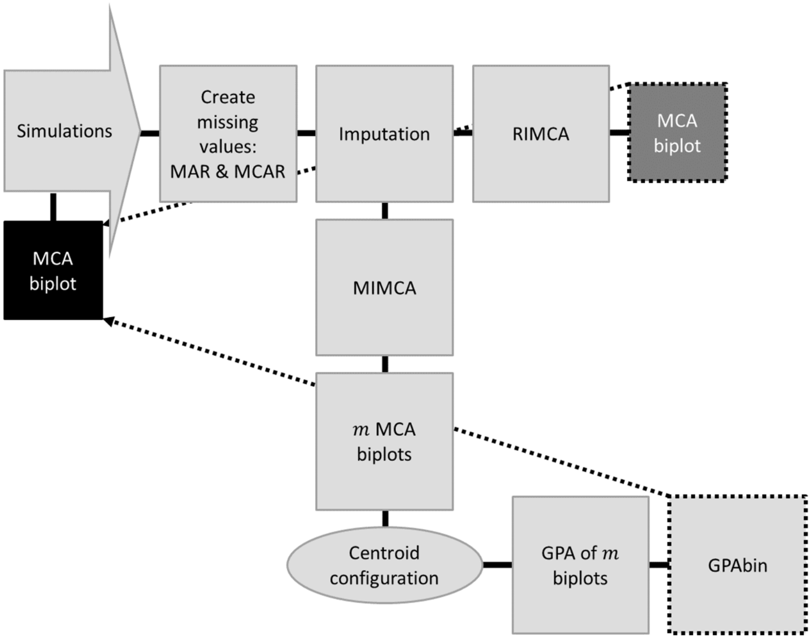


Figure 4.1 *Schematic representation of the GPAbin objective (cf. 1.4.1).*

The topics that are represented in Figure 4.1 for the first study objective are further discussed in Sections 4.2, 4.3 and 4.6.

The second study objective is summarised in Figure 4.2. The GPAbin application (cf. Figure 4.1) is replicated from the first study objective and is presented on the right side of the schematic representation of the sMCA study objective (cf. Figure 4.2). Focusing on the left steps after the missing data sets are created, firstly, additional CLs are created for the missing responses in the indicator matrix using single active handling (cf. 3.4.2.2), then sMCA is applied and only the observed CLs are displayed in the sMCA biplot which will be compared with the original complete MCA biplot. The GPAbin procedure is included in this study objective, since the difference between the quality of visual representations using MI and non-imputation techniques is investigated.

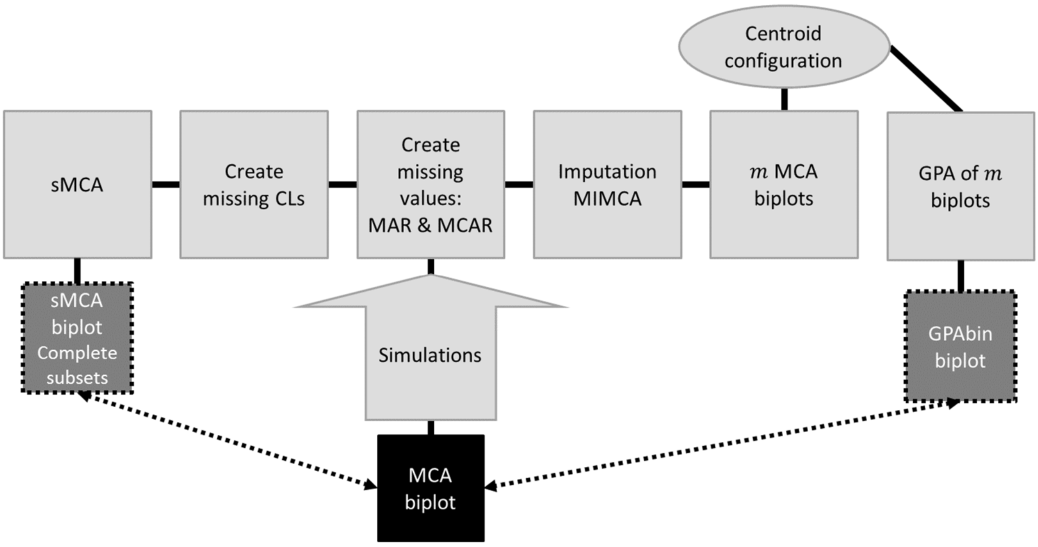


Figure 4.2 Schematic representation of the sMCA objective (cf. 1.4.2).

The topics that are represented in Figure 4.2 for the second study objective are discussed in Sections 4.2, 4.3, 4.5 and 4.6.

The third study objective aims at determining the success of the prediction of a multivariate categorical data set by using Euclidean distances between samples and CLPs in MCA biplots. The initial steps of the first objective (cf. Figure 4.1) are represented in Figure 4.3. After MIMCA is applied and MCA biplots are constructed for each completed data set, responses are predicted using the Euclidean distances between the sample points and CLPs in the biplots. The CLPs with the closest proximity to the sample points per variable are selected as the predicted response CLs. This is done for each of the MI MCA biplots. Therefore, ten completed predicted data sets are available for each initial simulation. The frequencies of the CLs per response across the predicted multiple imputed data sets are used to determine the final CL allocation. The CL with the highest frequency per response is assigned as the final predicted CL, if it happens that two or more modes occur, the final CL is randomly assigned between the levels that are in the majority. The completed predicted data set is again subjected to MCA in order to construct an MCA biplot to compare to the original complete simulated MCA biplot. This prediction approach is referred to as the Majority rule prediction method. Another prediction approach is followed by using the Euclidean distances between the sample points and CLPs in the GPAbin configuration. After a complete predicted data set is obtained, MCA is again applied to construct an MCA biplot for comparison with the original simulated complete MCA biplot. This prediction approach is referred to as the GPAbin prediction method.

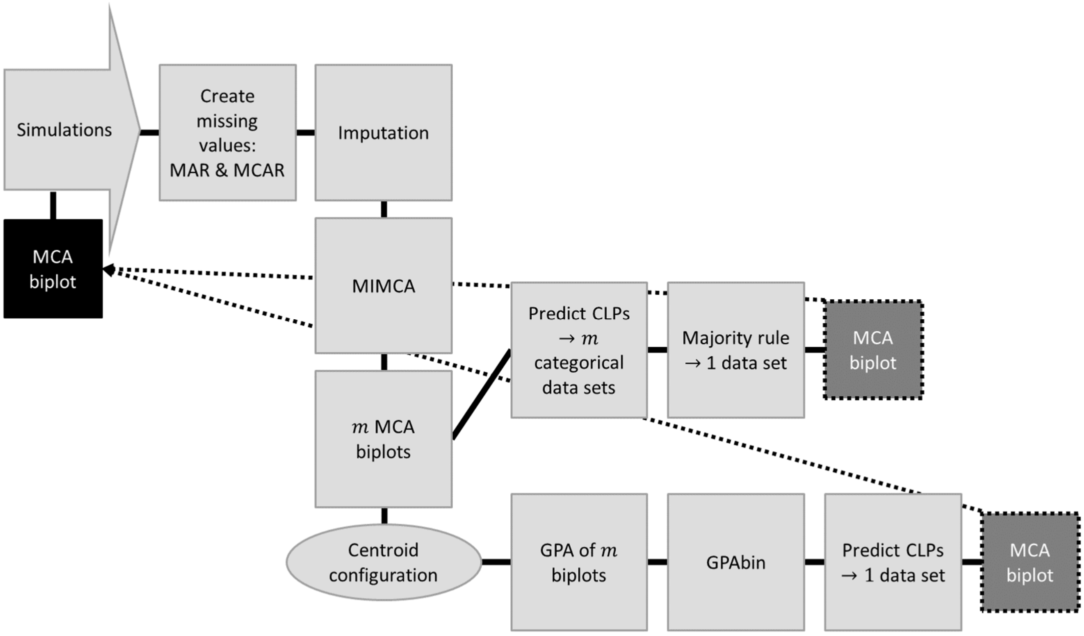

Simulations
Create missing values: MAR & MCAR
Imputation
MCA biplot
MIMCA
Predict CLPs → m categorical data sets
Majority rule → 1 data set
MCA biplot
m MCA biplots
Centroid configuration
GPA of m biplots
GPAbin
Predict CLPs → 1 data set
MCA biplot

Figure 4.3 Schematic representation of the prediction objective (cf. 1.4.3).

The topics that are represented in Figure 4.3 for the third study objective are discussed in Sections 4.2, 4.3, 4.4 and 4.6.

The fourth and final study objective is represented in Figure 4.4. The first few steps appear in the schematic representation of the second study objective (cf. Figure 4.3). This objective applies sMCA and visualises the missing subset in order to detect patterns or groupings that may indicate associations between variables which could aid in identifying the cause of missingness. The recoding of the indicator matrix is done by using both single- and multiple active handling (cf. 3.4.2.2), thereafter sMCA is applied and only the missing subsets are visualised in the sMCA biplots. The final step is to identify the MDM using clustering techniques and an available measure of fit as a criterion to determine the MDM.

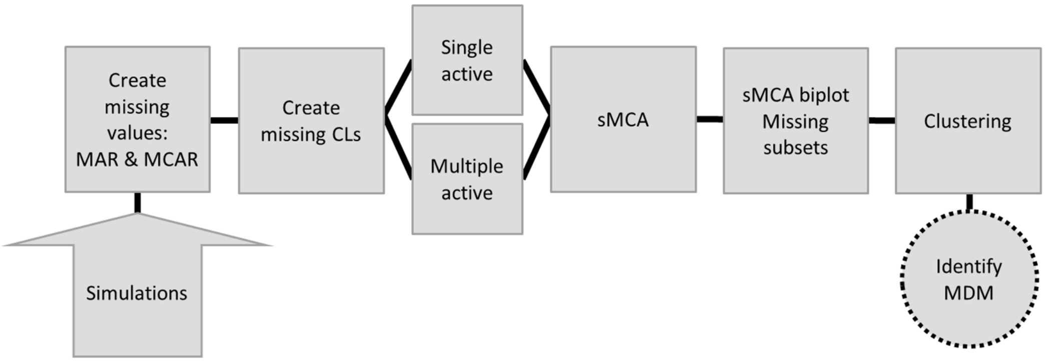


Figure 4.4 Schematic representation of the MDM objective (cf. 1.4.4).

The topics that are represented in Figure 4.4 for the fourth study objective are discussed in Sections 4.5 and 4.7.

The same example data set will be used throughout this chapter with the following specifications: five categorical variables for a 1000 samples simulated from a uniform distribution with 10% missing values inserted with a MAR MDM. An additional data set will be considered for the visualisations of Section 4.6.2 and Section 4.7 with the following specifications: five categorical variables for a 100 samples simulated from a uniform distribution with 50% missing values inserted with a MAR MDM. Details regarding the specific data simulations will be provided in Chapter 5.

4.2 Imputation methods

4.2.1 Regularised iterative multiple correspondence analysis

A RIMCA algorithm, dedicated to MAR and MCAR MDMs, is used as an SI model (Josse *et al.*, 2012). This algorithm is based on MCA and is therefore deemed a fitting choice for the categorical missing data problems that are the focus of this research project. It is conjectured that MI is superior over SI (Van Buuren, 2012; Zhang, 2003), but it is important to establish whether visualisations obtained from the GPAbin approach after MI is unbiased in comparison to configurations obtained from SI. The comparison of GPAbin and RIMCA will form part of the first study objective, referred to as the GPAbin objective.

The discussion of the theoretical aspects of the RIMCA algorithm will be discussed before reference is made to the available `R` material.

4.2.1.1 Algorithm: Regularised iterative multiple correspondence analysis

The RIMCA algorithm consists of three steps: initialisation, reconstruction and iteration. Consider a categorical multivariate data set, $\mathbf{X}$, with I samples and J variables ($p_j, j = \{1,2,\dots,J\}$) each with k_j categories such that $K = \sum_{j=1}^{J} k_j$. The indicator matrix, $\mathbf{G}$, will consist of I rows and K columns.

a) Initialisation ($\ell = 0$)

During the initialisation step of the algorithm the missing entries in the indicator matrix, $\mathbf{G}$, are substituted by fuzzy values (continuous values between zero and one) representing the proportions of the observed number of CLs over all samples. This approach is referred to as the missing fuzzy average method, as discussed in Section 3.4.2.3. The fuzzy category values, per missing response, are assigned such that the total of the fuzzy categories for a particular variable adds up to one. The updated indicator matrix consisting of fuzzy values, zeros and ones, is referred to as $\mathbf{G}^0$.

b) Reconstruction

The second step consists of applying MCA in terms of the weighted SVD of a current triplet of matrices ($\mathbf{X}^*, \mathbf{C}, \mathbf{R}$), then reconstructing the data using a predetermined number of dimensions from the MCA solution and lastly updating the indicator matrix after imputation.

As briefly mentioned in Section 2.6, multivariate data can be expressed in the form a triplet. The RIMCA algorithm expresses MCA as a weighted PCA of a matrix triplet. The elements of the triplet ($\mathbf{X}^*, \mathbf{C}, \mathbf{R}$) are obtained from the following weightings:

- $\mathbf{X}^* = I\mathbf{G}^{\ell-1}\left(\mathbf{D}_c^{\ell-1}\right)^{-1}$. Therefore, the indicator matrix is weighted according to the column margins, similar to CA (cf. 2.6), where $\mathbf{D}_c^{\ell-1}$ is the diagonal matrix of column margins of the indicator matrix.
- $\mathbf{C} = \frac{1}{IJ}\mathbf{D}_c^{\ell-1}$, which in terms of CA terminology, is the average column profile.

- $\mathbf{R} = \frac{1}{I}\mathbb{I}_I$, which is the weighting of the rows and $\mathbb{I}_I$ is the identity matrix with dimensions, $I \times I$.

Step (b.1) of the reconstruction step consists of performing the SVD of the data triplet, which is equivalent to obtaining the SVD of the following:

$$\left(I\mathbf{G}^{\ell-1}\left(\mathbf{D}_c^{\ell-1}\right)^{-1} - \mathbf{1}_I\mathbf{1}_K'\right) \times \sqrt{\frac{1}{IJ}\mathbf{D}_c^{\ell-1}} = \mathbf{U\Lambda V}',$$

using the triplet elements it simplifies to:

$$\left(\mathbf{X}^{*\,\ell-1} - \mathbf{1}_I\mathbf{1}_K'\right) \times \sqrt{\mathbf{C}} = \mathbf{U\Lambda V}'.$$

Step (b.2) of the reconstruction step consists of the data reconstruction and regularisation. The regularisation part of the algorithm refers to the singular values that are shrunk using the predetermined number of dimensions, S, as estimated by the `estim_ncpMCA()` function (cf. 4.2.1.2):

$$\sqrt{\hat{\lambda}_s^\ell} = \sqrt{\lambda_s^\ell} - \left[\left(\frac{1}{K-J-S}\sum_{s=S+1}^{K-J}\lambda_s^\ell\right)/\sqrt{\lambda_s^\ell}\right].$$

The reconstructed and regularised indicator matrix is obtained by using the following formula:

$$\widehat{\mathbf{X}}^{*\,\ell} = \left\{\mathbf{U}_{I\times S}\sqrt{\hat{\lambda}_s^\ell}\,\mathbf{V}_{K\times S}' \times \sqrt{IJ\left(\mathbf{D}_c^{\ell-1}\right)^{-1}}\right\} + \mathbf{1}_I\mathbf{1}_K'.$$

The updated values of the indicator matrix, $\widehat{\mathbf{G}}^\ell$, are obtained by rewriting the first triplet element, $\widehat{\mathbf{X}}^{*\,\ell}$:

$$\widehat{\mathbf{G}}^\ell = \frac{1}{I}\widehat{\mathbf{X}}^{*\,\ell}\mathbf{D}_c^{\ell-1}.$$

Only the initially missing values are to be updated by the algorithm and replaced by the fitted values, therefore a dummy matrix, $\mathbf{W}$, is constructed where elements with the value of zero refer to initially missing observations and elements with the value of one refer to initially observed observations. The single imputed data set is obtained from the following Hadamard ($*$) multiplications:

$$\mathbf{G}^\ell = \mathbf{W} * \mathbf{G} + (1-\mathbf{W}) * \widehat{\mathbf{G}}^\ell.$$

Step (b.3) of the reconstruction step consists of recalculating $\mathbf{X}^{*\,\ell}$, $\mathbf{D}_c^{\ell}$ and $\mathbf{C}^{\ell}$ for the imputed data set for step ℓ.

c) Iteration

The reconstruction step is repeated until the sum of the squared differences between the imputed indicator matrix of the current iteration, $\widehat{\mathbf{G}}^{\ell}$, and the previous iteration, $\widehat{\mathbf{G}}^{\ell-1}$, reaches a threshold of at most 0.001:

$$\sum_{i=1}^{I}\sum_{k=1}^{K}\left(\hat{g}_{ik}^{*\,\ell-1}-\hat{g}_{ik}^{*\,\ell}\right)^{2} \leq 0.001.$$

The final categorical data sets are obtained by allocating the category value to the CL with the largest fuzzy value between zero and one, which is regarded as a degree of membership or probability to be associated to a particular CL.

4.2.1.2 Code: Regularised iterative multiple correspondence analysis

The RIMCA algorithm is available in the `R` package, `missMDA` (Josse & Husson, 2016), and relies on two functions: `estim_ncpMCA()` and `imputeMCA()`.

The number of dimensions to retain in the reconstruction step of the MCA solution has to be determined before imputation using a cross-validation technique. The default parameters are given in the argument list of the `estim_ncpMCA()` function, which is available in the `R` package, `missMDA` (Josse & Husson, 2016):

```
estim_ncpMCA <- function (don, ncp.min = 0, ncp.max = 5, method
= c("Regularized","EM"), method.cv = c("Kfold", "loo"), nbsim =
100, pNA = 0.05, threshold = 1e-04, verbose = TRUE).
```

Determining the appropriate number of dimensions is important since it will have an impact on the success of the imputation procedure. Underestimation of the components could lead to the loss of information and overestimation could result in unstable predictions with the inclusion of unnecessary noise (Josse *et al.*, 2012). A range of proposed dimensions, between the specified '`ncp.min`' and '`ncp.max`', are explored and the number of dimensions that results in the smallest mean square error of prediction (MSEP) is retained and used in the RIMCA procedure. Inserting the range of dimensions in the argument list of the `estim_ncpMCA()` function becomes tedious when applying the algorithm to a large number of data sets with varying dimensions in a simulation study. This problem has been overcome by the addition of auxiliary functions (cf. Appendix L) to deal with the selection of the dimensions automatically using the `indcol()` and `indmat()` functions. The total number of columns in the updated indicator matrix containing the fuzzy values for the missing data entries is used to determine the maximum number of possible dimensions to retain.

In estimating the number of dimensions to retain, there is an option of two cross-validation procedures (`"Kfold"` and `"loo"`) which are used in conjunction with the option of two SI techniques (`"Regularized"` and `"EM"`). The number of dimensions that results in the smallest MSEP is retained. The Expectation-Maximisation (EM) method, applied in the iterative MCA algorithm, results in overfitting. In this context it means that the observed values are well represented in the MCA solutions, but the estimation of the missing information is of low quality since the MCA axes and components are poorly estimated. Possible reasons for the overfitting is when low correlations exist between variables, also when a high percentage of data entries are missing and a large number of dimensions is chosen to be retained (Josse *et al.*, 2012). An intuitive method to resolve the problem of overfitting is to decrease the selected number of dimensions with caution. The `"Regularized"` method, also referred to as the RIMCA, enforces a shrinkage method which results in reduced variances compared to the iterative MCA approach.

The cross-validation technique referred to as the leave-one-out method (`"loo")` removes the elements of the data matrix, one at a time, and determines the MCA solution. The number of dimensions that results in the smallest prediction error is retained for the imputation procedure. The `"Kfold"` method is a cross-validation technique in which a percentage of missing values is inserted with an MCAR MDM before predicting the indicator matrix. The procedure is repeated a 100 times (`nbsim=100`) until the predetermined threshold is reached. The authors (Audigier *et al.*, 2017) advise that even though a smaller number of simulations ('`nbsim`') will result in the procedure being less computationally intensive, the default parameter should be enforced to ensure that the simulation error remains low. Also, the default percentage of missing values (`pNA`=0.05) aids in preserving the structure of the original data set. The `"Kfold"` cross-validation method is preferred, since it is not as computationally intensive as the `"loo"` method (Audigier *et al.*, 2017).

The `estim_ncpMCA()` function is adapted and referred to as the `myestim_ncpMCA()` function available in Appendix F.2. The following changes are made:

- The threshold for all iterative algorithms in this study is set to 0.001.
- A seed parameter is included to enable the reproducibility of the results.
- In some cases, the `"Kfold"` method cannot estimate the number of dimensions to use when the number of missing values inserted in the estimation set and the original data set are equal. This problem is resolved by adding a statement in the function that repeats the deletion step using a different random starting point, by updating the seed value.

After the number of dimensions to retain in the regularisation is obtained, the imputation function can be used. The default arguments are as follows:

```
imputeMCA <- function (don, ncp = 2, method = c("Regularized",
"EM"), row.w = NULL, coeff.ridge = 1, threshold = 1e-06, seed =
NULL, maxiter = 1000).
```

The `"Regularized"` method is selected for all imputations, as discussed in the RIMCA algorithm steps (cf. 4.2.1.1).

The following modifications are made to the original `imputeMCA()` function. The adapted function is referred to as the `myimputeMCA()` function available in Appendix F.3:

- The original function does not make provision for cases where only one CL is observed per variable. Also, when the percentage of missing values increases, it might happen that most of the CLs are unobserved. The adapted function correctly handles such situations.
- If it happens that the algorithm does not converge due to the number of dimensions specified in the argument list, an error message is given suggesting a different number of dimensions. This becomes tedious when applying the function in a simulation study, since manual input is required to address the error. The function is updated to utilise the proposed number of dimensions given in the error message and automatically prompt the function again with the new parameter.

4.2.2 Multiple imputation using multiple correspondence analysis

The MIMCA method is derived from the RIMCA method (cf. 4.2.1) and was developed in collaboration with two of the RIMCA authors (Audigier *et al.*, 2017).

This will be the only MI technique applied in this research project, since the difference between available MI procedures is not the focus of the project, but rather the development of visualisations after handling missing data.

4.2.2.1 Algorithm: Multiple imputation using multiple correspondence analysis

The MIMCA algorithm also consists of three steps, similar to the preceding RIMCA algorithm (cf. 4.2.1.1):

a) Step 1

The weighting matrix in RIMCA, $\mathbf{R} = \frac{1}{I}\mathbb{I}_I$, is now determined by using a non-parametric bootstrap method. A simple random sample is selected with replacement to obtain a sample from the available number of samples (n) in the data set. The number of times the specific sample (n_i) is selected in the simple random sample is expressed as a proportion and is allocated as the weight of the specific sample (n_i).

A weight matrix, $\mathbf{R}_m$, is generated for each of the imputations required for the MI procedure. Now, the indicator matrices, $\mathbf{G}_m^0$, are updated according to the weights by replacing missing values with fuzzy values (proportions obtained from the non-parametric bootstrap samples). The variability of the parameters of the imputation model is incorporated by assigning the weights based on simple random samples. This is one of the requirements for a valid MI procedure (cf. 3.4.4.2).

b) Step 2

The RIMCA algorithm (cf. 4.2.1.1) is performed separately for each imputation (m), using the updated elements of the data triplet: ($\mathbf{X}_m^*, \mathbf{C}_m, \mathbf{R}_m$):

- $\mathbf{X}_m^* = I\mathbf{G}_m^{\ell-1}\left(\mathbf{D}_{(m)c}^{\ell-1}\right)^{-1}$,

- $\mathbf{C}_m = \frac{1}{IJ}\mathbf{D}_{(m)c}^{\ell-1}$,
- $\mathbf{R}_m$ (from step 1).

c) Step 3

After the algorithm has converged for the separate imputations, the categorical values are assigned to the fuzzy imputed values by sampling a CL from the available CLs based on the multinomial distribution of the final fuzzy values obtained in step 2. This incorporates the uncertainty of the imputations, which is also a requirement for a valid MI procedure (cf. 3.4.4.2).

4.2.2.2 Code: Multiple imputation using multiple correspondence analysis algorithm

The MIMCA algorithm is also available in the `R` package, `missMDA` (Josse & Husson, 2016), and relies on three functions: `estim_ncpMCA(), imputeMCA()` and `MIMCA()`.

The same procedure followed for RIMCA (cf. 4.2.1), to determine the number of dimensions to retain in the reconstruction and regularisation steps, is followed for MIMCA. The adapted functions of both existing functions (cf. 4.2.1.2) are also used for the MIMCA algorithm: `myestim_ncpMCA()` (cf. Appendix F.2) and `myimputeMCA()` (cf. Appendix F.3).

The default arguments for the MIMCA algorithm are as follows:

```
MIMCA <- function (X, nboot = 100, ncp, coeff.ridge = 1, threshold = 1e-06, maxiter = 1000, verbose = FALSE).
```

The number of dimensions ('`ncp`') to use for the MCA solution is determined by the `myestim_ncpMCA()` function.

The regularisation approach of the RIMCA algorithm (cf. 4.2.1.1) is applied by fixing the '`coeff.ridge`' argument to the value of 1, which is also the default setting for the MIMCA algorithm.

The following changes are made to the available functions and are presented in the `myMIMCA()` function in Appendix G.2:

- Similar to RIMCA, the function does not make provision for cases where only one CL is observed per variable. The adapted function correctly handles such situations.
- As is the case with the RIMCA algorithm, the function is updated to utilise the proposed number of dimensions given in the error message when the algorithm does not converge and automatically prompt the function again with the new parameter.
- The original `MIMCA()` function prompts the `imputeMCA()` function. This is changed to the adapted `myimputeMCA()` function.

4.2.2.3 Multiple correspondence analysis biplots of multiple imputed data sets

Figure 4.5 shows the MCA biplots for ten multiple imputed data sets obtained from the MIMCA algorithm. The example data set for this chapter (cf. 4.1) is used for the visualisations. As indicated in the legends of the panels of Figure 4.5, the samples are represented by open circle plotting characters and the CLPs are represented by filled triangle plotting characters.

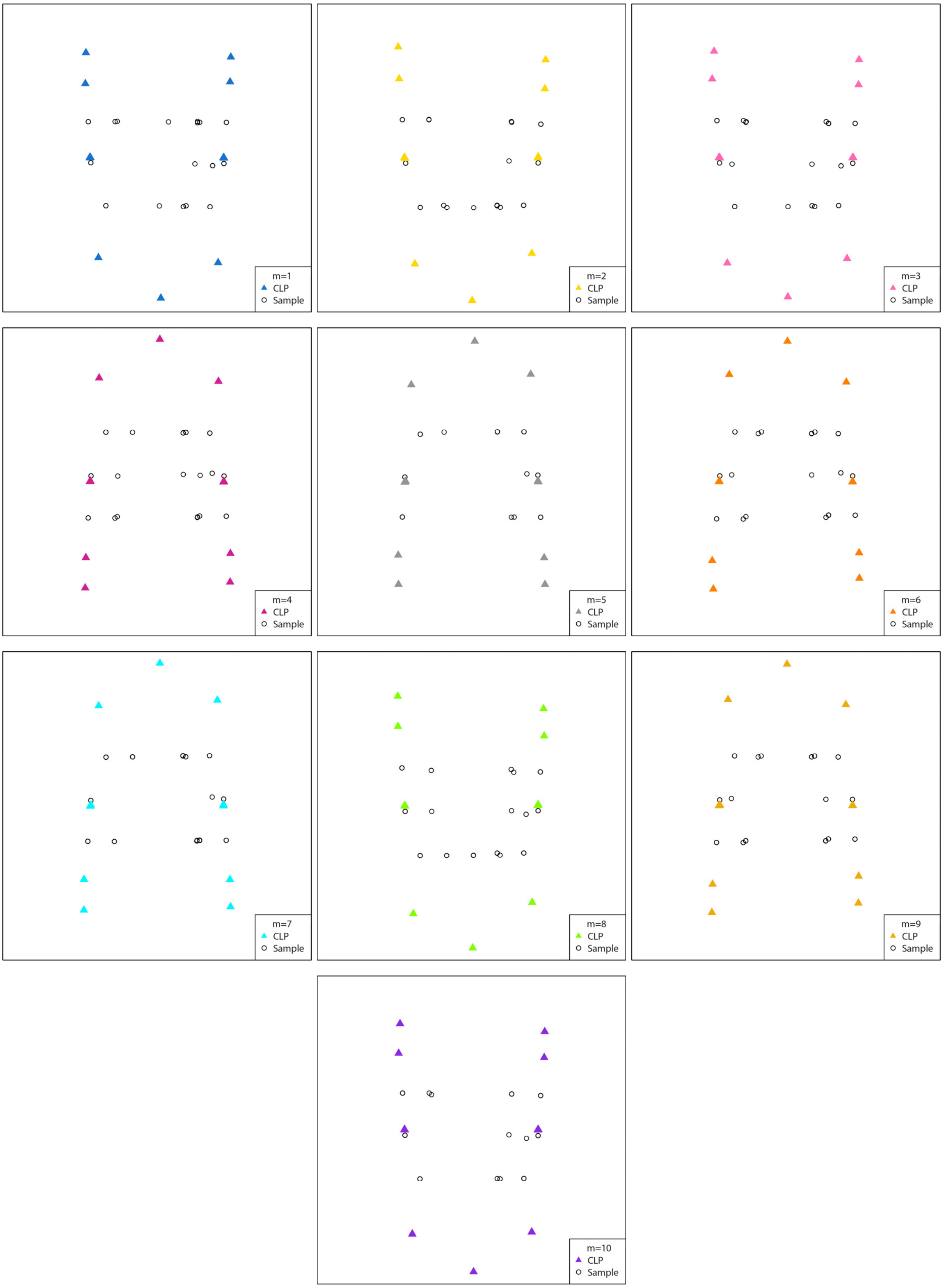


Figure 4.5 Illustration of MCA biplots constructed from ten MIs using MIMCA. Each panel represents an MCA biplot for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

The MCA biplots of the ten MIs result in similar general patterns of the samples and CLPs in Figure 4.5. The subtle differences between the configurations confirm that the MI procedure incorporates variation which is observed between imputations. However, the positions of the coordinates in the panels are not considerably different and therefore also show consistency of the imputation procedure. It is observed that imputations four ($m = 4$), five ($m = 5$), six ($m = 6$), seven ($m = 7$) and nine ($m = 9$) are reflections about the y-axis when compared to imputations one ($m = 1$), two ($m = 2$), three ($m = 3$), eight ($m = 8$) and ten ($m = 10$). Apart from the reflections, the same general patterns are observed between these configurations.

4.3 GPAbin

The GPAbin visualisation approach consists of iteratively aligning multiple MCA biplots, one for each imputation, to a centroid configuration of the multiple MCA biplots. Once the configurations are aligned, a combined final configuration is obtained by calculating the mean coordinate matrices after GPA. The R code for the methodology presented in this section is available in Appendix D and Appendix E.

4.3.1 Optimally aligning multiple imputed configurations

This section elaborates on the methodology for the ‘GPA’-part of the GPAbin procedure (cf. 2.9.1). After MCA is performed on each of the MIs, MCA biplots are constructed. The principal coordinates are used to plot the samples, $\hat{\mathbf{Z}}_m$, and standard coordinates are used for the CLPs, $\widehat{\mathbf{CLP}}_m$. The GPA procedure can be performed on either the samples or CLPs, since the final optimal biplot display can be obtained by transforming the remaining coordinate matrix to align with the optimally rotated coordinate matrix. The number of CLPs is typically less than the number of samples and it is expected that the display focussing on CLPs will be less cluttered and therefore ease visual inspection. The CLPs are also of particular interest, since the associations between responses could be indicative of the underlying MDM.

The iterative GPA procedure, in terms of the CLP coordinate matrices, minimizes the following sum of squares (SS):

$$SS = \sum_{m=1}^{M} \left\| \mathbf{s}_m(\widehat{\mathbf{CLP}}_m)\mathbf{Q}_m - \overline{\widehat{\mathbf{CLP}}} \right\|^{\mathbf{2}},$$

where $\mathbf{s}_m$ → optimal dilation factor for the m^{th} imputation,

$\widehat{\mathbf{CLP}}_m$ → coordinate matrix for the CLPs of the m^{th} imputation,

$\mathbf{Q}_m$ → optimal orthogonal rotation matrix for the m^{th} imputation and

$\overline{\widehat{\mathbf{CLP}}}$ → coordinate matrix for the current centroid configuration of the imputed CLPs.

The translation of the GPA is incorporated by centring the configurations at the origin before comparing the configurations (Gower & Dijksterhuis, 2004).

The target configuration, used to align the multiple configurations, is the mean of the current setting of the CLP coordinate matrices of the multiple imputed MCA biplots and is referred to as the centroid configuration:

$$\overline{\widehat{\mathbf{CLP}}} = \frac{1}{M}\sum_{m=1}^{M} s_m \widehat{\mathbf{CLP}}_m \boldsymbol{Q}_m.$$

The centroid configuration is updated on each iteration until the GPA algorithm converges.

The final coordinate matrices for each imputation are obtained from the following multiplication after the GPA algorithm has converged:

$$\widehat{\mathbf{CLP}}_m^* = s_m(\widehat{\mathbf{CLP}}_m)\mathbf{Q}_m$$

and

$$\hat{\mathbf{Z}}_m^* = s_m \hat{\mathbf{Z}}_m \mathbf{Q}_m.$$

The code for the `GPA()` function is presented in Appendix D.

4.3.2 Visual inspection

The GPA transformations of the CLPs of the multiple imputed MCA biplots are visualised in Figure 4.6. The transformed CLPs are represented by filled square plotting characters and the CLPs before GPA are represented by open triangle plotting characters.

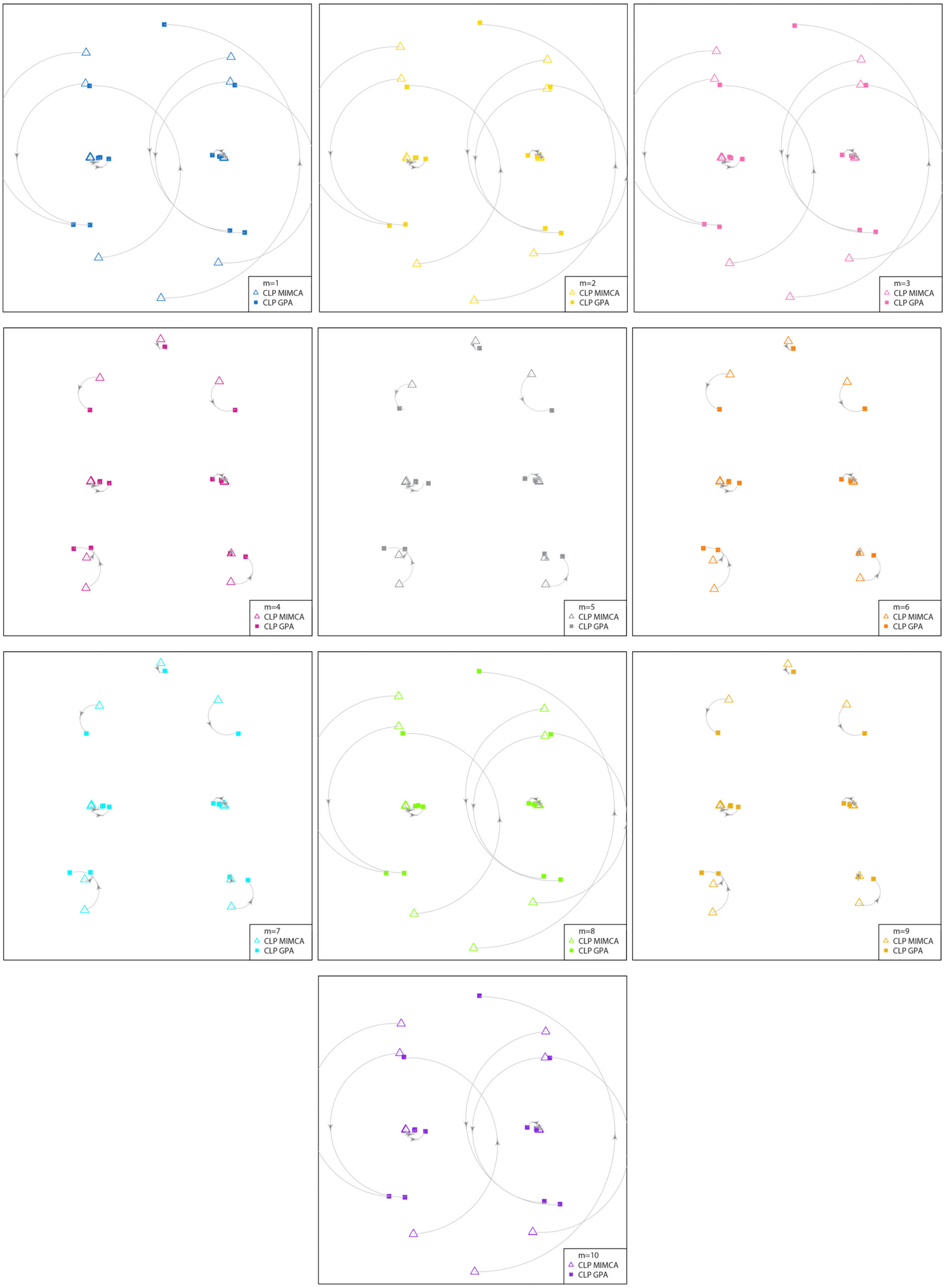


Figure 4.6 Illustration of CLPs before and after GPA for ten MIs. Each panel represents the CLPs for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

Minimal transformations are required for imputations four ($m = 4$), five ($m = 5$), six ($m = 6$), seven ($m = 7$) and nine ($m = 9$) with rotation and reflection transformations applied to imputations one ($m = 1$), two ($m = 2$), three ($m = 3$), eight ($m = 8$) and ten ($m = 10$).

The MCA biplots after GPA are presented in Figure 4.7 on the next page.

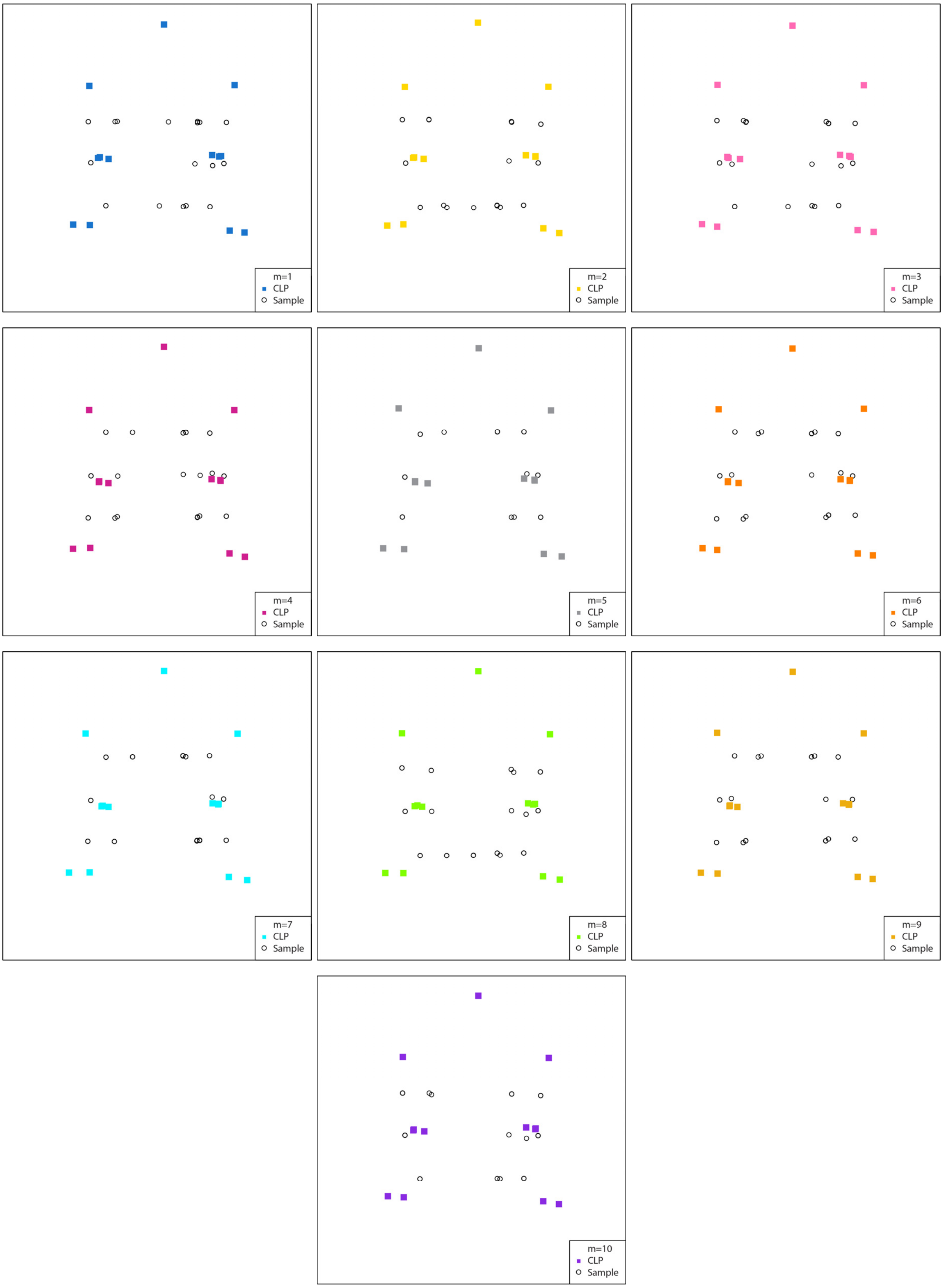


Figure 4.7 Illustration of transformed GPA MCA biplots for ten MIs. Each panel represents the GPA MCA biplot for a particular MI for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

The MCA biplots are now aligned and thus comparable. Minimal variation is observed between the panels of Figure 4.7, but the configurations are not identical which shows that the uncertainty incorporated by the MI procedure is still preserved after GPA.

The variation between imputations can be visually explored by constructing a plot of the transformed CLPs. The CLPs of the transformed GPA biplots are now superimposed onto one configuration in Figure 4.8 to evaluate the similarities between the aligned configurations.

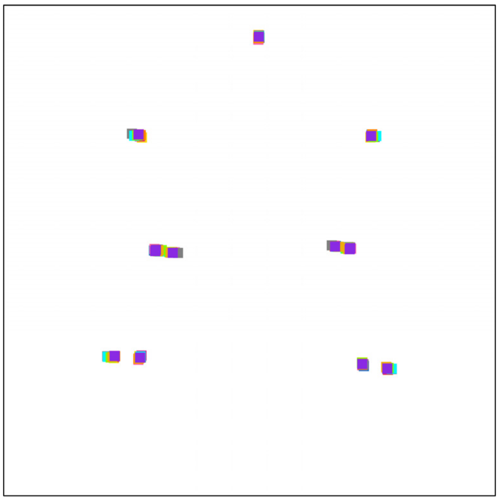


Figure 4.8 Superimposed CLPs from Figure 4.7 for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

After GPA the aligned CLPs in Figure 4.8 are in close proximity of particular locations which follow the same pattern. Therefore, the GPA procedure successfully aligns the configurations and improves visual interpretation between multiple configurations constructed from multiple imputed data sets.

4.3.3 Combining the aligned multiple imputed configurations

This section elaborates on the methodology for the ‘bin’-part of the GPAbin procedure.

Now, the mean coordinate matrices are calculated from the aligned GPA configurations. Combining the coordinates of visualisations of MIs mimics the use of Rubin’s rules (cf. 3.4.4.3) to combine estimates obtained from analyses on MIs. The coordinates are regarded as the estimates of interest since the focus is to visualise missing data. As stated by Schafer and Olsen (1998), the quantity of interest can refer to any estimated quantity of the population. Rubin’s rules rely on samples that are approximately normally distributed. However, the distribution of the coordinates is not of importance for the execution of the methodology presented in this project, since no statistical inferences are drawn from the (estimated) coordinates directly. This notion is also confirmed by Van Ginkel and Kroonenberg (2014) in the context of MI in PCA. The combined coordinates provide a holistic view of the imputations and therefore an assumption of normality is not required.

The combined set of coordinates (estimates) after optimally aligning the coordinates in the GPA procedure, are referred to as the GPAbin coordinates and are obtained from the following equations:

$$\widehat{\mathbf{CLP}}^* = \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathbf{CLP}}_m^*$$

and

$$\hat{\hat{\mathbf{Z}}}^* = \frac{1}{M}\sum_{m=1}^{M} \hat{\mathbf{Z}}_m^* \, .$$

The `GPAbin()` function is presented in Appendix E.

The GPAbin biplot of the example data set is presented in Figure 4.9. A total of 13 CLs are presented in Figure 4.9: variable A (‘one’, ‘two’), variable B (‘one’, ‘two’), variable C (‘one’, ‘two’), variable D (‘one’, ‘two’, ‘thr’) and variable E (‘one’, ‘two’, ‘thr’, ‘fou’).

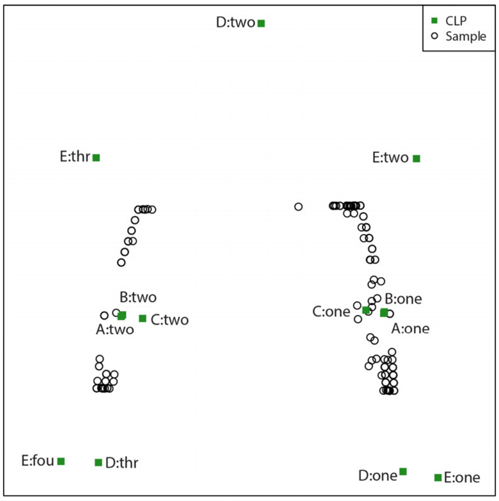


Figure 4.9 The GPAbin biplot for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

The GPAbin biplot (cf. Figure 4.9) enables a global representation of separate visualisations presented in Figure 4.5. It is observed that the responses for variables A, B and C are highly associated with the same CL values located in close proximity. The second CL for variable D (D: two) is not strongly associated to other responses. High associations between the responses for variables D and E are observed; the second CLs for both variables (D: two and E: two), as well as the third CL for variable D (D: thr) and the fourth CL for variable E (E: fou) are located in close proximity.

4.4 Prediction methods

The two prediction methods rely on the same methodology to predict categorical responses from visualisations by considering the Euclidean distances between a sample point and each CLP of a particular variable. The CLP per variable with the shortest Euclidean distance to the sample point is the CL assigned for the response in the predicted categorical data set. Therefore, considering an arbitrary sample i and variable j, the CLP with the shortest Euclidean distance to the sample point, will be the predicted CL, i.e. where the minimum of the expression:

$$\sqrt{\sum_{s}^{S}\sum_{k}^{K}(\mathrm{z}_{s}-\mathrm{clp}_{ks})^{2}},$$

occurs. The maximum number of dimensions available from the MCA solution is indicated by S and the letter, K, is the total number of CLs for a particular variable, j.

All available dimensions of the coordinate matrices are used to determine the shortest Euclidean distances between sample points and CLPs. Therefore, all available information is used for the estimation and not only the two-dimensional visualisation space. The responses that were initially observed in the data set containing missing observations are inserted in the predicted data set to ensure that the correct observed responses are maintained.

4.4.1 Majority rule prediction

The Majority rule prediction considers each multiple imputed MCA biplot separately and uses the distances (cf. 4.4) to predict the CLs of the missing responses. Therefore, multiple predicted categorical data sets are available and then combined into a final predicted categorical data set based on the frequencies of the predicted CLs. The CL per sample and per variable, for the multiple predicted data sets, with the highest frequency is assigned as the final predicted response. If there are multiple CLs with equal frequencies that are in the majority, a random CL is assigned from the plausible CLs with the highest frequency.

The Majority rule prediction is illustrated in Figure 4.10, showing a plot for each imputation and linking a chosen sample (third sample illustrated in Figure 4.10) to the CLPs allocated based on the shortest Euclidean distances. In some cases, the allocated CLP might not have the shortest distance to the illustrated sample in the approximated two-dimensional plot, since the allocation is based on the overall shortest Euclidean distance which is determined in higher dimension.

Overall, the MIs resulted in the same CL predictions with the exception of imputations one ($m = 1$) and four ($m = 4$), which resulted in a different CL for variable E than the other imputations. Since the CL with the highest frequency of predictions will be allocated as the final predicted response, the final predicted CLs for the third sample are: A: one, B: one, C: one, D: one and E: one.

The functions for the Majority rule prediction method are available in Appendix H. The `MajPred()` function predicts CLs separately for multiple visualisations and the `MajRule()` function combines the predicted CLs into a final predicted categorical data set.

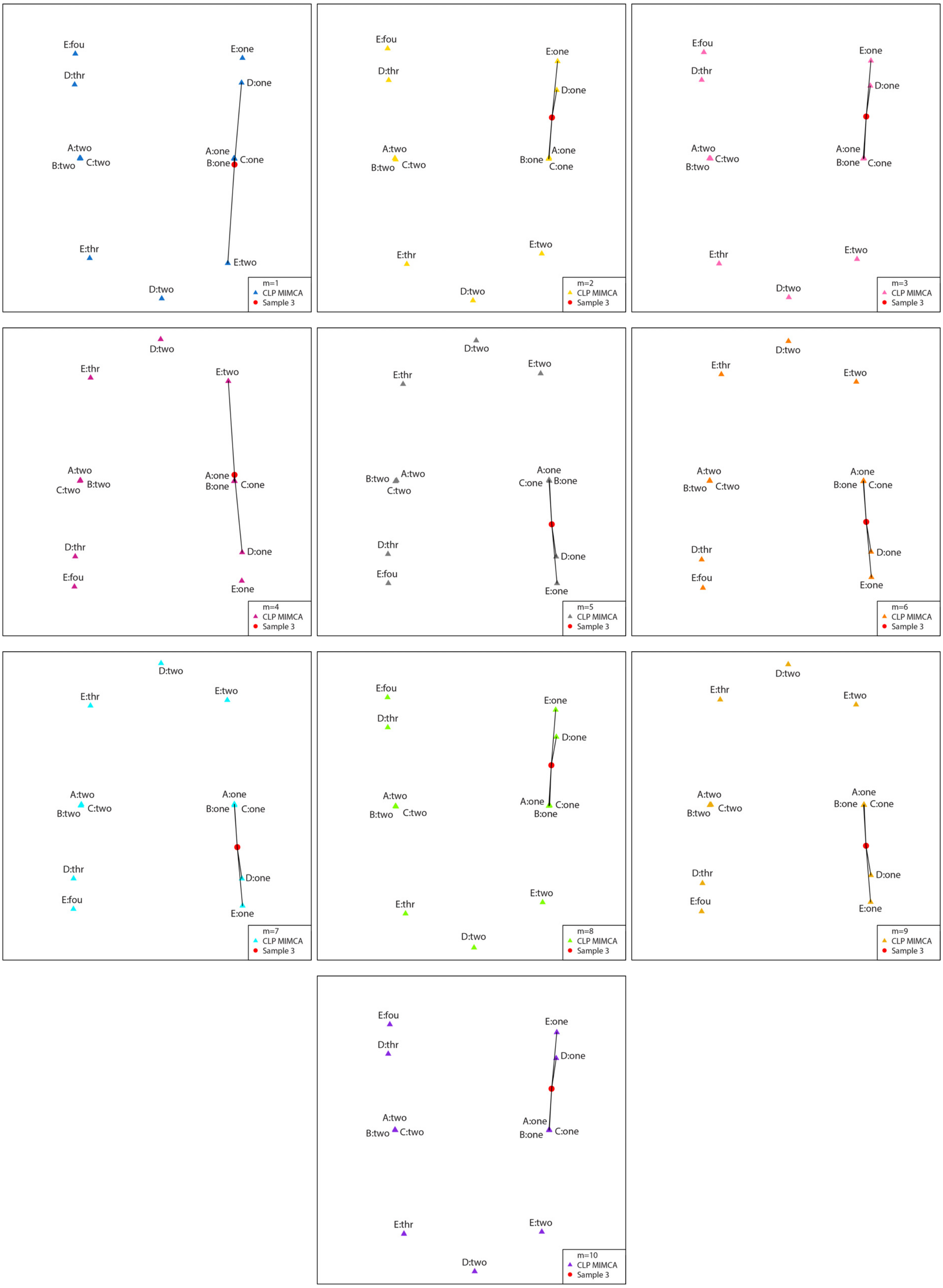


Figure 4.10 Illustration of predictions from MCA biplots after MIMCA (cf. Figure 4.5) for the third sample of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

4.4.2 GPAbin prediction

The GPAbin prediction method predicts a categorical data set from a GPAbin biplot (cf. 4.3). Predictions are made by utilising all available dimensions of the GPAbin biplot (cf. Figure 4.9). The predicted CLs are allocated per variable to CLPs with the shortest Euclidean distance to a particular sample (third sample illustrated in Figure 4.11). The prediction method is illustrated in two dimensions in Figure 4.11 by annotating one sample to convey the method.

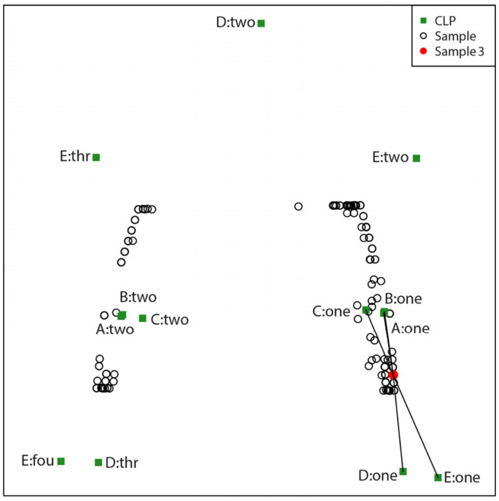


Figure 4.11 Illustration of predictions from GPAbin biplot (cf. Figure 4.9) for the third sample of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

In Figure 4.11 it is observed that the selected third sample is in the centre of the CLPs that resulted in the shortest Euclidean distance in higher dimension. The final predicted CLs for the third sample are: A: one, B: one, C: one, D: one and E: one. This is the same as the predictions made for the third sample using the Majority rule prediction approach (cf. 4.4.1).

The `GPAPred()` function (cf. Appendix I) contains the code to generate a predicted categorical data set from one visualisation, in this case from the GPAbin configuration.

4.5 Subset multiple correspondence analysis

This technique allows separate analyses on subsets of a multivariate categorical data set while maintaining the initial contribution of the CLs of the complete data set used for standard MCA. The column and row masses (marginal sums) obtained for standard MCA are fixed when performing sMCA on the chosen subset. By fixing the masses, the total variation (inertia) of the complete data set can be compartmentalised into the different subsets and therefore the initial contributions to the analysis are preserved (Greenacre, 2017; Greenacre & Pardo, 2006b). Greenacre and Pardo (2006a) propose to first perform MCA on the entire data set which can lead to the discovery of CLs that can be further investigated by focusing the analysis on a certain subset with sMCA. This form of MCA is particularly useful for the analysis of missing data, since the missing responses can be separated from the observed responses by active handling of missing values (cf. 3.4.2.2). Performing sMCA on a subset of missing information is equivalent to the missing passive modified margin method (cf. 3.4.2.1) (Josse *et al.*, 2012). The sMCA method to handle missing values is easy to perform since no imputation methods have to be applied and it has the advantage of maintaining the sample size, since no deletion is applied. Another benefit is that no observed information is forfeited or compromised, since the missing values are not estimated, but simply coded as an additional missing CL.

Apart from the benefits for missing data analysis, sMCA also provides a visualisation benefit in the sense that it improves visual representations where large data sets become uninterpretable due to cluttered displays. A set of visualisations can be used with different subsets of the data, whilst maintaining the explained variation of the analysis.

In order to expose the missing data structures, the indicator matrix is recoded by adding new CLs for missing responses using active data handling techniques (cf. 3.4.2.2). Van der Heijden and Escofier (2003) discuss the suitability of single and multiple active handling for specific MDMs. Since single active handling only creates one missing CL per variable, it is assumed that all samples with a missing response for a particular variable are similar. This assumption might not be a true reflection of the MDM and therefore single active handling could be inappropriate if the data are missing due to an MCAR MDM. Multiple active handling seems more representative of the population, since no assumptions are made regarding the samples, but since each unique CL will only consist of one response, samples with a large portion of missing observations might result in outliers.

This being said, in the context of this research, the single active handling is used to distinguish between missing and observed information irrespective of MDM and forms part of a step in the process of treating missing values before obtaining final inferences. Therefore, the single active handling enables the formation of two subsets which are available for separate investigation as opposed to using the actively handled data set as a whole for standard complete data analysis methods.

Again, MCA is applied by performing CA on the indicator matrix (cf. 2.7 and 2.7.1). However, for sMCA a subset of the recoded indicator matrix, $\mathbf{G}_h$, is isolated for the analysis. The subset indicator matrix, $\mathbf{G}_h$, is transformed by the diagonal matrices of row weights from the full recoded indicator matrix ,$\mathbf{G}_{(single\ active)}$, and column, $\mathbf{C}$, weights from the subset indicator matrix, $\mathbf{G}_h$:

$$\mathbf{S} = p^{-1/2}\mathbf{G}_{(n\times h)}\mathbf{C}_{(h\times h)}^{-1/2}.$$

The SVD of $\mathbf{S}$ results in the following:

$$\mathbf{S} = \mathbf{U}_{(n\times r)}\mathbf{\Lambda}_{(r\times r)}\mathbf{V}'_{(r\times h)},$$

where n is the number of samples, r the reduced dimension size and h the number of CLs in the particular subset. The singular vectors are represented in $\mathbf{U}$ and $\mathbf{V}$ with the singular values represented in decreasing order in $\mathbf{\Lambda}$ (Greenacre, 2017; Greenacre & Pardo, 2006b).

The sMCA biplot is constructed by plotting the samples using the first two columns of the following matrix multiplication:

$$p^{-1/2}\mathbf{U}_{(n\times 2)}\mathbf{\Lambda}_{(2\times 2)}.$$

The CLPs are obtained from using the first two columns of the following matrix multiplication:

$$\mathbf{C}_{(h\times h)}^{-1/2}\mathbf{V}_{(h\times 2)}$$

(Gower *et al.*, 2011).

4.5.1 Code: Subset multiple correspondence analysis

The original `mjca()` function (Nenadić & Greenacre, 2007) was adapted to handle small eigenvalues that are approximately equal to zero, but misrepresented by floating point number representation. The minor additions to the original `mjca()` function are presented in the `adap.mjca()` function in Appendix J.2.

Aside from the two changes made to the default `mjca()` function, sMCA can be performed by specifying the '`subsetcat`' argument in the following default argument list:

```
mjca <- function (obj, nd = 2, lambda = c("adjusted",
"indicator", "Burt", "JCA"), supcol = NA, subsetcat = NA, ps =
":", maxit = 50, epsilon = 1e-04, reti = FALSE, ...)
```

The `'subsetcat'` argument requires a vector containing the column numbers of the recoded indicator matrix (cf. 3.4.2.2) to be used for the sMCA.

The sMCA function calls are presented in the `mysMCA()` function in Appendix J.1. The `mjca()` function is available in the `R` package, `ca` (Nenadić & Greenacre, 2007).

The sMCA biplot enables a visualisation which isolates the observed responses after subsetting in Figure 4.12.

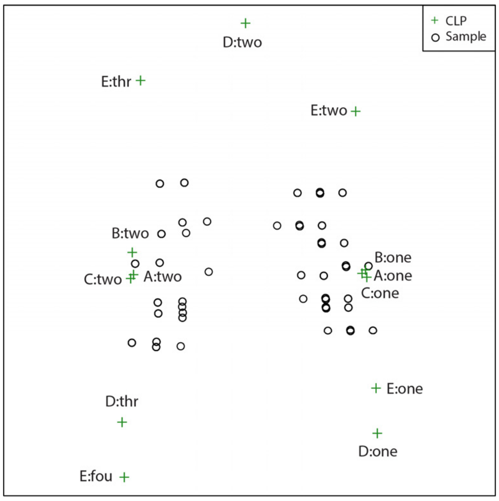


Figure 4.12 The sMCA biplot (observed subset) for the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

Certain CLP groupings are observed in Figure 4.12. Coordinates that are located in close proximity indicate strong associations between responses. The coordinate groupings will be further considered in the following section (cf. 4.6) on the comparison of configurations.

4.6 Comparison with simulated data

In order to determine the success of the proposed methods, MCA is also performed on the complete simulated data sets, $\mathbf{X}$, to enable a comparison between the configurations from the complete data sets and the completed data sets from the various proposed methods.

Figure 4.13 shows the visualisations for the complete MCA biplot (left panel) and the missing data visualisation approaches (GPAbin: middle panel and sMCA: right panel).

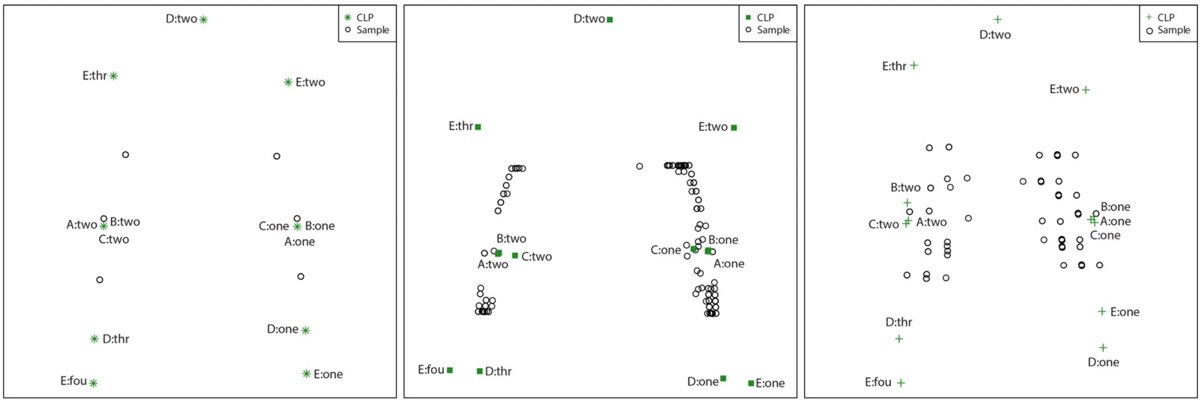


Figure 4.13 Comparison of visualisation approaches for missing data compared to the complete MCA biplot. Left panel: complete MCA biplot (five variables and 1000 samples simulated from a uniform distribution). Middle panel: GPAbin biplot (cf. Figure 4.9). Right panel: sMCA biplot (complete subset) (cf. Figure 4.12). Middle- and right panels: 10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution.

In general, the CLPs of the three panels in Figure 4.13 have similar placements and therefore the relationships between variables are preserved in the GPAbin biplot (cf. Figure 4.13: middle panel) and sMCA biplot (cf. Figure 4.13: right panel) that are observed in the complete MCA biplot (cf. Figure 4.13: left panel). The samples of the sMCA biplot (cf. Figure 4.13: right panel) show more horizontal variation compared to the samples in the GPAbin biplot (cf. Figure 4.13: middle panel) and complete MCA biplot (cf. Figure 4.13: left panel).

The similarities between the visualisations in Figure 4.13 could be due to the low percentage of missing values (10%) of the example data set.

The success of the missing data approaches can be quantified by performing OPA on the complete configuration and the missing data visualisation approaches separately. The simulated complete configuration is set as the target configuration and the GPAbin and sMCA configurations are set as the testee configurations, respectively. The following notation is adopted for the sample coordinate matrix, $\mathbf{Z}$, and CLPs coordinate matrix, $\mathbf{CLP}$, of the complete MCA biplot. The notation is generalised for the formulation of the measures of comparison, the target coordinate matrices are simply referred to as $\mathbf{Z}_{Target}$ and $\mathbf{CLP}_{Target}$ and the testee coordinate matrices are denoted by $\mathbf{Z}_{Testee}$ and $\mathbf{CLP}_{Testee}$, respectively.

The notation provided for OPA in Section 2.9 will now be redefined in terms of the coordinate matrices of the CLPs. The minimum of the following sum of squares (SS) is required:

$$SS = \left\| \mathbf{s}(\mathbf{CLP}_{Testee})\mathbf{Q} - \mathbf{CLP}_{Target} \right\|^2.$$

As noted previously, the translation is incorporated by centring the configurations at the origin before performing the Procrustes analysis.

The CLPs of the complete MCA biplot of the example data set (five variables and 1000 samples simulated from a uniform distribution) are presented as the target CLPs in Figure 4.14. The testee CLPs in Figure 4.14 are the CLPs of the GPAbin biplot of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution).

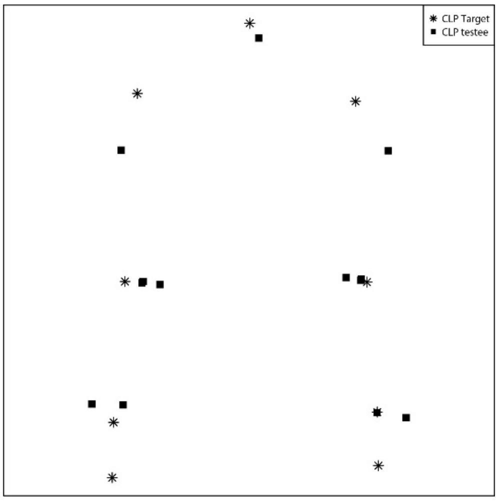


Figure 4.14 The CLP plot for the target configuration (CLPs from complete MCA) and testee (CLPs from GPAbin biplot) configuration before OPA.

The OPA between the complete MCA biplot CLPs and the GPAbin CLPs is illustrated in two steps in Figure 4.15: reflection and rotation (cf. Figure 4.15: left panel) and scaling (cf. Figure 4.15: middle panel). The translation is not illustrated, since the MCA coordinates are centred before applying OPA. The right panel of Figure 4.15 compares the updated testee configuration with the target configuration by superimposing the CLPs of the two configurations in the same figure.

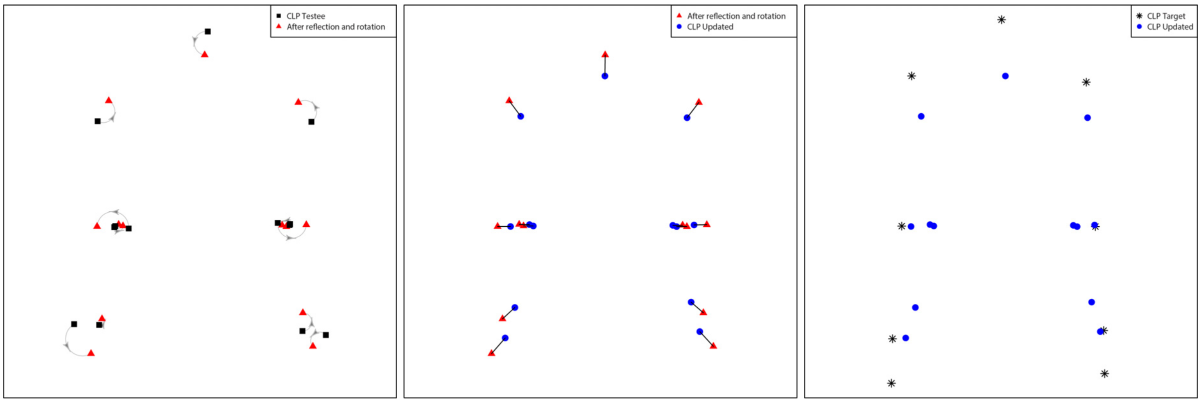


Figure 4.15 The OPA steps. Left panel: Reflection and rotation of testee CLPs (GPAbin CLPs). Middle panel: Scaling of updated CLPs (reflected and rotated). Right panel: Target CLPs compared to updated testee CLPs.

The updated testee configuration follows a similar CLP pattern as the target configuration (cf. Figure 4.15: right panel). The comparison of the right panel of Figure 4.15 with Figure 4.14 confirms that the OPA transformations successfully aligns the testee configuration to the target configuration.

4.6.1 Measures of comparison

Five measures of comparison are used to evaluate the proposed methods, by comparing the complete data and completed data configurations: Procrustes statistic (PS), congruence coefficient (CC), absolute mean bias (AMB), mean bias (MB) and root mean squared bias (RMSB). The function `fitMeas()` contains the necessary code to calculate the measures of comparison between two configurations (cf. Appendix C). These measures will be used to determine which methods succeed in preserving the relationships between the variables and samples across the variety of simulation scenarios. Also, to determine which proposed method best replicates the associations present in the original complete configuration. All MCA biplots are constructed in two-dimensional space. All measures of fit are computed in duplicate, once in two dimensions relating to the visual representation and once in higher dimensions using the maximum number of available dimensions in both the target and testee configurations. The difference between the two- and maximum dimensions might be regarded as comparing extremes, but this will establish whether the displayed two dimensions provide sufficient information regarding the fit of the configurations and also expose the effect of excluding the higher dimensions. This naturally progresses to the investigation of which dimensions should be selected to best resemble the target configuration. Methodology has been developed by Alves (2012) to select dimensions for predictive biplots in PCA for continuous data, which provides foundation for future research in categorical data. This however will not be considered in this research.

a) Procrustes statistic

The residual sum of squares distances between the updated (transformed) testee configuration and target configuration can be scaled and used as a measure of fit, referred to as the PS:

$$PS = \frac{tr\left(\left(s\mathbf{CLP}_{Testee}\mathbf{Q} - \mathbf{CLP}_{Target}\right)'\left(s\mathbf{CLP}_{Testee}\mathbf{Q} - \mathbf{CLP}_{Target}\right)\right)}{tr\left(\mathbf{CLP}_{Target}{}'\mathbf{CLP}_{Target}\right)}.$$

This measure evaluates the similarities between the target and the transformed testee after the application of the admissible transformations (translation, dilation, rotation and reflection). The PS is a measure between zero and one, with a smaller value indicating a good fit (Cox & Cox, 2001).

b) Congruence coefficient

The CC compares the Euclidean distances (d) between the coordinates of the target and testee configurations before the application of OPA (Borg & Groenen, 2005). Since distances and dissimilarities are non-negative, the CC will always be a value between zero and one (Tucker, Koopman & Linn, 1969; Zegers & Ten Berge, 1985).

The calculation of the CC is as follows:

$$CC = \frac{\sum_{s=1}^{S}\sum_{k=1}^{K} d_{ks}\left(\mathbf{CLP}_{Target}\right)d_{ks}\left(\mathbf{CLP}_{Testee}\right)}{\sqrt{\sum_{s=1}^{S}\sum_{k=1}^{K} d_{ks}^2\left(\mathbf{CLP}_{Target}\right)}\sqrt{\sum_{s=1}^{S}\sum_{k=1}^{K} d_{ks}^2\left(\mathbf{CLP}_{Testee}\right)}},$$

where K is the total number of CLPs and S is the number of dimensions used for the approximation of the MCA solutions that will be compared in OPA. The Euclidean distance between two CLPs (where clp_{is} and clp_{js} are the is^{th} and js^{th} elements of $\mathbf{CLP}$, respectively) over S dimensions is obtained by: $d_{ij}(\mathbf{CLP}) = \sqrt{\sum_{s=1}^{S}\left(\mathrm{clp}_{is} - \mathrm{clp}_{js}\right)^2}$. Again, only two values of S are considered for the purpose of this research; two and the maximum number of dimensions, as previously discussed.

This measure is interpreted analogous to the coefficient of determination in regression analysis. Therefore, it can be regarded as the amount of explained variation. A CC value close to one indicates configurations that are congruent. The CC has the disadvantage of taking on values close to one, even if the fit is not perfect and should be used in comparison with other measures and visualisations (Borg & Groenen, 2005).

c) Measuring bias

The MB measure is used to show systematic over- and underestimation (Van Ginkel & Kroonenberg, 2014; Walther & Moore, 2005). An estimate is regarded as being unbiased when the MB value is equal to zero or close to zero. The MB can be calculated using the following formula:

$$MB = \frac{1}{KS}\sum_{k=1}^{K}\sum_{s=1}^{S}\left(\mathrm{clp}_{ks\,(Testee)} - \mathrm{clp}_{ks\,(Target)}\right).$$

The square root of the squared distances between the coordinates of the testee and target configurations are used for the calculation of the RMSB:

$$RMSB = \sqrt{\sum_{k=1}^{K}\sum_{s=1}^{S}\left(\mathrm{clp}_{ks\,(Testee)} - \mathrm{clp}_{ks\,(Target)}\right)^{2}/KS}.$$

The RMSB measure provides insight into the magnitude of deviations of the testee coordinates from the target coordinates, similar to a standard deviation. Since the differences are squared, this measure of comparison can be influenced by outlier values. This potential problem is addressed by using the absolute difference as opposed to the squared difference (Walther & Moore, 2005).

This measure is referred to as the AMB:

$$AMB = \frac{1}{KS}\sum_{k=1}^{K}\sum_{s=1}^{S}\left|\mathrm{clp}_{ks\,(Testee)} - \mathrm{clp}_{ks\,(Target)}\right|.$$

Comparing different data set scenarios and methods, the smallest RMSB (or AMB) in comparison to other RMSB (or AMB) values will indicate unbiased representation of the target configuration.

The measures of comparison obtained from the example presented in Figure 4.14 and Figure 4.15, are as follows:

Table 4.1 Measures of comparison for the GPAbin CLPs of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution) compared to the CLPs of the complete MCA biplot (five variables and 1000 samples simulated from a uniform distribution).

Measures	Two dimensions	Maximum dimensions
PS	0.0358	0.3284
CC	0.9937	0.9842
AMB	0.1602	0.6798
MB	-0.0333	-0.1840
RMSB	0.2284	1.1261

The RMSB and AMB measures cannot be interpreted in isolation, but satisfactory results are obtained for the remaining three measures; the CC values are close to one, the PS values and MB values are close to zero. All of the measures of comparison perform slightly better when the configurations are compared in two dimensions. A detailed discussion of the results of the simulated data scenarios (cf. Chapter 5) is given in Chapter 6 (cf. 6.3).

4.6.2 Further visual comparison

Convex hulls are useful visual tools to highlight groupings and patterns that occur in configurations. Groupings that occur between CLPs in the simulated complete MCA biplot can be isolated by manually constructing convex hulls to enclose them. The CLPs that are enclosed in a particular convex hull indicate relative strong associations and are expected to also be in close proximity in the completed configurations. The CLPs that are used to construct the convex hulls in the complete biplot are also used to construct convex hulls in the completed configurations enabling the viewer to quickly grasp whether the natural groupings are preserved in the completed configurations without numerical confirmation from the measures of comparison.

The example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution) results in CLPs in close proximity and therefore convex hulls will not enhance the visual interpretation due to small coverage areas. An additional example data set is presented in this section with 50% missing values also with a MAR MDM with five variables and now with a 100 samples.

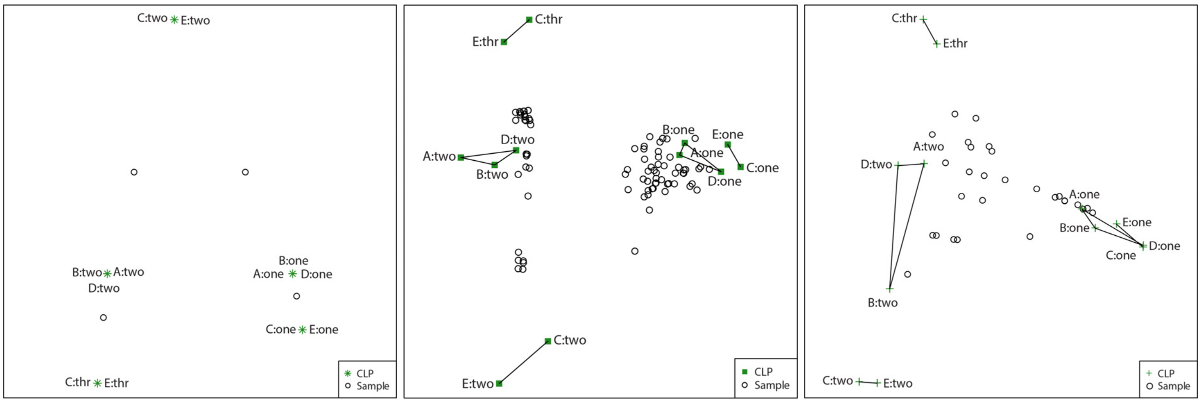


Figure 4.16 Manually constructed convex hulls to mimic groupings in complete biplots. Left panel: complete MCA biplot (five variables and 100 samples simulated from a uniform distribution). Middle panel: GPAbin biplot. Right panel: sMCA biplot (complete subset). Middle- and right panels: 50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution.

The right- and middle panels of Figure 4.16 include convex hulls that are manually constructed according to the groupings of CLPs that occur in the simulated MCA biplot in the left panel (cf. Figure 4.16). Since the CLPs of the simulated complete MCA biplot (cf. Figure 4.16: left panel) are in very close proximity, convex hulls are not constructed. However, the groupings are used as markers for the convex hulls that are constructed for the GPAbin- and sMCA biplots of the completed data (cf. Figure 4.16: middle- and right panels).

The higher percentage of missing values (50%) in the additional example data set results in less comparable missing data visualisations to the complete MCA biplot than was observed from the visualisations in Figure 4.13 with 10% missing values. The CLP groupings are preserved in both the sMCA biplot (cf. Figure 4.16: right panel) and GPAbin biplot (cf. Figure 4.16: middle panel) with slightly larger convex hull areas observed in the sMCA biplot (cf. Figure 4.16: right panel) than the GPAbin biplot (cf. Figure 4.16: middle panel). The samples in the GPAbin biplot (cf. Figure 4.16: middle panel) show a horizontal separation as is observed from the complete MCA biplot (cf. Figure 4.16: left panel). This sample separation is not reflected in the sMCA biplot (cf. Figure 4.16: right panel).

4.6.3 Similarity percentage

Another approach is followed to compare the complete- and completed data configurations which focus on the visual interpretation in two dimensions by using the distances between the sample points and CLPs. The visual interpretation refers to the plausible response value (i.e. CL) of a sample for a particular variable. The CLP with the shortest distance to a sample point, per variable, is regarded as the final interpretation of the association between a variable response and sample. After the strongest associations between the sample points and CLPs have been identified for all complete data configurations and completed data configurations (GPAbin, sMCA and RIMCA), the identified CLs of the completed configurations are compared to the CLs identified from the complete configuration. This resonates with the proposed projection procedures (cf. 4.4), but now the aim is to obtain the visual interpretation and not to predict a final categorical data set. Therefore, the actual responses of the original complete simulated categorical data set is not compared to the CLs that are identified from the completed configurations.

The number of identical CLs that are identified in the complete- and completed configurations are counted and expressed as a similarity percentage. The similarity percentage expresses the success of the visual interpretation from the completed data configurations in comparison to the complete simulated data configuration.

The `GPApred()` function is used for the prediction of the visual interpretation (cf. Appendix I) by only using the approximated coordinates in two dimensions. As mentioned in Section 4.4.2, the `GPApred()` function predicts CLs from a single configuration of samples and CLPs. The output of the `GPApred()` function is further analysed using the `sim.pct()` function (cf. Appendix C.2).

The similarity percentages of the configurations presented in Figure 4.13 of the example data set (10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution) are presented in Table 4.2.

Table 4.2 Similarity percentages obtained from GPAbin and sMCA when compared to the complete MCA biplot in two dimensions, respectively. Example data set: 10% missing values with a MAR MDM with five variables and 1000 samples simulated from a uniform distribution

	GPAbin	sMCA
Number of matches	4939/5000	4933/5000
Similarity percentage	98.78%	98.66%

Both approaches (GPAbin and sMCA) achieve high similarity percentages when compared to the complete MCA biplot (cf. Table 4.2). A slightly higher similarity percentage is obtained from the GPAbin approach. A detailed discussion of the results of the simulated data scenarios (cf. Chapter 5) is given in Chapter 6 (cf. 6.2).

4.7 Clustering

The sMCA biplot (cf. 4.5) of the missing subset is explored before a clustering technique is applied. In order to illustrate the techniques in this section the additional example data set considered in Section 4.6.2 will be presented (50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution). The visualisation of the methodology in this particular section will be enhanced by presenting the additional example data set, since a higher percentage of missing values results in a higher frequency of responses in the missing data subset. This statement will be resumed in the discussions of Chapter 8.

Based on the conditions of the simulated MAR MDM (cf. 5.4), one variable will be fully observed (in this example, variable A). Therefore, only four missing CLs are available when using the single active handling technique (cf. 3.4.2.2).

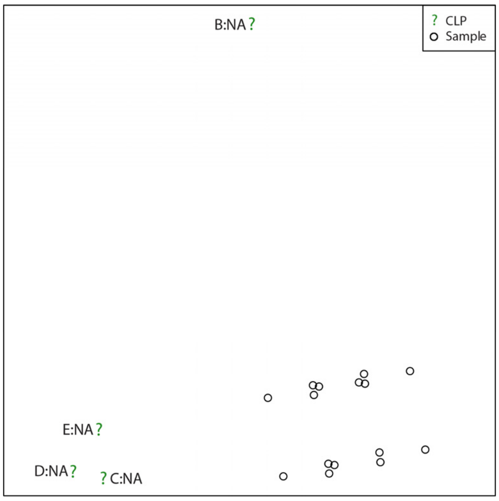


Figure 4.17 The sMCA biplot (missing subset) of the additional example data set (50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution).

The CLPs in Figure 4.17 occur in two groups: C: NA, D: NA and E: NA are situated in close proximity and B: NA is separated from the rest. Since a clear separation is visible in a two-dimensional visual representation, a clustering technique can be applied to the coordinates of the missing subset of CLPs to confirm whether a satisfactory clustering structure is present.

The well-known pam method (Kaufman & Rousseeuw, 1987) is implemented to identify distinguishable clusters between the CLPs in the sMCA biplots of the missing subsets using the dissimilarities (distances) between coordinates. A medoid is referred to as a representative object which has the shortest average distance to the other data points (or coordinates) of interest. The data points closest to the medoid form a cluster (Kaufman & Rousseeuw, 1987; Struyf, Hubert & Rousseeuw, 1997). The aim is to determine whether a sufficient clustering structure exists for the CLPs, since this could lead to emphasizing the association between missing responses and subsequently identifying the MDM. Since cluster analysis is applied on the reduced dimension sMCA solution, this is regarded as a tandem clustering approach (Mitsuhiro & Yadohisa, 2015).

In order to determine whether the number of predetermined medoids successfully discriminates between the clusters, the average silhouette width is evaluated. The silhouette value is obtained by first calculating the average dissimilarity of all objects in a specific cluster, say $C1$, to its medoid and then identifying the closest neighbouring cluster, say $C2$, for each object in $C1$. The silhouette width value provides a ratio between the allocated cluster and the second-best option for each object (CLP for the methodology presented in this chapter).

Silhouette width values are calculated for all coordinate points and then averaged to provide a global measure of fit, referred to as the average silhouette width or silhouette coefficient, $s(i)$. Silhouette coefficients take on values between -1 and 1 with the following interpretations (Struyf *et al.*, 1997):

- $s(i) \approx -1$, the classified CLP is closer to the second-best medoid than the allocated medoids which results in unsuccessful classification;
- $s(i) \approx 0$, the classified CLP lies between two medoids;
- $s(i) \approx 1$, the CLP is well classified.

Fixed guidelines to decide whether a silhouette coefficient reflects substantial clustering structures are not available. However, silhouette coefficients exceeding 0.5 is regarded as an above average measure reflecting the efficient identification of clustering structures. Struyf *et al.* (1997) propose that a silhouette value below 0.25 is an indication that no notable clusters are present in a data set. Carrying forward, a silhouette coefficient of at least 0.5, $s(i) \geq 0.5$, will be regarded as an indication of well separated clusters, which illustrates a high association between missing response CLs. A silhouette coefficient below 0.5, $s(i) < 0.5$, will be indicative of no substantial clustering structures and therefore low association of missing response CLs.

The pam method is available in the `R` package, `cluster` (Maechler *et al.*, 2017). All possible numbers of clusters are evaluated between two and $(K_{miss} - 1)$, where K_{miss} is the total number of missing CLs. Only the first two dimensions of the sMCA solution are used to correspond to the two-dimensional visualisations.

The additional example data set can therefore be clustered about two or three medoids and is presented in the panels of Figure 4.18. Different colours and symbols are used to indicate the group separations.

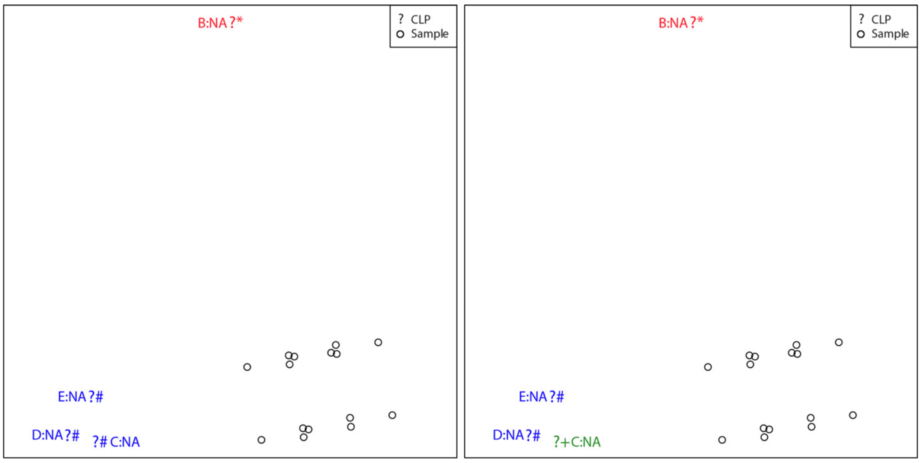


Figure 4.18 The pam clustering of CLPs in an sMCA biplot (missing subset) of the additional example data set (50% missing values with a MAR MDM with five variables and 100 samples simulated from a uniform distribution). Left panel: $k = 2 \rightarrow s(i) = 0.6800$ *. Right panel:* $k = 3 \rightarrow s(i) = 0.1848$*.*

Both the visual separation in the panels of Figure 4.18 and the silhouette coefficients suggest that the CLPs can be sufficiently separated into two clusters.

4.8 Conclusion

This chapter presented the complete methodology to achieve the aim and study objectives of this project as outlined in Chapter 1. Reference was made to the relevant R functions required to execute the proposed theory in each section and where appropriate the methodology was also illuminated by means of configurations of two example data sets.

The following chapter, Chapter 5, will present the simulation protocol of this research study. The simulation of different data set scenarios, computational challenges, utilisation of high performance computing (HPC) clusters and the reproducibility of the results in this research will be addressed.

Chapter 5
Simulation Protocol

5.1 Introduction

In this chapter the simulation protocol is discussed. Firstly, the generation of complete data sets are discussed using various approaches to ensure a variety of possible outcomes. Then artificial missingness is created in the complete data sets using varying percentages of missing values and different MDMs. These combinations merely provide plausible options that could occur in real applications.

5.2 Complete data sets: continuous

Continuous data sets are generated using three general distributions as a departure point with the following parameters:

- Number of samples (n):	100, 1000, 3000
- Number of variables (p):	5, 10, 15

The simulation distributions were chosen to create a variety of different data sets resulting in MCA biplots with varying patterns in order to evaluate the proposed methodology on a wide range of data possibilities.

5.2.1 Uniform distribution

The simulated uniform data sets are generated using the `runif()` function available in the `R` package, `stats` (R Core Team, 2017), to generate a specified number of samples for the number of variables. The variables are independently and univariately generated.

5.2.2 Skewed distribution

The Dirichlet distribution, also referred to as the multivariate generalised beta distribution, is chosen to generate a multivariate data set. The `rdirichlet()` function in the `R` package, `MCMCpack` (Martin, Quinn & Park, 2011), is used to generate the samples with corresponding shape parameters for the variables set to the default value of one ($\alpha = 1$).

If the shape parameters are all equal to one, the Dirichlet density is uniform over the simplex (Murphy, 2012; Schafer, 1997). However, when inspecting the univariate distributions of the generated variables, a positively skewed distribution is observed. Therefore, by proportionally assigning CLs, the first CLs will have a higher probability to be observed.

5.2.3 Symmetrical distribution

The multivariate normal distribution is used for the simulation of a symmetric distribution by using the `mvrnorm()` function in the `R` package, `MASS` (Venables & Ripley, 2002). In all scenarios the mean is set as a vector of zeros and a block covariance pattern is introduced to represent scenarios in which sets of variables are strongly correlated and have homogenous variance ($\sigma^2 = 1$). The strong correlations between variables were created to result in discerning patterns of sample coordinates and CLPs in the MCA biplots. It was found that unstructured covariance matrices or covariance matrices with homoscedastic uncorrelated variables result in MCA biplots with equally dispersed coordinates with no discerning response patterns. Two blocks of correlated variables are created in the covariance matrix: the first 66.7% of variables have the same correlation and the last 33.3% of variables are all equally correlated. All variables are however highly correlated with correlation coefficients exceeding 0.8.

5.3 Complete data sets: categorical

The categorical response variables are attained by sorting the simulated continuous responses and dividing the responses proportionally into the number of specified CLs per variable. This is executed by using the `cut()` function in the `R` package, `base` (R Core Team, 2017).

The number of CLs is fixed between two and five ([2; 5]), which are randomly assigned to a variable for each simulation. Consider the example in Table 5.1 for the allocation of two CLs for a continuous variable simulated from a uniform distribution with response values between zero and one. Observations with a value between (0; 0.5] will be allocated to the first CL and observations between (0.5; 1] will be allocated to the second CL.

Table 5.1 Allocating CLs from continuous values

Uniform response values	$(0; 0.5] \to A$ $(0.5; 1] \to B$	Categorical response values
0.2655		A
0.3721		A
0.5729		B
0.8984		B
0.4543		A

After the complete data sets are created the missing observations are inserted (cf. 5.4).

Selected examples of MCA biplots of the complete simulated data sets are presented in the figures below (cf. Figure 5.1 to Figure 5.3).

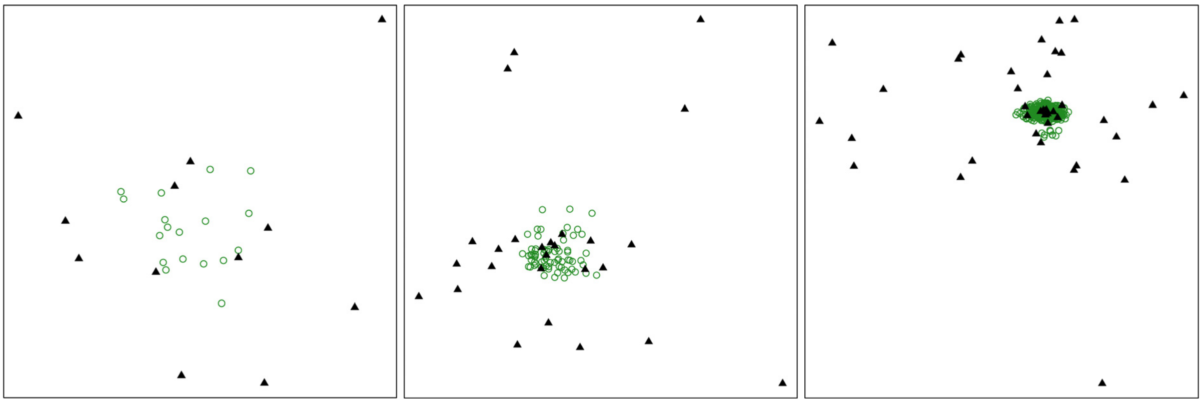


Figure 5.1 MCA biplots for Dirichlet simulations Left panel: $n = 100, p = 5$. Middle panel: $n = 100, p = 10$. Right panel: $n = 3000, p = 15$. Sample points are depicted by green open circles and CLPs by black triangles.

The MCA biplots of the Dirichlet simulations as presented in Figure 5.1 show no particular response patterns, since sample coordinates are centred with CLPs scattered within the plotting ranges of the configurations.

The MCA biplots of the Dirichlet distribution in Figure 5.1 show a homogenous placement of both sample coordinates and CLPs.

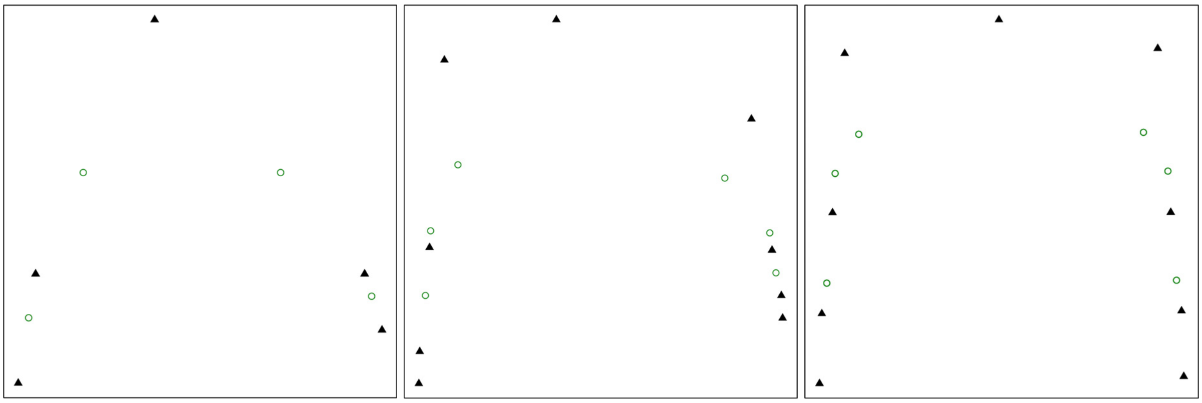


Figure 5.2 *MCA biplots for uniform simulations. Left panel:* $n = 100, p = 5$. *Middle panel:* $n = 100, p = 10$. *Right panel:* $n = 3000, p = 15$. *Sample points are depicted by green open circles and CLPs by black triangles.*

The uniform simulations resulted in a small number of unique response patterns, each with a high frequency, which are reflected in the coordinates of the MCA biplots in Figure 5.2. A majority of sample points and CLPs are equal and therefore plotted in the same locations.

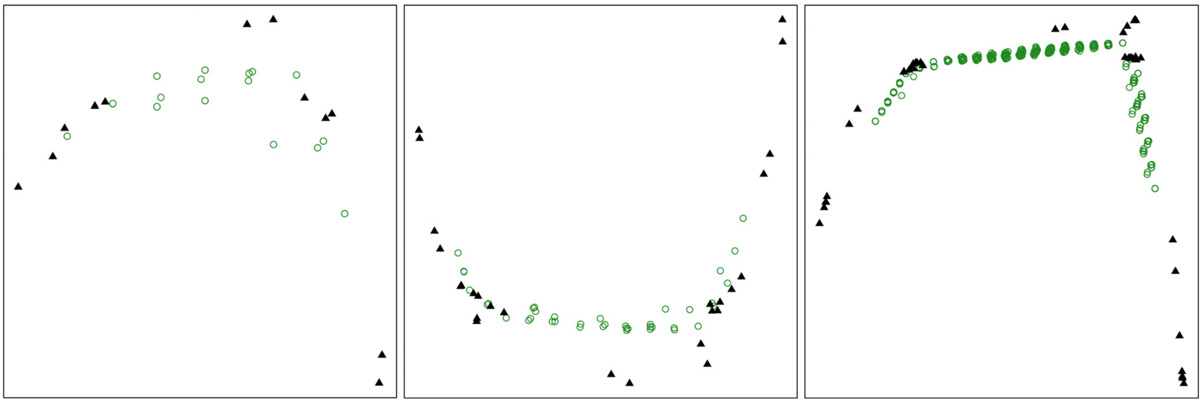


Figure 5.3 *MCA biplots for normal simulations. Left panel:* $n = 100, p = 5$. *Middle panel:* $n = 100, p = 10$. *Right panel:* $n = 3000, p = 15$. *Sample points are depicted by green open circles and CLPs by black triangles.*

The strong correlations between variables (cf. 5.2.3) result in the notable patterns in Figure 5.3. These patterns reflect the strong correlations between the CLs of each variable and is an illustration of the horseshoe effect, also known as the Guttman- or arch effect (Blasius & Thiessen, 2012). The CLPs are labelled for the MCA biplot presented in the left panel of Figure 5.3 and is given in Figure 5.4.

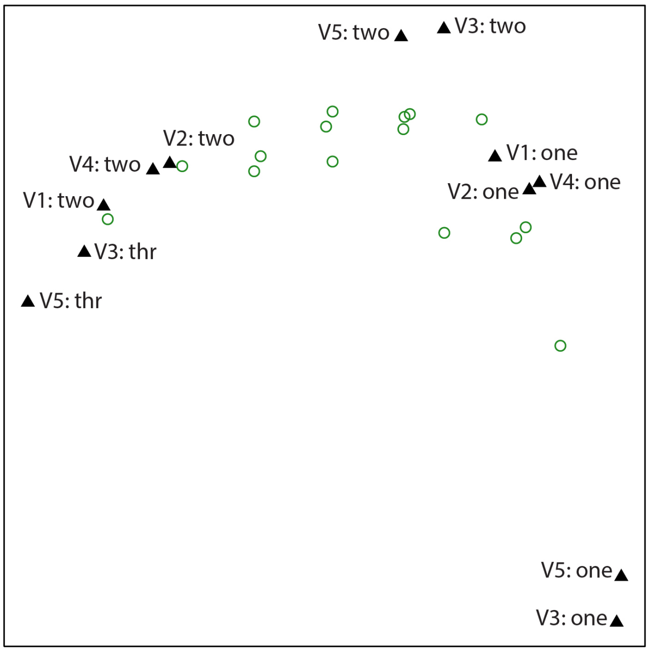


Figure 5.4 MCA biplot with CLP labels of Figure 5.3 (left panel)

Variables three (V3) and five (V5) have three CLs (‘one’, ‘two’, ‘thr’) each and variables one (V1), two (V2) and four (V4) only have two CLs (‘one’, ‘two’) per variable. The CLs of the variables with the same number of CLPs are in close proximity. Since the data are simulated from a normal distribution, the centre CLs are expected to occur more frequently. Consequently the first and last CLs are observed less frequently and occur to the end of the horseshoe pattern as would be the case with extreme CLs (Blasius & Thiessen, 2012).

The chosen simulation distributions result in a diverse range of simulated data sets. This is favourable to evaluate the proposed methodology on a variety of data possibilities.

5.4 Generating missingness

Artificial missingness is obtained by deleting specified percentages of missing values (10%, 30% and 50%) according to a MAR MDM or MCAR MDM.

The MCAR MDM is created by selecting a simple random sample with a size equivalent to the specified percentage of missing values from the complete data set. The responses of the selected sample are then deleted. The `myMCAR()` function to create missingness according to an MCAR MDM, is available in Appendix A.3.

In order to emulate the dependency of missing observations due to a MAR MDM a set of general conditions has been fixed that requires a data set to consist of categorical variables ($p_j, j = \{1,2,\ldots,J\}$) with at least two nominal CLs each. In the six conditions below, reference is made to the order of the variables and CLs, but this does not imply that the categorical variables are treated as ordinal:

- If the response of the first variable (p_1) is CL one or two, corresponding samples are randomly deleted for variables p_2 to p_{J-2} with increments of two.
- If the response of the centre variable ($p_{J/2}$) is one of the last two possible CLs, corresponding samples are randomly deleted for variable $p_{(J/2)+1}$.
- If the response of the second variable (p_2) is the median possible CL, corresponding samples are randomly deleted for variables p_{J-2} to p_J.
- If the response of the last variable (p_J) is equal to the last CL, corresponding samples are randomly deleted for the second variable, p_2.
- If the response of the third variable (p_3) is the median possible CL, corresponding samples are randomly deleted for variables p_2 to p_J with increments of two.
- If the first and last variables, p_1 and p_J, have the same response, corresponding samples are randomly deleted for variable p_J.

The conditions are constructed such that the first variable will always be fully observed. These conditions resonate with the joint- and conditional missingness patterns defined by (Fernstad, 2019) as discussed in Section 3.3.

The MAR code is available in the `myMAR()` function presented in Appendix A.2.

The six conditions are further explained by considering a data set with ten categorical variables (A, B,…,J) each with three CLs: one, two, three. The MAR conditions will be applied as follows:

- Considering variable ‘A’; if the responses are ‘one’ or ‘two’, corresponding samples for variables ‘B’, ‘D’, ‘F’ and ‘H’ are considered for deletion.

- Considering variable 'E'; if the responses are 'two' or 'three', corresponding samples for variable 'F' are considered for deletion.
- Considering variable 'B'; if the responses are 'two', corresponding samples for variables 'H', 'I' and 'J' are considered for deletion.
- Considering variable 'J'; if the responses are 'three', corresponding samples for variable 'B' are considered for deletion.
- Considering variable 'C'; if the responses are 'two', corresponding samples for variable 'B', 'D', 'F', 'H' and 'J' are considered for deletion.
- If variables 'A' and 'J' have the same responses, corresponding samples for variable 'J' are considered for deletion.

5.4.1 True percentages of missing values for missing at random simulations

The specified missing percentage is always reflected in the MCAR simulations, since a simple random sample is deleted. The true MAR missing percentages are however not always reflected, since the deletion is dependent on the six conditions (cf. 5.4). Higher percentages of missing values could be obtained by relaxing the MAR conditions to consider more response options for deletion. However, further investigation of the conditions will be considered in future research projects.

The true missing percentages that were obtained over 9000 data sets (1000 repetitions for each of the nine combinations of sample sizes and number of variables) are shown in Figure 5.5 to Figure 5.7. The upper panels of Figure 5.5 to Figure 5.7 present the percentages in ascending order to ease visual inspection. Therefore, the percentage point for the first simulation in the left panel is not necessarily the same data set represented by the first percentage points in the middle- and right panels. The lower panels of Figure 5.5 to Figure 5.7 are not sorted and distinguish between the sample sizes and number of variables for each percentage of missingness. A red horizontal line indicates the specified missingness percentage in the lower panels of Figure 5.5 to Figure 5.7.

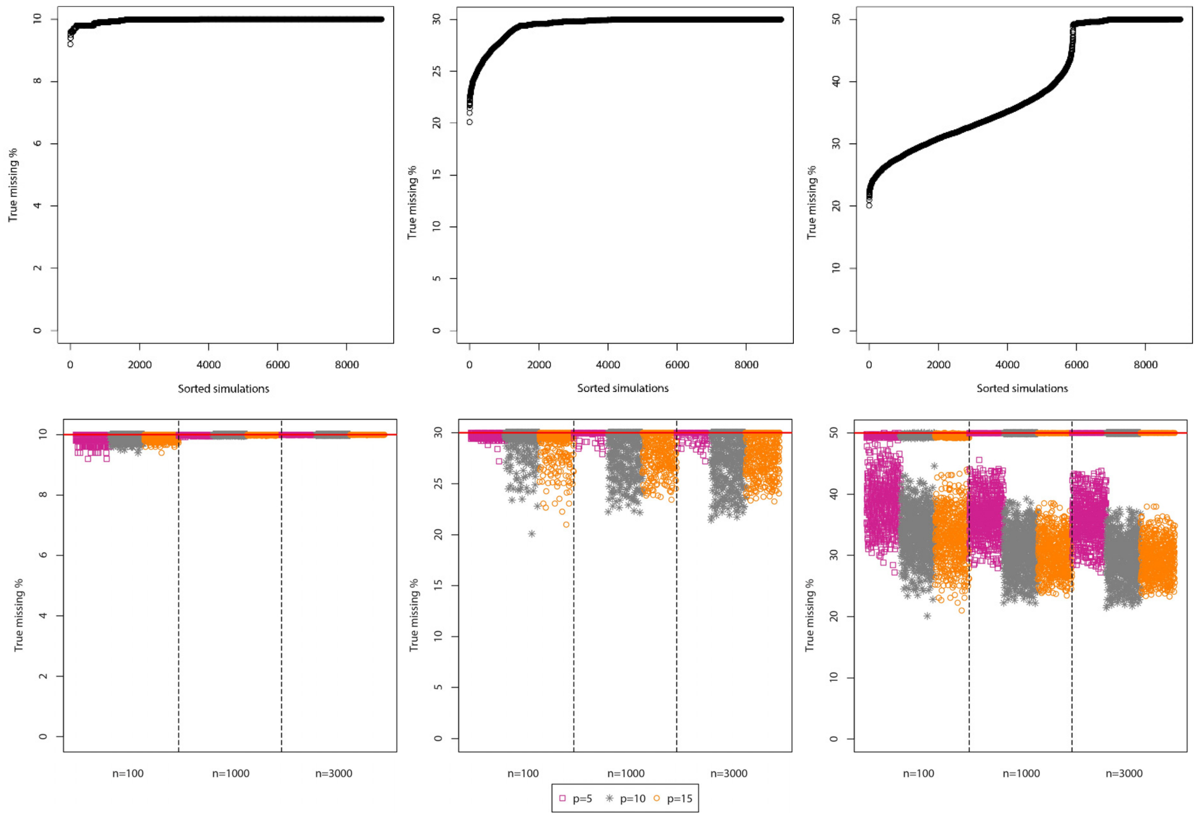


Figure 5.5 True percentage of missing values for Dirichlet simulations. Left panels: MAR 10% (mean missing values: 9.97%, minimum missing value percentage: 9.2%). Middle panels: MAR 30% (mean missing values: 29.4%, minimum missing value percentage: 20.1%). Right panels: MAR 50% (mean missing values: 38.8%, minimum missing value percentage: 20.1%). The simulations are sorted in the upper panels.

The Dirichlet distributed data sets have positively skewed variables which result in lower probabilities for observed frequencies for the last CLs per variable and a high probability for observed frequencies of the first CLs. The same MAR conditions (cf. 5.4) are applied to all simulated distributions and do not make concessions for cases where possible CLs are not observed. Since the majority of the MAR conditions (cf. 5.4) rely on the response of the first CLs or centre CLs, the initial condition for deletion might not be easily satisfied. This could be a possible explanation for the low missing percentages for the 50% missingness specification and the separation of the missing percentages observed in the scatterplot (cf. Figure 5.5: right panels).

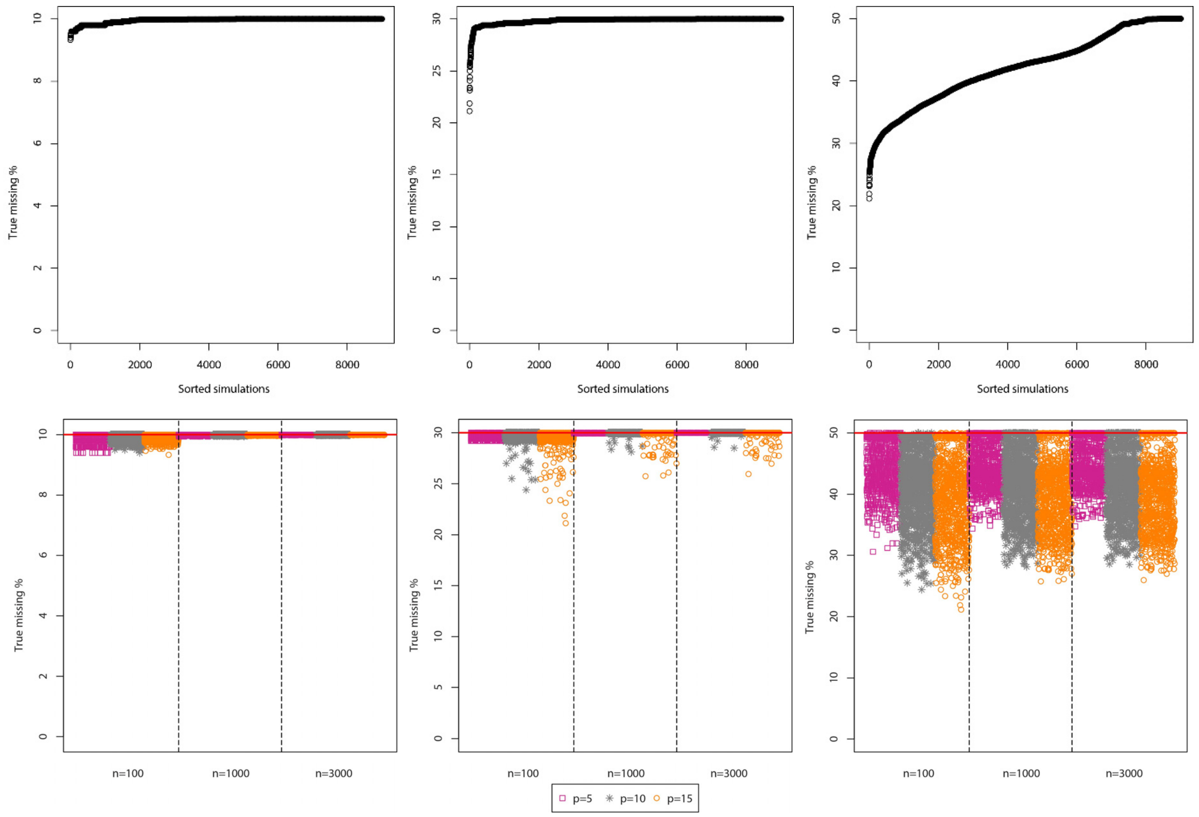


Figure 5.6 True percentage of missing values for uniform simulations. Left panels: MAR 10% (mean missing values: 9.96%, minimum missing value percentage: 9.3%). Middle panels: MAR 30% (mean missing values: 29.8%, minimum missing value percentage: 21.1%). Right panels: MAR 50% (mean missing values: 42.1%, minimum missing value percentage: 21.1%). The simulations are sorted in the upper panels.

The CLs of the uniform distribution variables occur uniformly and therefore the initial MAR conditions can be satisfied unlike the case for the Dirichlet distribution simulations. As the number of variables increases the percentage of missing values seem to decrease accordingly for the 30% and 50% missing value specifications (cf. Figure 5.6: middle- and right panels).

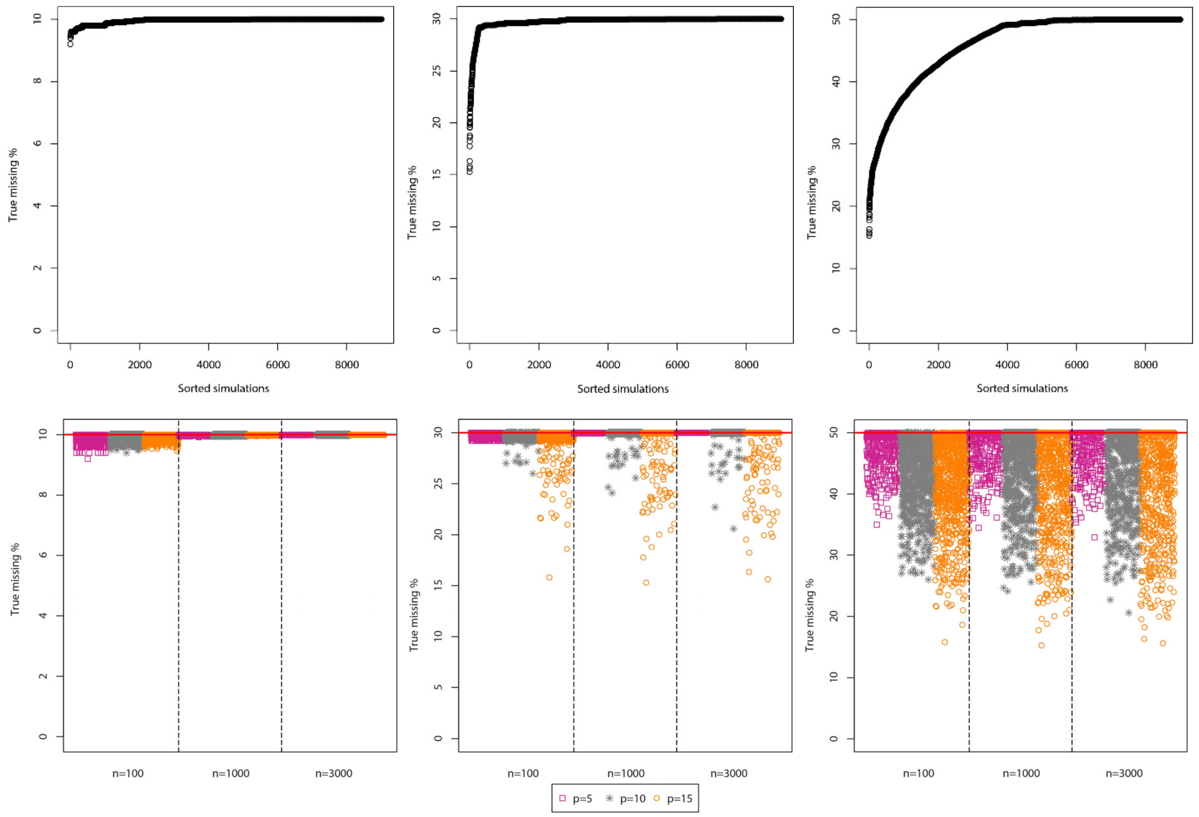


Figure 5.7 True percentage of missing values for normal simulations. Left panels: MAR 10% (mean missing values: 9.96%, minimum missing value percentage: 9.2%). Middle panels: MAR 30% (mean missing values: 29.8%, minimum missing value percentage: 15.3%). Right panels: MAR 50% (mean missing values: 46.0%, minimum missing value percentage: 15.3%). The simulations are sorted in the upper panels.

The normal distribution simulations consist of variables that are highly correlated with a symmetrical distribution of CLs, therefore the CLs in the centre will tend to occur more frequently. As was the case with the Dirichlet distribution, some initial MAR conditions might not be satisfied due to the lower probability of the first and last CLs per variable.

The uniform distribution simulations best reflected the specified missing percentages in comparison to the other two simulation distributions.

5.4.2 Reproducibility of results

All results in this research are reproducible by using a random seed vector of 1000 observations for the repetitions in the simulation study. Each procedure is repeated a 1000 times and is completed using the fixed seed value for a particular repetition. This however results in the same 'random' sample selection for the MCAR MDM for different distributions. The positions of the observations deleted with an MCAR MDM and a specific percentage of missing values for one simulated distribution for a specific repetition will result in the same positions of the observations deleted for the other simulated distributions for the same specific repetition. This has implications on the sMCA of the missing subsets which will be discussed in Chapter 8.

The random seed vector: `seed.vec <- round(runif(1000,1234,7777),0)`, was generated using the `runif()` function, which is available in the `R` package, `stats` (R Core Team, 2017).

5.5 High performance computing

The simulation study considers nine unique combinations of simulation parameters (sample size and number of variables) for three different simulation distributions and also two MDMs with three percentages of missing values. Therefore, 162 unique simulation scenarios are considered to evaluate the methodology presented in Chapter 4. A thousand repetitions of each of the 162 simulation scenarios are performed for all applications.

The simulations were initially executed using explicit parallelisation to improve computational time by utilising multiple processing cores on a local workstation. Due to limited local workstation capacity, the simulation study was performed and completed on the University of Stellenbosch Central Analytical Facilities' HPC clusters, HPC1 (Rhasatsha) and HPC2, during a five-month period. Parallel computing was also applied on the HPC clusters. `R` packages, `doSNOW` (Microsoft Corporation & Weston, 2017) and `parallel` (R Core Team, 2017), were used to perform repetitions on different processing cores in parallel.

The `R` statistical software is installed on the HPC clusters and any required packages can be requested and installed by the HPC support staff. The HPC clusters consist of a number of nodes with different specifications. A Windows application for remote computing, *MobaXterm*, enables console access in which jobs can be submitted to the server via command line. The home edition of *MobaXterm* can be downloaded for free and is developed by *Mobatek* (https://mobaxterm.mobatek.net/).

The HPC clusters implement a bash queueing system, referred to as PBS. This means that computational tasks are compiled in a script containing the necessary `R` code and functions which is submitted to the PBS. The required specifications are listed at the beginning of the script, which specify the expected walltime (or run time), number of cores to be used for the specific job and the required random-access memory (RAM). A script can be compiled using any text editor, but the script has to be exported in a UNIX format as a shell file (.sh) to be accepted by the PBS. Each job has a specific shell script which should be able to run independently once submitted in the queue.

Jobs are automatically assigned to computing nodes based on the specifications in the script. Each queue has access to a maximum number of cores to ensure that users in all queues have sufficient capacity available. Jobs are assigned to a queue according to the specified walltime: short (two hours), day (24 hours), week (168 hours), month (744 hours) and long (exceeding 31 days). The correct estimation of computational time is vital to ensure optimal usage of the HPC clusters. Jobs are not completed if insufficient time is specified and the waiting time in a longer queue is more due to less cores that are assigned to this specific queue. An example of a script file for a GPAbin (cf. 4.3) computation submitted to PBS is given in Appendix M.

The computational time depends on the particular task. Considering the GPAbin approach, the task consists of obtaining ten MIs with the MIMCA algorithm, applying MCA to the ten multiple imputed data sets, followed by GPA to align the MCA configurations of the ten MIs and finally obtaining the GPAbin configuration. Simulated data sets with small dimensions ($p = 5$ and $n = 100$) with only 10% missing values could complete a 1000 repetitions of the GPAbin approach with ten MIs within one hour using 24 cores and approximately 20 gigabyte (GB) RAM. Whereas a GPAbin procedure for higher dimensions ($p = 15$ and $n = 3000$) with 50% missing values completed a 1000 repetitions in 20 hours using 24 cores and approximately 44 GB RAM.

The prediction methods (cf. 4.4) were the most computationally intensive tasks and in some instances a 1000 repetitions had to be divided into tasks of 250 repetitions each. An example is the Majority rule prediction approach for normally distributed data ($p = 15$ and $n = 3000$) with 50% missing values and a MAR MDM, which completed 250 repetitions in 43 hours using 16 cores and 127 GB RAM.

In Table 5.2 (the number of cores), Table 5.3 (RAM) and Table 5.4 (walltime) summaries of one set of computations are presented. The sMCA computation specifications are not presented for specific combinations of sample size and number of variables, since less RAM was required than the other computations. Therefore, the sMCA computations could be completed for 9000 repetitions within one job on the HPC clusters. The prediction methods are abbreviated in the table: GPAbin prediction method is presented as 'GPAbin Pred' and the Majority rule prediction method as 'Maj.rule Pred'. The entries annotated by 'sim %' refers to the similarity percentage calculations (cf. 4.6.3).

Table 5.2 The number of cores used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM.

Number of cores	$p = 5$, $n = 100$	$p = 5$, $n = 1000$	$p = 5$, $n = 3000$	$p = 10$, $n = 100$	$p = 10$, $n = 1000$	$p = 10$, $n = 3000$	$p = 15$, $n = 100$	$p = 15$, $n = 1000$	$p = 15$, $n = 3000$
sMCA	16								
RIMCA	8	8	24	24	48	48	24	48	48
GPAbin	24	24	24	24	24	24	24	48	24
GPAbin Pred	8	8	8	8	8	8	8	$8^{\#}$	$16^{\#}$
Maj.rule Pred	8	8	8	8	16	8	16	16^{*}	16^{*}
sMCA sim %	8	8	8	8	8	8	8	8	$8^{\#}$
RIMCA sim %	8	8	8	8	8	8	8	8	$8^{\#}$
GPAbin sim %	8	8	8	8	8	8	8	8	$8^{\#}$

# 1000 repetitions were computed in two sets of 500 each.

* 1000 repetitions were computed in four sets of 250 each.

Table 5.3 The RAM (in GB) used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM.

RAM (in GB)	$p = 5,$ $n = 100$	$p = 5,$ $n = 1000$	$p = 5,$ $n = 3000$	$p = 10,$ $n = 100$	$p = 10,$ $n = 1000$	$p = 10,$ $n = 3000$	$p = 15,$ $n = 100$	$p = 15,$ $n = 1000$	$p = 15,$ $n = 3000$
sMCA	5								
RIMCA	2	2	5	6	13	16	8	20	26
GPAbin	19	20	24	26	28	55	33	41	49
GPAbin Pred	1	1	2	2	4	5	5	9[#]	22[#]
Maj.rule Pred	8	4	5	9	32	25	48	90[*]	118[*]
sMCA sim %	1	1	2	2	4	5	6	10	14[#]
RIMCA sim %	1	1	2	2	6	4	6	8	11[#]
GPAbin sim %	1	1	2	2	3	4	5	8	11[#]

[#] The average RAM of the two sets of 500 repetitions are reported.

[*] The average RAM of the four sets of 250 repetitions are reported.

Table 5.4 The walltime (in minutes) used for the computations of uniform distribution simulations with 50% missing values inserted with an MCAR MDM.

Walltime (in minutes)	$p = 5,$ $n = 100$	$p = 5,$ $n = 1000$	$p = 5,$ $n = 3000$	$p = 10,$ $n = 100$	$p = 10,$ $n = 1000$	$p = 10,$ $n = 3000$	$p = 15,$ $n = 100$	$p = 15,$ $n = 1000$	$p = 15,$ $n = 3000$
sMCA	80								
RIMCA	196	941	722	178	376	1037	714	966	1943
GPAbin	45	166	623	253	656	1079	710	1049	3595
GPAbin Pred	6	23	49	75	247	646	191	722[#]	1798[#]
Maj.rule Pred	60	167	363	604	849	3433	1878	8183[*]	11883[*]
sMCA sim %	6	21	49	82	270	505	371	1264	1045[#]
RIMCA sim %	6	20	44	73	270	499	284	843	774[#]
GPAbin sim %	6	20	41	56	149	331	317	1160	1292[#]

[#] The total walltime for the two sets of 500 repetitions are reported.

[*] The total walltime for the four sets of 250 repetitions are reported.

The walltime is effected by the RAM and number of cores specifications. It is not feasible to select the maximum available resources. Users do not have unlimited access to the available capacity on the HPC clusters and jobs are prioritised according to the combination of specifications. Therefore, sufficient specifications should be selected with a reasonable walltime in order to complete computations efficiently.

The walltimes as presented in Table 5.4 are equivalent to approximately 938 hours (39 days) of computation. This summary provides insight to the approximate time spent on the HPC computations. Eighteen of these sets (three simulation distributions, two MDMs and three percentages of missing values) were completed for the simulation study, which is roughly equivalent to 702 days of computational time on the HPC clusters. The measures of comparison (cf. 4.6.1), visualisations and the real data application (cf. Chapter 9) were less computationally intensive and could be completed on a local computer.

5.6 Conclusion

The simulation protocol for this research was developed to evaluate the performance of the proposed methodology (cf. Chapter 4) on data sets that result in diverse configurations.

This chapter presented the details of the simulation protocol, an overview of the procedure of the HPC clusters along with an estimation of the computational time for the approaches in this research project. The choices of the particular distributions used for the simulations were motivated and supported by MCA biplots. It was observed that the specified percentage of missing values is not always reflected when generating data sets with MAR MDMs, especially in scenarios where the expected percentage of missing values is as high as 50%.

The applications of the proposed methodology (cf. Chapter 4) on the simulated data, as described in this chapter, are presented in three consecutive chapters: missing data approaches (cf. Chapter 6), prediction methods (cf. Chapter 7) and the identification of the MDM (cf. Chapter 8).

The following Chapter 6 will illustrate and summarise the application of the missing data approaches: GPAbin, sMCA and RIMCA.

Chapter 6
Results: Missing data approaches

6.1 Introduction

Unwin et al. (2008) distinguishes between presentation- and exploratory graphical representations. Presentation graphics are to the point and should provide clear conclusions, whereas exploratory visualisations are used during the investigation phase of research in search of a specific conclusion. This principle is followed in the presentation of the results of the simulated data scenarios (cf. Chapter 5). In order to validate the methodologies presented in Chapter 4, a large number of exploratory graphics were generated, but only a selection of contributing figures will be presented. In many cases these results will seem repetitive, but by investigating the plots systematically, certain trends are observed which guides the further inspection of particular cases to reach final conclusions of the proposed methods.

Similarity percentages (cf. 4.6.3) and five measures of comparison (cf. 4.6.1) are used to summarise the results obtained from the three missing data approaches: GPAbin, sMCA and RIMCA.

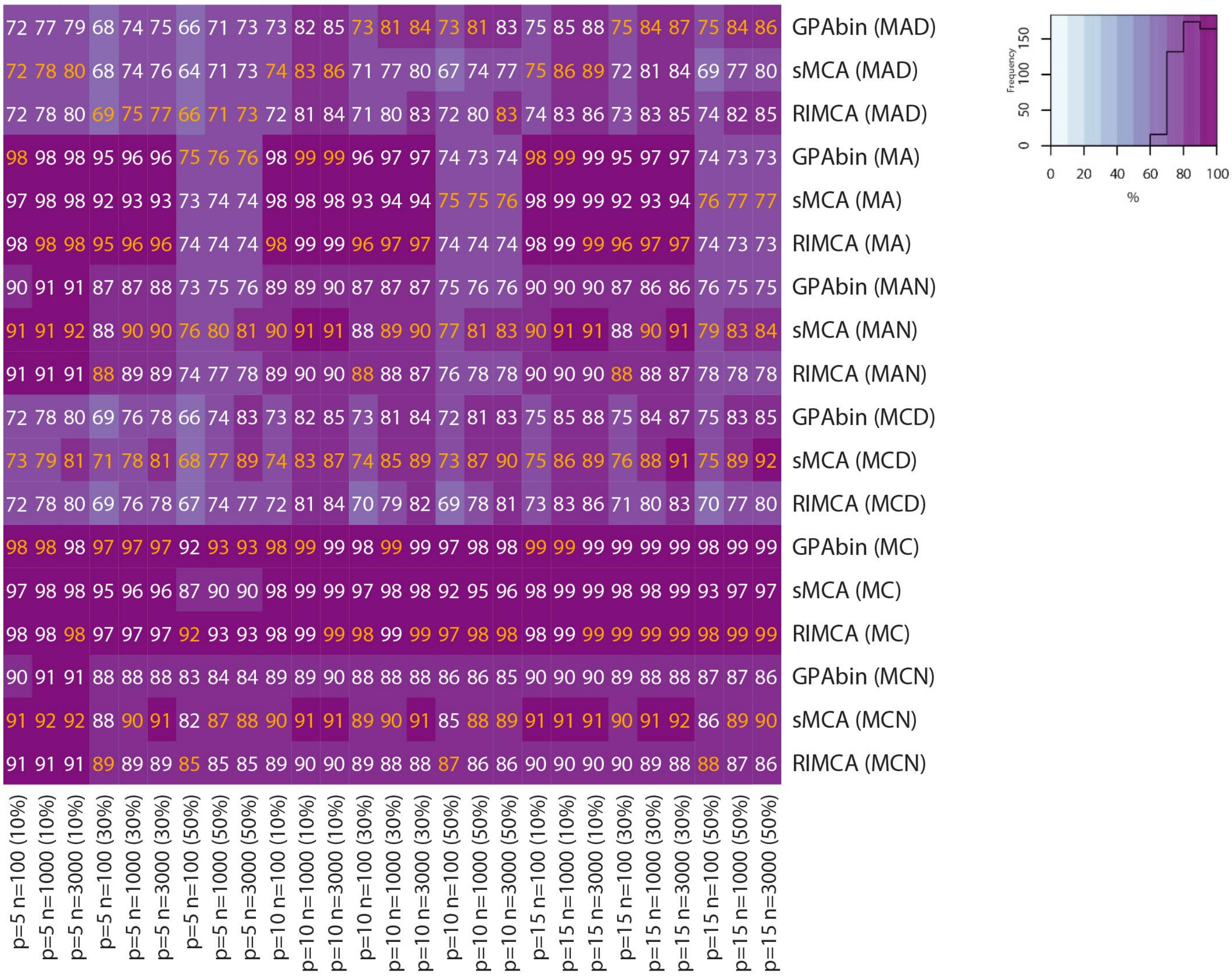
Throughout this chapter the abbreviations in Table 6.1 will be used for the simulation distributions and MDMs.

Table 6.1 Abbreviations for simulation distributions and MDMs

MAD	Data simulated from a Dirichlet distribution with a MAR MDM.
MA	Data simulated from a uniform distribution with a MAR MDM.
MAN	Data simulated from a normal distribution with a MAR MDM.
MCD	Data simulated from a Dirichlet distribution with an MCAR MDM.
MC	Data simulated from a uniform distribution with an MCAR MDM.
MCN	Data simulated from a normal distribution with an MCAR MDM.

6.2 Similarity percentages

The similarity percentages (cf. 4.6.3) are rounded to integers in the heat map in Figure 6.1. Each cell in the heat map represents the mean similarity percentage over 1000 repetitions per simulation scenario. The highest similarity percentage (based on the raw percentages) between the approaches (GPAbin, sMCA and RIMCA) per simulation scenario is annotated in orange. As an example consider the first column ($n = 5, p = 100$ and 10% missing values) and first three rows (MAR MDM with data simulated from a Dirichlet distribution): the highest similarity percentage is obtained from the sMCA approach (72.35%) compared to the GPAbin method (71.81%) and RIMCA method (71.99%). In the majority of cases the similarity percentages of the three different approaches do not differ notably per simulation scenario. The heat map key also includes a frequency distribution of the similarity percentages, illustrated by the solid black line. The frequency distribution summarises the similarity percentages across all simulation scenarios represented in the heat map. It is noted that a vast majority of similarity percentages are above 70% as reflected in the darker shades of purple cells in the heat map (cf. Figure 6.1). Heat maps are constructed using the `heatmap.2()` function available in the `R` package, `gplots` (Warnes, Bolker, Bonebakker, Gentleman, Liaw, Lumley, Maechler, Magnusson, Moeller, Schwartz & Venables, 2016).



	p=5 n=100 (10%)	p=5 n=1000 (10%)	p=5 n=3000 (10%)	p=5 n=100 (30%)	p=5 n=1000 (30%)	p=5 n=3000 (30%)	p=5 n=100 (50%)	p=5 n=1000 (50%)	p=5 n=3000 (50%)	p=10 n=100 (10%)	p=10 n=1000 (10%)	p=10 n=3000 (10%)	p=10 n=100 (30%)	p=10 n=1000 (30%)	p=10 n=3000 (30%)	p=10 n=100 (50%)	p=10 n=1000 (50%)	p=10 n=3000 (50%)	p=15 n=100 (10%)	p=15 n=1000 (10%)	p=15 n=3000 (10%)	p=15 n=100 (30%)	p=15 n=1000 (30%)	p=15 n=3000 (30%)	p=15 n=100 (50%)	p=15 n=1000 (50%)	p=15 n=3000 (50%)
GPAbin (MAD)	72	77	79	68	74	75	66	71	73	73	82	85	73	81	84	73	81	83	75	85	88	75	84	87	75	84	86
sMCA (MAD)	72	78	80	68	74	76	64	71	73	74	83	86	71	77	80	67	74	77	75	86	89	72	81	84	69	77	80
RIMCA (MAD)	72	78	80	69	75	77	66	71	73	72	81	84	71	80	83	72	80	83	74	83	86	73	83	85	74	82	85
GPAbin (MA)	98	98	98	95	96	96	75	76	76	98	99	99	96	97	97	74	73	74	98	99	99	95	97	97	74	73	73
sMCA (MA)	97	98	98	92	93	93	73	74	74	98	98	98	93	94	94	75	75	76	98	99	99	92	93	94	76	77	77
RIMCA (MA)	98	98	98	95	96	96	74	74	74	98	99	99	96	97	97	74	74	74	98	99	99	96	97	97	74	73	73
GPAbin (MAN)	90	91	91	87	87	88	73	75	76	89	89	90	87	87	87	75	76	76	90	90	90	87	86	86	76	75	75
sMCA (MAN)	91	91	92	88	90	90	76	80	81	90	91	91	88	89	90	77	81	83	90	91	91	88	90	91	79	83	84
RIMCA (MAN)	91	91	91	88	89	89	74	77	78	89	90	90	88	88	87	76	78	78	90	90	90	88	88	87	78	78	78
GPAbin (MCD)	72	78	80	69	76	78	66	74	83	73	82	85	73	81	84	72	81	83	75	85	88	75	84	87	75	83	85
sMCA (MCD)	73	79	81	71	78	81	68	77	89	74	83	87	74	85	89	73	87	90	75	86	89	76	88	91	75	89	92
RIMCA (MCD)	72	78	80	69	76	78	67	74	77	72	81	84	70	79	82	69	78	81	73	83	86	71	80	83	70	77	80
GPAbin (MC)	98	98	98	97	97	97	92	93	93	98	99	99	98	99	99	97	98	98	99	99	99	99	99	99	98	99	99
sMCA (MC)	97	98	98	95	96	96	87	90	90	98	99	99	97	98	98	92	95	96	98	99	99	98	98	99	93	97	97
RIMCA (MC)	98	98	98	97	97	97	92	93	93	98	99	99	98	99	99	97	98	98	98	99	99	99	99	99	98	99	99
GPAbin (MCN)	90	91	91	88	88	88	83	84	84	89	89	90	88	88	88	86	86	85	90	90	90	89	88	88	87	87	86
sMCA (MCN)	91	92	92	88	90	91	82	87	88	90	91	91	89	90	91	85	88	89	91	91	91	90	91	92	86	89	90
RIMCA (MCN)	91	91	91	89	89	89	85	85	85	89	90	90	89	88	88	87	86	86	90	90	90	90	89	88	88	87	86

Figure 6.1 Similarity percentages of all simulation scenarios for GPAbin, sMCA and RIMCA.

Overall the three methods perform well and manage to result in high similarity percentages, which means that the methods preserve the visual interpretation (associations between the samples and CLPs) of the simulated complete configurations in two dimensions. The weakest performance was observed from the sMCA approach with 63.7% (≈64%) similarity in visual interpretation with the following simulation parameters: $p = 5, n = 100$, 50% missing values created according to a MAR MDM and simulated from a Dirichlet distribution. A general overview of Figure 6.1 is that lower similarity percentages occur for Dirichlet distribution simulations with varying success between the three approaches for MAR MDMs. However, a clear success of the sMCA method is observed when compared to the similarity percentages of GPAbin and RIMCA for MCAR MDMs simulated from Dirichlet and normal distributions. The uniform distribution simulations with a MAR MDM result in similar percentages across approaches for 10% and 30% missing values, however GPAbin outperforms RIMCA and sMCA for 50% missing values when the number of variables is small ($p = 5$). The sMCA approach seems to preserve the interpretations of the original complete visualisations well when the number of variables is large ($p = 15$), but the similarity percentages are similar to those of RIMCA and GPAbin. The GPAbin and RIMCA approaches perform similarly for uniform distribution simulations with an MCAR MDM with noticeably lower similarity percentages achieved by sMCA.

Since the majority of similarity percentages for the three approaches per simulation scenario are similar, the focus will be placed on the scenarios resulting in a higher dispersion of similarity percentages between approaches. The coefficient of variation ($CV = \frac{s}{\bar{X}} \times 100$) is calculated per simulation scenario, where s is the sample standard deviation and $\bar{X}$ the sample mean of the similarity percentages. The simulation scenarios with a higher dispersion will be further investigated to determine which methods provide the best representation of the original visual interpretations (i.e. highest similarity percentage) for the particular scenarios. The coefficients of variation are presented in Table 6.2.

Table 6.2 Coefficients of variation of similarity percentages (GPAbin, sMCA and RIMCA) for all simulation scenarios. Values above 3% are indicated in ***bold*** *font.*

	$p = 5, n = 100$	$p = 5, n = 1000$	$p = 5, n = 3000$	$p = 10, n = 100$	$p = 10, n = 1000$	$p = 10, n = 3000$	$p = 15, n = 100$	$p = 15, n = 1000$	$p = 15, n = 3000$
10% missing values									
MAD	0.39%	0.35%	0.63%	1.01%	1.24%	1.24%	1.26%	1.59%	1.57%
MA	0.27%	0.23%	0.23%	0.15%	0.12%	0.11%	0.11%	0.10%	0.10%
MAN	0.24%	0.50%	0.58%	0.41%	0.68%	0.79%	0.44%	0.74%	0.87%
MCD	0.45%	0.55%	0.60%	1.08%	1.35%	1.36%	1.35%	1.73%	1.69%
MC	0.23%	0.19%	0.19%	0.07%	0.06%	0.06%	0.04%	0.03%	0.03%
MCN	0.23%	0.50%	0.61%	0.41%	0.72%	0.83%	0.48%	0.80%	0.92%
30% missing values									
MAD	0.76%	1.01%	1.29%	1.23%	2.71%	2.68%	1.88%	2.07%	1.62%
MA	1.97%	1.92%	1.90%	1.91%	2.07%	2.11%	1.82%	2.11%	2.14%
MAN	0.60%	1.42%	1.63%	0.73%	1.63%	2.12%	0.74%	1.97%	2.59%
MCD	1.29%	1.61%	1.73%	2.76%	**3.94%**	**3.98%**	**3.45%**	**4.89%**	**4.81%**
MC	1.14%	0.72%	0.69%	0.70%	0.36%	0.33%	0.51%	0.26%	0.25%
MCN	0.39%	1.24%	1.61%	0.53%	1.59%	2.06%	0.62%	1.75%	2.25%
50% missing values									
MAD	2.17%	0.58%	0.59%	**4.00%**	**4.78%**	**4.16%**	**4.23%**	**4.29%**	**3.92%**
MA	1.50%	1.22%	1.14%	0.98%	1.53%	1.46%	2.17%	2.98%	2.68%
MAN	1.80%	2.78%	2.84%	1.86%	**3.65%**	**4.54%**	2.23%	**5.14%**	**5.95%**
MCD	1.50%	2.61%	**7.26%**	**3.12%**	**5.74%**	**5.92%**	**4.45%**	**7.26%**	**7.24%**
MC	**3.28%**	1.82%	1.60%	**3.19%**	1.41%	1.15%	2.74%	1.16%	0.98%
MCN	1.32%	1.79%	2.60%	1.34%	1.59%	2.53%	1.38%	1.65%	2.71%

The simulation scenarios with coefficients of variation larger than 3% are further investigated in Table 6.3 and Table 6.4. The similarity percentage of each method is given as well as the ranking (1 – best, 2 – 2nd best, 3 – worst) for a particular simulation scenario. The rankings are allocated based on the raw similarity percentages. As an example consider the Dirichlet simulation ($p = 10$ and $n = 1000$) with a MAR MDM and 30% missingness as presented in Figure 6.1: the GPAbin approach has the highest similarity percentage (81%), followed by the RIMCA method (80%) and lastly the sMCA approach (77%).

*Table 6.3 Similarity percentages of simulation scenarios with 30% missing values with a coefficient of variation greater than 3% between approaches (**bold** values of Table 6.2).*

30% missingness	GPAbin	sMCA	RIMCA
$p = 5, n = 100$	None	None	None
$p = 5, n = 1000$	None	None	None
$p = 5, n = 3000$	None	None	None
$p = 10, n = 100$	None	None	None
$p = 10, n = 1000$	(2) MCD 81%	(1) MCD 85%	(3) MCD 79%
$p = 10, n = 3000$	(2) MCD 84%	(1) MCD 89%	(3) MCD 82%
$p = 15, n = 100$	(2) MCD 75%	(1) MCD 76%	(3) MCD 71%
$p = 15, n = 1000$	(2) MCD 84%	(1) MCD 88%	(3) MCD 80%
$p = 15, n = 3000$	(2) MCD 87%	(1) MCD 91%	(3) MCD 83%

The Dirichlet simulations with an MCAR MDM result in higher dispersion between the similarity percentages of the three approaches. For the scenarios presented in Table 6.3, sMCA was the best approach, followed by GPAbin and lastly RIMCA. It is noted that a small sample size ($n = 100$) results in lower similarity percentages.

*Table 6.4 Similarity percentages of simulation scenarios with 50% missing values with a coefficient of variation greater than 3% between approaches (**bold** values of Table 6.2).*

50% missingness	GPAbin	sMCA	RIMCA
$p = 5, n = 100$	(2) MC 92%	(3) MC 87%	(1) MC 92%
$p = 5, n = 1000$	None	None	None
$p = 5, n = 3000$	(2) MCD 83%	(1) MCD 89%	(3) MCD 77%
$p = 10, n = 100$	(1) MAD 73% (2) MCD 72% (2) MC 97%	(3) MAD 67% (1) MCD 73% (3) MC 92%	(2) MAD 72% (3) MCD 69% (1) MC 97%
$p = 10, n = 1000$	(1) MAD 81% (3) MAN 76% (2) MCD 81%	(3) MAD 74% (1) MAN 81% (1) MCD 87%	(2) MAD 80% (2) MAN 78% (3) MCD 78%
$p = 10, n = 3000$	(2) MAD 83% (3) MAN 76% (2) MCD 83%	(3) MAD 77% (1) MAN 83% (1) MCD 90%	(1) MAD 83% (2) MAN 78% (3) MCD 81%
$p = 15, n = 100$	(1) MAD 75% (2) MCD 75%	(3) MAD 69% (1) MCD 75%	(2) MAD 74% (3) MCD 70%
$p = 15, n = 1000$	(1) MAD 84% (3) MAN 75% (2) MCD 83%	(3) MAD 77% (1) MAN 83% (1) MCD 89%	(2) MAD 82% (2) MAN 78% (3) MCD 77%
$p = 15, n = 3000$	(1) MAD 86% (3) MAN 75% (2) MCD 85%	(3) MAD 80% (1) MAN 84% (1) MCD 92%	(2) MAD 85% (2) MAN 78% (3) MCD 80%

Only two uniform distribution simulations resulted in high dispersion between similarity percentages ($p = 5, 10$ with $n = 100$), which resulted in similar outcomes for RIMCA and GPAbin followed by the worst performance by sMCA. Again, the results from the Dirichlet distribution simulations are more dispersed. Simulations with an MCAR MDM perform better when using the sMCA approach, whereas MAR MDM Dirichlet distribution simulations perform better when using the GPAbin approach. In the cases where RIMCA outperforms GPAbin, the differences in the similarity percentages are negligible. Four scenarios of normal distribution simulations result in highly dispersed similarity percentages which all achieve the best performance when sMCA is applied, followed by RIMCA and lastly GPAbin.

The weakest methods are further summarised by identifying the simulation scenarios with a similarity percentage below 75%. The similarity percentages are summarised according to the percentage of missing values in Table 6.5 (10% missing values), Table 6.6 (30% missing values) and Table 6.7 (50% missing values).

Table 6.5 Similarity percentages below 75% for simulation scenarios with 10% missing values. Investigating the percentages of Figure 6.1.

10% missingness	GPAbin	sMCA	RIMCA
$p = 5, n = 100$	MAD 72% MCD 72%	MAD 72% MCD 73%	MAD 72% MCD 72%
$p = 5, n = 1000$	None	None	None
$p = 5, n = 3000$	None	None	None
$p = 10, n = 100$	MAD 73% MCD 73%	MAD 74% MCD 74%	MAD 72% MCD 72%
$p = 10, n = 1000$	None	None	None
$p = 10, n = 3000$	None	None	None
$p = 15, n = 100$	None	None	MAD 74% MCD 73%
$p = 15, n = 1000$	None	None	None
$p = 15, n = 3000$	None	None	None

A small sample size ($n = 100$) and simulations from a Dirichlet distribution (irrespective of MDM) do not easily replicate the visual interpretation of the simulated complete configurations. There are small differences between the simulations that result in similarity percentages below 75% (cf. Table 6.5). The sMCA method results in slightly higher similarity percentages for MCAR MDMs with GPAbin resulting in a slightly higher similarity percentage than RIMCA for both MDMs when $p = 10$. All similarity percentages are however still close to 75% similarity when the percentage of missing values is low (10%).

Table 6.6 Similarity percentages below 75% for simulation scenarios with 30% missing values. Investigating the percentages of Figure 6.1.

30% missingness	GPAbin	sMCA	RIMCA
$p = 5, n = 100$	MAD 68% MCD 69%	MAD 68% MCD 71%	MAD 69% MCD 69%
$p = 5, n = 1000$	MAD 74%	MAD 74%	None
$p = 5, n = 3000$	None	None	None
$p = 10, n = 100$	MAD 73% MCD 73%	MAD 71% MCD 74%	MAD 71% MCD 70%
$p = 10, n = 1000$	None	None	None
$p = 10, n = 3000$	None	None	None
$p = 15, n = 100$	None	MAD 72%	MAD 73% MCD 71%
$p = 15, n = 1000$	None	None	None
$p = 15, n = 3000$	None	None	None

The Dirichlet distribution simulations with small sample sizes ($n = 100$) still provide the weakest replication of the original visual interpretations. As the percentage of missing values increases (now, 30%) a few additional simulation scenarios are considered (cf. Table 6.6). Now, also the GPAbin and sMCA approaches with $p = 5$ and $n = 1000$ resulted in similarity percentages of 74% for the Dirichlet distribution with missingness caused by a MAR MDM, but these methods are not notably different from RIMCA which achieved a similarity percentage of only 75% for this particular simulation scenario. As is observed from Table 6.6, the GPAbin method results in higher similarity percentages than RIMCA for $p = 10$ and $n = 100$. Also, the sMCA approach for $p = 15$ and $n = 100$ results in a similarity percentage less than 75% for simulations from a Dirichlet distribution with a MAR MDM. Again, in the scenarios where the sMCA method resulted in a similarity percentage below 75%, the similarity percentages are slightly higher for sMCA for MCAR MDMs compared to RIMCA and GPAbin. When comparing the similarity percentages for 10% missingness (cf. Table 6.5) and 30% missingness (cf. Table 6.6), the results are stable for $p = 10$ and $p = 15$, but for a small number of variables ($p = 5$) the similarity percentages decrease with an increase in the percentage of missing values.

Table 6.7 Similarity percentages below 75% for simulation scenarios with 50% missing values. Investigating the percentages of Figure 6.1.

50% missingness	GPAbin	sMCA	RIMCA
$p = 5, n = 100$	MAD 66% MAN 73% MCD 66%	MAD 64% MA 73% MCD 68%	MAD 66% MA 74% MAN 74% MCD 67%
$p = 5, n = 1000$	MAD 71% MCD 74%	MAD 71% MA 74%	MAD 71% MA 74% MCD 74%
$p = 5, n = 3000$	MAD 73%	MAD 73% MA 74%	MAD 73% MA 74%
$p = 10, n = 100$	MAD 73% MA 74% MCD 72%	MAD 67% MCD 73%	MAD 72% MA 74% MCD 69%

Table 6.7 continued

50% missingness	GPAbin	sMCA	RIMCA
$p = 10, n = 1000$	MA 73%	MAD 74%	MA 74%
$p = 10, n = 3000$	MA 74%	None	MA 74%
$p = 15, n = 100$	MA 74%	MAD 69%	MAD 74% MA 74% MCD 70%
$p = 15, n = 1000$	MA 73%	None	MA 73%
$p = 15, n = 3000$	MA 73%	None	MA 73%

As the percentage of missing values increases (50%), the other simulated distributions (normal and uniform) also result in similarity percentages below 75%. The increase in the percentage of missing values might result in the absence of responses for a particular CL. Therefore, the missing CL with no responses, will have no response profile to be used for the imputation and will not be represented in the imputed data sets. Consequently, this will result in missing information and could lead to biased imputations. This also translates to the MCA biplots, since the unobserved CLs will not be displayed as coordinates in the MCA biplots.

Also, responses for a particular variable with unobserved responses will result in a majority of the same CLs for samples and therefore no discrimination between the samples will be perceived in the MCA biplots. Thus, it is expected that the similarity percentages will be less for imputation methods as the percentage of missing values increases.

To conclude, the best similarity percentages are obtained from uniform distribution simulations with overall, considering all three approaches, mean similarity percentages of 89% (MAR uniform) and 97% (MCAR uniform). The normal distribution simulations perform slightly worse than the uniform distribution simulations with overall mean similarity percentages of 85% (MAR normal) and 89% (MCAR normal). From the discussions in this section it is clear that the Dirichlet distribution simulations provide the worst similarity percentages with overall mean values of 77% (MAR Dirichlet) and 79% (MCAR Dirichlet). Further, if the distributions of the CLs of variables are skewed and the MDM is known to be MCAR, the sMCA approach will result in higher similarity percentages than the other approaches. Similar performance is observed from GPAbin and RIMCA. However, higher similarity percentages are obtained from GPAbin for simulated data with the following specifications: a large number of variables ($p = 10$ and $p = 15$), a small sample size ($n = 100$), simulated from a Dirichlet distribution and for all percentages of missing values with a MAR MDM. Therefore, the GPAbin approach appears to be reliable in instances where the available information (sample size) is limited.

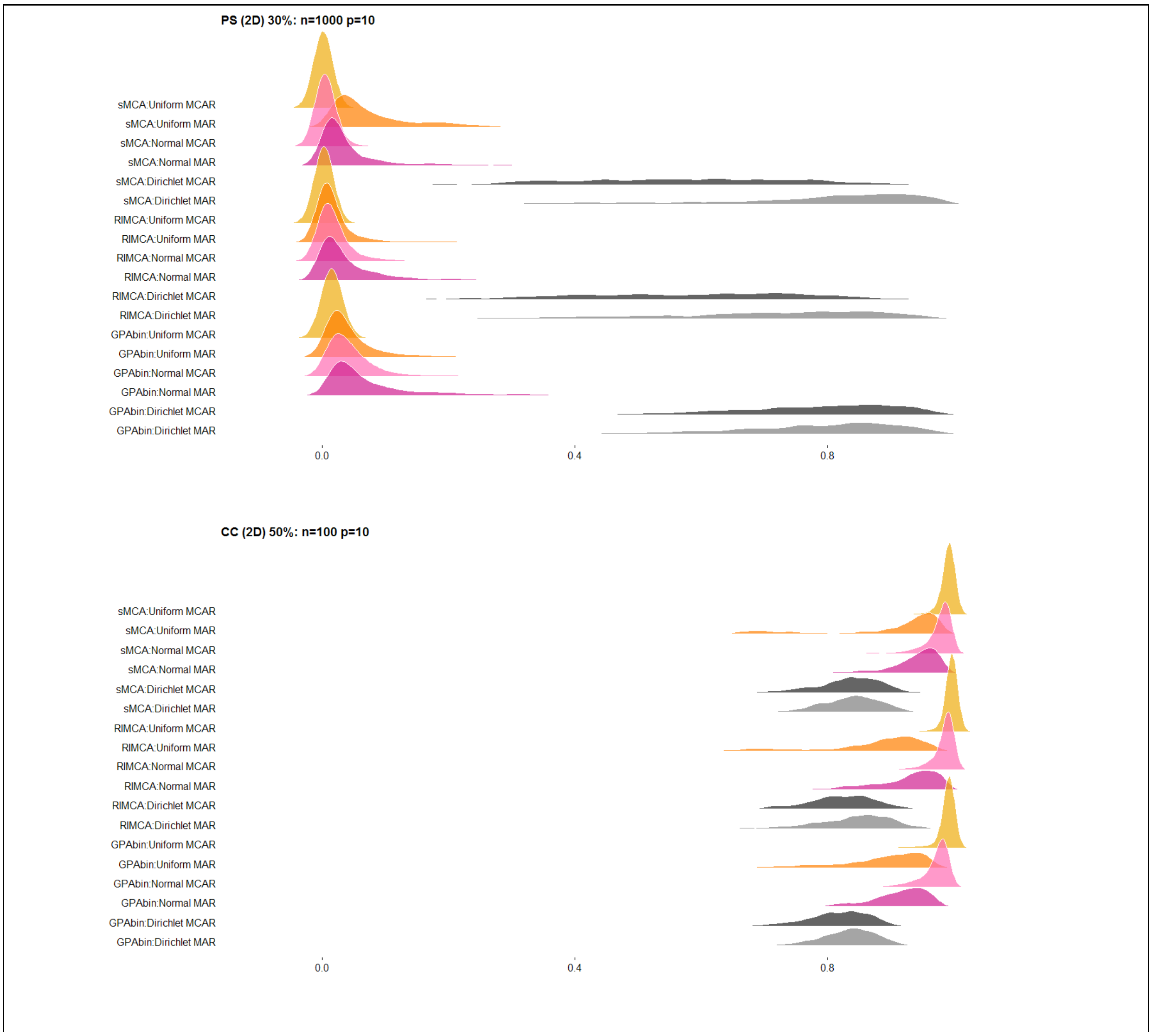
6.3 Measures of comparison

The following measures (cf. 4.6.1) have been calculated using two dimensions (visual space) and also the maximum available dimensions that agree in both the simulated complete MCA coordinates and MCA coordinates obtained from the three approaches (GPAbin, sMCA, RIMCA). The emphasis is however on the success of the visual approximation in two dimensions and how it compares to the visualisation obtained from the original simulated complete data configurations. The difference between the measures of comparison obtained from the two dimension- and maximum dimension configurations will establish the magnitude of information that is forfeited when approximating in lower dimension.

Different visualisation techniques have been explored to efficiently summarise the vast amount of results: heat maps for an overview of the results; scatterplots to visualise trends observed in the measures of comparison per simulation scenario and density plots to compare the distributions across simulation scenarios.

Only a limited selection of these visualisations will be presented to highlight important findings, with supplementary visualisations available in Appendix O.

It was found through visual inspection that the measures of comparison have skewed distributions with a selected example for each measure of comparison shown in Figure 6.2. Staggered density plots are generated using the `R` package, `ggridges` (Wilke, 2018). Figure 6.2 consists of separate panels for each measure of comparison (cf. 4.6.1) for a selection of simulation scenarios. Eighteen staggered density plots of a 1000 repetitions each are given in each panel. Take note that in order to enhance the visualisations, the full observation ranges (cf. Table 6.8) of the bias measures (AMB, MB, RMSB) are not used in the density plots (cf. Figure 6.2).



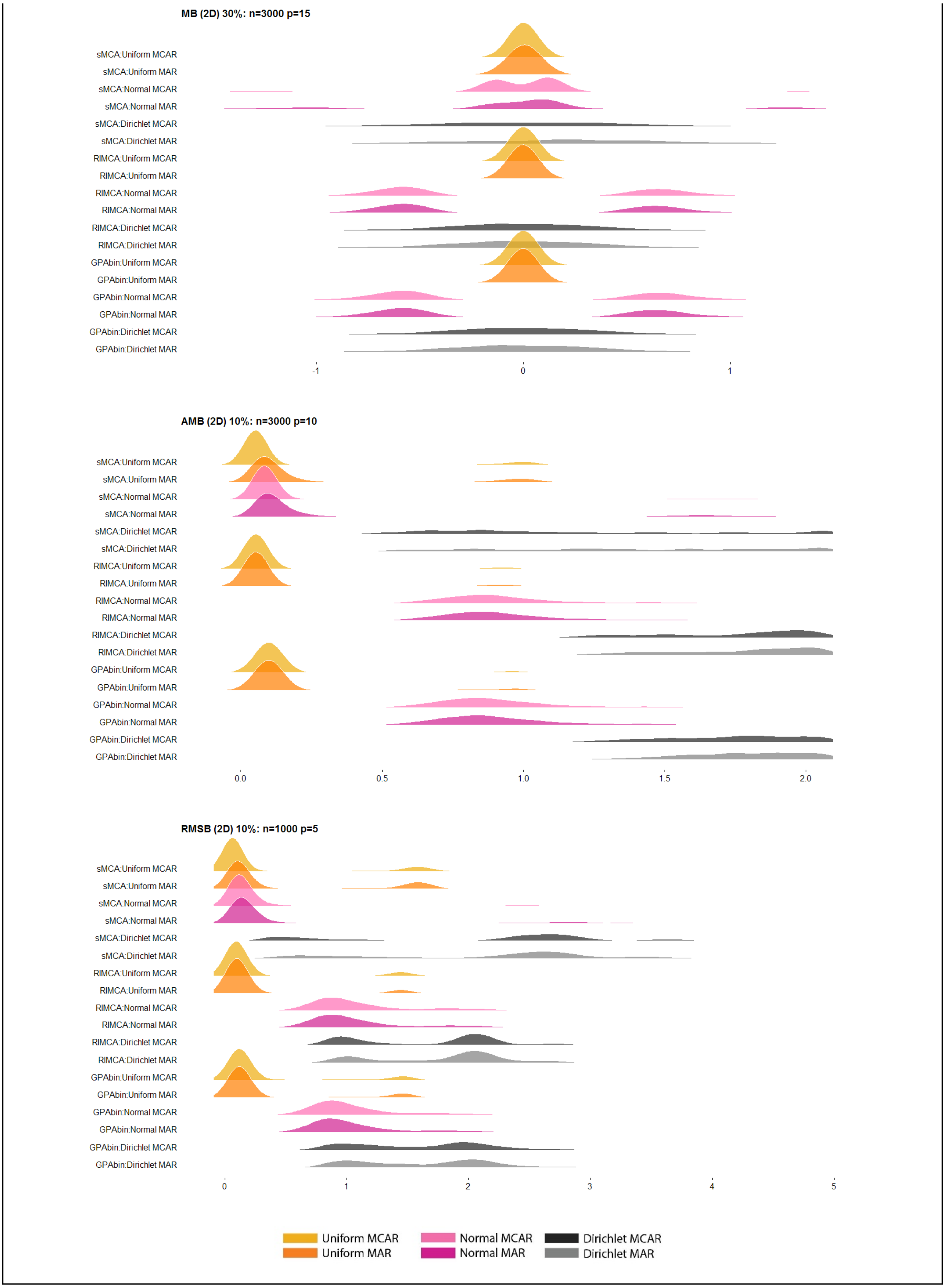


Figure 6.2 Density ridge plots to illustrate skewness of measures of comparison. A selection of simulation scenarios are illustrated.

The differences between the simulation distributions (Dirichlet, normal and uniform) will be discussed in more depth. It is however already evident from the selection in Figure 6.2 that the uniform distribution simulations provide consistent measures of comparison over 1000 simulations, whereas more dispersion is observed from the normal distribution simulations and especially the Dirichlet distribution simulations. Due to the skewness of the results, both the mean and median measures of comparison per 1000 repetitions are summarised in heat maps in the following sections (cf. 6.3.1 and 6.3.2). Heat maps are constructed by dividing the classification intervals (colours) into ten equally-spaced intervals determined by the range of each measure of comparison (cf. Table 6.8). However, the approach only applies to the bias measures (AMB, MB, RMSB) since the possible range for PS and CC values are known to be between zero and one.

The following ranges ([lower bound; upper bound]) are observed for the measures of comparison and given in Table 6.8:

Table 6.8 Ranges for measures of comparison: missing data approaches

Measures	Two dimensions	Maximum dimensions
PS	$[0,0.9997]$	$[0; 0.9659]$
CC	$[0.3248; 1]$	$[0.3522; 1]$
AMB	$[0.0106; 13.6548]$	$[0.0558; 6.9534]$
MB	$[-10.5105; 12.3802]$	$[-3.2818; 2.9193]$
RMSB	$[0.0121; 26.4073]$	$[0.0814; 18.9926]$

The top 10% of the AMB, MB and RMSB measures of comparison across all simulated scenarios for GPAbin, RIMCA and sMCA are used to define the cut-off values for the top 10% of measures of comparison.

The 10% coverage intervals are calculated as follows:

- $AMB < \left(0 + \frac{max(AMB)-\min(AMB)}{100} \times 10\right)$
- $\left(0 - \frac{max(MB)-\min(MB)}{100} \times 5\right) < MB < \left(0 + \frac{max(MB)-\min(MB)}{100} \times 5\right)$

- $RMSB < \left(0 + \frac{max(RMSB) - \min(RMSB)}{100} \times 10\right)$

The PS and CC measures of comparison can only result in values between zero and one, therefore the lower 10% of possible values for PS and the upper 10% of possible values for CC are used for the top 10% coverage intervals:

- $PS < 0.1$
- $CC > 0.9$

The ranges for the top 10% of measures of comparison are given in Table 6.9:

Table 6.9 Top 10% ranges for measures of comparison: missing data approaches

Measures	Two dimensions	Maximum dimensions
PS	$[0; 0.1)$	$[0; 0.1)$
CC	$(0.9; 1]$	$(0.9; 1]$
AMB	$[0; 1.3644)$	$[0; 0.6898)$
MB	$(-1.1445; 1.1445)$	$(-0.3101; 0.3101)$
RMSB	$[0; 2.6395)$	$[0; 1.8911)$

Selected simulation distributions will be further evaluated by using scatterplots. A figure of scatterplots consists of a number of panels to visualise the measurements of a particular measure of comparison. One panel of a figure presents the 1000 measures of comparison obtained for each combination of sample size and number of variables from a specific approach (GPAbin, sMCA, RIMCA) for a specific percentage of missing values and MDM. The scale of the y-axis is the same for a particular measure of comparison across all scatterplots in one figure. The optimal measures of comparison are indicated with a horizontal red line in each scatterplot and the optimal range (top 10% as specified in Table 6.9) is illustrated as a horizontal black dotted line. Vertical lines divide each panel into three sections, one for each sample size. Three colours and plotting characters are used to distinguish between the number of variables within a particular sample size.

6.3.1 Measures of bias

Separate heat maps for the means and medians of the three bias measures, over a 1000 repetitions per simulation scenario, are presented in this section in Figure 6.3 to Figure 6.8. Each cell in these heat maps (cf. Figure 6.3 to Figure 6.8) represents the median (or mean) measure of comparison over 1000 repetitions per simulation scenario. Ideally a majority of cells in a heat map will be associated with the interval regarded as the top 10% of possible values (cf. Table 6.9). The top 10% ranges for AMB (cf. Figure 6.3) and RMSB (cf. Figure 6.8) measures are indicated by the lightest classification shade in the heat maps. The classification colours in the centre of the colour key (cf. Figure 6.5) indicate the top 10% ranges for MB values.

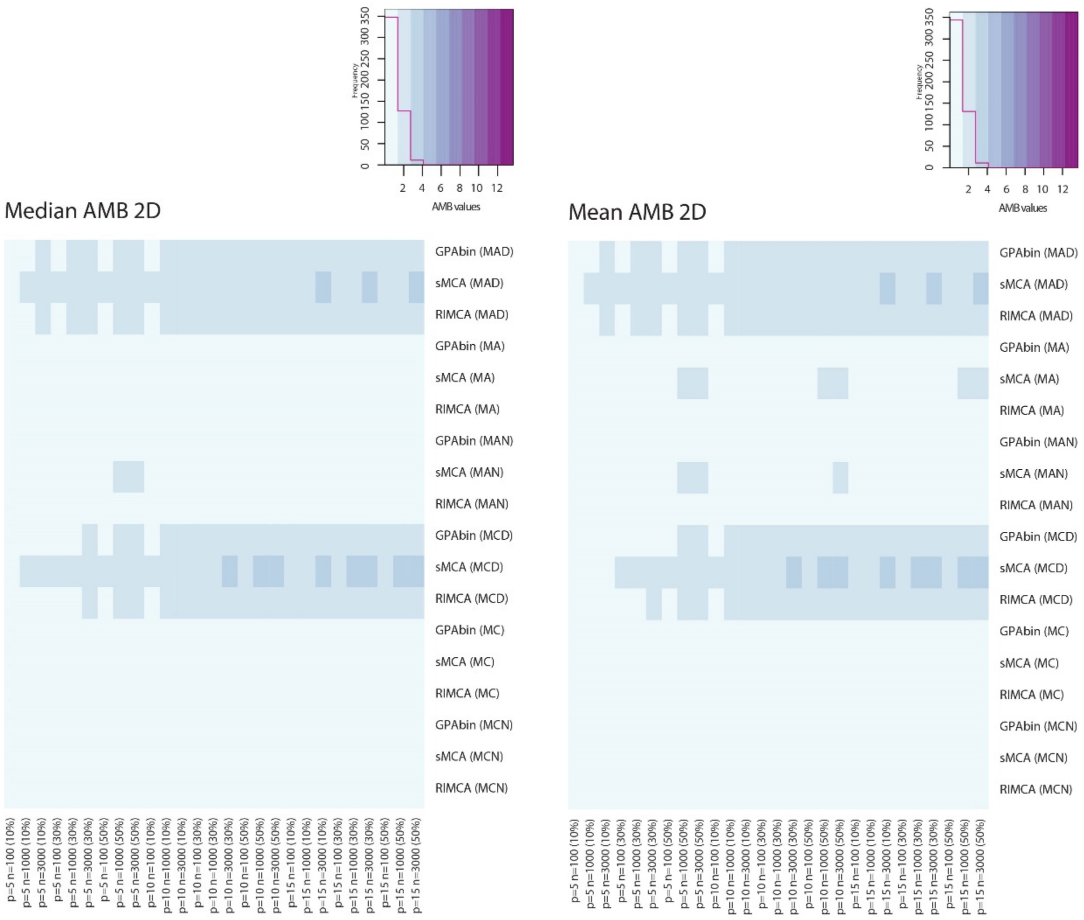


Figure 6.3 AMB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median AMB values. Right panel: Mean AMB values.

Even though the distribution of the measures of comparison were expected to be skewed, there are no severe deviations between the heat maps using either the mean (cf. Figure 6.3:

right panel) or median (cf. Figure 6.3: left panel) to summarise the results over 1000 repetitions for the AMB values. The frequency distributions of both summaries (mean and median) are similar as presented in the colour key of the heat maps. A majority of cells in the heat maps are classified in the top 10% interval (lightest shade) with darker shades of colours occurring for the Dirichlet distribution simulations, especially for the sMCA method. Some improvement is observed for the Dirichlet distribution simulations when the parameters of the data sets are smaller (i.e. $n = 100$ and $p = 5$). Overall, the Dirichlet distribution simulations are biased compared to the other two simulation distributions. The simulations obtained from normal and uniform distributions result in unbiased representation with the exception of the selected sMCA cases with higher bias values occurring when the percentage of missing values is large (50%). Scatterplots of the sMCA approach for 50% missing values with a MAR MDM are presented in Figure 6.4 for further inspection.

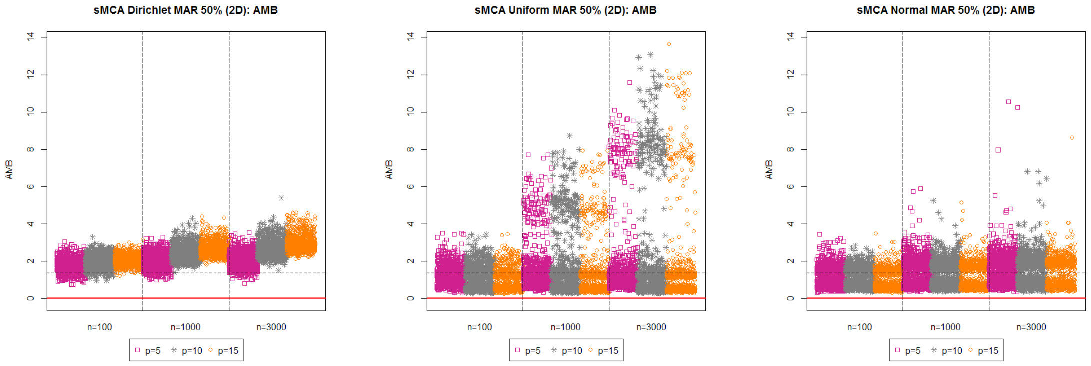


Figure 6.4 sMCA approach: 50% missing values with a MAR MDM for the three simulation distributions.

The scatterplots (cf. Figure 6.4) confirm the colour classifications of the heat maps presented in Figure 6.3. The AMB values increase as the sample size increases. The Dirichlet simulations result in a majority of AMB values outside the top 10% range with some exceptions when the number of variables is small ($p = 5$). The uniform distribution simulations result in more dispersion of AMB values, but a majority still occur within the optimal 10% range. The normal distribution simulations have AMB values with lower dispersion than the uniform simulations, however a large number of AMB values occur above the 10% cut-off when the sample sizes are large ($n = 1000$ and $n = 3000$).

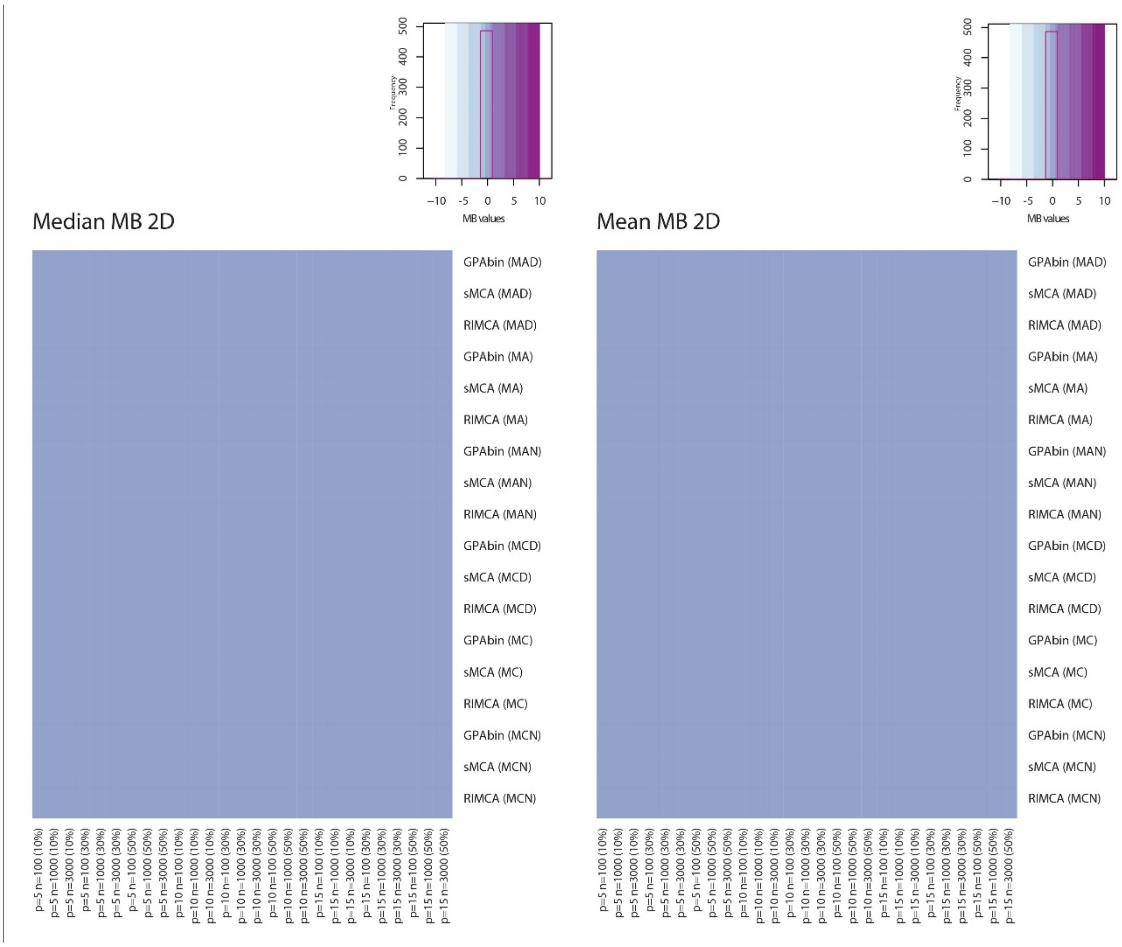


Figure 6.5 MB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median MB values. Right panel: Mean MB values.

All cells in both heat maps in Figure 6.5 occur in one of the optimal classes surrounding zero and therefore it can be concluded that all approaches are unbiased across the different simulations scenarios with regard to MB values. The differences that occur between the 1000 repetitions per simulation scenario are further investigated using scatterplots (cf. Figure 6.6). A horizontal inspection of the figures allows the evaluation of the performance of a particular method for different simulation scenarios as the percentage of missing values increases. Whereas, a vertical inspection enables the comparison of the MB values across different simulation distributions. The performance of the GPAbin and RIMCA approaches are consistent over all simulations irrespective of the MDM. Scatterplots depicting the MB values of these methods are presented in Appendix O. The imputation procedures (GPAbin and RIMCA), as presented in Appendix O.1 (MAR MDM) and O.2 (MCAR MDM), result in MB values closely concentrated around zero for the uniform distribution with less dispersion as the number of variables increases to $p = 15$. The MB values for the Dirichlet distribution become more dispersed as the sample size and the number of variables increase, but are still closely scattered around zero with selected cases occurring outside the top 10% range. The normal distribution simulations showed the most dispersion around zero. As the sample size increases the MB values move away from zero, showing more under- and overestimation, but still occurring within the top 10% bounds. The percentage of missing values does not seem to affect the MB values. The sMCA method shows more dispersion between the MB values per simulation scenario and are presented in Figure 6.6 (MAR MDM) and Figure 6.7 (MCAR MDM).

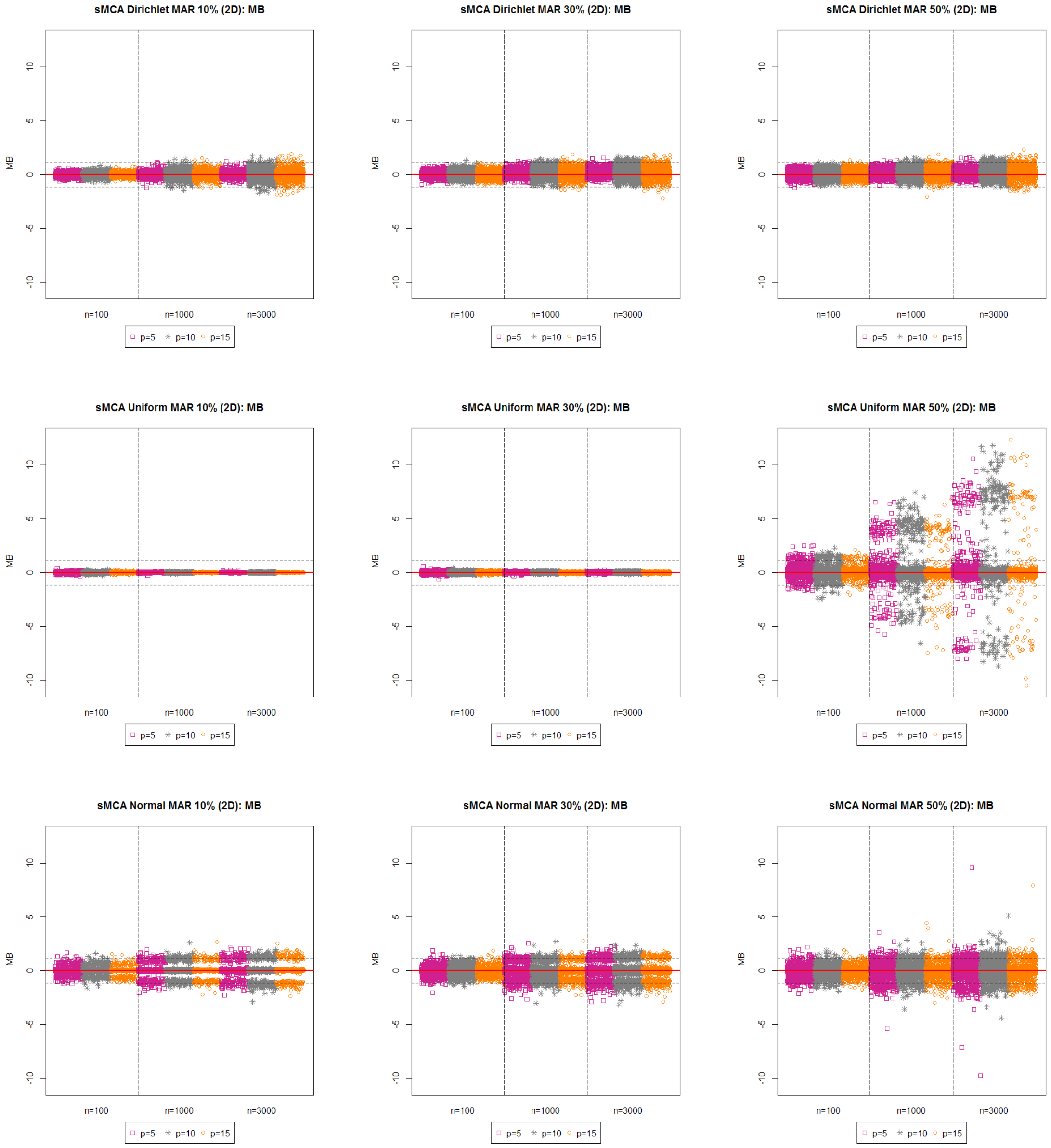


Figure 6.6 Scatterplots of the MB values for the sMCA method (MAR MDM). Left panels: 10% missing values. Middle vertical panels: 30% missing values. Right panels: 50% missing values. Upper panels: Dirichlet distribution. Middle horizontal panels: uniform distribution. Lower panel: normal distribution.

The sMCA method results in higher bias in selected simulations when the percentage of missing values is high (50%), especially for the uniform distribution simulations as can be observed from Figure 6.6 (middle right panel). The normal distribution simulations result in three groupings of MB values for 10% missing values as the sample size increases, which shows an overall MB of approximately zero in Figure 6.6 (lower left panel). The separation in the MB values for the normal distribution reduces as the percentage of missing values increases (cf. Figure 6.6: lower middle- and right panel).

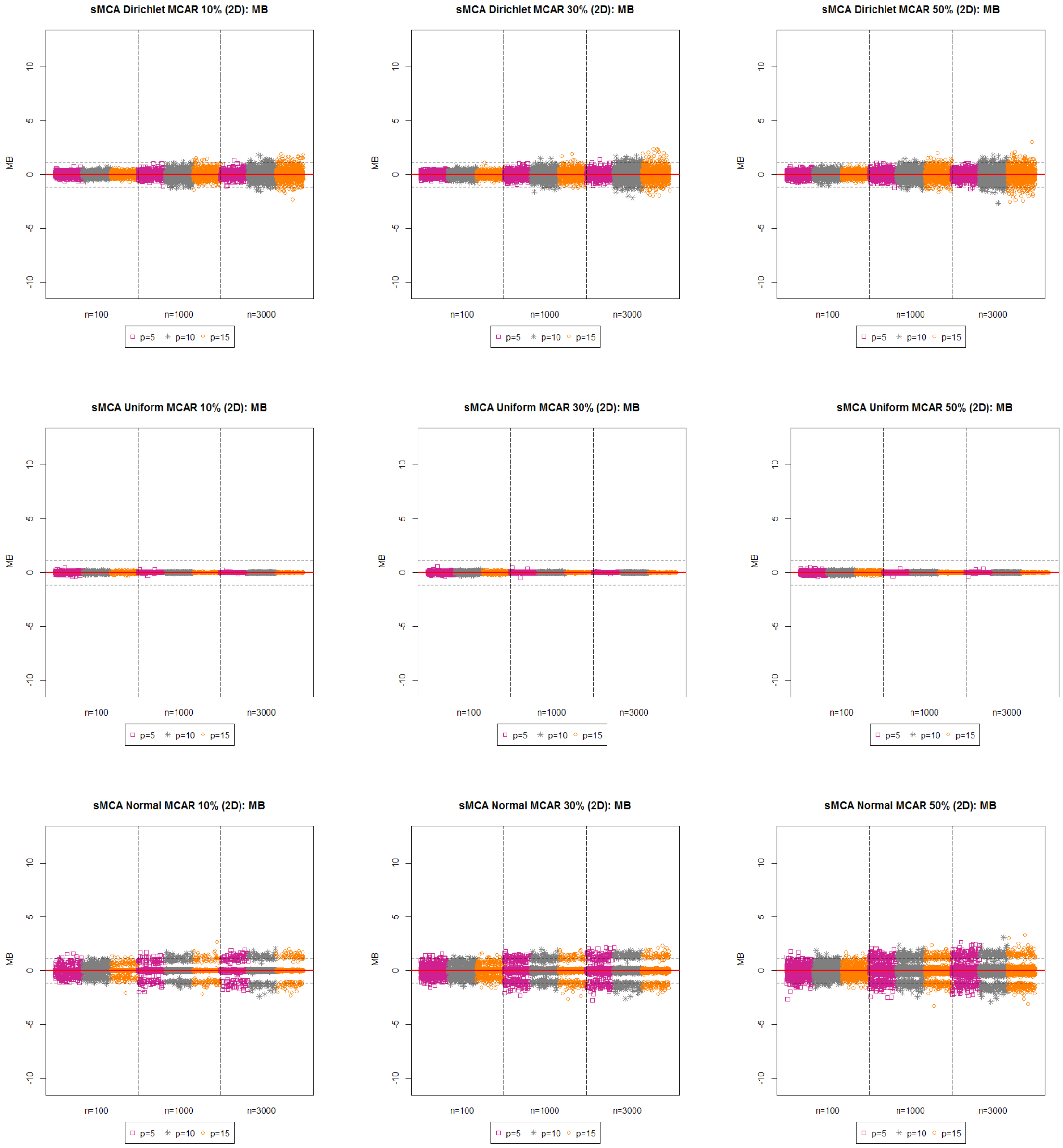


Figure 6.7 Scatterplots of the MB values for the sMCA method (MCAR MDM). Left panels: 10% missing values. Middle vertical panels: 30% missing values. Right panels: 50% missing values. Upper panels: Dirichlet distribution. Middle horizontal panels: uniform distribution. Lower panel: normal distribution.

The MCAR simulations in Figure 6.7 result in more consistent MB values in comparison to the scatterplots presented for the MAR simulations in Figure 6.6. Again, three groupings occur for MB values for the normal distribution simulations, but now for all percentages of missing values considered. The impact of the MDM on the sMCA method is clearly illustrated by the severe biased simulations that are observed for the sMCA method and MAR simulations in Figure 6.6 (middle right panel), but resulted in unbiased representation for the MCAR MDM (cf. Figure 6.7: middle right panel).

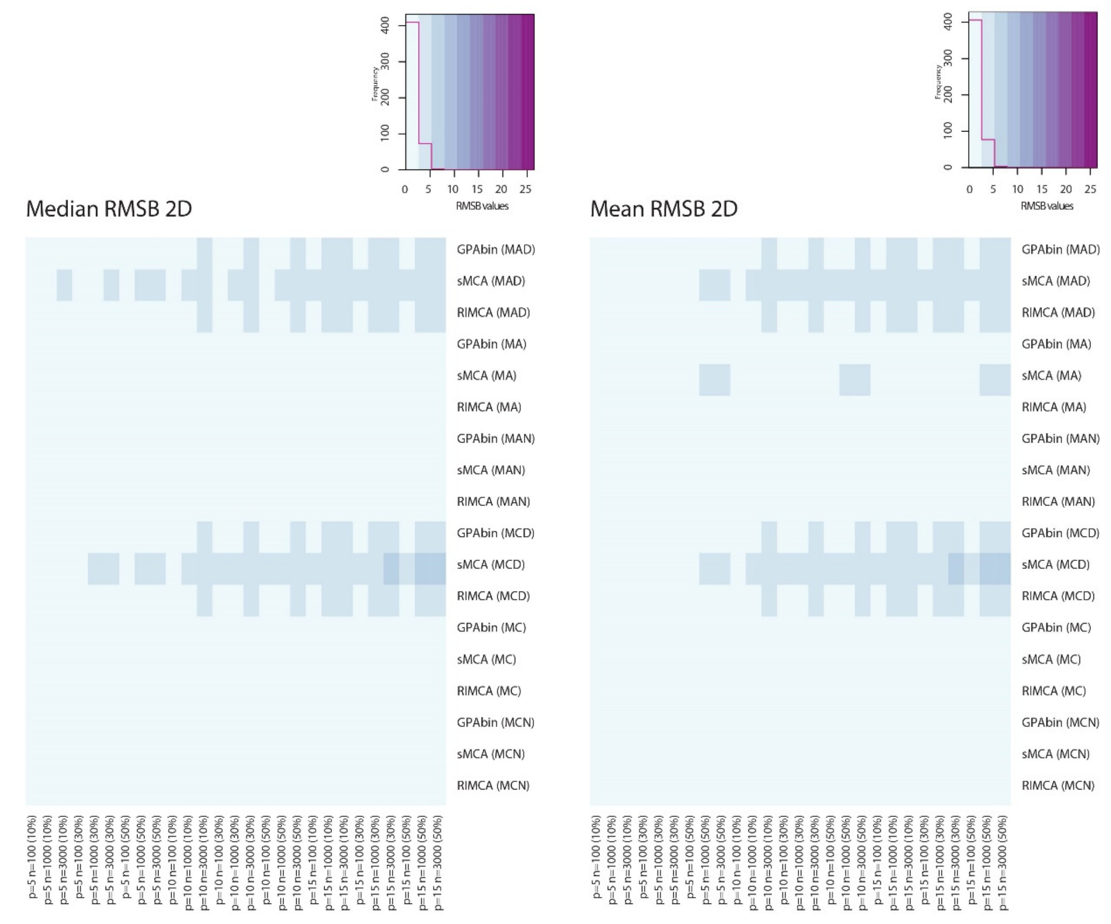


Figure 6.8 RMSB values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median RMSB values. Right panel: Mean RMSB values.

Again, the mean and median summaries are similar in Figure 6.8 for the RMSB values. The uniform distribution simulations are again unbiased across all simulation scenarios and missing data approaches; with the exception of the sMCA approach resulting in biased representation when the percentage of missing values is high (50%) with a MAR MDM (cf. Figure 6.8: right panel). The normal distribution simulations also result in unbiased representation, which showed slight bias with respect to the AMB values in Figure 6.3. The

Dirichlet distribution simulations are again biased and perform especially poorly when using the sMCA approach with a high percentage of missing values and large sample sizes ($n = 1000$ and $n = 3000$).

Overall the AMB (cf. Figure 6.3) and RMSB measures (cf. Figure 6.8) result in similar conclusions, however the AMB evaluation is slightly firmer than the RMSB evaluation and result in less classifications in the top 10% ranges.

6.3.2 Measures of fit

This section will also present heat maps and scatterplots to summarise the measures of fit. Again, selected scatterplots will follow heat maps to illuminate certain results. Each cell in the heat maps (cf. Figure 6.9 to Figure 6.13) represents the median (or mean) measure of comparison over 1000 repetitions per simulation scenario. The PS values (cf. Figure 6.9) indicated by the lightest classification shade and CC values (cf. Figure 6.13) indicated by the darkest classification shade are associated with the interval regarded as the top 10% of possible values (cf. Table 6.9).

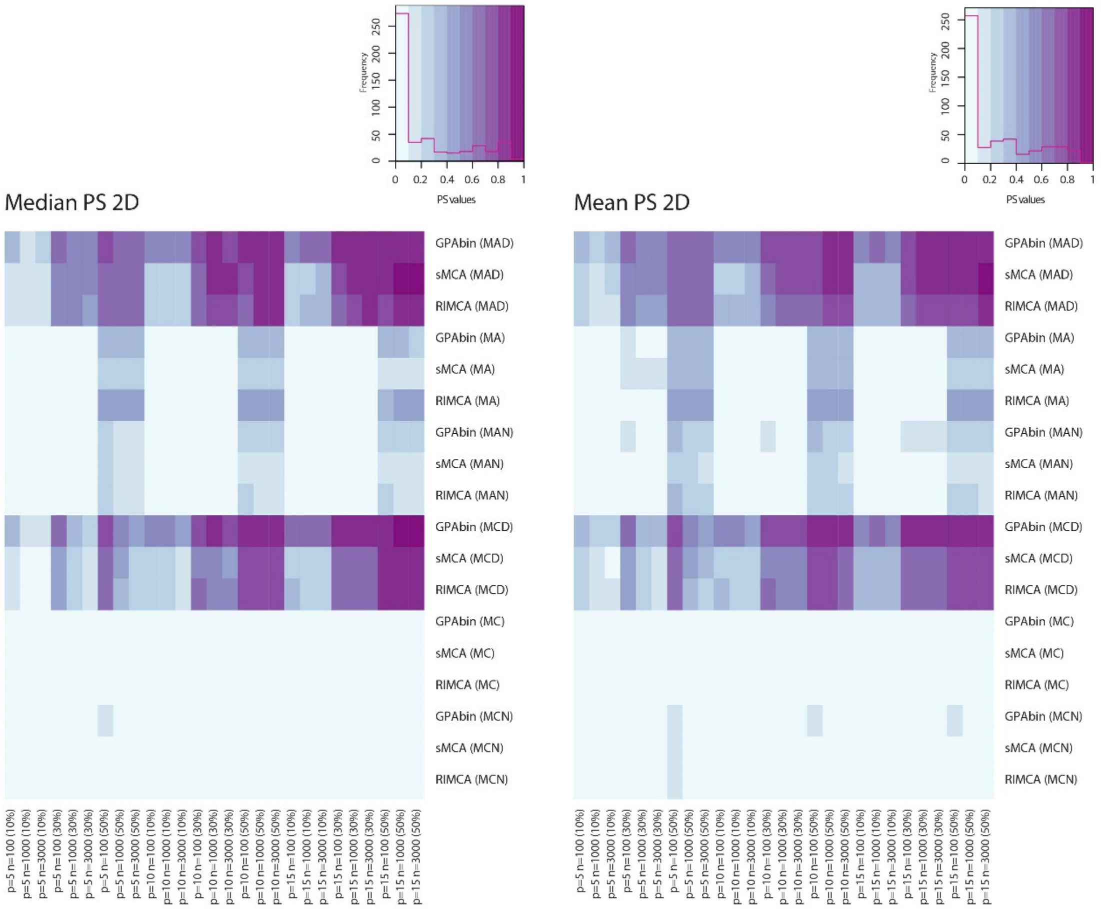


Figure 6.9 PS values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median PS values. Right panel: Mean PS values.

The PS values should be close to zero when two configurations are similar, which is regarded as a measure of good fit (cf. 4.6.1). The heat map in Figure 6.9 provides the same trend as observed in the heat maps of the bias measures in Figure 6.3 (AMB), Figure 6.5 (MB) and Figure 6.8 (RMSB). The uniform distribution simulations result in good fit measurements (close to zero) for all simulation scenarios, with the exception of slightly lower PS values for the MAR MDM with 50% missingness for all approaches. The sMCA approach applied to uniform and normal distributed data sets with 50% missingness and a MAR MDM results in slightly lower PS values than the imputation methods; illustrated by the lighter colour classification of the cells in Figure 6.9. In general, the cells in the heat map associated with the normal distribution simulations for MAR MDMs are slightly lighter for higher percentages of missing values than the uniform distribution PS values. The GPAbin method applied to normal distribution simulations with a small sample size ($n = 100$) results in slightly poorer fit for the following instances; higher percentages of missing values (30% and 50%) with a

MAR MDM and 50% missing values with an MCAR MDM, irrespective of the number of variables.

Overall, the approaches perform well for uniform and normal distribution simulations. The MCAR MDM simulations consistently result in smaller PS values compared to the MAR MDM simulations. The Dirichlet distribution simulations do not result in optimal PS values, but show some improvement for a small percentage of missing values as was observed from the bias measures (cf. 6.3.1) and also when the MDM is MCAR. Regarding the MAR MDMs; some higher PS values (median and mean) are observed when the percentage of missing values is large (50%) and when the percentage of missing values is 30% for a small sample size ($n = 100$) considering all number of variables.

The performance of each method is further investigated in the scatterplots in Figure 6.10 to Figure 6.12. These figures consist of 18 panels, since the general trends are of importance and not the scrutinisation of each observation. One figure contains the PS values over all simulation scenarios per approach (GPAbin, sMCA, RIMCA). The discussion will precede the figures.

Similar trends are observed from the three approaches: GPAbin (cf. Figure 6.10), RIMCA (cf. Figure 6.11) and sMCA (cf. Figure 6.12). Only the notable differences between the approaches will be discussed, along with general remarks.

It is noted that as the percentage of missing values increases the PS values are more dispersed. A majority of values occur outside the top 10% range when the percentage of missing values is high (50%) and a MAR MDM is considered. Overall, the Dirichlet distribution simulations do not result in good fit, but slightly better performance is observed when the number of variables is small ($p = 5$). By evaluating the panels separately, it can be seen that there is an improvement in the PS values for the normal and uniform distributions as the sample size and number of variables increase when considering the MCAR MDMs. Also, for the normal and uniform distribution simulations, it is clear that the MCAR MDMs result in better fit measures compared to the MAR MDMs, especially when the percentage of missing values is large (50%).

The sMCA approach (cf. Figure 6.12) result in fit measures below the top 10% range when applied to normal distribution simulations with an MCAR MDM (10% and 30% missing values).

This is not the case for the PS values obtained from the GPAbin (cf. Figure 6.10) and RIMCA approaches (cf. Figure 6.11), as the values are more dispersed in comparison to the scatterplots of the sMCA method (cf. Figure 6.12).

Regarding the uniform distribution simulations with 10% missing values and a MAR MDM; fit measures are most dispersed for the sMCA approach (cf. Figure 6.12) and most concentrated measures are obtained from the RIMCA method (Figure 6.11). Large portions of the PS values for the sMCA method (cf. Figure 6.12) for a larger percentage of missing values (30% and 50%) occur outside the top 10% range.

The PS values of the RIMCA method (cf. Figure 6.11) are slightly more concentrated and situated in closer proximity to the top 10% range than the GPAbin method (cf. Figure 6.10). The slight dispersion of the GPAbin fit measures could be attributed to the additional uncertainty that is incorporated by MI.

As was observed from the similarity percentages (cf. 6.2) and measures of bias (cf. 6.3.1), the uniform distribution simulations result in the best representation of the simulated complete visualisations compared to the other simulation distributions. The normal distribution simulations result in satisfactory representation for a majority of simulation scenarios and the Dirichlet distribution simulations result in the worst fit.

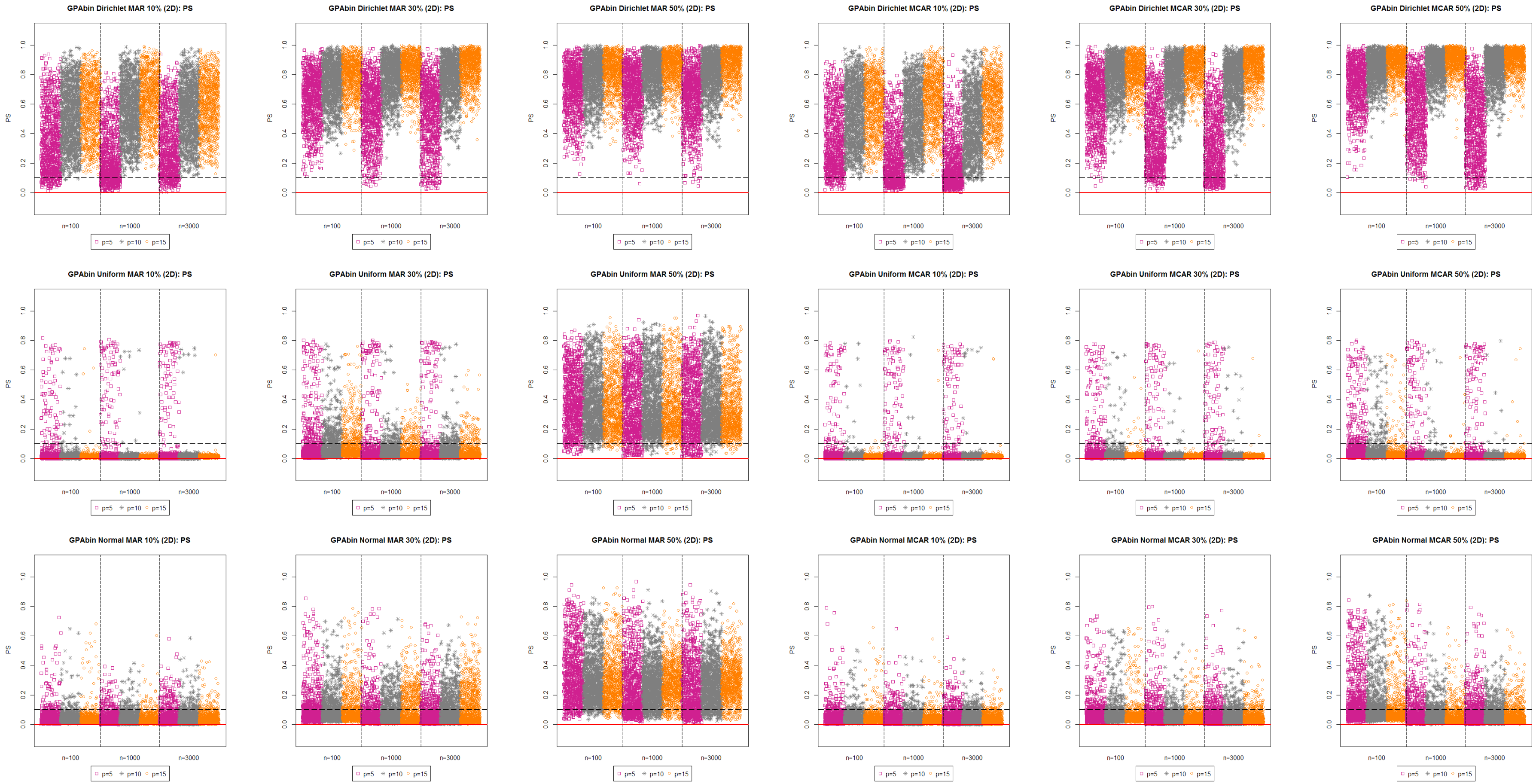


Figure 6.10 Scatterplots of PS values for the GPAbin approach across all simulation scenarios.

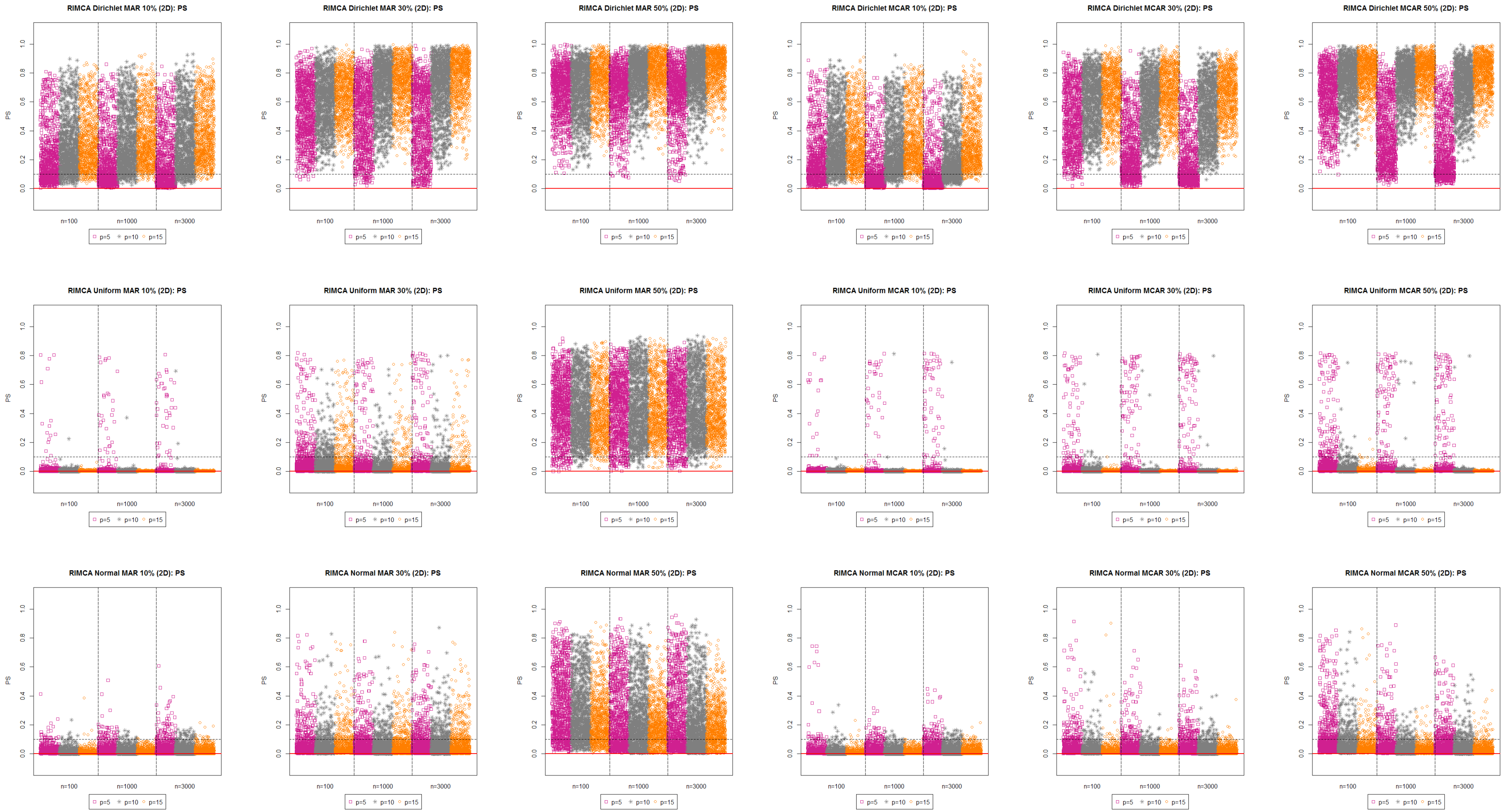


Figure 6.11 Scatterplots of PS values for the RIMCA approach across all simulation scenarios.

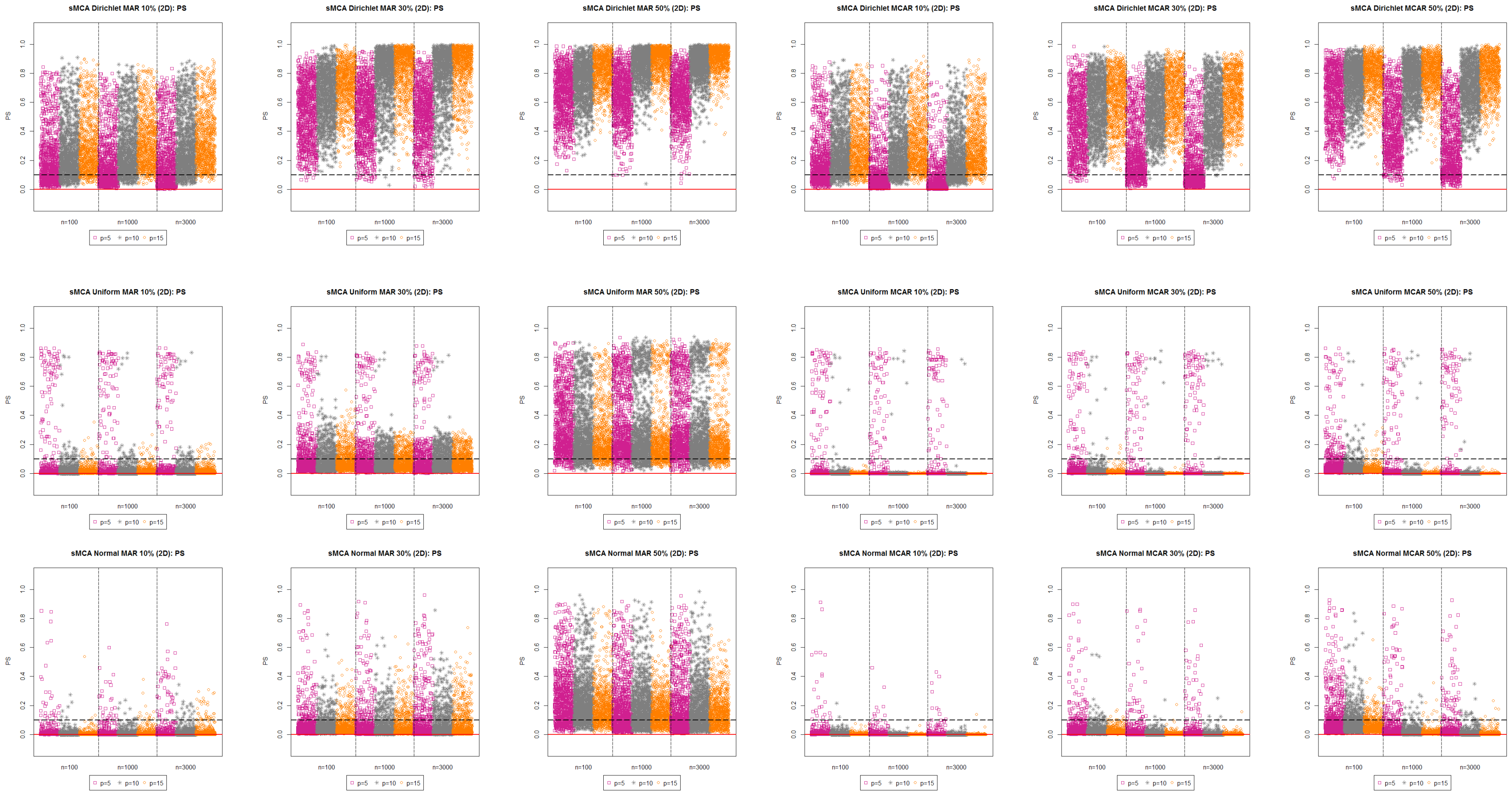


Figure 6.12 Scatterplots of PS values for the sMCA approach across all simulation scenarios.

The CC results (cf. Figure 6.13) confirm the statement that the coefficients commonly occur close to one even if the two configurations are not an exact match (Borg & Groenen, 2005). Apart from a few additional lighter shaded cells occurring in the mean CC heat map(cf. Figure 6.13: right panel), the displays of the mean and median (cf. Figure 6.13:left panel) heat maps are regarded as similar. The majority of values occur in the optimal classification interval ($0.9 < CC \leq 1$) with some lighter colour classifications occurring for the Dirichlet distribution. Small CC values result from uniform distribution simulations with a MAR MDM and 50% missing values, as well as selected normal distribution simulations with the same missingness. As was previously observed in Figure 6.9 and Figure 6.10, the GPAbin method performs slightly worse for normal distribution simulations with a high percentage of missing values (50%) with the following dimensions: $p = 15$ and $n = 1000, 3000$.

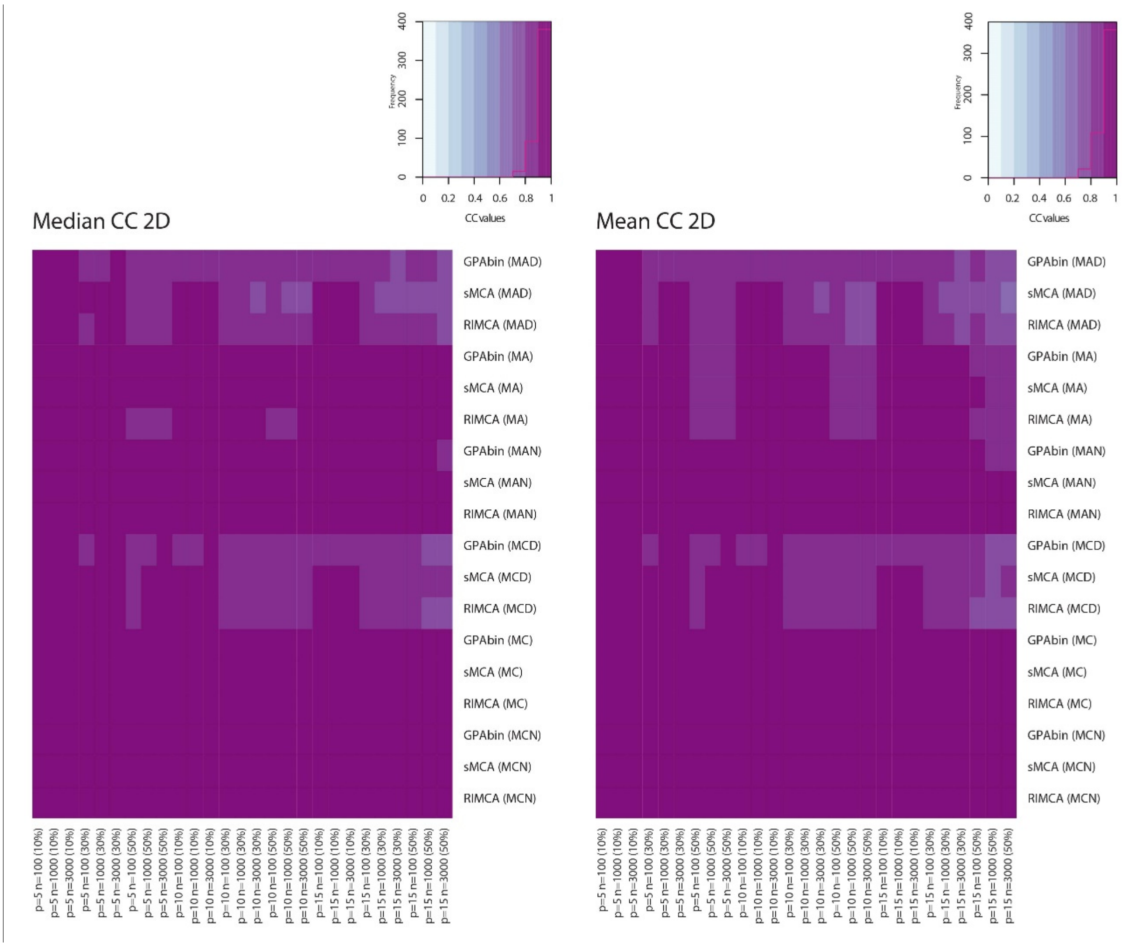


Figure 6.13 CC values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Median CC values. Right panel: Mean CC values.

The normal distribution simulations with a MAR MDM and 50% missing values are further evaluated in Figure 6.14. The GPAbin approach was the only method that did not result in optimal classifications in Figure 6.13 (left- and right panel) for these specifications and therefore the differences between the approaches are investigated.

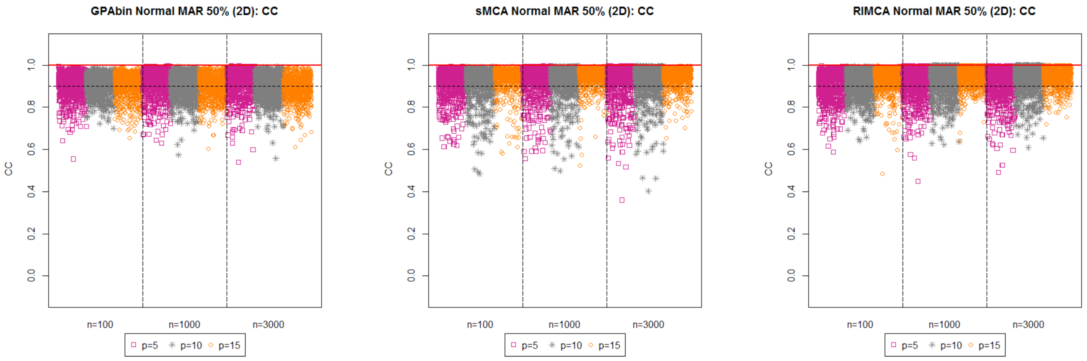


Figure 6.14 Scatterplots of normal distribution simulations: 50% missing values and MAR MDMs. Left panel: GPAbin. Middle panel: sMCA. Right panel: RIMCA.

A majority of CC values for the GPAbin method are still within the top 10% range (cf. Figure 6.14: left panel), however the RIMCA and sMCA methods show more concentrated CC values within the top 10% range confirms the colour classifications presented in Figure 6.13. The GPAbin approach (cf. Figure 6.14: left panel) results in consistent CC values across different simulation scenarios when compared to the CC values of the other approaches (cf. Figure 6.14: middle- and right panel).

In conclusion, the MB and CC values easily result in optimal values (cf. Table 6.9), which are not necessarily reflected by the other measures of comparison. For this reason, when deciding upon a successful method to visualise data containing missing values, these measures are not to be considered in isolation.

The results obtained from the bias measures (AMB and RMSB) and the PS fit measures are in agreement with the trends observed in the similarity percentages (cf. 6.2). All approaches (GPAbin, sMCA, RIMCA) successfully preserve the associations of the original simulated MCA biplots from a uniform distribution. The approaches also perform well with simulations from the normal distribution, with the exception of slightly poorer fit measures for the sMCA and GPAbin approaches in high percentages of missing values (50%) and MAR MDMs. All methods applied to Dirichlet distribution simulations perform poorly with respect to bias and goodness of fit, however slight improvement is observed when the sample size and number of variables are small. The GPAbin and RIMCA methods consistently result in unbiased configurations of the normal and uniform distribution simulations. The sMCA method performs well in lower percentages of missing values and an MCAR MDM for both uniform and normal distribution simulations. Overall, the performance of the methods improve when applied to data sets with MCAR MDMs.

6.4 Comparison in maximum available dimensions

The focus of this study is to obtain unbiased visual representation of multivariate categorical data containing missing values. Higher (more than three) dimensions cannot be visualised and is therefore not aligned with the scope of this research study. However, as multidimensional data sets are approximated in lower dimension, consequently information that is only available in higher dimension is forfeited. In order to establish the effect of lower approximation on the bias and fit of completed data visualisations compared to the complete data visualisations, the measures of comparison are evaluated in the full space.

Only the mean measures of comparison will be further considered, since the distributions of the median and mean summaries did not differ notably when compared in two dimensions (cf. 6.3). The MB and CC values will not be visualised, since these measures did not result in discriminating scenarios. All MB values and the majority of CC values were observed in the top 10% range.

The heat maps for the mean AMB and RMSB are presented in Figure 6.15.

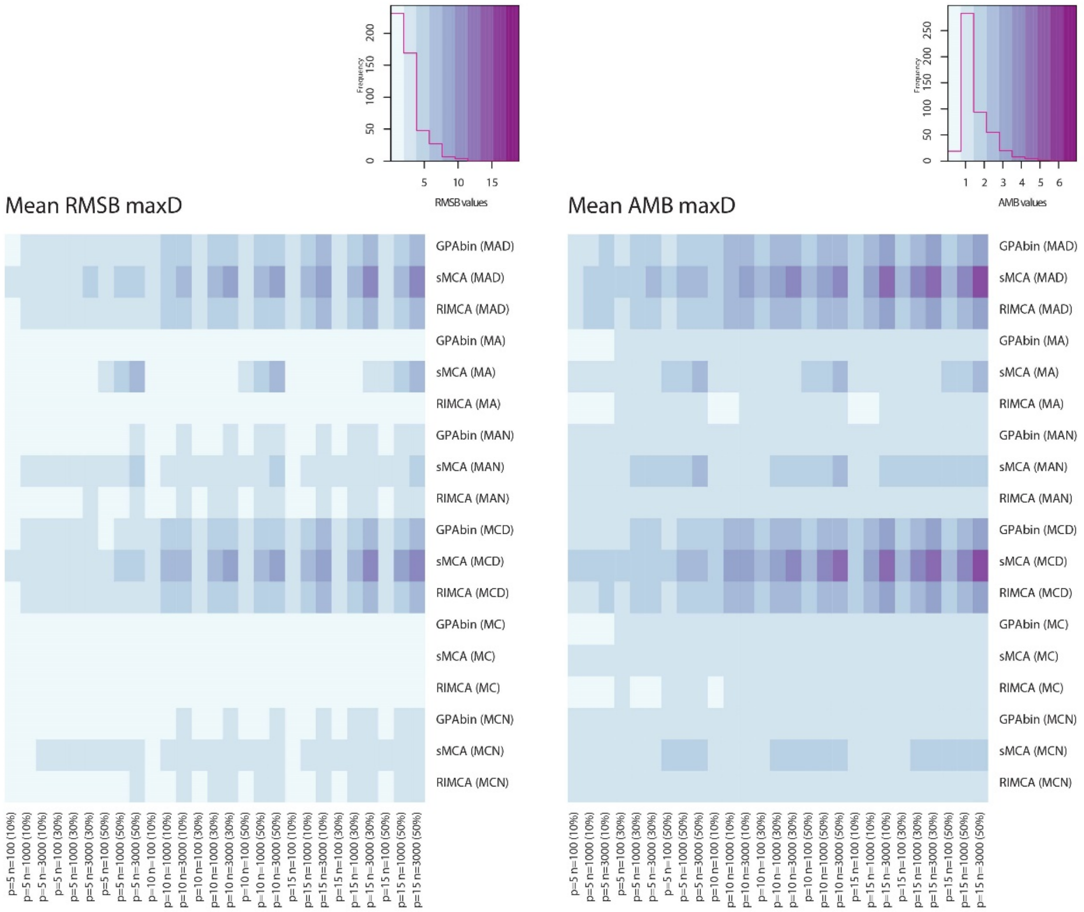


Figure 6.15 Measures of comparison values per simulation scenario over 1000 repetitions (comparison in maximum dimensions). Left panel: Mean RMSB values. Right panel: Mean AMB values.

Similar patterns are again observed in both heat maps (cf. Figure 6.15), as was the case for the two-dimensional AMB (cf. Figure 6.3) and RMSB (cf. Figure 6.8) values.

Considering the RMSB values (cf. Figure 6.15: left panel), more simulation scenarios (cells) in the heat map are classified outside the top 10% range when configurations are compared in maximum dimension than in two dimension (cf. Figure 6.8). The normal distribution simulations result in bias when compared in maximum dimensions for both MDMs, which was not the case for comparisons in two dimensions (cf. Figure 6.8). The uniform distribution simulations remain unbiased when compared in maximum dimension.

The AMB values (cf. Figure 6.15: right panel) show less unbiased classifications with only selected simulation scenarios of GPAbin and RIMCA resulting in mean AMB values within the top 10% range. It is clear that the bias increases for distributions when configurations are compared in higher (more than two) dimensions.

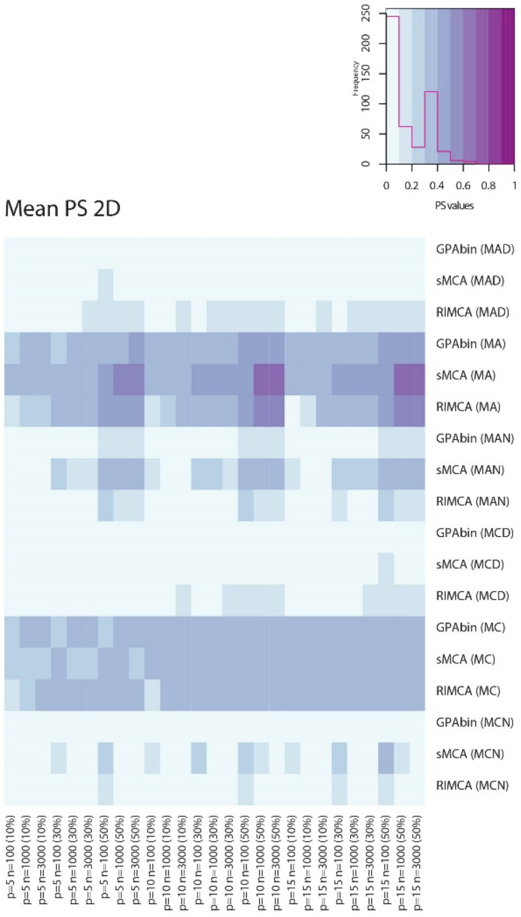


Figure 6.16 Mean PS values per simulation scenario over 1000 repetitions (comparison in maximum dimensions).

The mean PS measures (cf. Figure 6.16) improve for the Dirichlet distribution when compared in maximum dimension, whereas the uniform distribution simulations perform poorly when compared in higher dimension. The normal distribution simulations still perform well, but the sMCA method resulted in better fit measures when compared in two dimensions (cf. Figure 6.9).

Therefore, the two-dimensional representation of configurations are less biased and results in good fit when compared to the original configurations simulated from uniform and normal distributions. The Dirichlet distribution simulations are better represented in higher dimension with regard to the measures of fit, but still result in biased representations. Data with skewed distributions containing missing values are therefore poorly approximated in lower dimension and visualisation techniques should be interpreted with caution.

6.5 Conclusion

This chapter evaluated the influence of different simulation parameters on the success of three approaches: GPAbin, sMCA and RIMCA. Similarity percentages (cf. 6.2), measures of fit (cf. 6.3.2) and bias (cf. 6.3.1) were used to evaluate the approaches. The best visual representations were obtained from uniform distribution simulations, followed by normal distribution simulations and the poorest representations were obtained from Dirichlet distribution simulations.

Configurations were only compared in two- or maximum dimensions. It was found that the configurations were biased when compared in higher dimension. The fit measures improved for the Dirichlet distribution simulations when compared in maximum dimensions, whereas the uniform distribution simulations resulted in better fit when compared in two dimensions. The sMCA method for normal distribution simulations resulted in larger PS values (poor fit) when compared in higher dimension. Assessing the best number of dimensions to compare configurations was beyond the scope of this research, but will be considered for future research projects.

In general, it was found that for a high percentage of missing values all approaches (GPAbin, sMCA and RIMCA) perform better for missing data with MCAR MDMs than MAR MDMs. This is in agreement with the literature, since even deletion techniques can result in unbiased inference for MCAR MDMs (Raghunathan *et al.*, 2001; Schafer, 1997; Van Buuren, 2012). The sMCA method result in satisfactory bias and fit measures when the percentage of missing values is low (10%), but overall the GPAbin and RIMCA methods achieve consistent results across simulation scenarios as noted in detail in the preceding sections of this chapter.

The effect of the number of MIs used before the GPAbin method has not been investigated and will also be considered in forthcoming research.

Chapter 7 will focus on the plausibility of predicting responses from multiple imputed MCA biplots using two prediction approaches: GPAbin prediction and the Majority rule prediction.

Chapter 7
Results: Prediction of categorical data sets

7.1 Introduction

This chapter explores the validity of predicting a final multivariate categorical data set from results obtained from standard data analysis techniques applied to multiple imputed data sets. Since the main objective of this research study is to expand methodology on the visualisation of multivariate categorical data containing missing values, the standard data analysis technique again refers to the construction of MCA biplots.

All available dimensions are used to predict a final data set (cf. 4.4). As stated by Gower and Hand (1996) the prediction of responses by using the nearest CLP in the biplot would result in prediction errors due to the approximation in two dimensions for data sets of higher dimensions. A better approach would be to construct convex prediction regions in the full space for each variable to classify the responses of samples positioned in prediction regions (Gower *et al.*, 2011; Gower & Hand, 1996).

The Majority rule prediction (cf. 4.4.1) utilises all possible dimensions of each multiple imputed MCA solution and the GPAbin prediction approach (cf. 4.4.2) utilises all dimensions of the combined MCA solution obtained from the GPAbin approach (cf. 4.3). The success of the approaches is based on the visual interpretation in two dimensions. In order to evaluate the performance of the prediction methods, an MCA biplot is constructed for the predicted data set and compared to the MCA biplot of the complete simulated data set in two dimensions.

Schafer (1997) advises that imputed data should not be mistakenly regarded as complete data. Disregarding the fact that unobserved information is substituted with plausible responses could lead to misleading inference that does not reflect the uncertainty of the imputation task, resulting in underestimation of variation. The aim of imputation methods is not to recover the original complete data, but rather to preserve the inference that would have been obtained from the complete data analysis. Imputation is not regarded as prediction methods, in the context of regression analysis, in which the differences between the true observations and the predicted observations are minimised (Van Buuren, 2012). Therefore, only the configurations (i.e. estimates) obtained from the complete and predicted data sets are compared and not the complete and completed data matrices.

7.2 Measures of comparison

The same measures of comparison are used to evaluate the differences between the two prediction approaches as in Section 6.3. The ranges ([lower bound; upper bound]) of the measures of comparison for the two prediction methods are given in Table 7.1 below:

Table 7.1 Ranges for measures of comparison: prediction methods.

Measures	Two dimensions	Maximum dimensions
PS	$[0; 0.9959]$	$[0; 0.9736]$
CC	$[0.2456; 1]$	$[0.3285; 1]$
AMB	$[0; 12.4441]$	$[0; 6.9211]$
MB	$[-11.0468; 11.9155]$	$[-2.6771; 3.4384]$
RMSB	$[0; 27.6457]$	$[0; 20.2072]$

The top 10% of the AMB, MB and RMSB measures of comparison across all simulated scenarios for the GPAbin prediction and Majority rule prediction are used to define the cut-off values for the top 10% of the measures of comparison as was done for the missing data approaches in Section 6.3. The ranges for the top 10% of the measures of comparison are presented in Table 7.2.

Table 7.2 Top 10% ranges for measures of comparison: prediction methods

Measures	Two dimensions	Maximum dimensions
PS	$[0; 0.1)$	$[0; 0.1)$
CC	$(0.9; 1]$	$(0.9; 1]$
AMB	$[0; 1.2444)$	$[0; 0.6921)$
MB	$(-1.1481; 1.1481)$	$(-0.3058; 0.3058)$
RMSB	$[0; 2.7646)$	$[0; 2.0207)$

The top 10% interval widths presented in Table 7.2 are similar to the intervals of the missing data approaches in Table 6.9, with a slightly narrower interval for the AMB values and slightly wider intervals for the RMSB and MB values.

The heat maps presented in Chapter 6 cannot be directly compared to the heat maps in this chapter, since the colour classifications depend on the ranges (cf. Table 6.9 and Table 7.2) determined by the observed measures of comparison (cf. Table 6.8 and Table 7.1) for each specific chapter.

In the figures of this chapter the GPAbin prediction method is referred to as 'GPAbin Pred' and the Majority rule prediction is referred to as 'Maj.rule Pred'. Again, the columns of the heat maps (cf. Figure 7.1 and Figure 7.4) denote a specific combination of sample size, number of variables and percentage of missing values. The same abbreviations (MAD, MA, etc.) as defined in Table 6.1 are used for the row annotations along with the names of the two prediction methods.

As the mean and median summaries of the measures of comparisons were not notably different in Section 6.3, only the mean of the 1000 repetitions per simulation scenario will be considered for the evaluation of the prediction methods. Again, each cell of a heat map (cf. Figure 7.1 and Figure 7.4) will represent the mean of a 1000 measures for a specific simulation scenario.

7.3 Measures of bias

As was observed from the missing data approaches (cf. 6.3.1); the mean MB values result in a heat map (cf. Appendix P.1) with no discrimination between methods with all the mean MB values of the prediction methods occurring in the optimal colour classification range. Therefore, only the mean AMB and RMSB values are considered and summarised to evaluate the bias of the prediction methods.

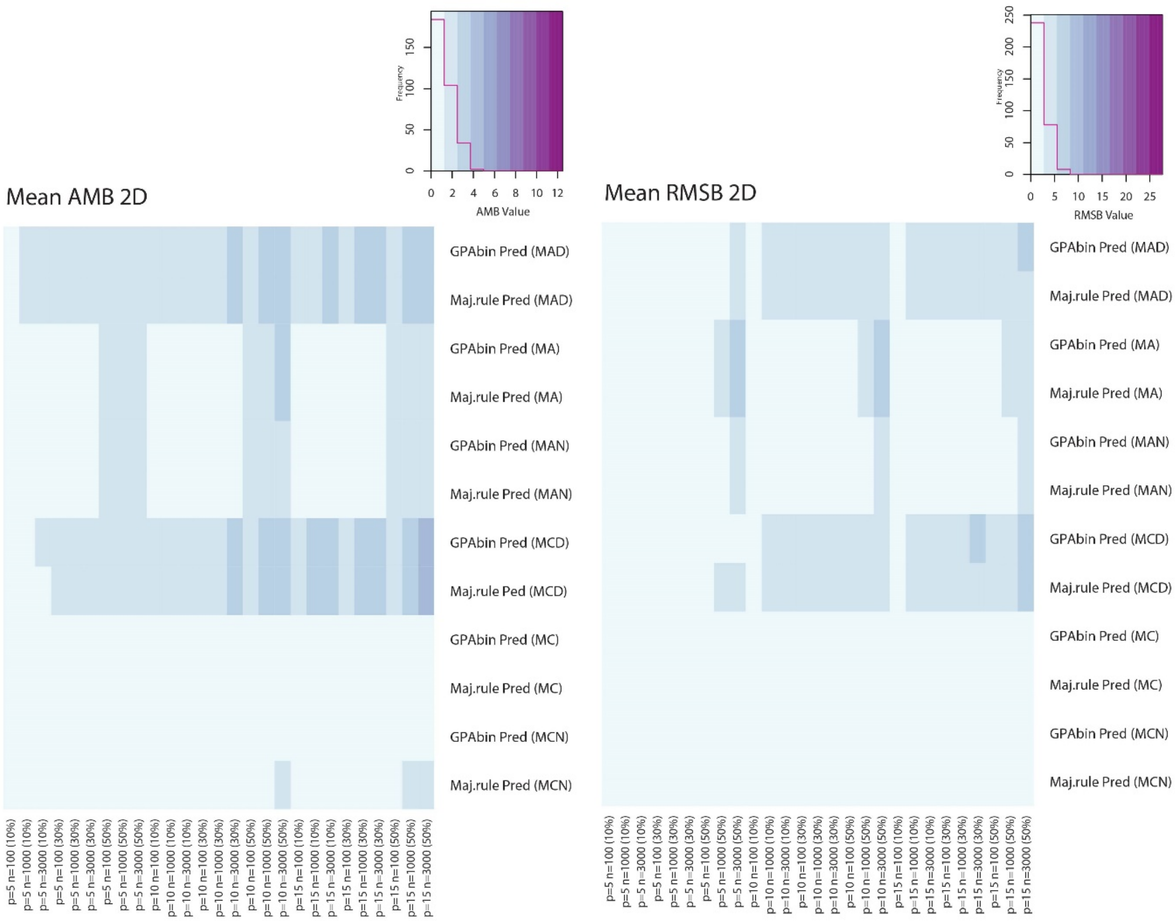


Figure 7.1 Measures of comparison values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Mean AMB values. Right panel: Mean RMSB values.

The two prediction methods result in similar bias measures with AMB again providing a stricter evaluation of bias than the RMSB (cf. 6.3.1). Methods applied to simulations from a Dirichlet distribution result in biased representation of the original simulated MCA biplots when compared to simulations from uniform and normal distributions. However, the RMSB summary (cf. Figure 7.1: right panel) shows that the prediction methods are unbiased for selected cases of the Dirichlet distribution simulations when the number of variables is small ($p = 5$). The prediction methods applied to uniform and normal distribution simulations are more biased when the percentage of missing values is large (50%). Prediction methods applied to simulations with an MCAR MDM result in unbiased representation with the exception of the Majority rule prediction for simulations from a normal distribution (cf. Figure 7.1: left panel). Overall, predictions for data sets with a high percentage of missing values (50%) and a MAR MDM result in biased representation for all simulation distributions (uniform, normal and Dirichlet).

Since the evaluation of the AMB values (Figure 7.1: left panel) is slightly firmer than the RMSB values (Figure 7.1: right panel), density estimates of the AMB values are further investigated.

The two prediction methods result in similar kernel density estimates using the default specifications of the `density()` function available in the `R` package, `stats` (R Core Team, 2017). The x-axis does not make provision for the entire observation range (cf. Table 7.1), since the focus is on the density of the simulations within the top 10% range (cf. Table 7.2). The cut-off value for the AMB results for the prediction methods is 1.2444.

Notable differences between the densities of the two prediction methods are only observed for normal simulations with 30% missing values and an MCAR MDM for the following data sets: $n = 1000, p = 10$ and $n = 3000, p = 5, 10, 15$ (cf. Figure 7.2). Also, slight differences between the densities of the prediction methods are observed for uniform simulations with a sample size of $n = 1000$ and number of variables: $p = 5, 15$ with 50% missing values and an MCAR MDM (cf. Figure 7.3).

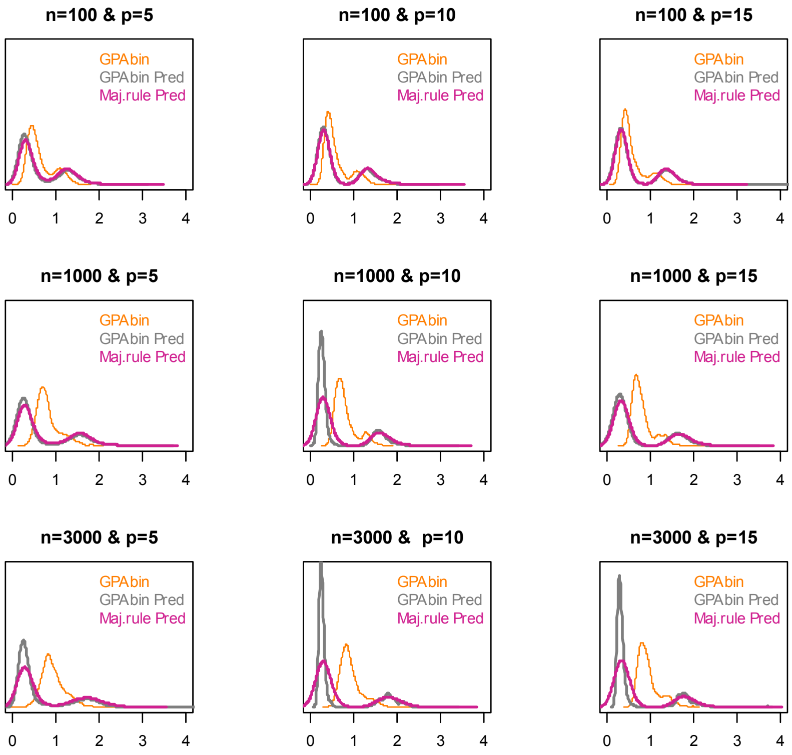


Figure 7.2 Kernel density estimates of AMB values for prediction methods (MCN 30% missing values).

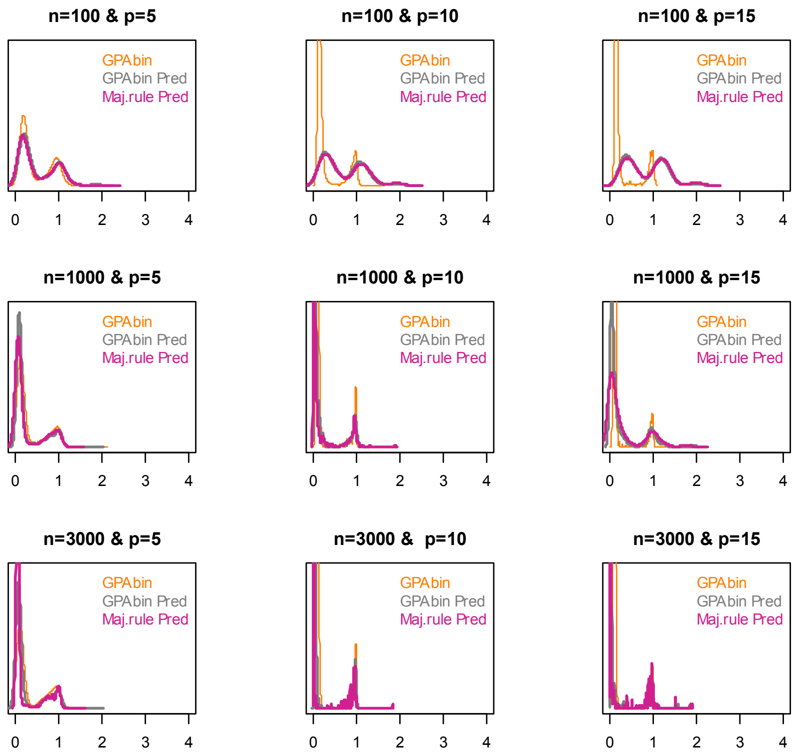


Figure 7.3 Kernel density estimates of AMB values for prediction methods (MC 50% missing values).

The density estimates of the GPAbin approach are also presented in Figure 7.2 and Figure 7.3 with a thinner plotting line with an orange colour. It is expected that the measures of comparison of the prediction methods will be similar to the GPAbin approach (cf. 6.3) since the prediction methods rely on the GPAbin biplot and also the multiple imputed MCA biplots used for GPAbin. Being mindful of the fact that the final visualisations are based on the MCA solution of one completed (predicted) data set, it could lead to biased representation of the original complete visualisations. The predictions are made from results based on MI and are therefore not directly comparable to SI methods. However, it is possible that the variation could be underestimated in the prediction visualisations, which is expected to be reflected in improved fit measurements when compared to the GPAbin results. This expectancy is based on the literature that SI procedures could distort the true distribution of the observed responses and lead to underestimation of the variance (Van Buuren, 2012).

The following discussion highlights the important findings from the kernel density estimates from all simulation scenarios. The visualisations are presented in Appendix P.2.

The prediction methods for Dirichlet distribution simulations result in higher bias than the GPAbin method.

The prediction methods and GPAbin approach for uniform simulations result in similar bias in a low percentage of missing values (10%). The AMB values for the prediction methods increase when the percentage of missing values increases to 30% with a MAR MDM for all simulation parameters with the exception of $n = 3000$ and $p = 5$. The uniform distribution simulations with an MCAR MDM and 30% or 50% missing values result in biased predictions for a small sample size ($n = 100$) with a larger number of variables ($p > 5$). Predictions made for uniform distribution simulations with 50% missing values with a MAR MDM are biased.

The normal distribution simulations show consistent bias measures for GPAbin within the top 10% range for both MDMs and 10% missing values. The prediction methods show a high density of unbiased values (lower than GPAbin), but also result in a biased portion of simulations beyond the 10% cut-off value. As the percentage of missing values increases to 30% with a MAR MDM, the density of unbiased values decreases for the prediction methods, showing both biased and unbiased values from the 1000 repetitions per simulation scenario. The GPAbin prediction method shows consistent results, still achieving a majority of AMB values within the top 10% range. The GPAbin method results in unbiased representation for 50% missing values for both MDMs. The bias of the prediction methods improve slightly for MCAR MDMs.

7.4 Measures of fit

The measures of fit (PS and CC) are presented in Figure 7.4.

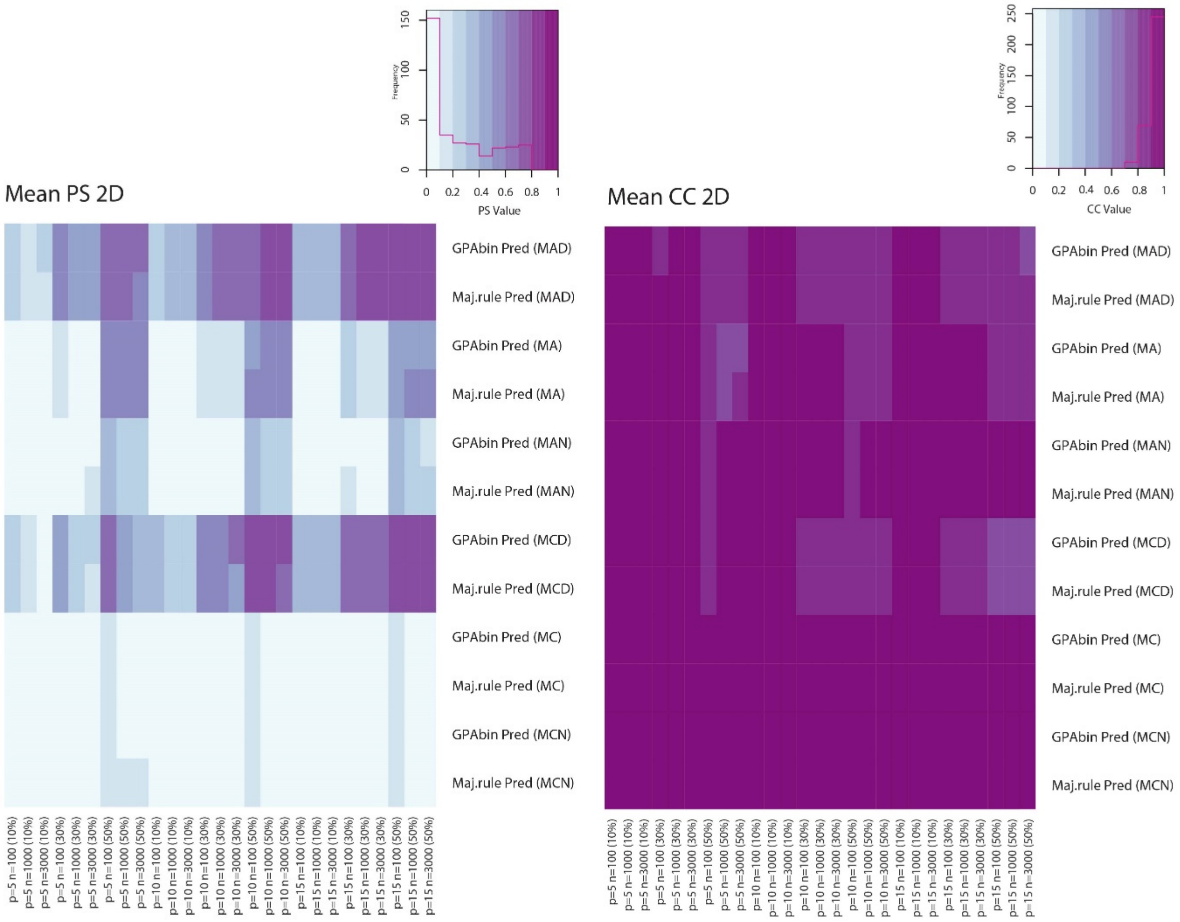


Figure 7.4 Measures of fit values per simulation scenario over 1000 repetitions (comparison in two dimensions). Left panel: Mean PS values. Right panel: Mean CC values.

The CC measures (cf. Figure 7.4: right panel) result in a majority of values in the top 10% range ($CC > 0.9$) with some lighter shades of simulation scenarios occurring for higher percentages of missing values. The PS measures (cf. Figure 7.4: left panel) show that the fit is poor when the percentage of missing values is high (50%), in particular for small sample sizes ($n = 100$) across all simulation scenarios and methods. The MCAR MDM simulations result in better fit than the MAR MDM simulations with small percentages of missing values (10%) resulting in good fit and therefore good prediction of a data set which preserves the configuration of the original simulated data set when visualised as an MCA biplot.

Again the GPAbin approach is depicted against the prediction methods in Figure 7.5 to Figure 7.7, now in terms of the PS values. The figures consist of 18 panels to evaluate the fit of the two prediction methods and GPAbin approach per distribution. The discussion will precede the visualisations.

The prediction methods for Dirichlet distribution simulations (cf. Figure 7.5: upper- and middle panels) show improved fit when compared to the GPAbin approach (cf. Figure 7.5: lower panels). The improved fit could be a result of using all available dimensions for the predicted data set, since better fit between the GPAbin approach and the simulated complete MCA biplots was observed using all possible dimensions (cf. 6.4).

Figure 7.6 shows similar fit for all methods for 10% missing values for both MDMs, but as the percentage of missing values increases to 30%, the GPAbin approach results in better fit than the prediction methods. All methods result in poor fit for 50% missing values and a MAR MDM. Improved PS values are observed for all methods for MCAR MDMs.

The prediction methods for the normal distribution simulations present in Figure 7.7 result in fit measures that are closely situated to the zero horizontal line for 10% missing values for both MDMs. Slightly more dispersion is observed for the GPAbin PS values. As the percentage of missing values increases (30% and 50%) for a MAR MDM the PS values are more dispersed for all methods, however more concentrated PS values occur for the prediction methods with an increase in sample size ($n = 1000$ and $n = 3000$).

In selected cases, the uniform (cf. Figure 7.6) and normal (cf. Figure 7.7) distribution simulations result in more dispersed PS values for the GPAbin approach when compared to the prediction methods. This could be a reflection of the realistic estimation of the variation when imputing missing values.

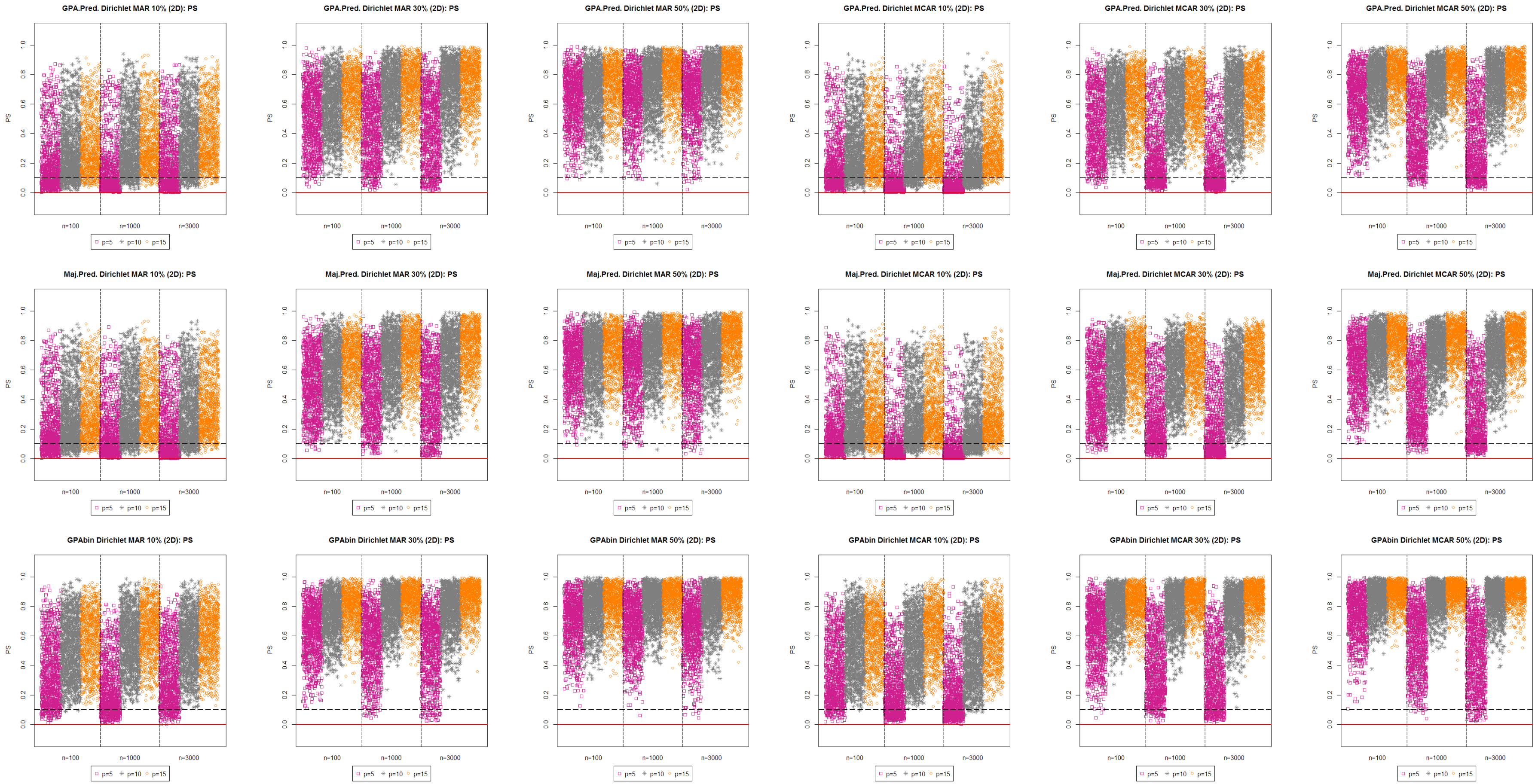
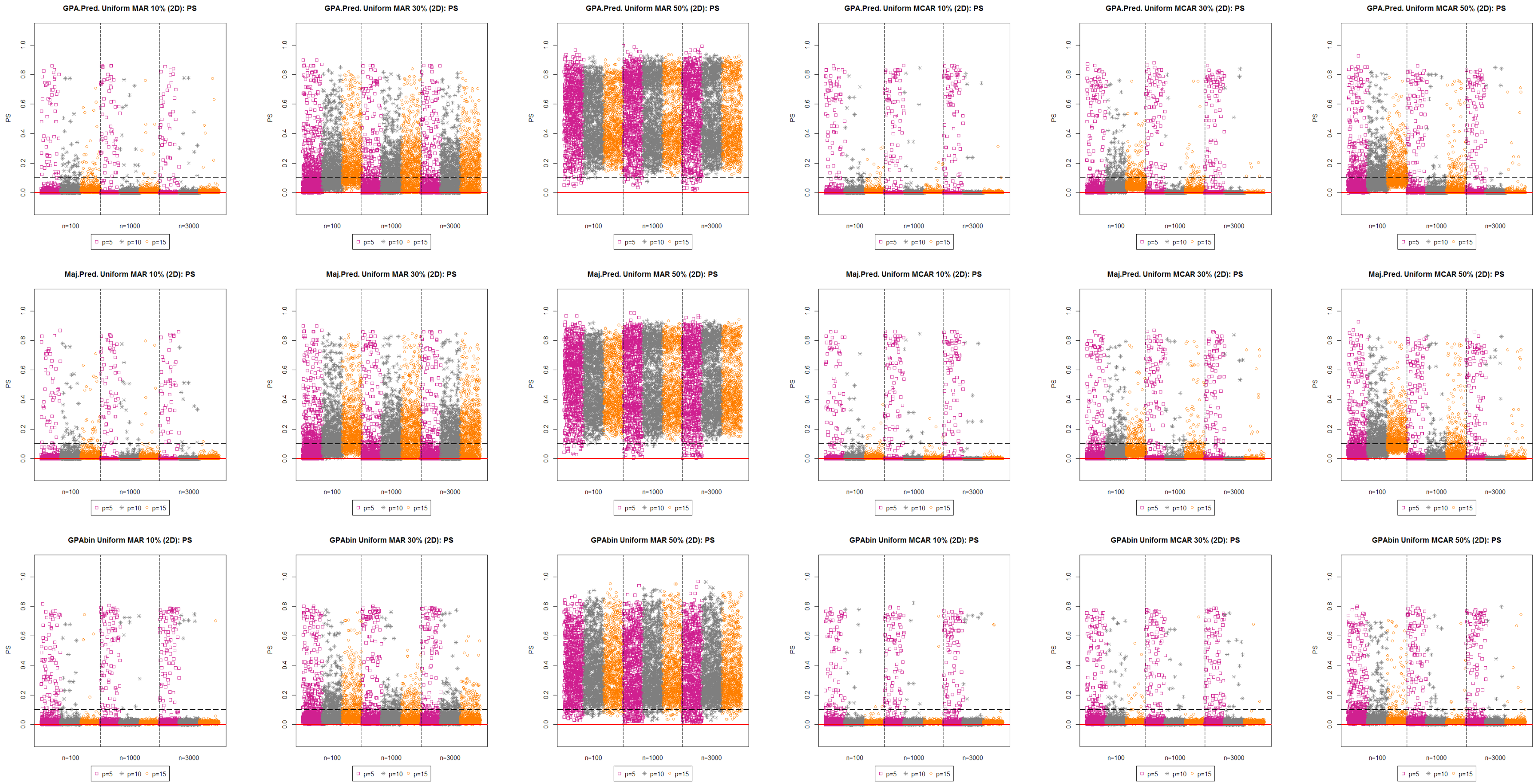


Figure 7.5 PS values: Dirichlet distribution (GPAbin prediction, Majority rule prediction and GPAbin)



GPA.Pred. Uniform MAR 10% (2D): PS

GPA.Pred. Uniform MAR 30% (2D): PS

GPA.Pred. Uniform MAR 50% (2D): PS

GPA.Pred. Uniform MCAR 10% (2D): PS

GPA.Pred. Uniform MCAR 30% (2D): PS

GPA.Pred. Uniform MCAR 50% (2D): PS

Maj.Pred. Uniform MAR 10% (2D): PS

Maj.Pred. Uniform MAR 30% (2D): PS

Maj.Pred. Uniform MAR 50% (2D): PS

Maj.Pred. Uniform MCAR 10% (2D): PS

Maj.Pred. Uniform MCAR 30% (2D): PS

Maj.Pred. Uniform MCAR 50% (2D): PS

GPAbin Uniform MAR 10% (2D): PS

GPAbin Uniform MAR 30% (2D): PS

GPAbin Uniform MAR 50% (2D): PS

GPAbin Uniform MCAR 10% (2D): PS

GPAbin Uniform MCAR 30% (2D): PS

GPAbin Uniform MCAR 50% (2D): PS

Figure 7.6 PS values: Uniform distribution (GPAbin prediction, Majority rule prediction and GPAbin)

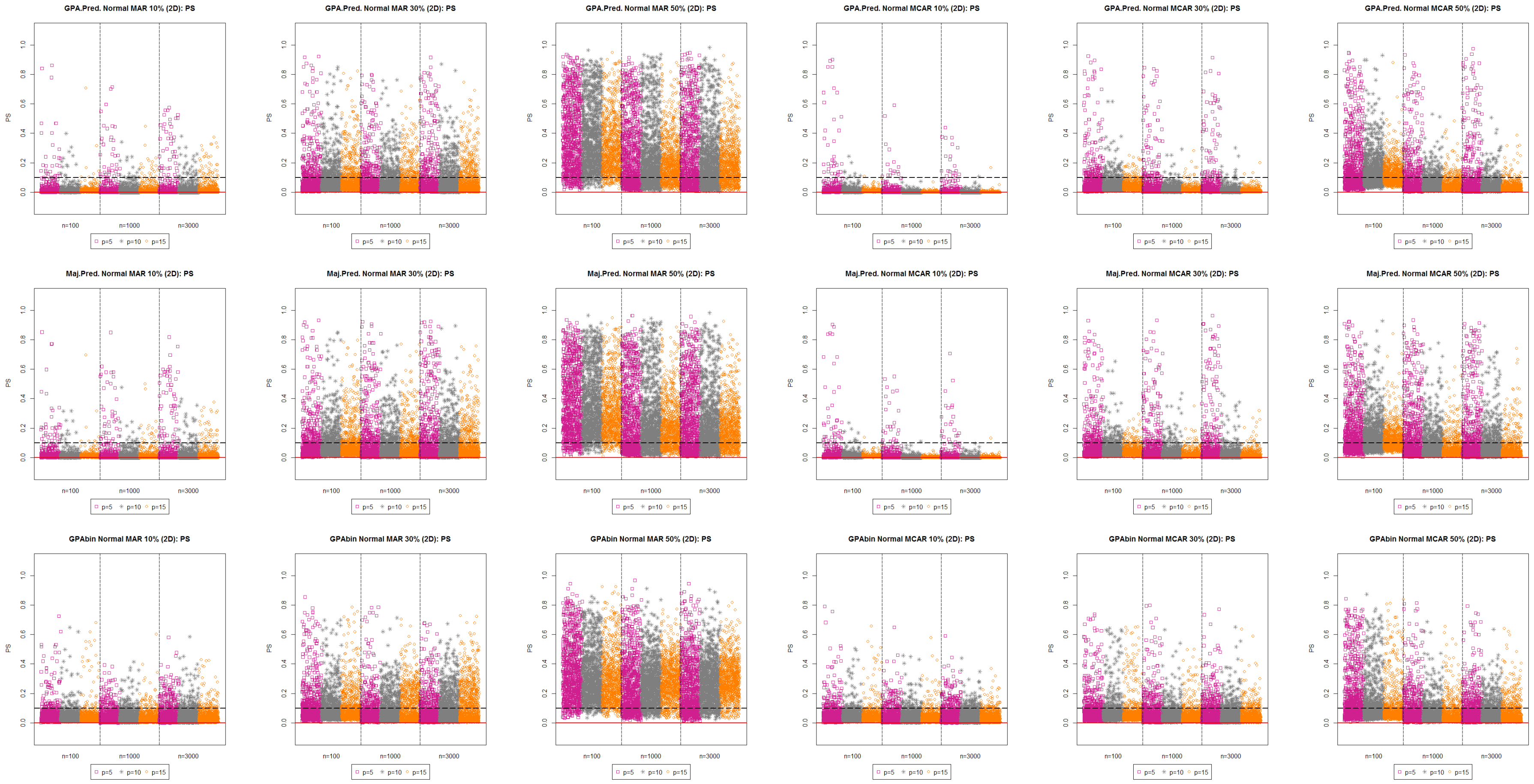


Figure 7.7 PS values: Normal distribution (GPAbin prediction, Majority rule prediction and GPAbin)

7.5 Conclusion

There is no notable difference in the performance of the two prediction methods. Accurate predictions of the original data set can be made from the MI based visualisations (GPAbin and multiple imputed MCA biplots) when the percentage of missing values is low (10%) for uniform and normal distribution simulations, irrespective of the MDM. The most ideal scenario to predict data sets from visualisations obtained from either MCA biplots of MIs or a GPAbin biplot would be for an MCAR MDM with a low percentage of missing values (10%) and a large number of variables ($p = 10$ or $p = 15$). It is not advisable to predict data sets from missing data with large portions (more than 10%) of missing values and a MAR MDM.

Chapter 8 will explore the visualisations of the missing subsets of simulated data using sMCA biplots in combination with clustering techniques to identify the MDM.

Chapter 8
Results: Identification of the missing data mechanism

8.1 Introduction

The aim of this chapter is to expose the difference in clustering structures between the MAR and MCAR MDMs within different simulated scenarios. This is achieved by clustering the CLPs of the missing subsets in the sMCA biplots for all simulated data scenarios (cf. Chapter 5). It is noted by Templ *et al*. (2012) that the assumption of the MDM is complex for multivariate data and even more so when the variables are heterogeneous and the data are skewed. The variety of simulated data sets (cf. Chapter 5) that are considered in this study addresses this remark and the results are expected to highlight the differences between the MDM structures for the different simulation scenarios. Determining the MDM with more certainty would aid in selecting appropriate handling techniques and analyses (Fielding, Fayers & Ramsay, 2009). Selecting the correct handling technique for missing data impacts the validity of the inference obtained from completed data. If the missingness is due to an MCAR MDM, a majority of missing data techniques will be suitable and provide valid inference (Jamshidian & Jalal, 2010; Little, 1988). This being said, the conditions for MCAR are strict and often violated, which again lead to the assumption of MAR being acceptable in most scenarios. If there is sufficient evidence to conclude that the assumption of MCAR is violated, care should be taken with the chosen handling technique and subsequent analyses. Simulation studies have shown that MI procedures perform well even when the MAR assumption is not certain (Schafer & Olsen, 1998). Albeit a simple approach to handling missing data, case deletion has been shown to be ineffective for multivariate data. This is due to the fact that a small number of item non-responses (cf. 3.1) across different samples could result in large proportions of data being deleted (Schafer & Graham, 2002).

Tests to determine the MDM are focused on the MCAR MDM, since it is a stronger assumption and not as sensitive to the misspecification of the distribution of the data (Little, 1988). Little (1988) developed a hypothesis test to determine whether the missingness is due to an MCAR MDM. This test is suitable for multivariate continuous data and utilises a global test statistic that is asymptotically χ^2distributed. This addresses the problem of an increasing Type I error when performing multiple comparisons. The means of the observed and missing observations per variable are compared to determine whether there is a significant difference between the visible spread of the subsets of observations. A significant difference suggests that there is sufficient evidence against the assumption of MCAR. Another hypothesis test is proposed by Jamshidian and Jalal (2010) to confirm the MCAR MDM. The data set is divided into groups defined by missing data patterns and then the homogeneity of the means and covariances across the allocated groups are determined. Both these techniques do not focus on exploratory analysis and are also not suitable for multivariate categorical data sets.

Exploratory analysis is regarded as a valuable tool to explore the missingness in the data (Templ *et al.*, 2012). Hidden relationships and structures become visible when suitable visualisations are applied (Beh & Lombardo, 2014).

As noted by Fernstad (2019) the `R` package, `VIM` (Kowarik & Templ, 2016) is the most recent development in the visualisation of missing data. Briefly, the `VIM` package includes interactive multivariate data visualisations, such as scatterplots, boxplots, histograms, matrix plots, mosaic plots, parallel coordinate plots and also introduce the visualisation of missing values in geographical maps (Templ *et al.*, 2012). These visualisations aim at recognising the distribution of missing values in order to identify the MDM, typically by depicting the proportion of missing values per variable. Another option is to visualise the missing values in a missingness map which also provides insight of the missing values that occur in specific variables (Honaker *et al.*, 2011). Van Buuren (2012) uses boxplots and density plots to investigate the difference in the distributions of observed and missing partitions of continuous data sets. This resonates with the hypothesis test proposed by Little (1988).

The above mentioned techniques again rely on obtaining sufficient evidence to discard the assumption of MCAR, as suggested by Little (1988). If the spreading of missing and observed partitions are homogenous it is acceptable to assume that the missingness is due to observations being MCAR.

It is thus plausible to expect that an sMCA biplot of the missing subsets can reveal structures leading to a decision on the underlying MDM. The positions and proximities of the missing CLPs in the sMCA biplot may be indicative of the associations between missing values and therefore lead to the identification of the MDM. Since the association between the CLPs could aid in understanding the MDM, the subset of incomplete CLPs is subjected to a clustering technique. It is hypothesized that randomly scattered, or evenly spaced, CLPs in an sMCA biplot will not result in sufficient clustering structures. This is in a sense comparable to variables with similar spreadings and homogenous distances between coordinates as proposed by the authors mentioned in the preceding discussion. Therefore, the lack of sufficient clusters are associated with independence and an MCAR MDM, whereas sufficient clustering structures between CLPs are expected to occur when the missingness is due to a MAR MDM in which the close proximities of the CLPs indicate associations.

8.2 Presentation of results

The sMCA biplots of the missing subsets are first evaluated and presented in Figure 8.1 to Figure 8.9. Each figure consists of twelve plots with the rows distinguishing between the percentage of missing values and the columns distinguishing between the simulated distribution and MDM. In all sMCA biplots the green open circles represent the sample points and the black triangle plotting characters represent the CLPs.

As mentioned in Section 5.4.2, a fixed random seed is used to ensure the reproducibility of results. Creating an artificial MCAR MDM requires the missing values to be independent of any observed or missing information and therefore represents an unbiased sample from what would have been the complete data set. A 1000 randomly generated seed values are used throughout the simulations, which means that each first simulation (irrespective of data scenario) uses the first seed value, the second simulation uses the second seed value until the final simulation is completed. Utilisation of the same seed values results in similar positions of the missing observations for the same repetition of the simulations, irrespective of the simulated distribution and data scenario.

The recoded indicator matrix using single active handling (cf. 3.4.2.2) for the three simulated distributions will be similar, since similar positions of missing values occur due to the fixed random seed. The missing subsets of the recoded indicator matrix will consequently result in similar sMCA solutions and clustering structures. The MCAR visualisations appear to be identical, but minor differences do occur due to rounding differences of coordinate points. It was found that in some instances differences occurred beyond the tenth decimal place ($\times 10^{-10}$), which resulted in a slightly different clustering allocation and consequently slightly different silhouette coefficients. Rounding was not applied on the coordinate matrices obtained from sMCA before applying the clustering algorithm, which resulted in at most 0.2% different silhouette coefficients when compared across the simulated distributions. Albeit, the occurrence of slightly different silhouette coefficients in 0.2% of the simulations, the conclusions remain the same. It is advisable to use rounding to ensure that coordinates are identical before applying sensitive techniques.

Due to the similarity of the visualisations obtained from the MCAR MDMs, irrespective of the simulated distribution, only one MCAR distribution per missing percentage is visualised along with the sMCA biplots of the missing subsets obtained from the MAR simulations (normal, uniform and Dirichlet) to ease the visual interpretation. Only one repetition is illustrated in Section 8.3 with additional examples available in Appendix Q. The visible patterns in the sMCA biplots aid as a precursor of the possible clustering structure of the CLPs.

After the visual inspection of the sMCA biplots, the overall average silhouette widths ($s(i)$) obtained from the pam clustering are summarised using three types of visualisations in this chapter: scatterplots (cf. 8.4.1 to 8.4.3), stacked bar plots (cf. 8.4.1 to 8.4.3) and heat maps (cf. 8.4.4).

The figures in Section 8.4.1 to 8.4.3 (cf. Figure 8.10 to Figure 8.18) consist of four panels. The left panels of the figures comprise of nine scatterplots, each scatterplot represents a specific sample size and number of variables simulated from one of the three distributions (Dirichlet, normal or uniform) and consisting of missing values with a particular missing percentage and MDM. The scatterplots are divided into vertical intervals to distinguish between the silhouette coefficients obtained from the number of clusters that are specified for the 1000 repetitions for each simulated scenario. Each scatterplot has the same number of vertical intervals to enhance the comparison between different panels. The number of intervals is set to fourteen, since the maximum number of missing CLPs available for the MCAR simulations is fifteen (cf. 4.7). In the MAR simulations, one variable is always fully observed based on the conditions specified in Section 5.4. Therefore, the number of populated intervals for the MAR simulations is one less than the number of intervals for the MCAR simulations. A solid horizontal line is visualised where $s(i) = 0.5$ which enables a quick appreciation of the trend of silhouette widths below and above the 0.5 benchmark value (cf. 4.7). The focus is on identifying the difference in patterns / trends between the MAR and MCAR MDMs.

The stacked bar plots visualise the frequency distribution of the overall average silhouette widths ($s(i)$) using the following intervals: $[-0.25; 0)$, $[0; 0.25)$, $[0.25; 0.5)$, $[0.5; 0.75)$ and $[0.75; 1)$. Different gradients are used within the bars to visualise the frequency of silhouette widths within each specified interval, as defined in the legend of each stacked bar plot. The modal interval containing the majority of occurrences is visualised using a solid colour to enable fast identification of the interval with the highest frequency. The stacked bar plots are presented in the right panels of Figure 8.10 to Figure 8.18 and also consist of nine figures per panel.

The scatterplots and stacked bar plots of a MAR and MCAR MDM with a specific percentage of missing values and simulated distribution are presented and labelled as a single figure with four panels. For the scatterplots, the overall average silhouette widths are given in abbreviated form as 'silhouette widths' for the y-axis label.

The heat maps provide a combined summary of all simulation scenarios in a single figure. Each simulation scenario and number of specified clusters are considered separately, by determining the number of silhouette coefficients above a fixed value (e.g. 0.5) and expressing the frequency as a percentage. The percentage for each simulation scenario and number of clusters are presented as a cell in the heat map which relates to a colour associated with percentages between 0% and 100%, as defined in the key of the map. The abbreviations of the simulation distributions and MDMs (cf. Table 6.1) are again used to annotate the rows of the heat maps.

8.3 Subset multiple correspondence analysis biplots: missing subsets

The evaluation of both the sample points and the CLPs are of importance and could lead to an initial identification of the possible MDM. In general, discerning patterns or groupings are of interest and establishing whether there is a notable difference between the patterns of MAR and MCAR MDMs. The visual interpretation precedes the application of a clustering technique. It is expected that the silhouette coefficients will support the visual separations (cf. 8.4).

Structured sample points are observed in some of the visualisations, which could be a reflection of recurring response patterns due to the recoding of the indicator matrix using single active handling (cf. 3.4.2.2). It is also plausible that the structured sample points indicate the categorical scale of the data, since only qualitative responses are possible.

Each figure (cf. Figure 8.1 to Figure 8.9) represents a combination of sample size and number of variables for a particular example from the 1000 repetitions.

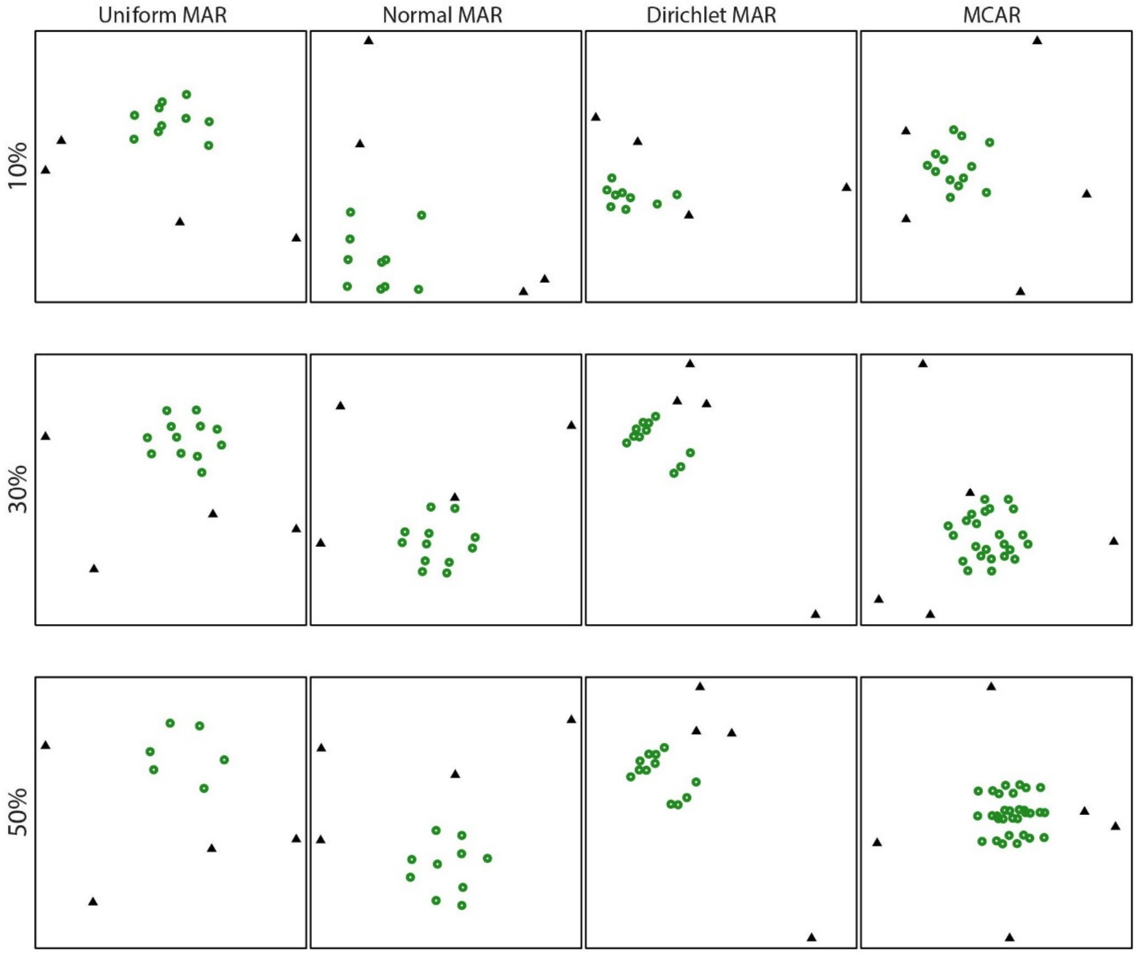


Figure 8.1 *sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 5$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).*

The sMCA biplots of simulations with a small number of variables ($p = 5$), do not show clear differences between the MAR and MCAR MDMs as can be seen in Figure 8.1. The sample points in the sMCA biplots of the MAR MDMs are slightly dispersed for 10% missing values, but with an increase in the percentage of missing values, the sample points are in closer proximity with a majority of overlapping points. Since the MAR MDMs are created by using a set of deletion conditions (cf. 5.4), similar missing data sets are created for 30% and 50% missingness due to the limited deletion possibilities for data sets with small dimensions. This is reflected in the similar sMCA biplots of 30% and 50% missingness for each simulated distribution in Figure 8.1.

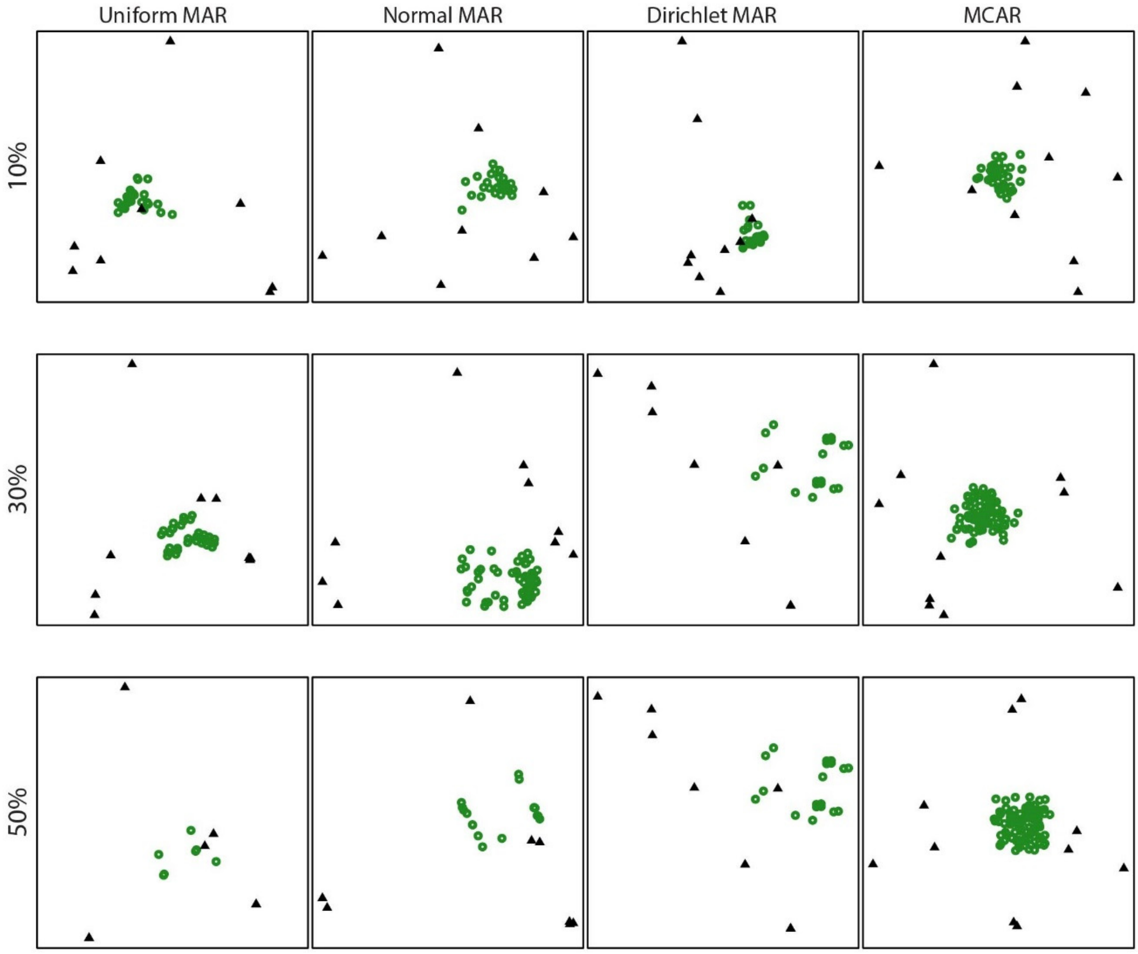


Figure 8.2 sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

In Figure 8.2 a larger number of variables ($p = 10$) are considered, which show visible differences between MAR and MCAR visualisations. Again, as the percentage of missing values increases, the sample points for the MAR MDM overlap. Additional to the sample point behaviour, it is noticeable that the CLPs also show separation with overlapping points in the MAR configurations. This is not the case for the MCAR visualisations, where the samples form a circular cloud of points with approximately equally scattered CLPs which suggest that there is no particular separation.

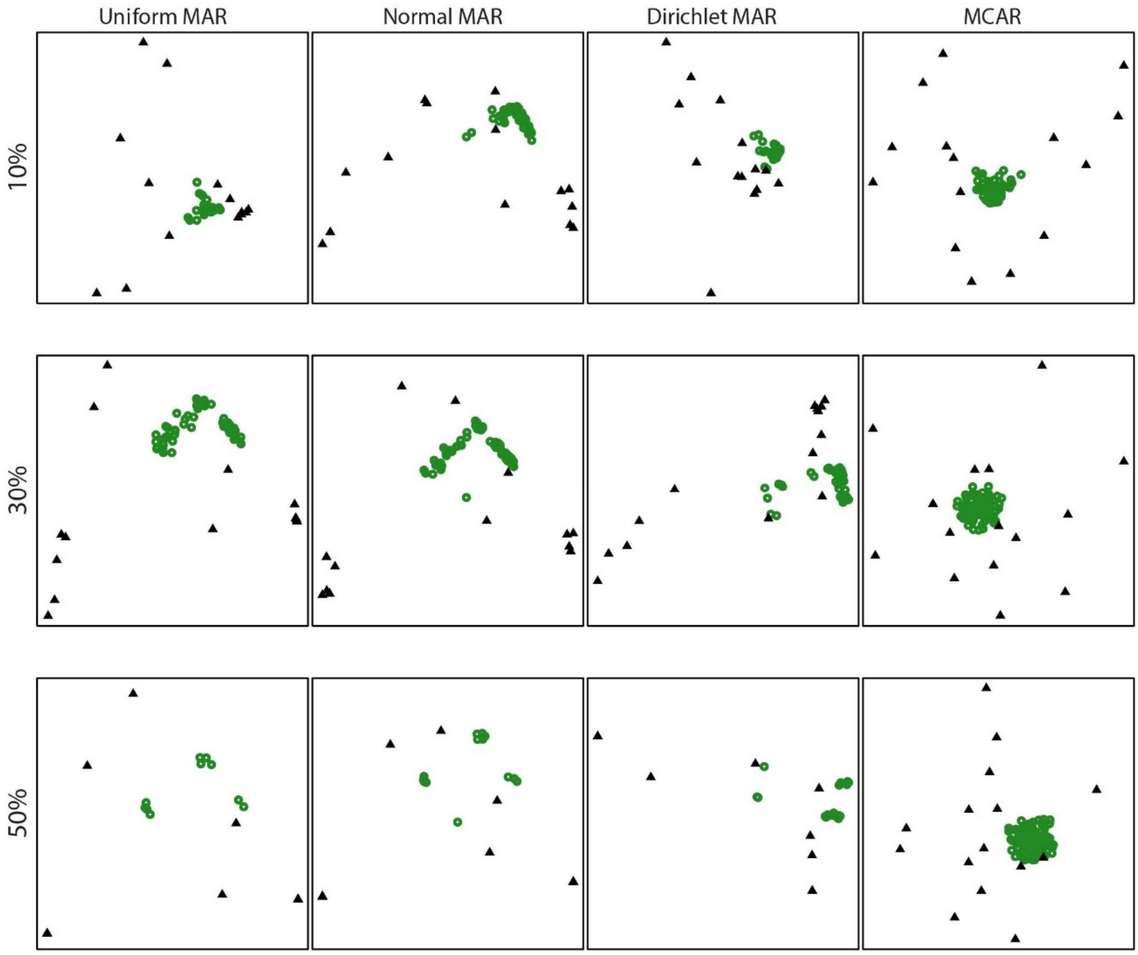


Figure 8.3 sMCA biplots for the missing subsets of a selected repetition ($n = 100, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

The conclusions drawn from Figure 8.1 and Figure 8.2 are also evident with a further increase in the number of variables in Figure 8.3. Overlapping sample points and CLPs with distinct groupings occur for the MAR MDMs as the percentage of missing values increases to 50%. There are no particular groupings in the MCAR sMCA biplots and a slightly more dispersed cluster of samples is observed as the percentage of missing values increases.

Figure 8.4 to Figure 8.6 will be presented before the combined discussion on the sMCA biplots obtained from a 1000 samples, considering all number of variables.

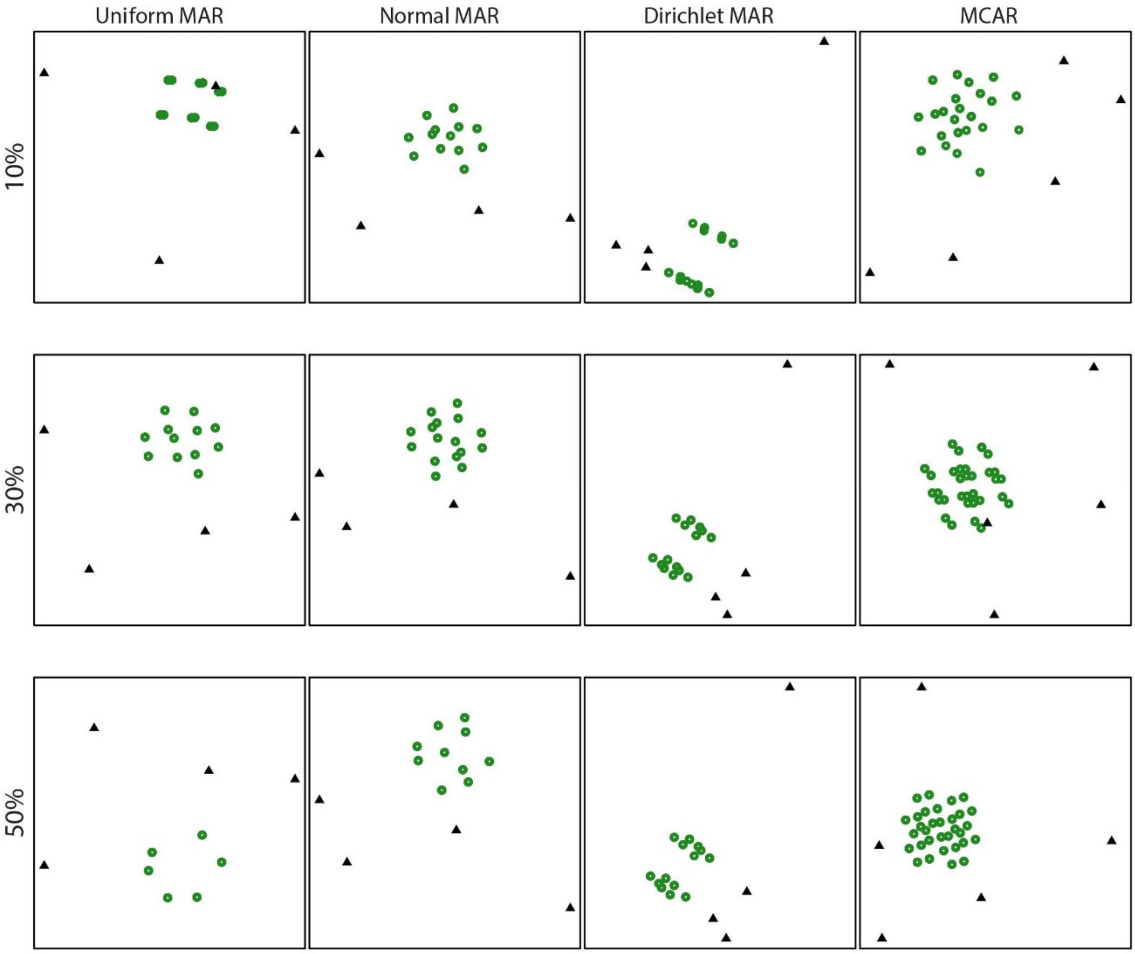


Figure 8.4 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 5$)*. Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).*

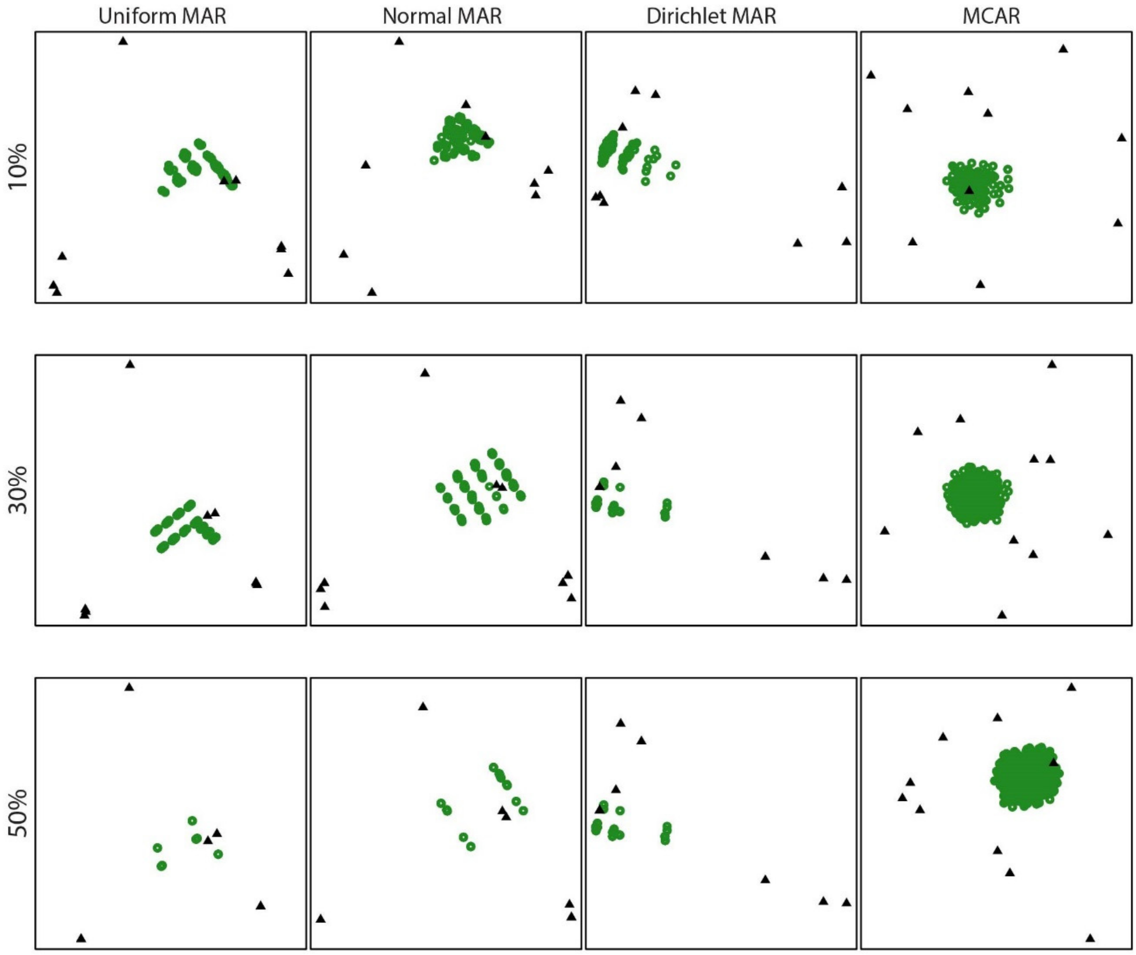


Figure 8.5 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

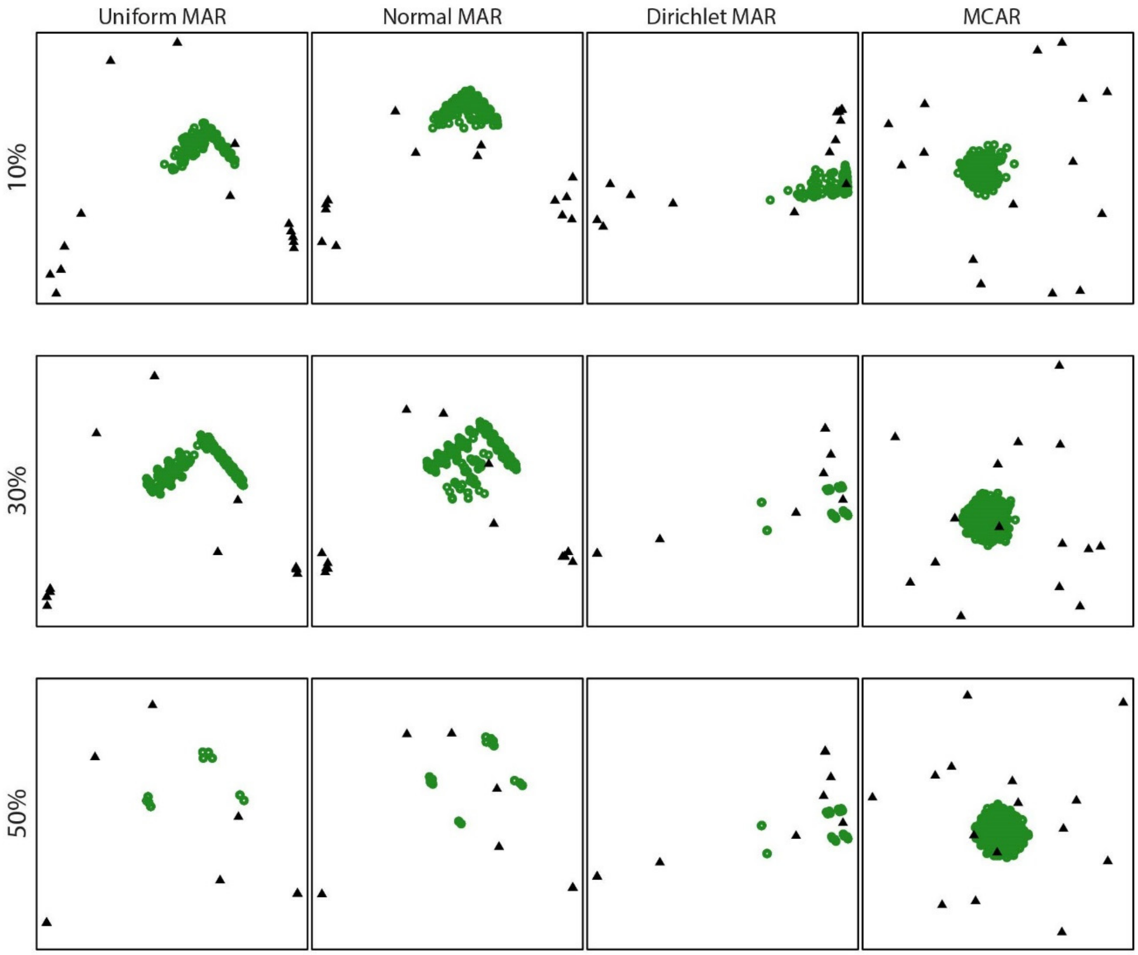


Figure 8.6 sMCA biplots for the missing subsets of a selected repetition ($n = 1000, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

Even with an increase in the sample size ($n = 1000$), a small number of vairables do not show clear differences between the MDMs (cf. Figure 8.4). It is again evident that the MAR samples and CLPs tend to overlap and separate into groupings as the percentage of missing values increases. The MCAR CLPs are randomly situated without notable groupings occurring. As was observed from the smaller sample sizes in Figure 8.1 to Figure 8.3, the sample points of the MCAR sMCA biplots occur in one group with no separation.

Finally, Figure 8.7 to Figure 8.9 will be presented before the combined discussion on the sMCA biplots obtained from 3000 samples, considering all number of variables.

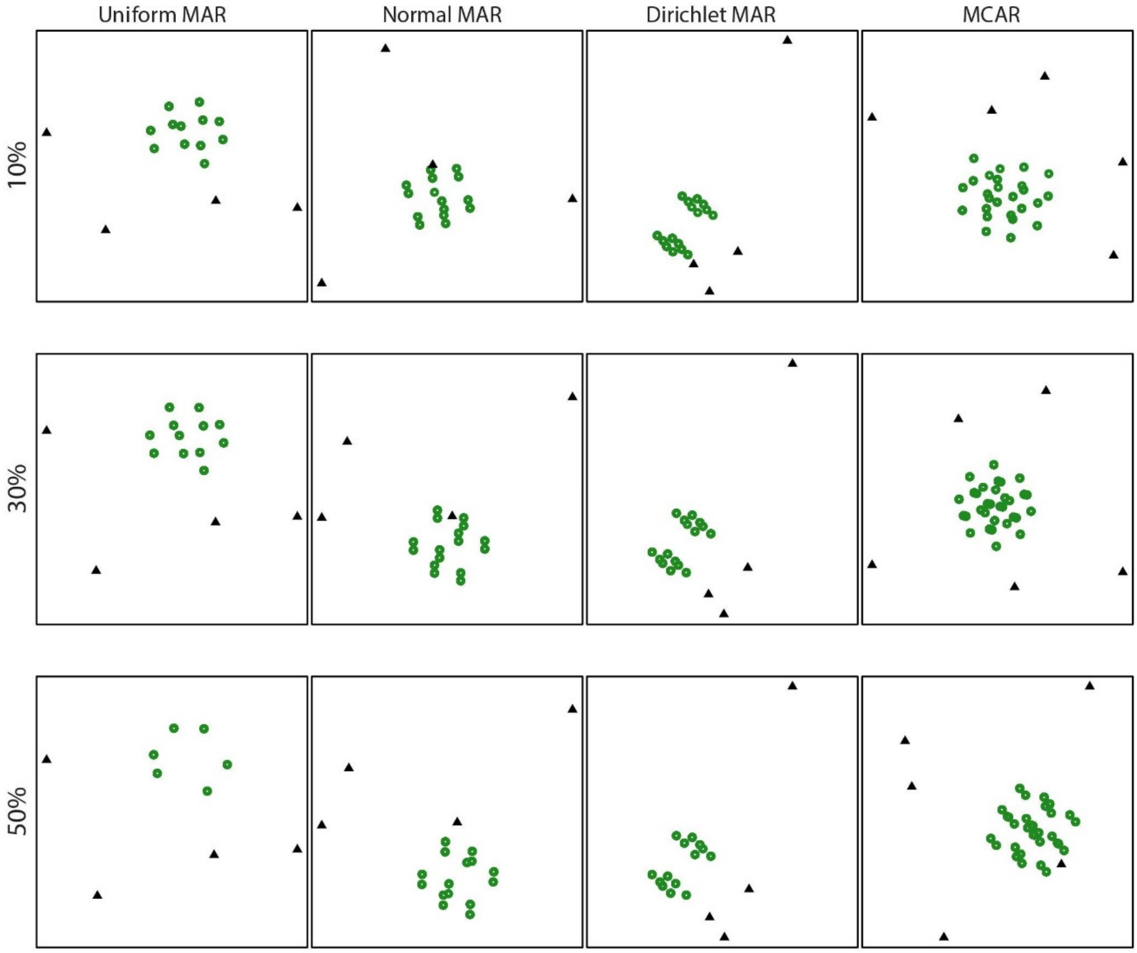


Figure 8.7 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 5$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

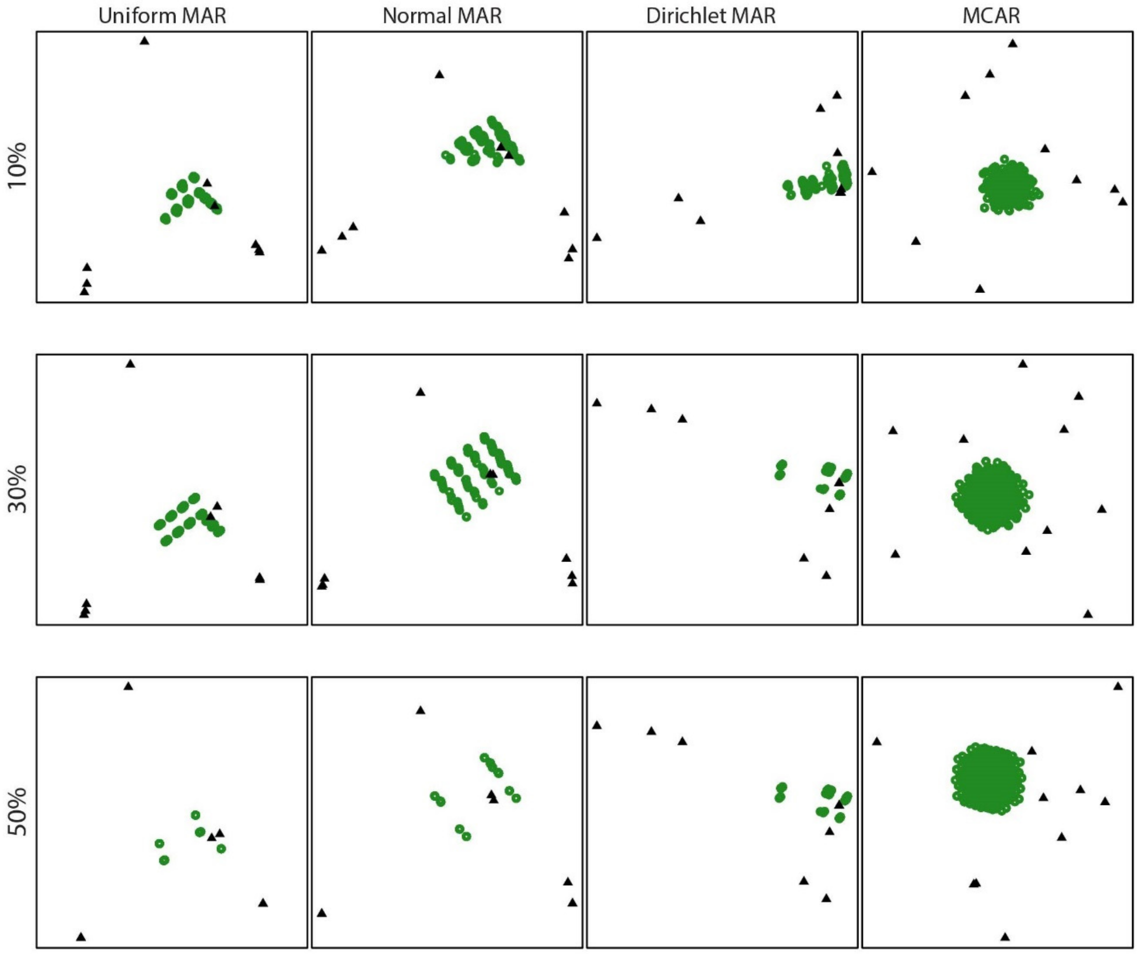


Figure 8.8 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 10$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

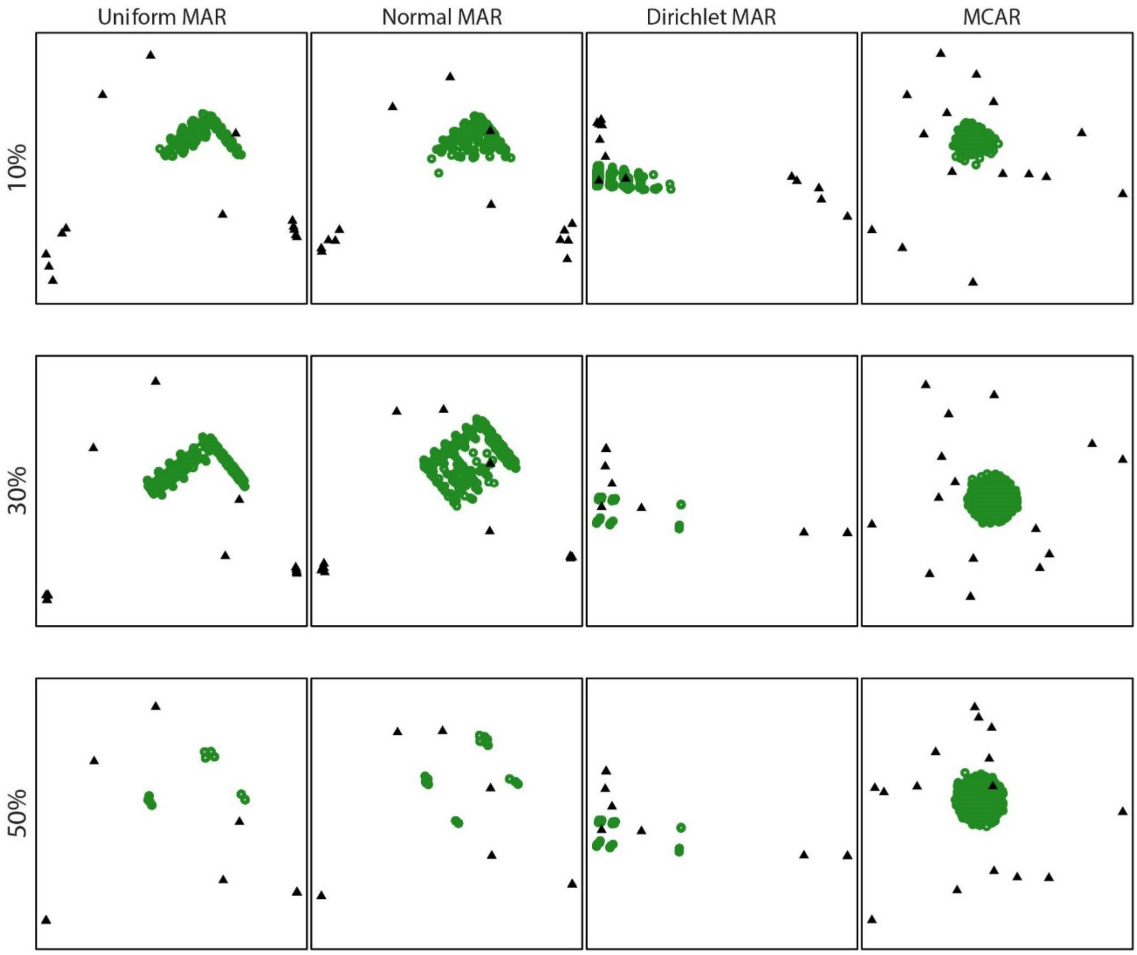


Figure 8.9 sMCA biplots for the missing subsets of a selected repetition ($n = 3000, p = 15$). Sample points are depicted by green open circles and CLPs by black triangles. Take note that the number of CLPs of the MAR simulations is one less than the number of MCAR CLPs (cf. 5.4).

The biplots representing the larger sample size of $n = 3000$ in Figure 8.7 to Figure 8.9, reiterate the conclusions of the preceding figures.

To summarise, it is evident that overlapping sample points and CLPs tend to occur in the MAR sMCA biplots with visible separation as the percentage of missing values increases. As the number of variables increases, separation is visible between the CLPs in the MAR sMCA biplots, but a random spread of CLPs is noted in the MCAR sMCA biplots. The random cloud of points resembles homogeneous spread which upholds the definition of the MCAR MDM. The visible separation between CLPs in the MAR sMCA biplots is expected to result in higher silhouette coefficients than MCAR sMCA biplots.

8.4 Visualisations of silhouette coefficients

As previously mentioned (cf. 8.1), the silhouette coefficients are summarised using three visualisations. The scatterplots expose the trend of silhouette coefficients obtained from a specific number of clusters for each simulated scenario. The stacked bar plots provide the frequency distributions of the silhouette coefficients, which quantifies the trends observed from the scatterplots. The three plots complement each other: both the scatterplots and stacked bar plots present the silhouette coefficients for specific simulation scenarios, whereas the heat maps enable an encapsulating view of the silhouette coefficients by incorporating all simulation parameters in a single figure.

The visualisations in Sections 8.4.1, 8.4.2 and 8.4.3, consider each simulated distribution and particular percentage of missing values separately. In order to achieve an overall impression of the silhouette coefficients obtained from the various simulation scenarios, heat maps are presented in 8.4.4.

8.4.1 Scatterplots and stacked bar plots (10% missingness)

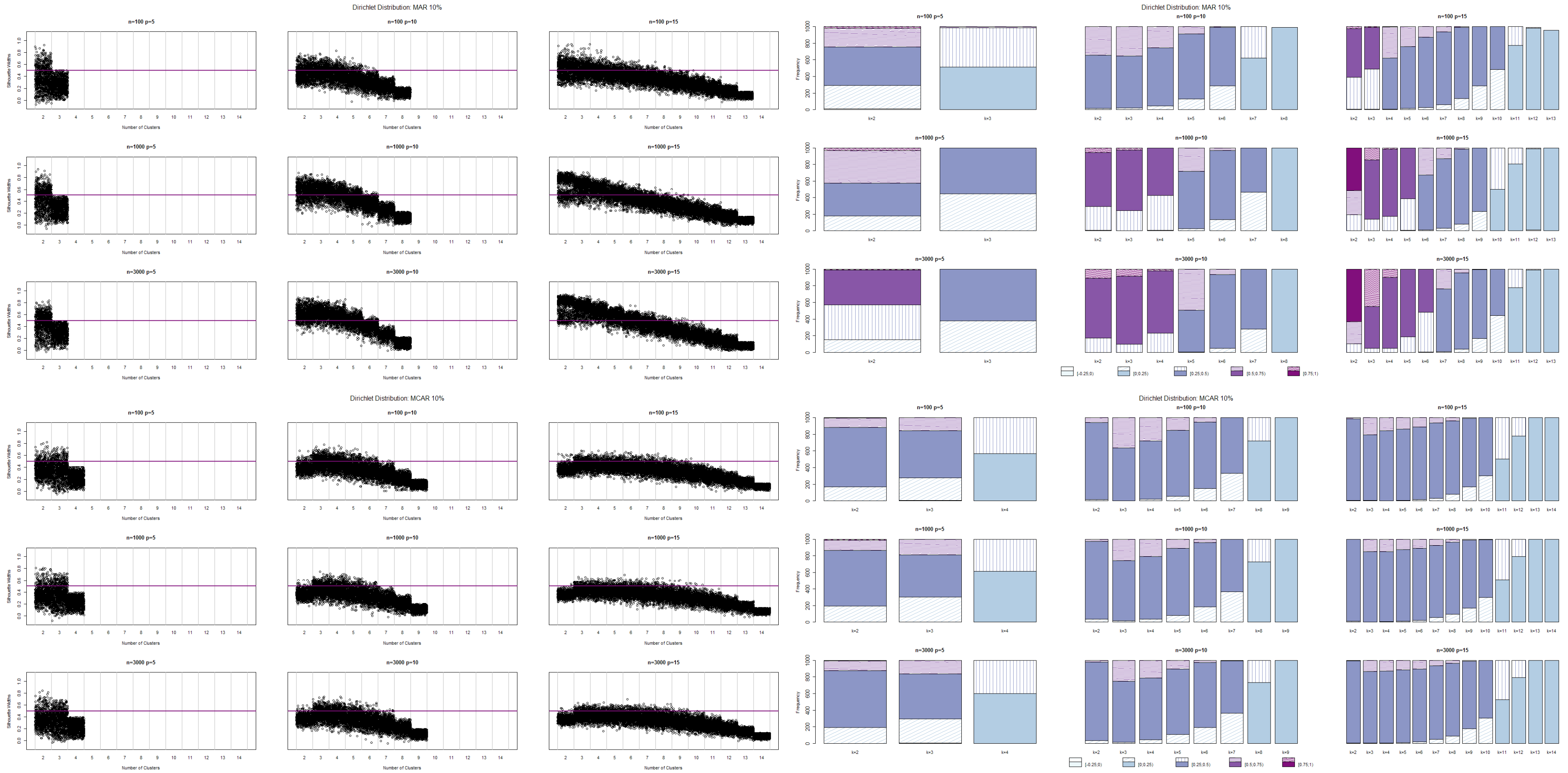


Figure 8.10 Silhouette coefficients of the Dirichlet distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

The nine figures in the upper left panel of Figure 8.10 representing the simulation scenarios of the MAR MDM simulated from the Dirichlet distribution, are considered first. Data sets with lower dimension ($p = 5$) tend not to have strong clustering structures, due to the small number of possible missing CLPs that are available for clustering. As the sample sizes increase, the silhouette coefficients per cluster (interval) seem to be more concentrated with less dispersion for $n = 3000$. It is notable that when specifying three clusters, the clustering structure is not as well separated as the specification of two clusters, since the majority of silhouette coefficients appear below the 0.5 horizontal line. Considering the scatterplots with $p = 10$, it is observed that as the sample sizes increase, large silhouette coefficients occur with the bulk of the points occurring above the 0.5 horizontal line. The same trend is observed from the simulations with $p = 15$, since larger sample sizes result in higher silhouette coefficients. It is observed in all scenarios that the silhouette coefficients decrease when the specified number of clusters increases, due to the CLPs that are not substantially separated.

Considering the lower left panel of Figure 8.10, the MCAR MDM does not result in substantially different silhouette coefficients when compared across the sample sizes for a particular number of variables as was observed from the MAR scatterplots in the upper left panel of Figure 8.10. It is clear that a majority of silhouette coefficients obtained from the MCAR MDM occur below the 0.5 benchmark value, also with a more gradual decreasing trend in silhouette coefficients as the number of clusters increases than was observed for the MAR MDM.

Evaluating the stacked bar plots in the right panel of Figure 8.10 the first noticeable difference between the upper right panel (MAR MDM) and lower right panel (MCAR MDM) is the colours of the solid sections of the stacked bars. The modal classes of the silhouette coefficients of the lower panel (MCAR MDM) occur in the $[0.25; 0.5)$ and $[0; 0.25)$ intervals. Whereas, a variation of modal classes appears in the upper panel (MAR MDM) with more solid sections indicated with darker shades of purple associated with the silhouette coefficients above 0.5. Due to the deletion conditions for the MAR simulations (cf. 5.4), not all MAR simulations have the same number of missing CLPs, which in some cases lead to repetitions of simulation scenarios that cannot be separated using the same number of clusters (k). The following simulation scenarios could not be clustered: 13 simulations ($n = 100, p = 10, k = 8$), two simulations ($n = 100, p = 15, k = 11$), ten simulations ($n = 100, p = 15, k = 12$) and 45 simulations ($n = 100, p = 15, k = 13$).

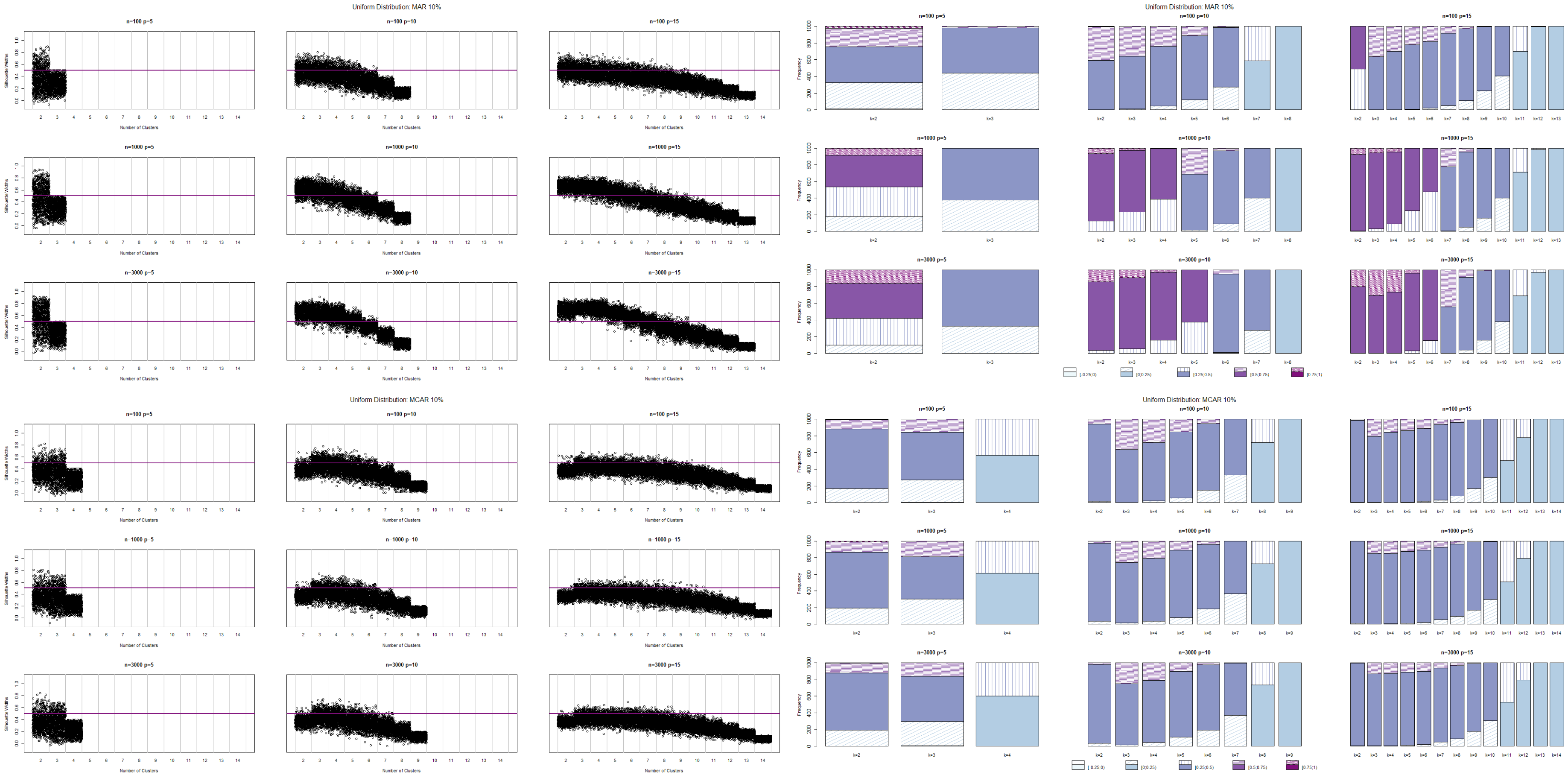


Figure 8.11 Silhouette coefficients of the uniform distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

Considering the left panels of Figure 8.11, the simulations from the uniform distribution again confirm that an increase in the sample size improves the clustering structure of the missing CLPs. The trend of the silhouette coefficients obtained from $n = 3000$ and $p = 15$ in the upper left panel shows a noted increase of silhouette coefficients for $k = 3$ which starts to decrease for $k = 5$. The lower left panel shows a similar gradual decrease in silhouette coefficients as was observed from the Dirichlet distribution (cf. Figure 8.10: lower left panel), which again concludes that the trends in the MDMs differ with silhouette coefficients tending to be above 0.5 for a MAR MDM.

Considering the upper right panel of Figure 8.11, lower modal class intervals occur for a small sample size ($n = 100$), but in general higher modal class intervals are observed for the MAR scenarios than for the MCAR scenarios. The following simulation scenarios could not be clustered: one simulation $(n = 100, p = 10, k = 8)$ and one simulation $(n = 100, p = 15, k = 13)$.

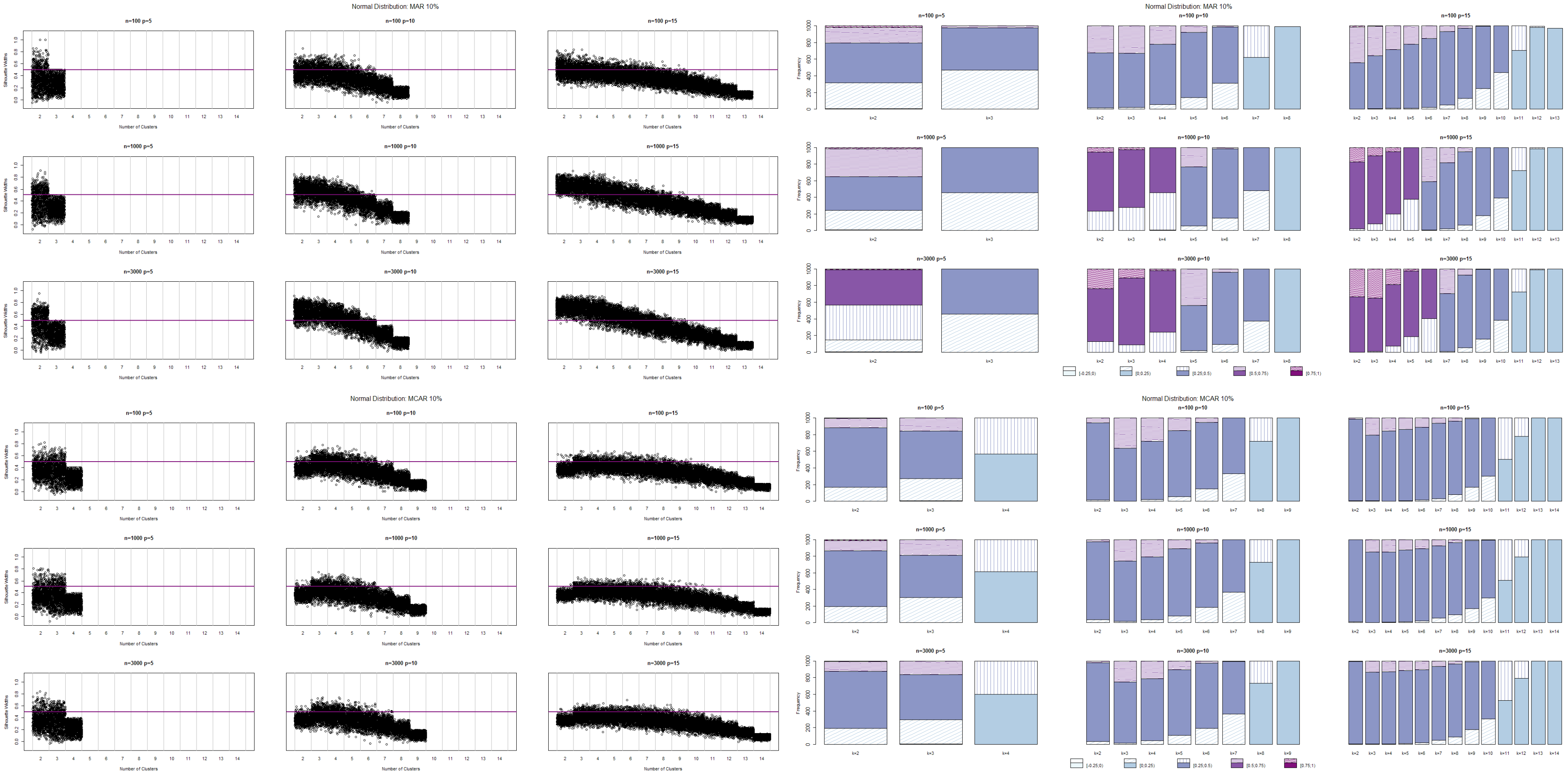


Figure 8.12 Silhouette coefficients of the normal distribution (10% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

The simulations from the normal distribution in the left panel of Figure 8.12 result in the same conclusions as the uniform and Dirichlet scatterplots, since an increase in sample size again leads to an increase in the overall silhouette coefficients. Also, for the normal distribution the MAR simulations show higher silhouette coefficients than the MCAR simulations. It is interesting to take note that two simulations resulted in silhouette coefficients equal to one for $n = 100$ and $p = 5$, which is visible in the first plot of the upper left panel of Figure 8.12. This is however not aligned with the trend of the clustering structures obtained from data sets with small dimensions.

The stacked bar plots in the right panel of Figure 8.12 confirm the findings of the previous simulated distributions, in which a lower sample size and number of variables result in lower silhouette coefficients and the modal classes of the MAR simulations tend to be higher than those of the MCAR simulations. The following simulation scenarios could not be clustered: one simulation $(n = 100, p = 5, k = 3)$, 13 simulations $(n = 100, p = 10, k = 8)$, two simulations $(n = 100, p = 15, k = 11)$, six simulations $(n = 100, p = 15, k = 12)$ and 29 simulations $(n = 100, p = 15, k = 13)$.

8.4.2 Scatterplots and stacked bar plots (30% missingness)

In order to avoid duplication, the discussion of the silhouette coefficients for the simulation scenarios with 30% missingness will be combined for the three simulated distributions after presenting Figure 8.13 to Figure 8.15.

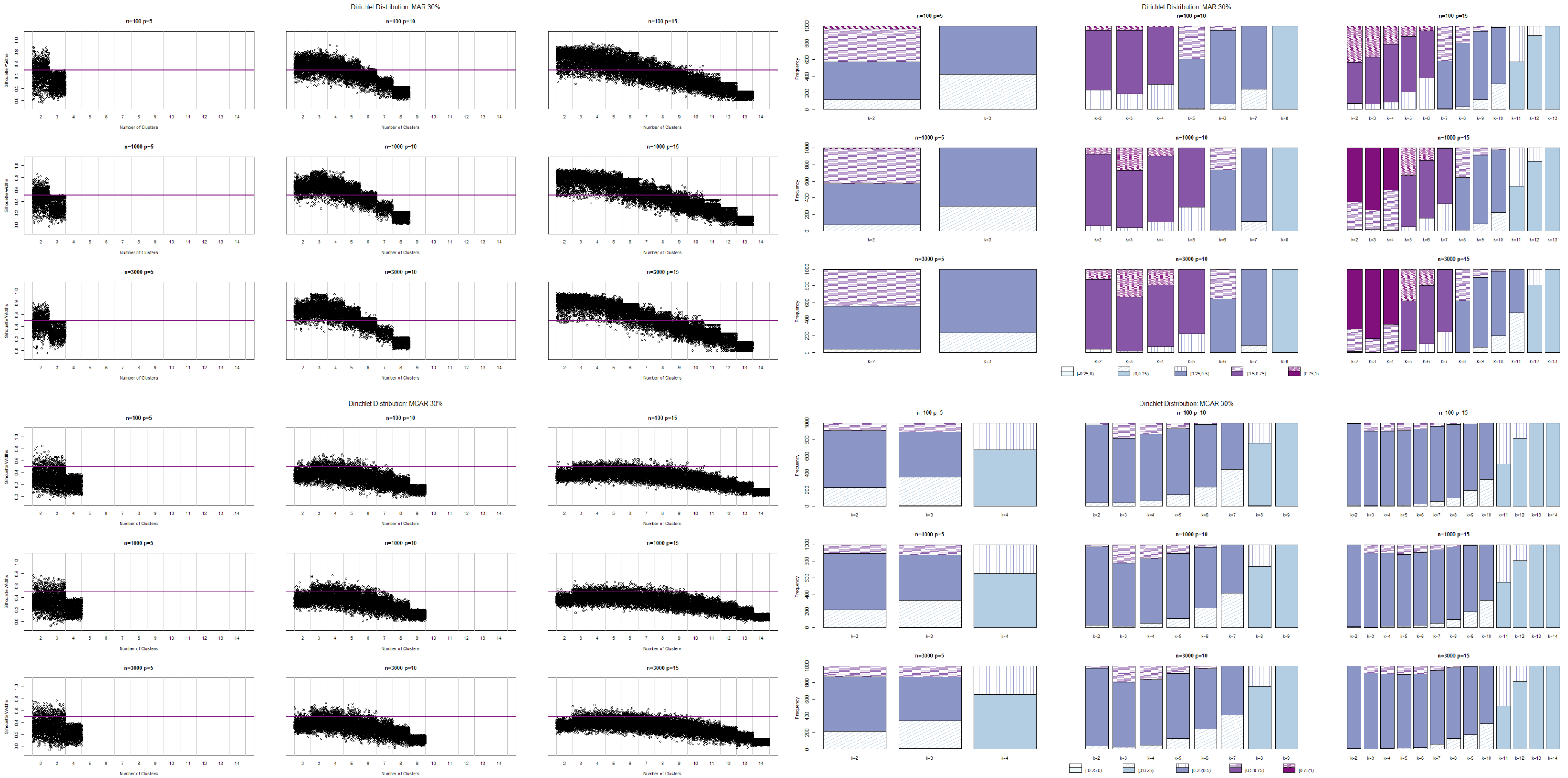


Figure 8.13 Silhouette coefficients of the Dirichlet distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

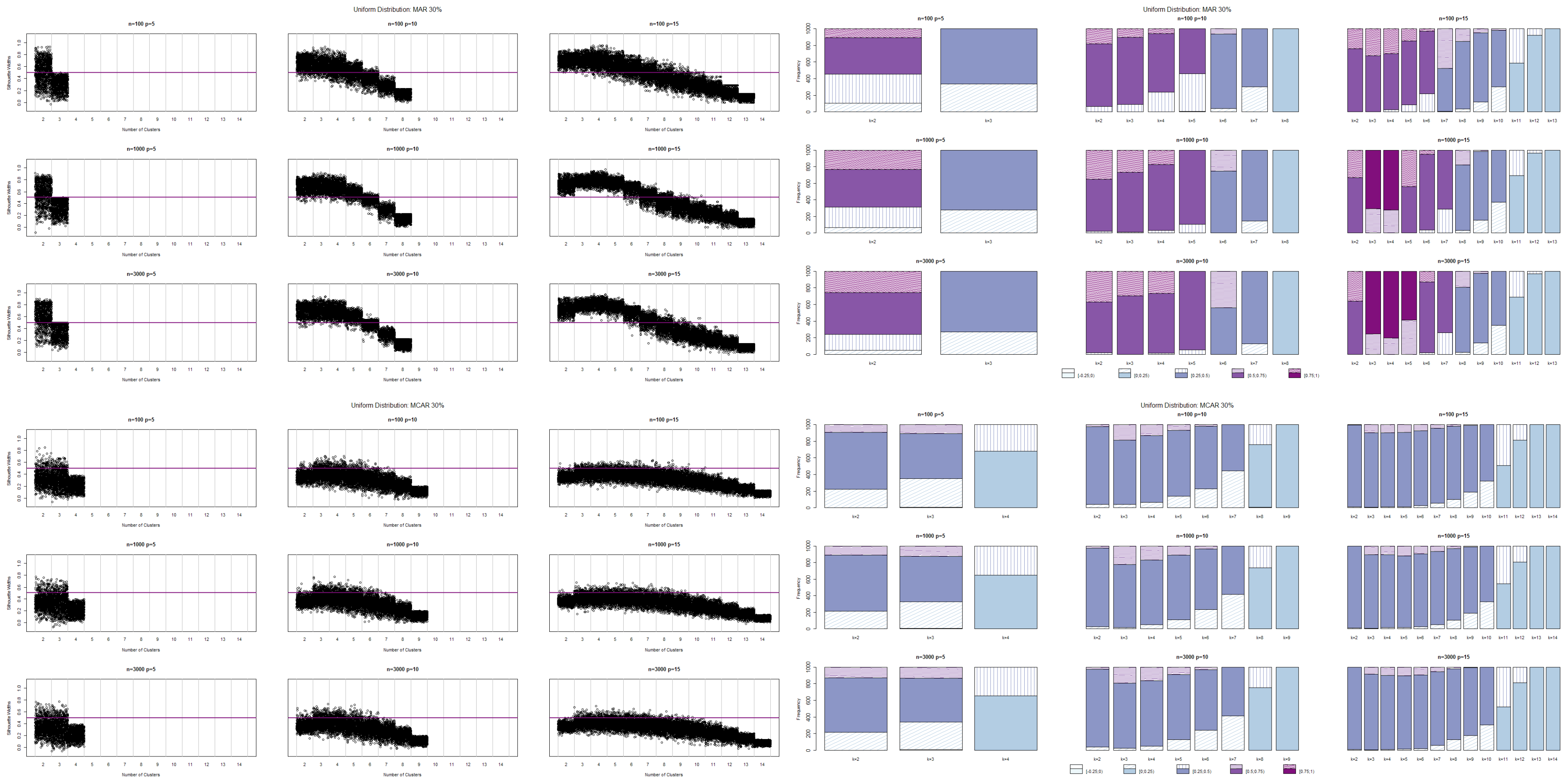


Figure 8.14 Silhouette coefficients of the uniform distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

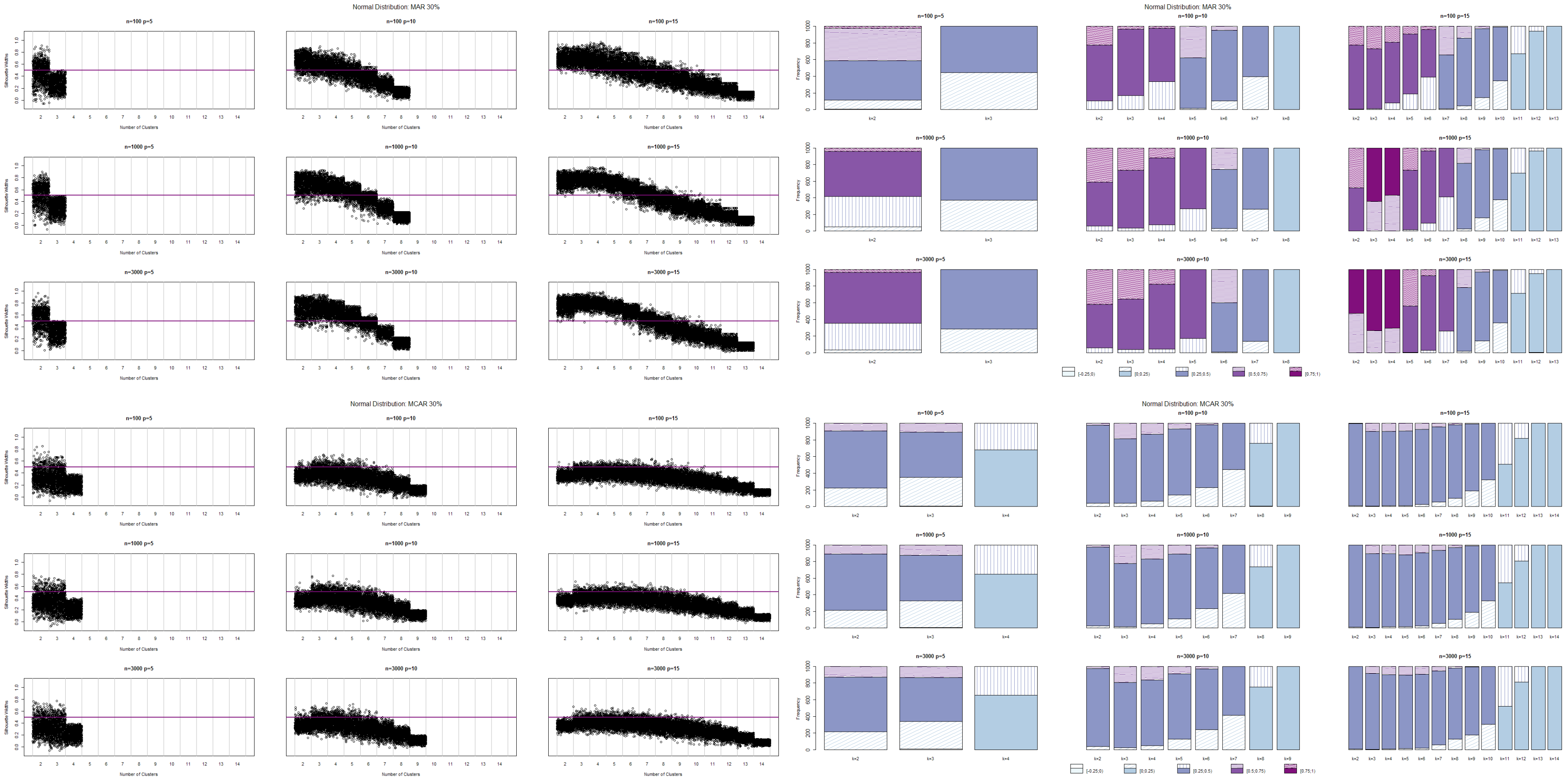


Figure 8.15 Silhouette coefficients of the normal distribution (30% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

In general, the same overall trends are observed, especially that an increase in the sample size improves the classification of the CLPs to clusters. However, now that more observations are missing due to an increase in the percentage of missing values, there are more data points available for the cluster analysis, which leads to an overall increase in silhouette coefficients in the MAR simulations for all simulated distributions.

The trends observed from the Dirichlet distribution (cf. Figure 8.13) are similar to the 10% missingness simulations (cf. Figure 8.10), with an upward shift in silhouette coefficients. The notable increase in silhouette coefficients observed in the upper left panel of Figure 8.11 ($n = 3000, p = 15$) for the uniform distribution, is now observed in the upper left panels of both Figure 8.14 and Figure 8.15 with a more discerning trend of an increase in silhouette coefficients until $k = 5$. The pattern is especially evident when $p = 15$ and $n = 1000$ and 3000. The stacked bar plots of the MAR simulations also reflect the increase in silhouette coefficients with an increase in modal class intervals with higher silhouette coefficients.

The percentage of missing values does not seem to affect the identification of clustering structures in the MCAR simulations. Again, the gradual decrease in silhouette coefficients is observed in all three simulation distributions with no vertical increase in the overall silhouette coefficients obtained from the MCAR simulations. Also, the frequency distributions presented in the lower right panels of Figure 8.13 to Figure 8.15 are similar to the stacked bar plots of Section 8.4.1.

8.4.3 Scatterplots and stacked bar plots (50% missingness)

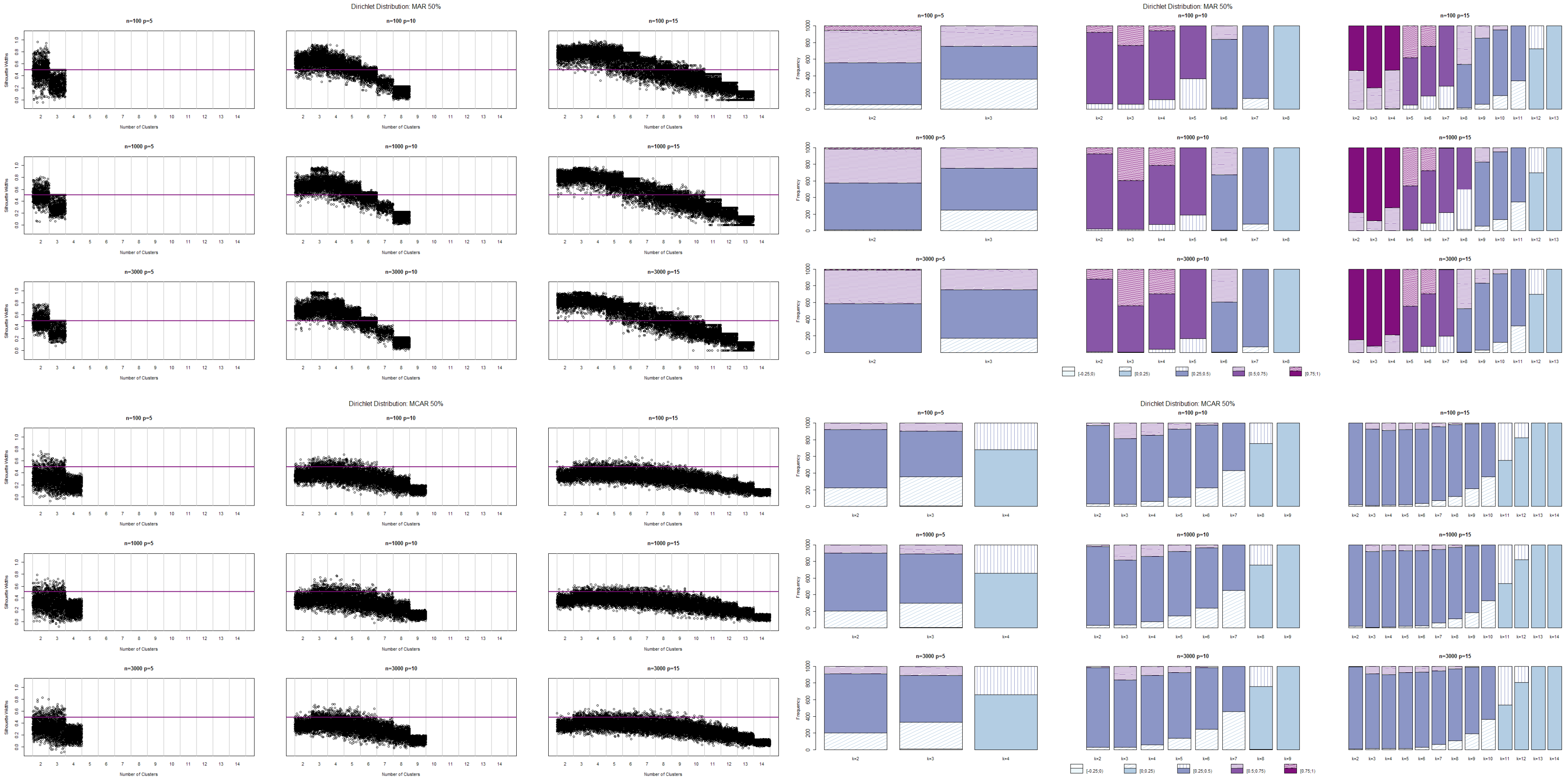


Figure 8.16 Silhouette coefficients of the Dirichlet distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

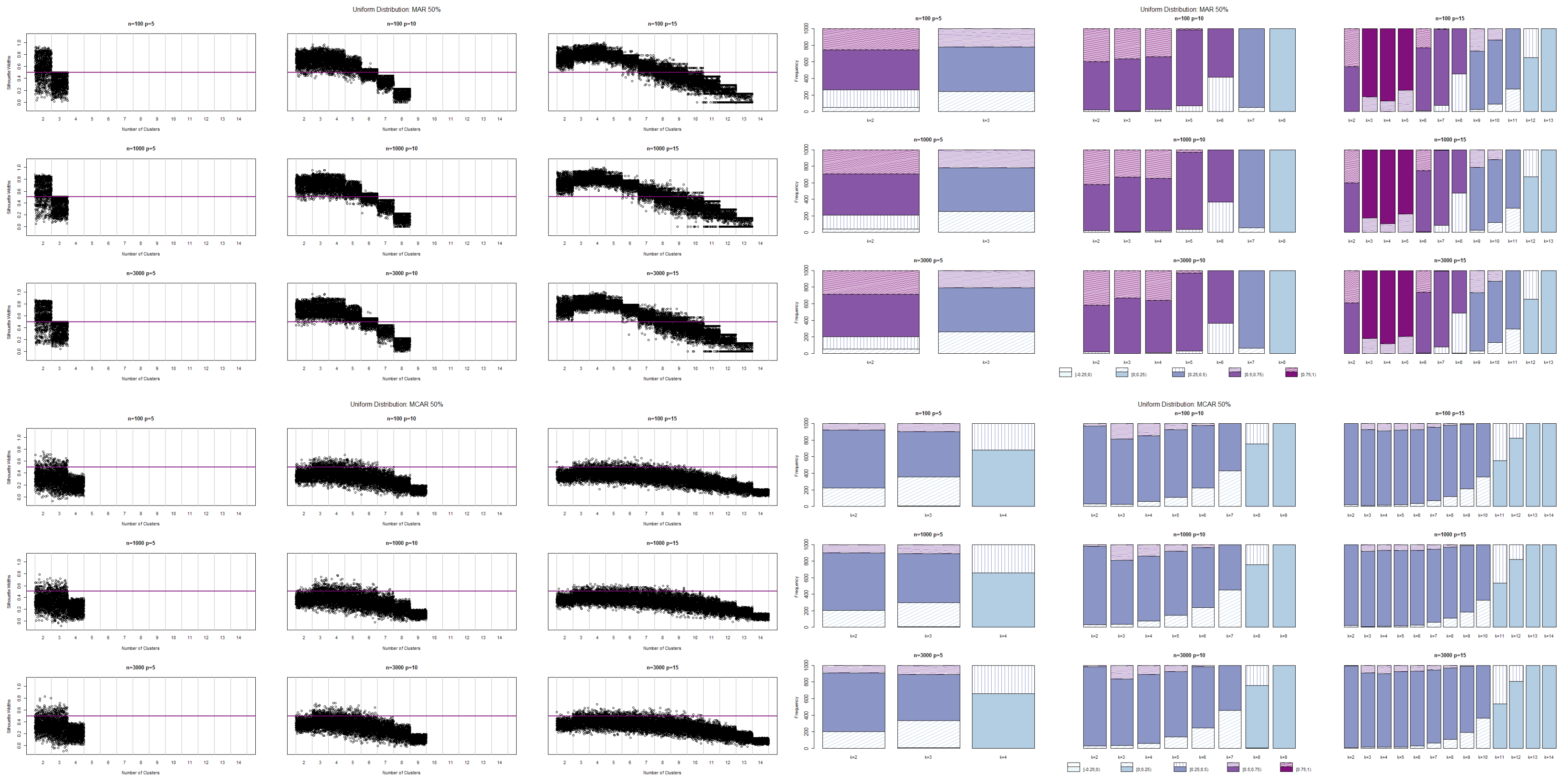


Figure 8.17 Silhouette coefficients of the uniform distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

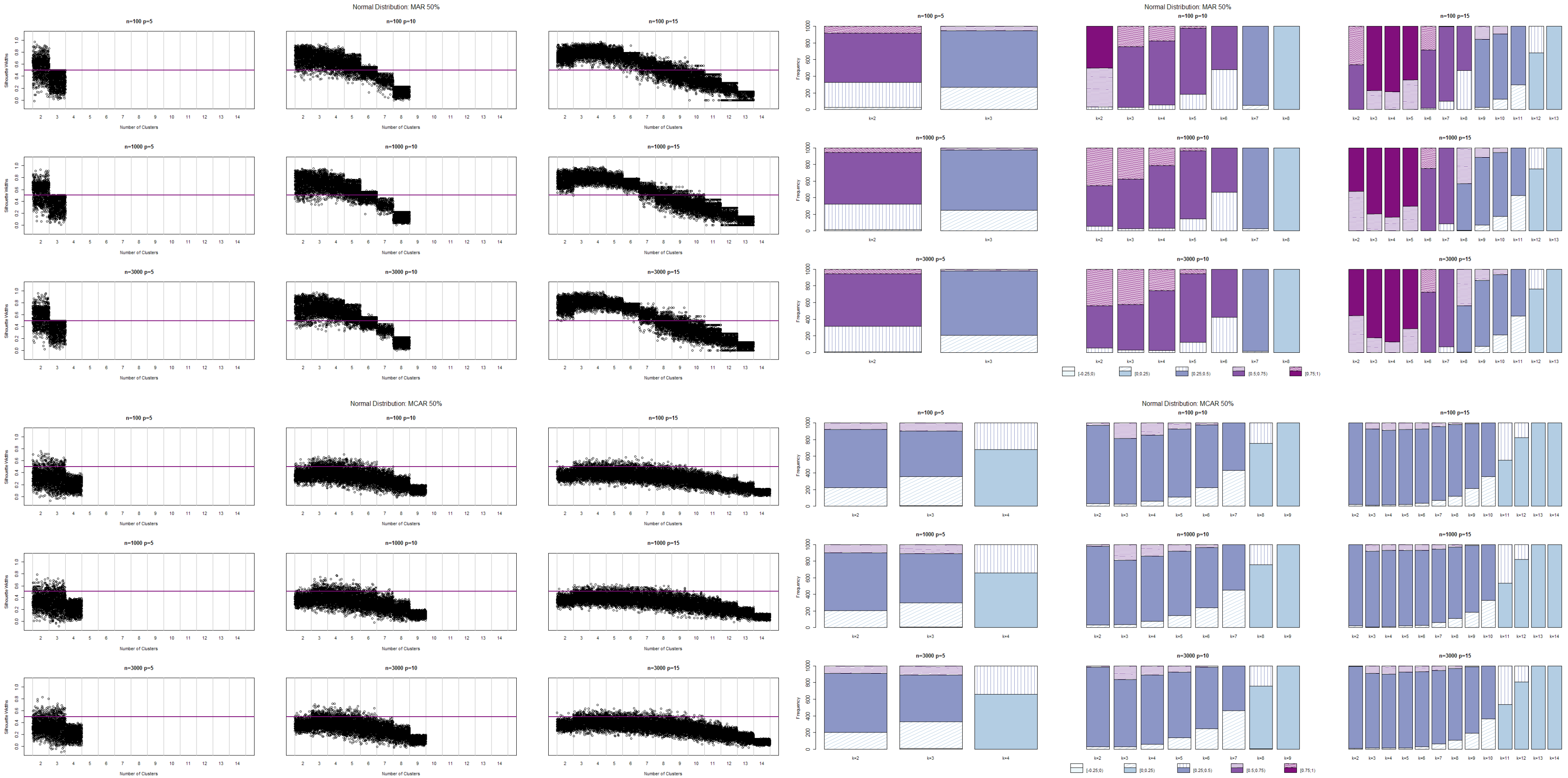


Figure 8.18 Silhouette coefficients of the normal distribution (50% missingness). Upper panels: MAR MDM. Lower panels: MCAR MDM. Left panels: scatterplots. Right panels: stacked bar plots.

The increasing trends observed from the normal (cf. Figure 8.17: upper left panel) and uniform distributions (cf. Figure 8.18: upper left panel) as observed in the previous section (cf. 8.4.2) are more distinctive with a higher percentage of missing values. The patterns now occur for all sample sizes and as the sample sizes increase the silhouette coefficients per interval (number of clusters) are more concentrated. As was observed from Section 8.4.2, the downward trend in silhouette coefficients starts for $k = 5$.

The curved patterns of the uniform and normal distributions are not visible in the Dirichlet distribution (cf. Figure 8.16: upper left panel), but there is a clear increase in the silhouette coefficients, especially for clusters before $k = 5$ with only a few silhouette coefficients occurring below the 0.5 line.

Again, the increase in the percentage of missing values does not affect the clustering structures of the MCAR simulations.

8.4.4 Heat maps of silhouette coefficients

The scatterplots and stacked bar plots (cf. 8.4.1 to 8.4.3) show that there are notable differences between the clustering structures of MAR and MCAR MDMs. However, an overall visual interpretation of the differences between the percentages of missing values cannot easily be deduced from the separate scatterplots and stacked bar plots, which focus on a particular percentage of missing values at a time. The first heat map, Figure 8.19, provides a different representation of the silhouette coefficients obtained from simulated data sets with 30% missing values. The silhouette coefficients that occur above the horizontal 0.5 line in the scatterplots (cf. Figure 8.13, Figure 8.14 and Figure 8.15) are summarised together in the heat map given in Figure 8.19. The first heat map is given as an example in order to make an association between the colours and the percentage of silhouette coefficients above 0.5 for a specific simulation scenario. The cells in the heat map are labelled with the percentages to enhance the visual interpretation of this visualisation, however as the number of elements of the heat map increases (cf. Figure 8.20 and Figure 8.21), the labels will not be included and the colours provided in the key should be used for interpretation. The number of clusters exceeding ten is omitted from the heat maps in this section, since no silhouette widths occur above 0.5 for all scenarios beyond this number of clusters.

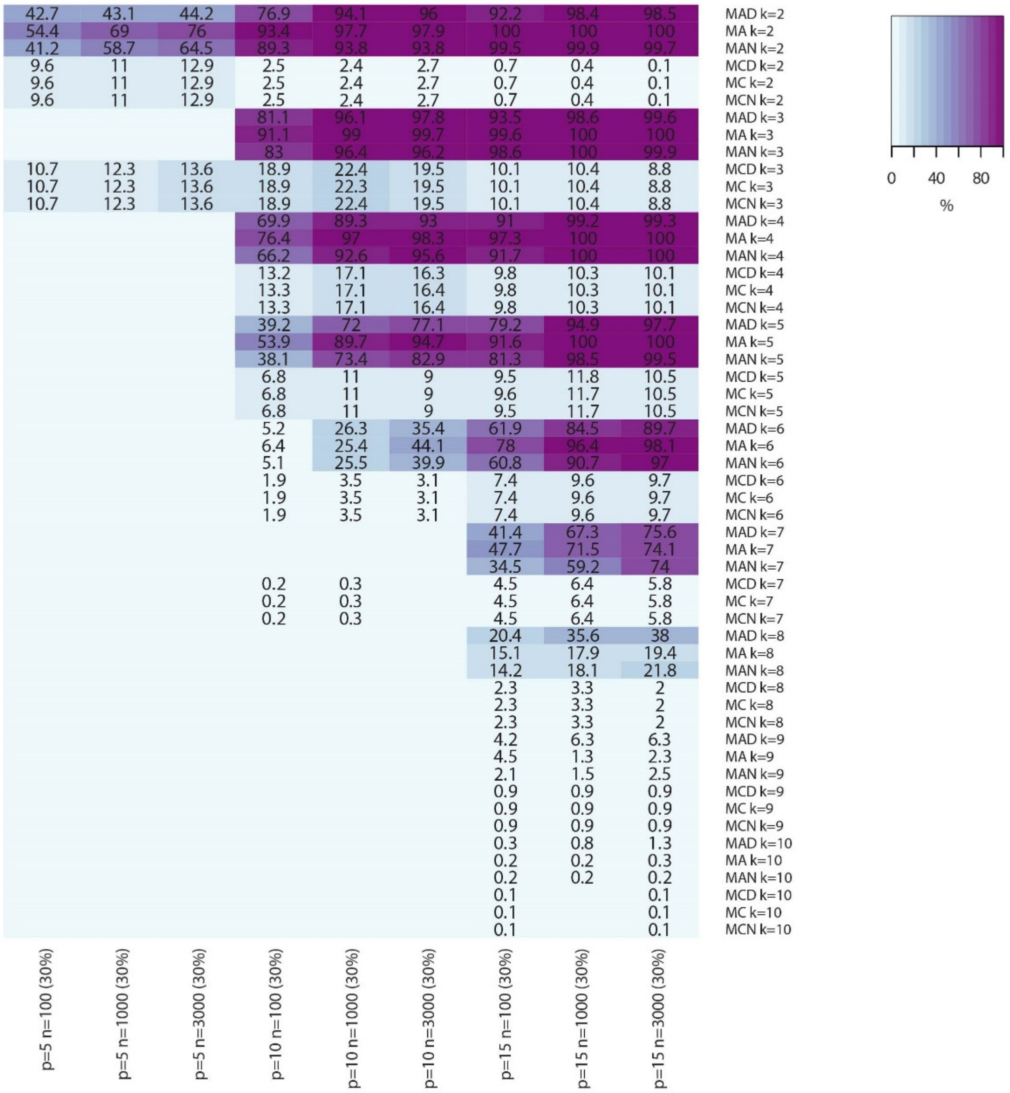


Figure 8.19 Heat map of the percentage of silhouette coefficients above 0.5 for 30% missingness.

The differences observed between the MAR and MCAR MDMs in the previous visualisations (cf. 8.4.1 to 8.4.3) are again confirmed by the clear separation of colours between the MDMs when considering a specific number of clusters. As the number of variables increases (use the labels on the x-axis for guidance) the clustering structures improve as can be seen from the darker shades of purple. Only small differences between the percentages of MCAR MDMs are observed in a few instances as was previously discussed. Again, the MCAR MDMs in this simulation study are not affected by an initial simulated distribution.

The three simulated distributions for a specific MDM and number of clusters result in similar percentages, with slightly higher percentages occurring for the uniform MAR simulations (indicated by MA in the heat map) for simulations up to and including six clusters. The MCAR

MDMs consistently achieve lower percentages of silhouette coefficients above 0.5. The exception occurs when the number of clusters is equal to three for data sets with $p = 5$. This could be due to the simulation conditions of MAR, since only four missing CLPs are available when $p = 5$. The clustering of four CLPs is not successfully separated using three clusters and results in small silhouette coefficients.

Now, all percentages of missing values are presented in Figure 8.20 without labelling each cell with the associated percentage value.

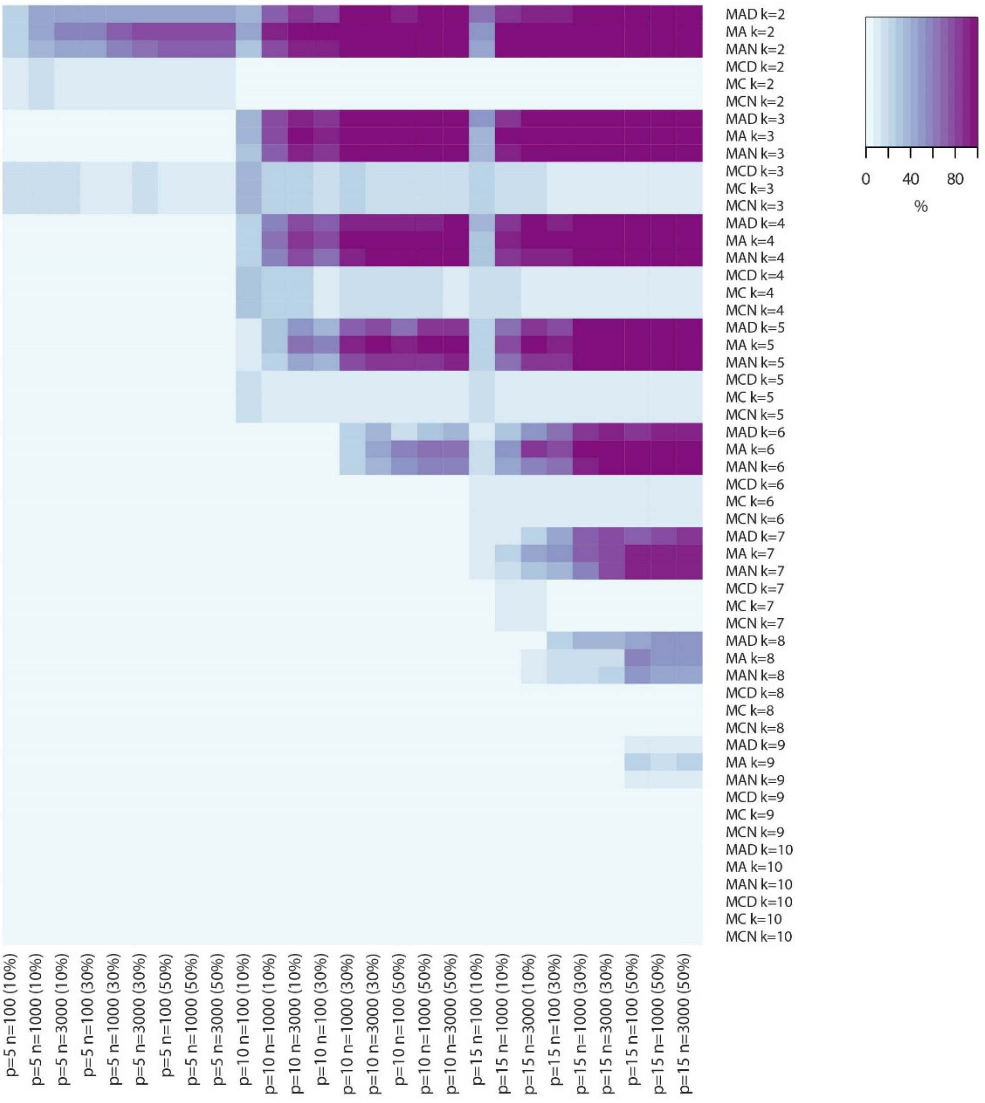


Figure 8.20 Heat map of the percentage of silhouette coefficients above 0.5.

There are fewer missing observations and therefore less missing CLPs to cluster when the percentage of missing values is small. This is reflected in the heat map, as a visual difference in trends is easier distinguished when the percentage of missing values is larger.

The clustering structures are less evident for a small sample size ($n = 100$), small number of variables ($p = 5$) and small percentage of missing values (10%). The darker shades of purple occur when the sample sizes, numbers of variables and percentage of missing values are larger: $n \geq 1000$, $p \geq 10$ and 30% or more missingness.

It is clear that the bulk of silhouette coefficients for MAR MDMs frequently occur above 0.5. This was also evident from the scatterplots (cf. 8.4.1 to 8.4.3) with large proportions of silhouette coefficients occurring above the horizontal line. In order to distinguish between the two MDMs based on silhouette coefficients, a stricter benchmark of 0.6 is proposed to suggest a MAR MDM. It is conjectured that the frequency of silhouette widths above 0.6 for MAR MDMs will be considerably more than the frequency for MCAR MDMs. The percentages of silhouette widths above 0.6 from the 1000 repetitions per simulation scenario are determined and illustrated in Figure 8.21.

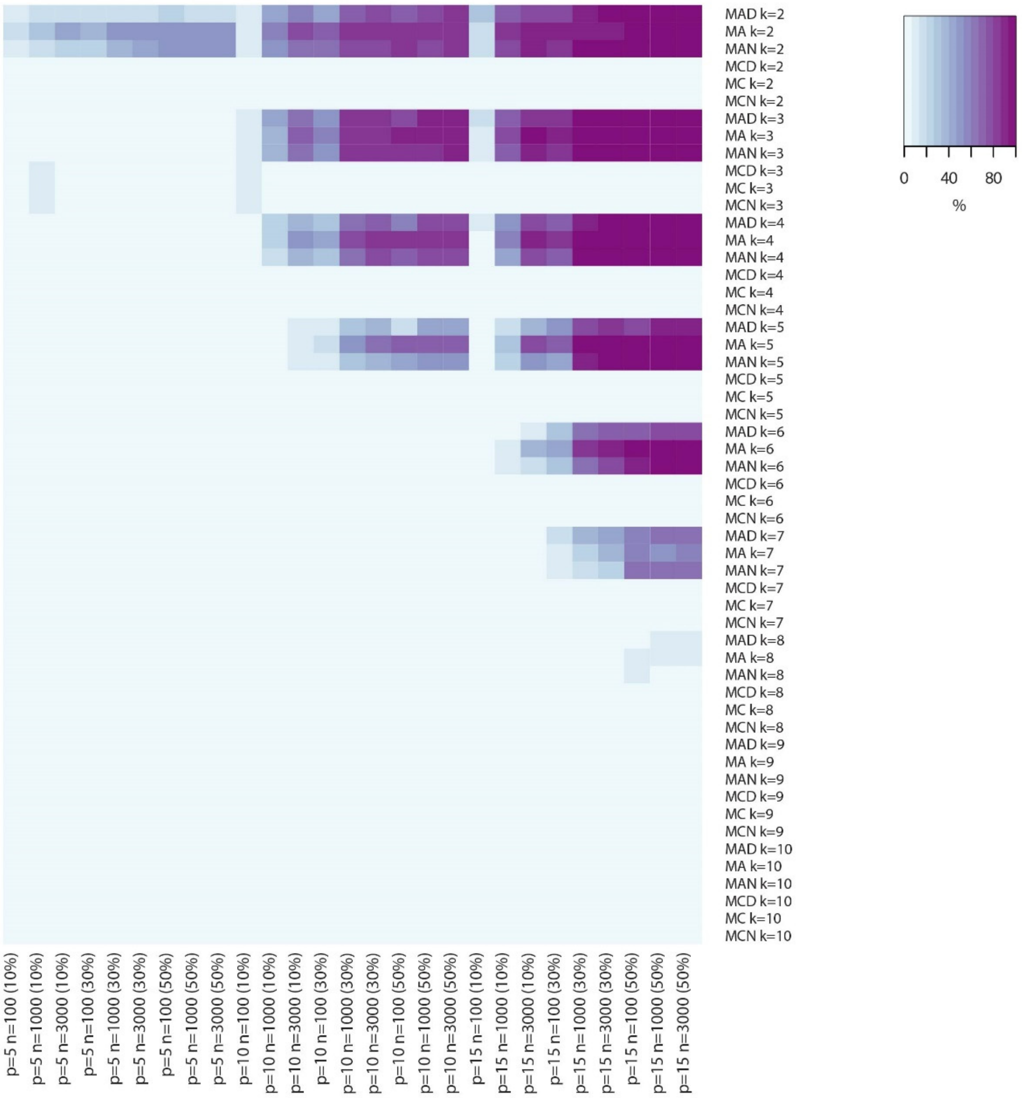


Figure 8.21 Heat map of the percentage of silhouette coefficients above 0.6.

The MCAR MDMs are easily distinguished from the MAR MDMs with lighter colours occurring in the cells related to the MCAR MDMs. The horizontal separation is due to the poor clustering structures that are present when the sample size is small ($n = 100$) and the percentage of missing values is 10%. The specific separations with lighter colours occur for: $n = 100$ with $p = 10$ and $n = 100$ with $p = 15$.

It can be conjectured that an optimal number of clusters for the evaluation of the MDM is approximately a third of the available number of missing CLPs in the recoded indicator matrix. Since the number of missing CLPs for the MAR MDMs is one less than the MCAR MDMs, the former is used to maintain a one-to-one comparison between the MDMs. Therefore, there are four available missing CLPs when $p = 5$ (three missing CLPs available for clustering), nine available missing CLPs when $p = 10$ (eight missing CLPs available for clustering) and 14 available missing CLPs when $p = 15$ (13 missing CLPs available for clustering). A third of the possible missing CLPs when $p = 5$, is equal to one which will not allow the investigation of the separation of CLPs. Hence, the smallest number of clusters, $k = 2$, is proposed when the number of variables is small. The proposed number of clusters when $p = 10$ is $k = 3$ and when $p = 15$ is $k = 4$.

The visualisations in this chapter show that overall, lower silhouette coefficients are obtained for data sets with smaller dimensions and the differences between the MAR and MCAR clustering structures are not as substantial as observed in cases with higher dimensions. Also, when the percentage of missing values and the sample size are small, the difference between the clustering structures of MAR and MCAR is more apparent when using fewer clusters ($k = 2$).

Using the proposed number of clusters, the frequency distributions of the stacked bar plots (cf. 8.4.1 to 8.4.3) show that all modal class intervals of the MCAR MDMs are in the interval$[0.25; 0.5)$, whereas the modal classes vary for the MAR MDMs with approximately 27% of simulation cases in the interval $[0.25; 0.5)$, approximately 54% in the interval $[0.5; 0.75)$ and approximately 19% in the interval $[0.75; 1)$.

The uniform distribution simulations result in slightly higher percentages compared to the other simulated distributions and especially the Dirichlet distribution simulations perform poorly when the number of variables is small ($p = 5$). The percentages of the silhouette coefficients above 0.6 for the MAR MDMs (cf. Figure 8.21) are ranked to determine which simulated distributions more frequently result in higher silhouette coefficients. Considering all number of clusters when 10% of observations are missing, the uniform simulations obtain the highest percentages in approximately 60% of simulation scenarios and the Dirichlet and normal distributions are ranked equally with each achieving the highest percentages in 20% of simulation scenarios. In the presence of 30% missing values, the uniform simulations are again ranked the best with approximately 51% of higher percentages and the normal distribution simulations with 30% and Dirichlet with 19%. Lastly, when 50% of missingness occurs, the uniform distribution simulations obtain higher values in approximately 57% of scenarios, whereas the Dirichlet distribution ranked second with 26% and the normal simulations with 17%. Therefore, it seems as if the simulated distribution of the data sets contributes to the success of the separation of the clusters in the missing CLPs, with the simulations from the uniform distribution outperforming the other two distributions.

8.5 Limited multiple active results

The current capacity of the HPC clusters at Stellenbosch University is not sufficient to complete the clustering of the missing subsets after recoding the indicator matrices with multiple active handling. Only a few cases with lower dimensions managed to terminate and this does not allow for a conclusive overview of the difference between the clustering of structures present in single active- or multiple active handled data sets (cf. 3.4.2.2). A future investigation will include the further optimisation of code to reduce the required RAM in consideration with the parallelisation of the SVD and an incremental SVD approach of Iodice D'Enza and Markos (2015).

8.6 Conclusion

This chapter proposed the use of sMCA biplots for the missing subsets of data handled with a single active recoding procedure for an initial inspection of the inherent structures of the samples and CLPs of missing responses. It was found that data sets with small dimensions ($n = 100$ and $p = 5$) do not result in discernible differences between MAR and MCAR MDMs. However, with an increase in the size of the data set, separation is observed between the CLPs of MAR MDMs with overlapping points in the separate groups. The sample points also result in structured patterns which converge to overlapping separated groups as the percentage of missing values increases. To the contrary, MCAR MDM sMCA biplots do not show clear separation between CLPs and the sample points tend to form a cloud of points, especially in a higher percentage of missing values. Therefore, the visual differences between the MDMs are noticeable. In order to confirm the visual interpretations, cluster analysis is applied to the CLPs of the sMCA biplots.

The silhouette coefficients confirmed that the clustering structure of data sets is more evident for larger data sets, both sample size and number of variables, with a large percentage of missing values. It was observed that the data sets simulated from a uniform distribution achieved slightly higher silhouette coefficients than the Dirichlet and normal distributions.

However, similar patterns in silhouette coefficients were observed for the uniform and normal distributions.

An optimal number of clusters to use with respect to the parameters of a given data set has been identified. In general the smallest number of clusters (usually, $k = 2$) is proposed when attempting to identify the MDM of a data set with small dimensions and a low percentage of missing values. However, as a general rule of thumb a third of the available number of missing CLPs can be used as the optimal number of clusters to deduce whether the MDM is possibly MAR of MCAR. If the silhouette coefficient obtained from the number of clusters equivalent to a third of the number of missing CLPs is above 0.6, it is strongly suggestive of a MAR MDM.

The next chapter will present the application of the proposed methodology (cf. Chapter 4) to a real data application.

Chapter 9
Real data application

9.1 Introduction

The International Social Survey Program (ISSP 1994) is considered for the real application. This survey investigated the family perspectives of changing gender roles in Germany using 11 questions with three possible CLs (agree, neutral and disagree), as well as five demographic variables: region, gender, age, marital status and education. The adapted survey as used by Greenacre and Pardo (2006b) is available from: http://www.carme-n.org/?sec=data. The results presented in this chapter are based on the survey consisting of 3291 samples of which 811 samples contain non-responses, which relates to missing values in approximately 25% of the samples. Missing values only occur in the survey questions while the demographic information is fully observed. The analysis is only carried out on the survey questions, since some of the demographic variables are not measured on a nominal scale. The demographic information will however be used to enhance the biplot display and improve the visual interpretation.

The results obtained from the simulation study (cf. Chapter 6 and Chapter 8) will aid in the interpretation of the real application results. The overall percentage (4.58%) of missing values in the real data set is less than the simulated cases, which considered between 10% and 50% missing values. However, the findings of lower percentages of missing values (approximately 10%) with $n = 3000$ and $p = 10$ in the simulation study can be carried forward as a benchmark.

First, the responses of the variables are investigated to determine the possible distribution of the data set in Figure 9.1 (cf. 9.2). Thereafter the possible MDM will be investigated in Section 9.3. As was found in Chapter 8, data sets with larger dimensions result in noticeable differences between the clustering structures of MAR and MCAR MDMs. Even with a small percentage of missing values, sample points of MAR MDMs tend to occur in patterns and not clouds of points as is the case with the sample points of MCAR MDMs. The CLPs of MAR MDMs show separation with overlapping points occurring in a higher percentage of missing values, which is not expected to be the case for this real data application. After the visual inspection, cluster analysis is applied and a third of the available missing CLPs are used to determine the optimal number of clusters to specify for the procedure. Using single active handling (cf. 3.4.2.2), 11 missing CLPs are available for this real data application, thus four clusters are used for the final evaluation of the MDM.

A suitable missing data approach will be applied in Section 9.4 after the identification of the MDM (cf. 9.3).

9.2 Variable information

The variable information (Greenacre & Pardo, 2006b: 200) is summarised in Table 9.1 and Table 9.2.

Table 9.1 Variable information for survey questions of ISSP 1994.

Variable	Survey questions	% of missing values
A	"A working mother can establish just as warm and secure a relationship with her children as a mother who does not work."	3.13
B	"A pre-school child is likely to suffer if his or her mother works."	3.49
C	"All in all, family life suffers when the woman has a full-time job."	2.86
D	"A job is all right, but what most women really want is a home and children."	6.81
E	"Being a housewife is just as fulfilling as working for pay."	6.93
F	"Having a job is the best way for a woman to be an independent person."	4.44
G	"Most women have to work these days to support their families."	4.07

Table 9.1 continued		
Variable	Survey questions	% of missing values
H	“Both the man and woman should contribute to the household income.”	3.92
I	“A man’s job is to earn money; a woman’s job is to look after the home and family.”	2.34
J	“It is not good if the man stays at home and cares for the family and the woman goes out to work.”	5.01
K	“Family life often suffers because men concentrate too much on their work.”	7.35
Overall percentage of missingness: 4.58%		

Table 9.2 Variable information for demographic questions of ISSP 1994.

Variable	Demographic questions
c	Region: West Germany (DW) or East Germany (DE)
g	Gender: Male (M) or Female (F)
a	Age: A1 (up to 25), A2 (26-35), A3 (36-45), A4 (46-55), A5 (56-65) or A6 (66 and over)
m	Marital status: MA (married), WI (widowed), DI (divorced), SE (separated) or SI (single)
e	Education: E0 (none), E1 (incomplete primary), E2 (primary), E3 (incomplete secondary), E4 (secondary), E5 (incomplete tertiary) or E6 (tertiary)

Survey questions A, F and G are formulated in favour of women pursuing a career, whereas survey questions B, C, D, E, I and J are formulated opposing women to pursue a career. Survey questions H and K are not directly aimed at supporting or opposing women who work.

The distribution of the variable responses could aid in deciding on the best approach to follow for the data with missing values. The frequency distributions of the data containing missing values are presented in bar graphs in Figure 9.1.

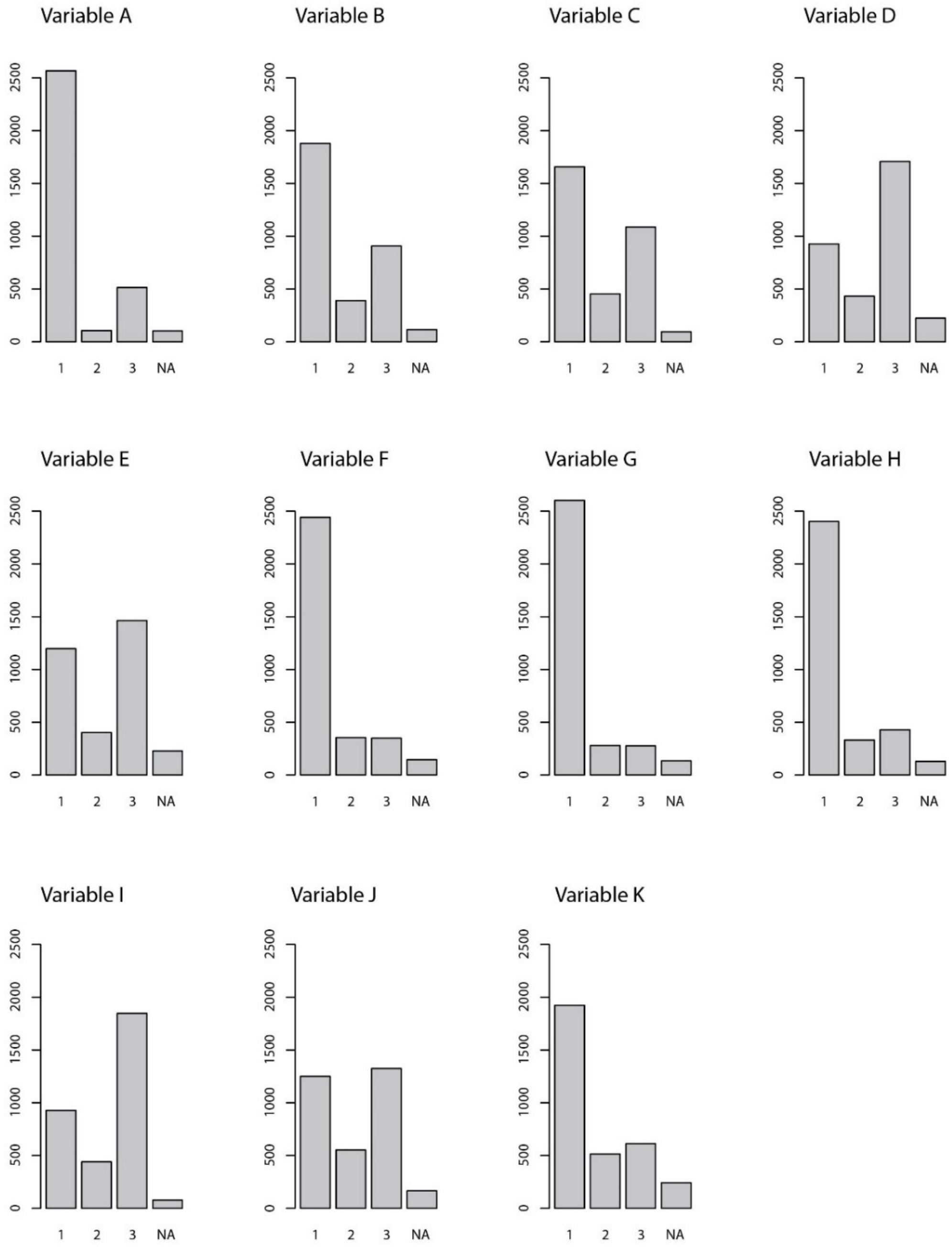


Figure 9.1 Bar graphs of each variable in the adapted survey: 1 – agree, 2 – neutral, 3 – disagree, NA – missing

The bar graphs in Figure 9.1 do not necessarily reflect the distributions of the fully observed variables, but suggests plausible response distributions of variables. The majority of variables in Figure 9.1 will probably remain skewed even if the missing values are attributed to other CLs. This indicates that the distribution of the variables could be skewed and therefore the results obtained from the Dirichlet distribution can be used as a guideline (cf. Chapter 6 and Chapter 8).

9.3 MDM identification

The sMCA biplot of the missing subset of the real data application is presented in Figure 9.2. The right panel shows that the demographic variables can be used to enhance the visualisation by constructing convex hulls. This enables a focused view of the dispersion of sample points with respect to a specific classification. The education level variable is the only demographic variable that showed noticeable differences in the spread of sample points.

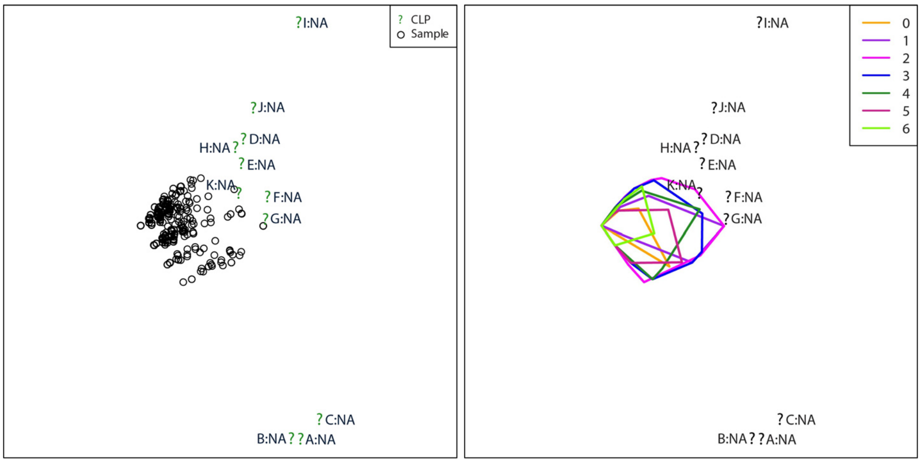


Figure 9.2 sMCA biplot: missing subset of real data application. Left panel: standard sMCA biplot display. Right panel: sMCA biplot with samples colour coded according to educational level and represented as convex hulls.

The CLPs in Figure 9.2 show a vertical separation with three visible groupings: (1) A: NA, B: NA, C: NA, (2) D: NA, E: NA, F: NA, G: NA, H: NA, J: NA, K: NA and (3) I: NA. The separation of the CLPs is a first indication of a MAR MDM. The sample points do not form a cloud of points as was observed from the simulations in Chapter 8. Both the overlapping sample points that occur to the left and the triangular pattern of the sample points are also indications of a MAR MDM.

The convex hulls confirm that the positions of the samples are influenced by the education level, which causes the bulk of overlapping sample points to the left of the sMCA biplot.

Therefore, the visualisations suggest that the MDM is possibly MAR.

In order to confirm the visual interpretation of the MDM, cluster analysis is applied using the minimal number of clusters ($k = 2$) up to a third of the possible number of missing CLPs ($k = 4$). The allocated clusters are represented in different colours in Figure 9.3.

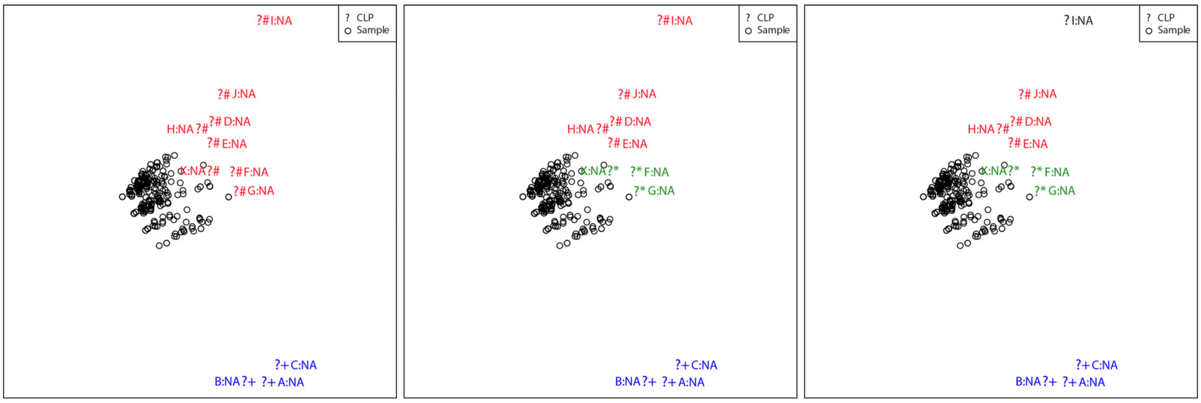


Figure 9.3 Illustrating allocated clusters for the sMCA biplot (missing subset) of the real data application. Left panel: $k = 2 \rightarrow s(i) = 0.7832$*. Middle panel:* $k = 3 \rightarrow s(i) = 0.5016$*. Right panel:* $k = 4 \rightarrow s(i) = 0.5805$*.*

The simulation results confirmed that using a third of the possible missing CLPs, a silhouette coefficient above 0.6 is indicative of a MAR MDM (cf. Chapter 8). Separating the missing CLPs into four clusters, result in a silhouette coefficient of approximately 0.6. Therefore, the silhouette coefficient concurs with the visual indication in the right panel of Figure 9.3, which conjectures a MAR MDM.

9.4 Missing data approach for real application

It was found that all three approaches (GPAbin, sMCA and RIMCA) achieved similar results (cf. Chapter 6) in a low percentage of missing values (10%) with a MAR MDM and $n = 3000$ and $p = 10$. The sMCA method achieved a slightly higher overall similarity percentage (86%) than the GPAbin (85%) and RIMCA (84%) methods (cf. Figure 6.1). The bias measures were similar for the three approaches with the sMCA method achieving slightly better fit values than the GPAbin and RIMCA approaches. Only the novel approaches, GPAbin and sMCA, will be applied to the real application.

9.4.1 Multiple imputation and generalised orthogonal Procrustes analysis

The MIMCA procedure is used to generate ten imputations. An MCA biplot is constructed for each imputed data set and then optimally aligned using GPA (cf. 2.9.1 and 4.3.1). Figure 9.4 presents the transformations of the CLPs obtained from MIMCA (triangle plotting character) to the CLPs obtained from GPA (square plotting character).

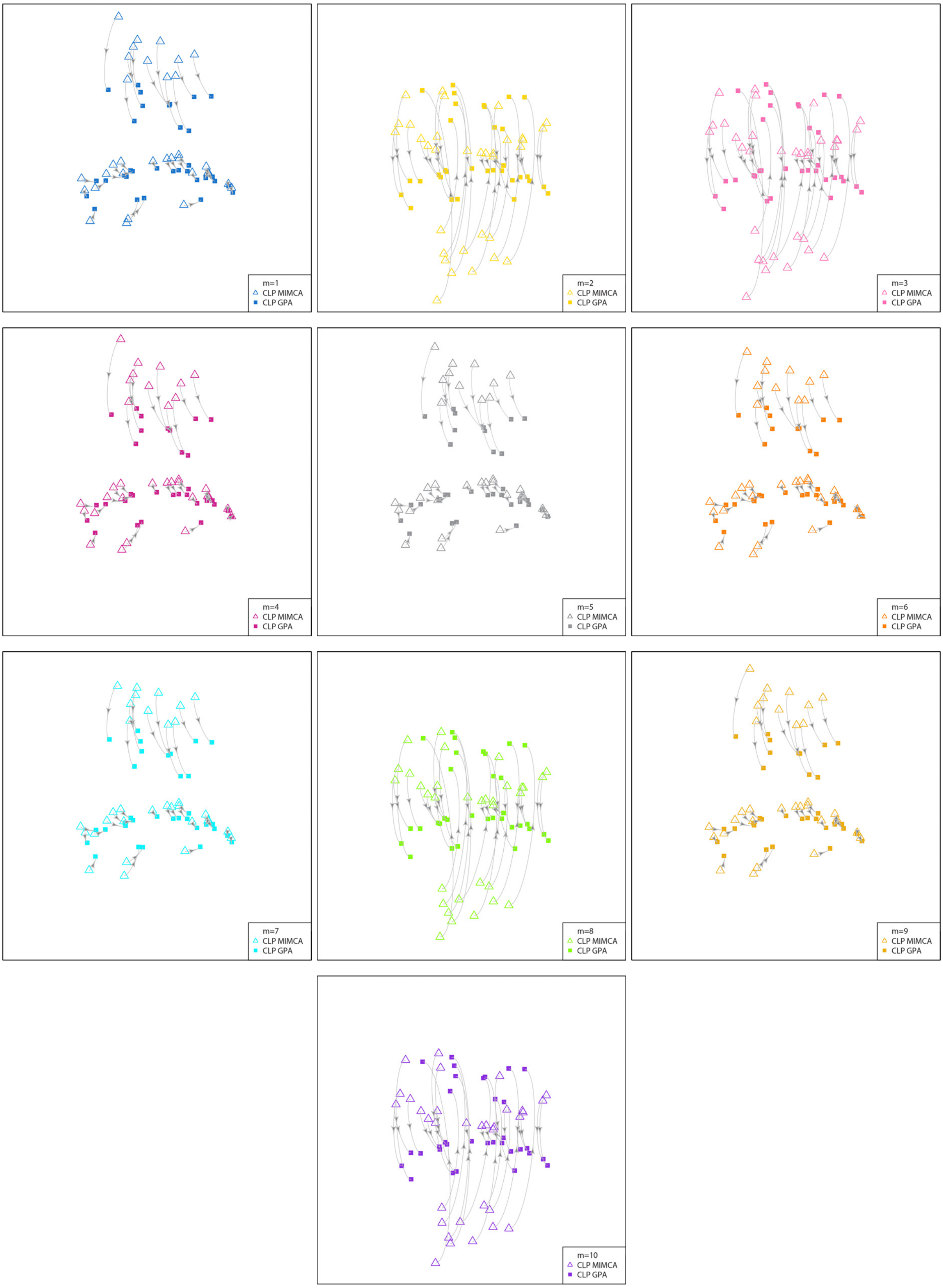


Figure 9.4 Illustration of CLPs before and after GPA for ten MIs. Each panel represents the CLPs for a particular MI for the real data application.

Minimal transformations are observed for imputations one ($m = 1$), four ($m = 4$), five ($m = 5$), six ($m = 6$), seven ($m = 7$) and nine ($m = 9$) with reflections and rotations performed on imputations two ($m = 2$), three ($m = 3$), eight ($m = 8$) and ten ($m = 10$).

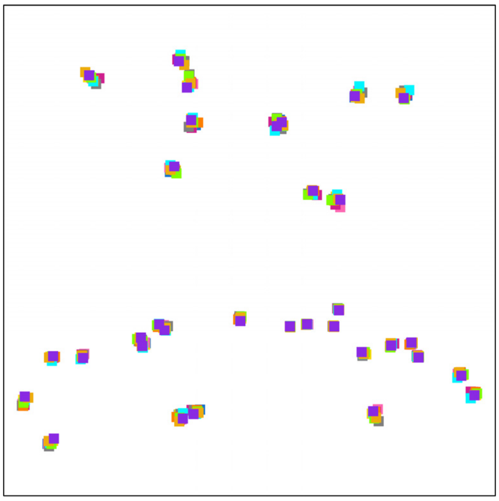


Figure 9.5 Superimposed transformed CLPs from Figure 9.4 for the real data application.

The superimposed CLPs in Figure 9.5 are closely grouped together after GPA and therefore indicate low variability between the MIs.

Convex hulls are used in Figure 9.6 to visually separate the MIs of a particular CLP, which improves the exploratory analysis of the between-imputation variation. The size of the area of the convex hull is an indication of the variation between the MIs and might be related to the percentage of missing values for a particular variable. Similar PCA applications appear in the literature where convex hulls are used to investigate the uncertainty between the factor loadings of PCA from MIs (Van Ginkel & Kroonenberg, 2014). Another example is given by Josse and Husson (2012) where both ellipses and convex hulls are used in factor maps to visualise the uncertainty between imputations.

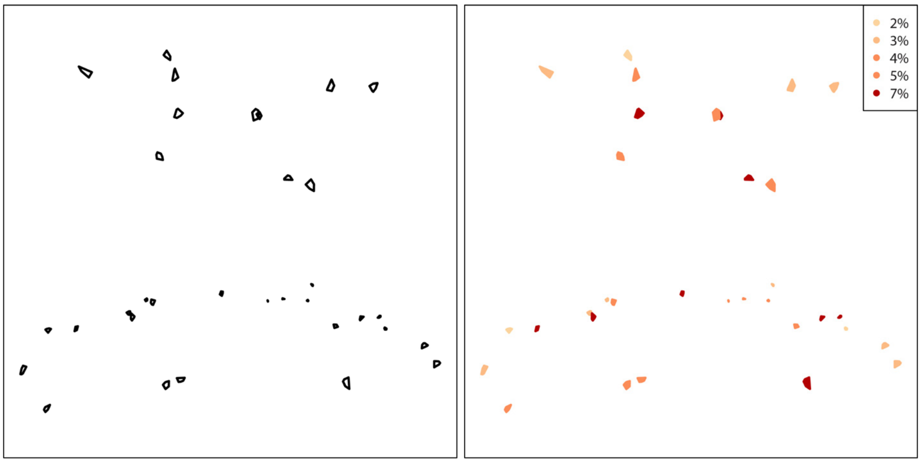


Figure 9.6 Left panel: Convex hulls of superimposed CLPs after GPA. Right panel: Colour coded convex hulls with increasing colour intensity according to the percentage of missing values

The percentages of missing values per variable (cf. Table 9.1) are used for the colour representation in the right panel of Figure 9.6. Therefore, the colours reflect the percentage of missing values in a particular variable and not for a specific CL. The right panel of Figure 9.6 shows that the area of the convex hulls presented in two dimensions cannot be used as an accurate reflection of the percentage of missing values, since a larger area does not imply a higher percentage of missing values for a particular variable.

9.4.2 GPAbin and subset multiple correspondence analysis biplots

The biplot visualisations are considered and compared for the GPAbin (cf. Figure 9.7) and sMCA (cf. Figure 9.8) approaches.

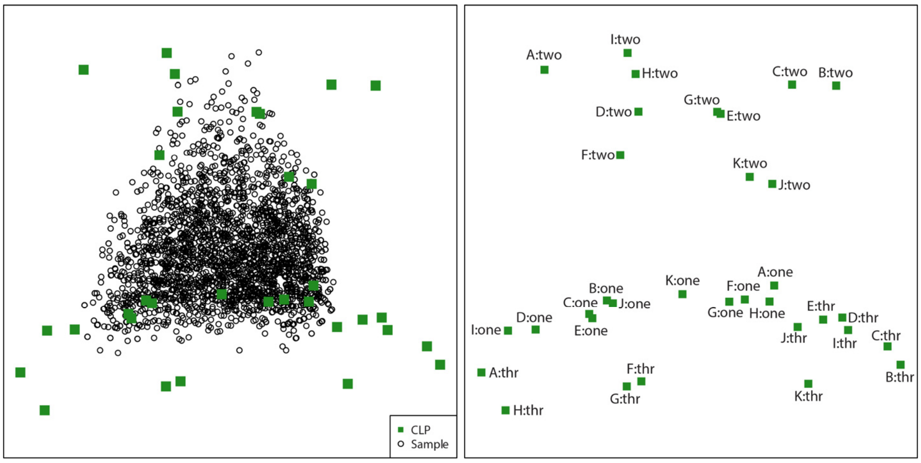


Figure 9.7 Left panel: GPAbin biplot. Right panel: CLP descriptions of GPAbin biplot.

Considering the GPAbin biplot (cf. Figure 9.7: left panel) a vertical separation is visible between the CLPs. The CLP labels in Figure 9.7 (right panel) show that the CLPs related to neutral responses (second CL) are separated from the response levels that reflect an opinion (agree and disagree).

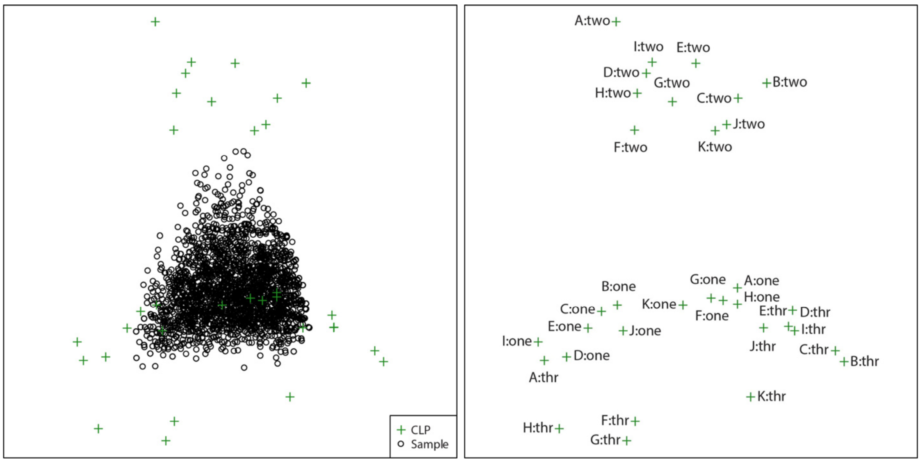


Figure 9.8 Left panel: sMCA biplot. Right panel: CLP descriptions of sMCA biplot.

The sMCA biplot in Figure 9.8 results in a similar pattern of samples and CLPs as observed in the GPAbin biplot in Figure 9.7. This confirms the similarity of the techniques when the percentage of missing values is small (cf. Chapter 6). Again, the neutral responses (second CL) are separated from the first and third CLs, as was observed in the GPAbin biplot (cf. Figure 9.7).

Based on the description of the variables, it is expected that 'agree' responses (first CL) for questions against women pursuing a career will be closely associated to 'disagree' responses (third CL) for questions supporting women to stay at home. This is confirmed by the positions of the CLPs annotated in the right panels of Figure 9.7 and Figure 9.8, where some of the 'disagree' responses (third CL) are situated closely to some of the 'agree' (first CL) responses. The neutral responses (second CL) are all grouped together in the upper half of the configurations. The response levels can be further highlighted by using colour coded plotting characters, as presented in Figure 9.9.

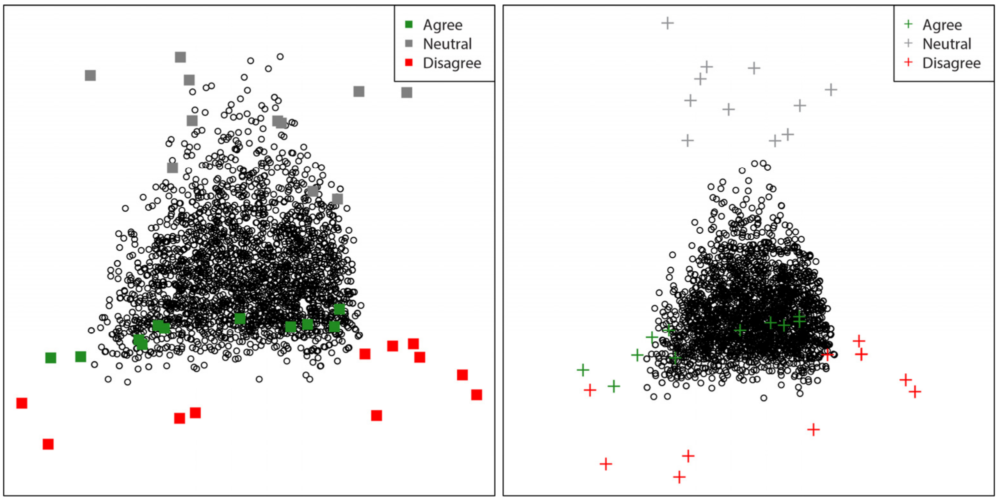


Figure 9.9 Colour coded CLPs according to the response level. Left panel: GPAbin biplot. Right panel: sMCA biplot.

The use of colour enhances the visualisations by drawing attention to the response type (agree, neutral or disagree). Subtle differences are noted between the two visualisations in Figure 9.9 with respect to the placement of CLPs in the upper half of the configurations and the scaling of the samples. However, the patterns of sample points are congruent and the two visualisations are deemed to be similar.

The demographic variables can be used to separate the samples in the biplots to explore the association between samples and responses to the survey questions. Separate biplots (cf. GPAbin: Figure 9.10 and sMCA: Figure 9.11) are presented for each of the demographic variables in the survey. To avoid cluttered displays, convex hulls are used to visualise the samples based on the observed demographic information.

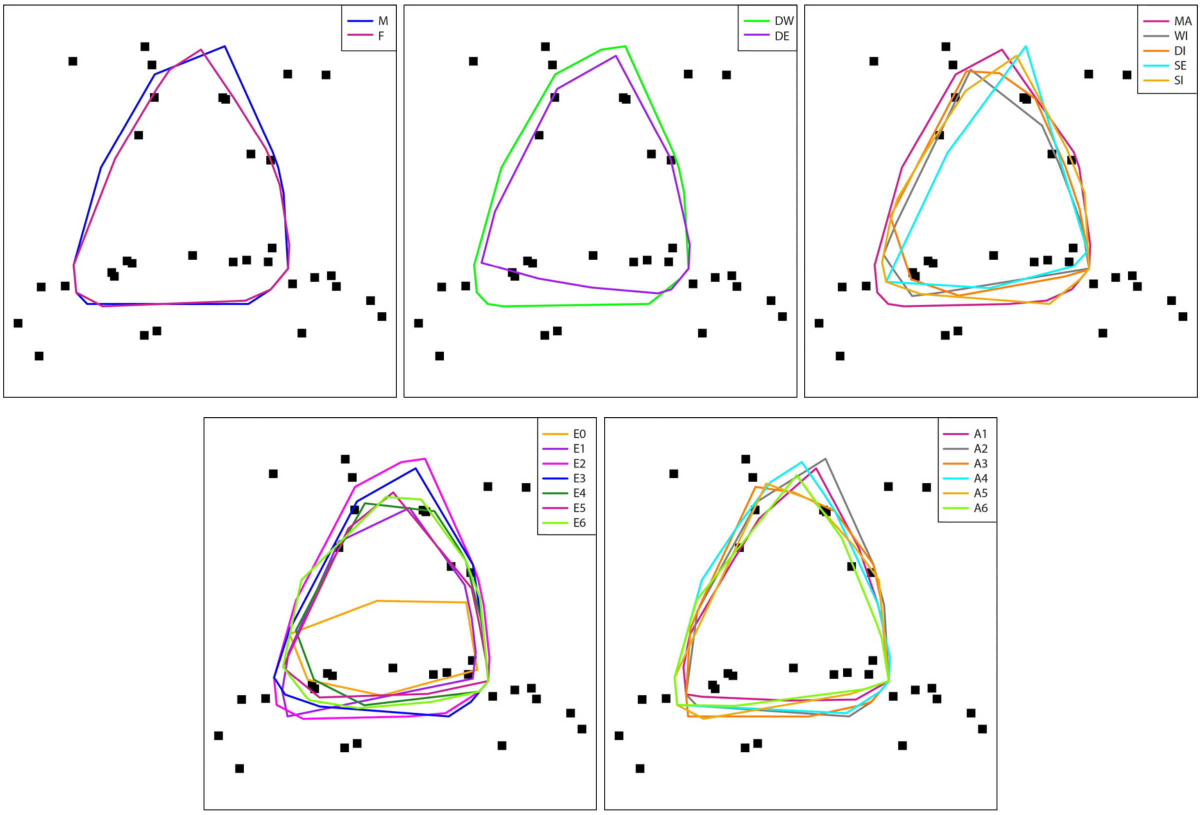


Figure 9.10 GPAbin biplots with convex hulls for sample separation based on demographic information. Top panels: Gender (left), Region (middle) and Marital status (right). Lower panels: Education (left) and Age (right).

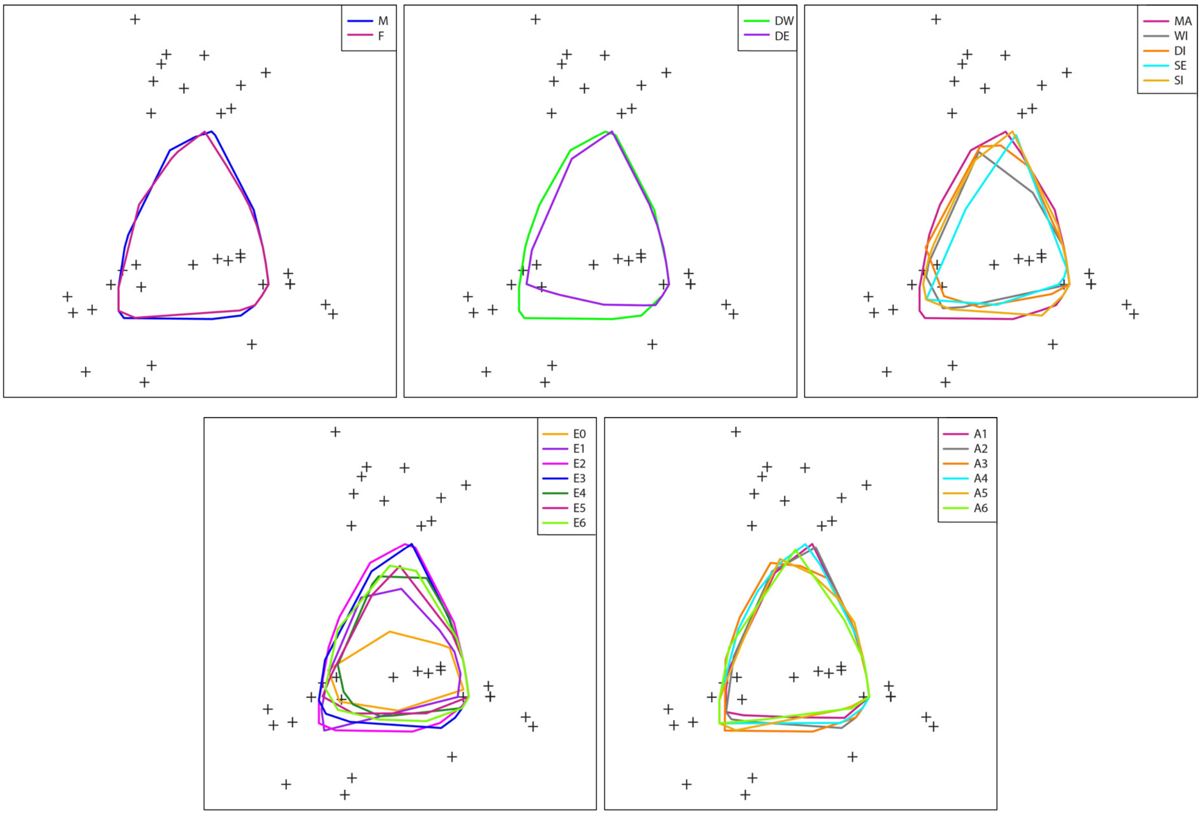


Figure 9.11 sMCA biplots with convex hulls for sample separation based on demographic information. Top panels: Gender (left), Region (middle) and Marital status (right). Lower panels: Education (left) and Age (right)

The only clear separation occurs for the education variable in both approaches (cf. Figure 9.10 and Figure 9.11: lower left panels). The samples with the lowest level of education (E0 – none) are concentrated and positioned closer to the response levels linked to a specific opinion (disagree or agree).

9.4.3 Comparison of GPAbin and subset multiple correspondence analysis biplots

In order to establish how similar the final visualisations of the GPAbin and sMCA approaches are, the measures of comparison (cf. 4.6.1) have been calculated and are presented in Table 9.3.

Table 9.3 Measures of comparison: GPAbin and sMCA real application

Measures	Two dimensions
PS	0.0749
CC	0.9815
AMB	0.4140
MB	-0.1285
RMSB	0.6023

The PS value is close to zero and the CC value is close to one, which confirms similarity between the two configurations. The bias values should be approximately equal to zero to indicate unbiased representation. Since there are no indices to establish severe bias, the top 10% ranges defined by the simulation study results (cf. Table 6.9) are used as a guideline. All bias measures (AMB, MB, RMSB) occur within the ranges, therefore implying that the sMCA and GPAbin configurations are similar.

9.5 Conclusion

The results from the preceding chapters were applied to analyse a real data application. In practice, the possible MDM should be confirmed before applying a missing data technique. It was conjectured that the MDM is MAR and that the distributions of a majority of variables are possibly skewed. Therefore, the conclusions of the Dirichlet distribution simulations were used to decide on the most suitable approach to visualise the real data application. The sMCA approach performed slightly better than the GPAbin method for the specific simulation parameters considered in Chapter 6. However, the difference between the methods is not discerning and both approaches (GPAbin and sMCA) were applied and illustrated in this chapter.

The similarity between the approaches for the specific application confirmed the findings of the simulations (cf. Chapter 6).

The subsequent chapter will conclude the findings of this research.

Chapter 10
Concluding remarks

"The greatest value of a picture is when it forces us to notice what we never expected to see." (Tukey, 1977: vi)

10.1 Review

The occurrence of missing data is not unique to any particular research application; it should be regarded as a multi-disciplinary phenomenon. Every data analyst should acquire the skills for the proper handling of missing data.

This research amalgamated three core methodologies, namely multivariate categorical data analysis, missing data techniques and biplot visualisations. The overall aim of this study was to develop unbiased visualisations for multidimensional categorical data sets containing missing values. The correct visualisation can expose patterns and provide insight into the multidimensional relationships between samples and variables in a data set. Tukey's (1977) comment on the contribution of visualisations is particularly true for the visualisation of data sets containing missing values, as missing observations become 'visible'.

Almost three decades have passed since Wainer (1990: 346) made the statement that visualisations are "an evolving invention". This research was in agreement with Wainer's (1990) statement by developing novel visualisations to enhance the exploration of multivariate categorical data with missing values.

Four study objectives (cf. 1.4) were stipulated in order to achieve the aim of this research:

- The GPAbin objective: Obtaining an unbiased single visualisation after MI
- The sMCA objective: Determining the applicability of visualisations without prior imputation
- The prediction objective: Obtaining a single completed categorical data set using predictions from visualisations
- The MDM objective: Identifying the MDM using visualisation.

In order to attain these objectives, known procedures had to be extended and novel procedures developed. These new proposals were evaluated by means of an extensive simulation study (cf. Chapter 5) with the results summarised in three separate chapters:

- The GPAbin objective and sMCA objective were discussed in Chapter 6.

In general, missing data techniques perform better for MCAR MDMs than for MAR MDMs. It is therefore important to understand the cause of missingness before applying relevant techniques. While the non-technical practitioner may apply the sMCA approach when the percentage of missing values is small (10% or less), it is advisable to use an MI technique in conjunction with the GPAbin method when dealing with larger proportions of missing data.

- Chapter 7 considered the prediction objective.

It was found that albeit useful to obtain a single complete data set after MIs using the prediction methods, the relationships between the samples and CLPs are only preserved for data sets with a small proportion of missing values (10%) and a small number of variables ($p = 5$).

- Results on the MDM objective were reported in Chapter 8.

It was established that approximately a third of the missing CLPs had to be used as the number of medoids to separate the missing CLPs with the pam clustering technique. A silhouette coefficient in excess of 0.6 was found to suggest a MAR MDM.

The real application in Chapter 9 emulated the typical data analysis steps when working with multivariate categorical data containing missing observations. It is advised to start with the single active data coding followed by investigating the sMCA biplot of the missing subset. After establishing the MDM, a suitable missing data technique can be applied.

10.2 Future work

This research project opens up several topics for future work:

- The code developed for the execution of all methodology in this research, as presented in the Appendix, needs to be made available as a proper `R` package. This will enable data practitioners to implement suitable exploratory analyses for multivariate categorical data with missing observations.
- Only nominal scaled multivariate categorical data were considered in this research. The methodology should be extended for further application on ordinal scaled multivariate categorical data as well as continuous data.
- The application of the Burt matrix approach as opposed to using the indicator matrix could be considered.
- The difference between the distribution of the simulated data before and after deletion was not investigated. It could be useful to evaluate the effect of the structure of the missing data on the success of missing data techniques. Currently, only the effect of the initial distribution of the data was evaluated in this study.
- The effect of the number of MIs on the success of the GPAbin method could be investigated.
- Variations of the GPAbin method could be explored by investigating for example the difference in performance when (1) aligning both the sample coordinates and the CLPs simultaneously in the GPA algorithm and (2) aligning the sample coordinates in the GPA algorithm and then transforming the CLPs according to the final transformations.
- Currently, all available dimensions are used in the visualisations of the GPAbin algorithm in this study. A criterion should be developed to determine the optimal number of dimensions to use to combine MCA configurations. The MI of large data sets is computationally intensive and therefore reducing the dimension of the data sets for the alignment in GPA could be beneficial.

- The performance of the GPAbin method when using different MI techniques could be compared.
- The GPAbin method has been extended in a pilot study to compositional data by aligning log-ratio biplots as opposed to MCA biplots after MI. The preliminary results are promising and will be evaluated in an imminent research project.
- In order to extend the evaluation of visualisation techniques, additional measures of comparison should be developed and considered.
- Optimisation of code and incremental SVD can be considered to reduce the computational burden of the MDM identification when using multiple active data handling.

10.3 Impact of this research

The simulation study covered a variety of data distributions, sample sizes, number of variables and percentages of missingness that aid as guidelines for expected outcomes when confronted with a multivariate categorical data set comprising of missing observations.

It was found that there was a notable difference between the clustering structures of missing CLPs of sMCA biplots. Conditions were defined to assist in the identification of a particular MDM.

In certain circumstances, as discussed in Chapter 6, the non-technical practitioner has the option of utilising the sMCA technique as opposed to MI approaches.

The GPAbin approach enables the exploration of a single configuration that captures the between-imputation variation from MI and provides an unbiased visual representation when compared to the original simulated configuration.

The aim and study objectives as outlined in Chapter 1 were successfully achieved. This research delivers a tool pack for the unbiased visualisation of incomplete multivariate categorical data sets that will equip the modern data analyst with essential skills to deal with missing observations.

Reference list

Abdi, H. 2007. Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In N.J. Salkind (ed.) *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): SAGE. 907-912.

Agresti, A. 1990. *Categorical Data Analysis*. New York: John Wiley & Sons, Inc.

Agresti, A. 2002. *Categorical Data Analysis*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Agresti, A. 2007. *An Introduction to Categorical Data Analysis*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Agresti, A. 2013. *Categorical Data Analysis*. 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Akande, O., Li, F. & Reiter, J. 2017. An Empirical Comparison of Multiple Imputation Methods for Categorical Data. *The American Statistician*. 71(2):162–170.

Allison, P.D. 2002. *Missing data*. Thousand Oaks (CA): SAGE.

Alves, M.R. 2012. Evaluation of the predictive power of biplot axes to automate the construction and layout of biplots based on the accuracy of direct readings from common outputs of multivariate analyses: 1. application to principal component analysis. *Journal of Chemometrics*. 26(5):180–190.

Ambler, G., Omar, R.Z. & Royston, P. 2007. A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome. *Statistical Methods in Medical Research*. 16(3):277–298.

Arnold, G.M., Gower, J.C., Gardner-Lubbe, S. & Le Roux, N.J. 2007. Biplots of free-choice profile data in generalized orthogonal Procrustes analysis. *Journal of the Royal Statistical Society. Series C: Applied Statistics*. 56(4):445–458.

Audigier, V., Husson, F. & Josse, J. 2017. MIMCA: multiple imputation for categorical variables with multiple correspondence analysis. *Statistics and Computing*. 27(2):501–518.

Autonne, L. 1902. Sur les groupes linéaires, réels et orthogonaux. *Bulletin de la Société Mathématique de France*. 30:121–134.

Autonne, L. 1913. Sur les matrices hypohermitiennes et les unitaires. *Comptes Rendus de l'Académie des Sciences*. 156:858–860.

Bartlett, M.S. 1935. Contingency Table Interactions. *Supplement to the Journal of the Royal Statistical Society*. 2(2):248–252.

Beh, E.J. & Lombardo, R. 2012. A genealogy of correspondence analysis. *Australian and New Zealand Journal of Statistics*. 54(2):137–168.

Beh, E.J. & Lombardo, R. 2014. *Correspondence Analysis: Theory, Practice and New Strategies*. West Sussex, United Kingdom: John Wiley & Sons, Ltd.

Benzécri, J. 1969. Statistical analysis as a tool to make patterns emerge from data. In S. Watanabe (ed.) *Methodologies of Pattern Recognition*. New York: Academic Press. 35-74.

Benzécri, J. 1973. *L'Analyse des Données. Tome 1: La Taxinomie. Tome 2: L'Analyse des Correspondances*. Paris: Dunod.

Biemer, P.P. & Lyberg, L.E. 2003. *Introduction to Survey Quality*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Bishop, Y.M.M., Fienberg, S.E. & Holland, P.W. 1975. *Discrete Multivariate Analysis: Theory and Practice*. Cambridge, Massachusetts: The MIT Press.

Blasius, J. & Greenacre, M. 2006. Correspondence Analysis and Related Methods in Practice. In M. Greenacre & J. Blasius (eds.) *Multiple Correspondence Analysis and Related Methods*. Boca Raton: Chapman & Hall/CRC. 3-40.

Blasius, J. & Thiessen, V. 2012. *Assessing the Quality of Survey Data*. London: SAGE Publications Ltd.

Blasius, J., Eilers, P.H.C. & Gower, J. 2009. Better biplots. *Computational Statistics and Data Analysis*. 53(8):3145–3158.

Bodner, T.E. 2008. What Improves with Increased Missing Data Imputations? *Structural Equation Modeling: A Multidisciplinary Journal*. 15(4):651–675.

Borg, I. & Groenen, P.J.F. 2005. *Modern Multidimensional Scaling*. 2nd ed. New York: Springer.

Browne, E.T. 1930. The characteristic roots of a matrix. *Bulletin of the American Mathematical*

Society. 36(10):705–710.

Buhi, E.R., Goodson, P. & Neilands, T.B. 2008. Out of Sight, Not Out of Mind: Strategies for Handling Missing Data. *American Journal of Health Behavior*. 32(1):83–92.

Burt, C. 1950. The factorial analysis of qualitative data. *British Journal of Mathematical and Statistical Psychology*. 3(3):166–185.

Cochran, W.G. 1954. Some Methods for Strengthening the Common χ2 Tests. *Biometrics*. 10(4):417–451.

Collins, L.M., Schafer, J.L. & Kam, C. 2001. A Comparison of Inclusive and Restrictive Strategies in Modern Missing Data Procedures. *Psychological Methods*. 6(4):330–351.

Cox, T.F. & Cox, M.A.A. 2001. *Multidimensional Scaling*. 2nd ed. Boca Raton: Chapman & Hall/CRC.

Darroch, J.N. 1962. Interactions in Multi-Factor Contingency Tables. *Journal of the Royal Statistical Society. Series B*. 24(1):251–263.

Doove, L.L., Van Buuren, S. & Dusseldorp, E. 2014. Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational Statistics and Data Analysis*. 72:92–104.

Dramalidis, A. & Markos, A. 2016. Subset Multiple Correspondence Analysis as a Tool for Visualizing Affiliation Networks. *Journal of Data Analysis and Information Processing*. 4:81–89.

Eaton, C., Plaisant, C. & Drizd, T. 2005. Visualizing Missing Data: Graph Interpretation User Study. In M.F. Costabile & F. Paternò (eds.). Rome, Italy *Human-Computer Interaction - INTERACT 2005. Lecture Notes in Computer Science, vol 3585.* Springer, Berlin, Heidelberg. 861-872.

Eckart, C. & Young, G. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*. 1(3):211–218.

Eckart, C. & Young, G. 1939. A principal axis transformation for non-Hermitian matrices. *Bulletin of the American Mathematical Society*. 45(12):118–121.

Fernstad, S.J. 2019. To identify what is not there: A definition of missingness patterns and evaluation of missing value visualization. *Information Visualization*. 18(2):230–250.

Fielding, S., Fayers, P.M. & Ramsay, C.R. 2009. Investigating the missing data mechanism in quality of life outcomes: a comparison of approaches. *Health and Quality of Life Outcomes*. 7:57.

Fisher, R.A. 1922. On the Interpretation of χ2 from Contingency Tables. *Journal of the Royal Statistical Society*. 85(1):87–94.

Fisher, R.A. 1934. *Statistical Methods for Research Methods*. Edinburgh: Oliver and Boyd.

Fisher, R.A. 1940. The precision of discriminant functions. *Annals of Eugenics*. 10(1):422–429.

Di Franco, G. 2016. Multiple correspondence analysis: one only or several techniques? *Quality and Quantity*. 50(3):1299–1315.

Friendly, M. 2008. A Brief History of Data Visualization. In C. Chen, W. Härdle, & A. Unwin (eds.) *Handbook of Data Visualization*. Berlin: Springer Verlag Berlin Heidelberg. 15-56.

Friendly, M. & Denis, D. 2005. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*. 41(2):103–130.

Gabriel, K.R. 1971. The biplot graphic display of matrices with application to principal component analysis. *Biometrika*. 58(3):453–467.

García-Laencina, P.J., Sancho-Gómez, J. & Figueiras-Vidal, A.R. 2010. Pattern classification with missing data: a review. *Neural Computing and Applications*. 19(2):263–282.

Gardner, S. 2001. Extensions of biplot methodology to discriminant analysis with applications of non-parametric principal components. Doctoral dissertation. Stellenbosch University.

Goodman, L.A. 1964. Interactions in Multidimensional Contingency Tables. *The Annals of Mathematical Statistics*. 35(2):632–646.

Goodman, L.A. 1969. On Partitioning χ2 and Detecting Partial Association in Three-Way Contingence Tables. *Journal of the Royal Statistical Society. Series B.* 31(3):486–498.

Goodman, L.A. 1971. Partitioning of Chi-Square, Analysis of Marginal Contingency Tables, and Estimation of Expected Frequencies in Multidimensional Contingency Tables. *Journal of the American Statistical Association*. 66(334):339–344.

Goodman, L.A. 1985. The Analysis of Cross-Classified Data Having Ordered and/or Unordered

Categories: Association Models , Correlation Models , and Asymmetry Models for Contingency Tables With or Without Missing Entries. *The Annals of Statistics*. 13(1):10–69.

Goodman, L.A. 1996. A single general method for the analysis of cross-classified data: Reconciliation and synthesis of some methods of Pearson, Yule, and Fisher, and also some methods of correspondence analysis and association analysis. *Journal of the American Statistical Association*. 91(433):408–427.

Goodman, L.A. 2000. The analysis of cross-classified data: Notes on a century of progress in contingency table analysis, and some comments on its prehistory and its future. In 1st ed. C.R. Rao & G.J. Székely (eds.) *Statistics for the 21st Century: Methodologies for Applications of the Future*. New York: Marcel Dekker. 189-231.

Goodman, L.A. & Kruskal, W.H. 1979. *Measures of Association for Cross Classifications*. New York: Springer-Verlag.

Gower, J.C. 2006. Divided by a Common Language: Analyzing and Visualizing Two-Way Arrays. In M. Greenacre & J. Blasius (eds.) *Multiple Correspondence Analysis and Related Methods*. Boca Raton: Chapman & Hall/CRC. 77-105.

Gower, J.C. & Dijksterhuis, G.B. 2004. *Procrustes Problems*. Oxford: Oxford University Press.

Gower, J.C. & Hand, D.J. 1996. *Biplots*. London: Chapman & Hall.

Gower, J.C. & Harding, S.A. 1988. Nonlinear biplots. *Biometrika*. 75(3):445–455.

Gower, J., Lubbe, S. & Le Roux, N. 2011. *Understanding Biplots*. West Sussex, England: John Wiley & Sons Ltd.

Graham, J.W. 2009. Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology*. 60:549–576.

Graham, J.W., Olchowski, A.E. & Gilreath, T.D. 2007. How Many Imputations are Really Needed? Some Practical Clarifications of Multiple Imputation Theory. *Prevention Science*. 8(3):206–213.

Green, P.E. & Carroll, J.D. 1976. *Mathematical Tools for Applied Multivariate Analysis*. New York: Academic Press.

Greenacre, M. 2010. *Biplots in Practice*. Fundación BBVA.

Greenacre, M. 2013. The contributions of rare objects in correspondence analysis. *Ecology*. 94(1):241–249.

Greenacre, M. 2017. *Correspondence Analysis in Practice*. 3rd ed. Boca Raton: Chapman & Hall/CRC Interdisciplinary Statistics Series.

Greenacre, M.J. 1984. *Theory and Applications of Correspondence Analysis*. London: Academic Press Harcourt Brace Jovanovich.

Greenacre, M. & Pardo, R. 2006a. Subset Correspondence Analysis: Visualizing Relationships Among a Selected Set of Response Categories From a Questionnaire Survey. *Sociological Methods & Research*. 35(2):193–218.

Greenacre, M. & Pardo, R. 2006b. Multiple Correspondence Analysis of Subsets of Response Categories. In M. Greenacre & J. Blasius (eds.) *Multiple Correspondence Analysis and Related Methods*. Boca Raton: Chapman & Hall/CRC. 197-217.

Guttman, L. 1941. The quantification of a class of attributes. In P. Horst (ed.). New York *The Prediction of Personal Adjustment*. Social Science Research Council: 321-347.

Hayashi, C. 1950. On the quantification of qualitative data from the mathematico-statistical point of view. *Annals of the Institute of Statistical Mathematics*. 2(1):35–47.

Hayashi, C. 1951. On the prediction of phenomena from qualitative data and the quantification of qualitative data from the mathematico-statistical point of view. *Annals of the Institute of Statistical Mathematics*. 3(2):69–98.

Hayashi, C. 1953. Multidimensional quantification. *Annals of the Institute of Statistical Mathematics*. 5(2):121–143.

Hendry, G., North, D., Zewotir, T. & Naidoo, R.N. 2014. The application of subset correspondence analysis to address the problem of missing data in a study on asthma severity in childhood. *Statistics in Medicine*. 33(22):3882–3893.

Hirschfeld, H.O. 1935. A Connection between Correlation and Contingency. *Mathematical Proceddings of the Cambridge Philosophical Society*. 31(4):520–524.

Honaker, J., King, G. & Blackwell, M. 2011. Amelia II: A Program for Missing Data. *Journal of*

Statistical Software. 45(7):1–47.

Horst, P. 1935. Measuring Complex Attitudes. *The Journal of Social Psychology*. 6(3):369–374.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika*. 28(3–4):321–377.

Hron, K., Templ, M. & Filzmoser, P. 2010. Imputation of missing values for compositional data using classical and robust methods. *Computational Statistics and Data Analysis*. 54(12):3095–3107.

Husson, F., Lê, S. & Pagès, J. 2011. *Exploratory Multivariate Analysis by Example Using R*. Boca Raton: Chapman & Hall/CRC Computer Science and Data Analysis Series.

Ientilucci, E.J. 2003. Using the Singular Value Decomposition. *Rochester Institute of Technology, New York*. (Technical Report).

Iodice D'Enza, A. & Markos, A. 2015. Low-dimensional tracking of association structures in categorical data. *Statistics and Computing*. 25(5):1009–1022.

Jamshidian, M. & Jalal, S. 2010. Tests of homoscedasticity, normality, and missing completely at random for incomplete multivariate data. *Psychometrika*. 75(4):649–674.

Josse, J. & Husson, F. 2012. Handling missing values in exploratory multivariate data analysis methods. *Journal de la Société Française de Statistique*. 153(2):79–99.

Josse, J. & Husson, F. 2016. missMDA: A Package for Handling Missing Values in Multivariate Data Analysis. *Journal of Statistical Software*. 70(1):1–31.

Josse, J. & Reiter, J.P. 2018. Introduction to the Special Section on Missing Data. *Statistical Science*. 33(2):139–141.

Josse, J., Chavent, M., Liquet, B. & Husson, F. 2012. Handling Missing Values with Regularized Iterative Multiple Correspondence Analysis. *Journal of Classification*. 29(1):91–116.

Kaufman, L. & Rousseeuw, P.J. 1987. Clustering by means of medoids. In Y. Dodge (ed.) *Statistical Data Analysis Based on the L1 Norm and Related Methods*. Amsterdam: North-Holland. 405-416.

Keim, D.A. 2002. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*. 7(1):100–107.

Klema, V.C. & Laub, A.J. 1980. The Singular Value Decomposition: Its Computation and Some

Applications. *IEEE Transactions on Automatic Control*. 25(2):164–176.

Kowarik, A. & Templ, M. 2016. Imputation with R Package VIM. *Journal of Statistical Software*. 74(7):1–16.

Kurtz, A.K. & Edgerton, H.A. 1939. *Statistical Dictionary of Terms and Symbols*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Lall, R. 2016. How Multiple Imputation Makes a Difference. *Political Analysis*. 24(4):414–433.

Le Roux, B. & Rouanet, H. 2004. *Geometric Data Analysis: From Correspondence Analysis to Structured Data Analysis*. Dordrecht, Kluwer: Springer Science and Business Media, Inc.

Little, R.J.A. 1988. A Test of Missing Completely at Random for Multivariate Data With Missing Values. *Journal of the American Statistical Association*. 83(404):1198–1202.

Little, R.J.A. & Rubin, D.B. 2002. *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. & Hornik, K. 2017. cluster: Cluster Analysis Basics and Extensions. *https://cran.r-project.org/*. R package(version 2.0.6).

Mantel, N. & Haenszel, W. 1959. Statistical Aspects of the Analysis of Data From Retrospective Studies of Disease. *Journal of the National Cancer Institute*. 22(4):719–748.

Martin, A.D., Quinn, K.M. & Park, J. 2011. MCMCpack: Markov Chain Monte Carlo in R. *Journal of Statistical Software*. 42(9):1–21.

Michailidis, G. & De Leeuw, J. 1998. The Gifi System of Descriptive Multivariate Analysis. *Statistical Science*. 13(4):307–336.

Microsoft Corporation & Weston, S. 2017. doSNOW: Foreach Parallel Adaptor for the “snow” Package. *https://cran.r-project.org/*. R package(version 1.0.16).

Mitsuhiro, M. & Yadohisa, H. 2015. Reduced k-means clustering with MCA in a low-dimensional space. *Computational Statistics*. 30(2):463–475.

Murphy, K.P. 2012. *Machine Learning A Probabilistic Perspective*. Cambridge, Massachusetts: The MIT Press.

Nenadić, O. & Greenacre, M. 2007. Correspondence Analysis in R, with Two- and Three-dimensional Graphics: The ca Package. *Journal of Statistical Software*. 20(3):1–13.

Nishisato, S. 1980. *Analysis of categorical data: Dual scaling and its applications*. Toronto: University of Toronto Press.

Nordholt, E.S. 1998. Imputation: Methods, simulation Experiments and Practical Examples. *International Statistical Review*. 66(2):157–180.

Ohmann, J.L., Gregory, M.J., Henderson, E.B. & Roberts, H.M. 2011. Mapping gradients of community composition with nearest-neighbour imputation: extending plot data for landscape analysis. *Journal of Vegetation Science*. 22(4):660–676.

Pearson, K. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*. 2(11):559–572.

Pearson, K. 1904. Mathematical Contributions to the Theory of Evolution. XIII. On the Theory of Contingency and its Relation to Association and Normal Correlation. *Drapers' Company Research Memoirs: Biometric Series*. I.

Penn, D.A. 2007. Estimating Missing Values from the General Social Survey: An Application of Multiple Imputation. *Social Science Quarterly*. 88(2):573–584.

Plackett, R.L. 1962. A Note on Interactions in Contingency Tables. *Journal of the Royal Statistical Society. Series B*. 24(1):162–166.

Poor, D.D.S. & Wherry, R.J. 1976. Invariance of multidimensional configurations. *British Journal of Mathematical and Statistical Psychology*. 29(1):114–125.

R Core Team. 2017. R: A Language and Environment of Statistical Computing. *R Foundation for Statistical Computing*. [Online], Available: https://www.r-project.org/.

Raghunathan, T.E., Lepkowski, J.M., Van Hoewyk, J. & Solenberger, P. 2001. A Multivariate Technique for Multiply Imputing Missing Values Using a Sequence of Regression Models. *Survey Methodology*. 27(1):85–95.

Rencher, A.C. 2002. *Methods of Multivariate Analysis*. 2nd ed. New York: John Wiley & Sons, Inc.

Richardson, M.W. & Kuder, G.F. 1933. Making a Rating Scale that Measures. *Personnel Journal*. 12:36–40.

Roy, S.N. & Kastenbaum, M.A. 1956. On the Hypothesis of No "Interaction" In a Multi-way

Contingency Table. *The Annals of Mathematical Statistics*. 27(3):749–757.

Royston, P. 2004. Multiple imputation of missing values. *The Stata Journal*. 4(3):227–241.

Rubin, D.B. 1976. Inference and missing data. *Biometrika*. 63(3):581–592.

Rubin, D.B. 1987. *Multiple Imputation for Nonresponse in Surveys*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Rubin, D.B. 1996. Multiple Imputation After 18+ Years. *Journal of the American Statistical Association*. 91(434):473–489.

Rubin, D.B. 2003. Discussion on Multiple Imputation. *International Statistical Review*. 71(3):619–625.

Rubin, D.B. & Schenker, N. 1986. Multiple Imputation for Interval Estimation From Simple Random Samples With Ignorable Nonresponse. *Journal of the American Statistical Association*. 81(394):366–374.

Schafer, J.L. 1997. *Analysis of Incomplete Multivariate Data*. United States of America: Chapman & Hall/CRC.

Schafer, J.L. 2003. Multiple Imputation in Multivariate Problems When the Imputation and Analysis Models Differ. *Statistica Neerlandica*. 57(1):19–35.

Schafer, J.L. & Graham, J.W. 2002. Missing Data: Our View of the State of the Art. *Psychological Methods*. 7(2):147–177.

Schafer, J.L. & Olsen, M.K. 1998. Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective. *Multivariate Behavioral Research*. 33(4):545–571.

Sibson, R. 1978. Studies in the Robustness of Multidimensional Scaling: Procrustes Statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*. 40(2):234–238.

Stewart, G.W. 1993. On the Early History of Singular Value Decomposition. *Society of Industrial and Applied Mathematics*. 35(4):551–566.

Struyf, A., Hubert, M. & Rousseeuw, P.J. 1997. Integrating robust clustering techniques in S-PLUS. *Computational Statistics & Data Analysis*. 26(1):17–37.

Tang, W., He, H. & Tu, X.M. 2012. *Applied Categorical and Count Data Analysis*. Boca Raton: Chapman & Hall/CRC Texts in Statistical Science.

Templ, M., Alfons, A. & Filzmoser, P. 2012. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*. 6(1):29–47.

Ten Berge, J.M.F. 1977. Orthogonal procrustes rotation for two or more matrices. *Psychometrika*. 42(2):267–276.

Tenenhaus, M. & Young, F.W. 1985. An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data. *Psychometrika*. 50(1):91–119.

Tucker, L.R., Koopman, R.F. & Linn, R.L. 1969. Evaluation of factor analytic research procedures by means of simulated correlation matrices. *Psychometrika*. 34(4):421–459.

Tukey, J.W. 1977. *Exploratory data analysis*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.

Unwin, A., Chen, C. & Härdle, W.K. 2008. Introduction. In C. Chen, W. Härdle, & A. Unwin (eds.) *Handbook of Data Visualization*. Berlin: Springer Verlag Berlin Heidelberg. 3-14.

Van Buuren, S. 2012. *Flexible Imputation of Missing Data*. Boca Raton: Chapman & Hall/CRC Interdisciplinary Statistics Series.

Van Buuren, S., Brand, J.P.L., Groothuis-Oudshoorn, C.G.M. & Rubin, D.B. 2006. Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*. 76(12):1049–1064.

Van de Geer, J.P. 1993. *Multivariate Analysis of Categorical Data: Theory*. California: Sage Publications, Inc.

Van der Heijden, P.G.M. & Escofier, B. 2003. Multiple Correspondence Analysis with missing data. In B. Escofier (ed.) *Analyse de Correspondances: Recherches au cœur de l'analyse des données.* Rennes: Presses Universitaire de Rennes. 152-170.

Van Ginkel, J.R. & Kroonenberg, P.M. 2014. Using Generalized Procrustes Analysis for Multiple Imputation in Principal Component Analysis. *Journal of Classification*. 31(2):242–269.

Venables, W.N. & Ripley, B.D. 2002. *Modern Applied Statistics with S*. 4th ed. Springer.

Von Hippel, P.T. 2009. How to Impute Interactions, Squares, and other Transformed Variables. *Sociological Methodology*. 39(1):265–291.

Von Hippel, P.T. 2018. How Many Imputations Do You Need? A Two-stage Calculation Using a Quadratic Rule. *Sociological Methods and Research*. (OnlineFirst):1-20. DOI: 10.1177/0049124117747303.

Wainer, H. 1990. Graphical Visions from William Playfair to John Tukey. *Statistical Science*. 5(3):340–346.

Walther, B.A. & Moore, J.L. 2005. The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature review of estimator performance. *Ecography*. 28(6):815–829.

Wang, H. & Wang, S. 2007. Visualization of the Critical Patterns of Missing Values in Classification Data. In G. Qiu, C. Leung, X. Xue, & R. Laurini (eds.). Shanghai, China *Advances in Visual Information Systems. VISUAL 2007. Lecture Notes in Computer Science, vol 4781.* Springer, Berlin, Heidelberg. 267-274.

Warnes, G.R., Bolker, B., Bonebakker, L., Gentleman, R., Liaw, W.H.A.L., Lumley, T., Maechler, M., Magnusson, A., et al. 2016. gplots: Various R Programming Tools for Plotting Data. *https://cran.r-project.org/*. R Package(version 3.0.1).

Wayman, J.C. 2003. *Multiple Imputation For Missing Data: What Is It And How Can I Use It?* Chicago: Annual Meeting of the American Educational Research Association.

White, I.R., Royston, P. & Wood, A.M. 2011. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*. 30(4):377–399.

Wilke, C.O. 2018. ggridges: Ridgeline Plots in “ggplot2”. *https://cran.r-project.org/*. R package(version 0.5.1).

Yau, N. 2013. *Data Points: Visualization That Means Something*. Indianapolis, Indiana: John Wiley & Sons, Inc.

Yule, G.U. 1900. On the Association of Attributes in Statistics. *Philosophical Transactions of the Royal Society of London. Series A*. 194:257–319.

Yule, G.U. 1906. On a Property which holds good for all groupings of a Normal Distribution of Frequency for Two Variables, with Applications to the Study of Contingency-Tables for the Inheritance of Unmeasured Qualities. *Proceedings of the Royal Society A.* 77(517):324–336.

Yule, G.U. 1912. On the Methods of Measuring Association between Two Attributes. *Journal of the Royal Statistical Society*. 75(6):579–652.

Zegers, F.E. & Ten Berge, J.M.F. 1985. A family of association coefficients for metric scales. *Psychometrika*. 50(1):17–24.

Zhang, P. 2003. Multiple Imputation: Theory and Method. *International Statistical Review*. 71(3):581–592.

Appendix

Appendices A to N consist of code written for R statistical software. The code presents original functions and in some cases changes made to existing functions to produce the methodology of this research. The functions are ready to be compiled and published as a package to ensure the use of the proposed methodology to a wider audience.

Appendices O to Q illustrates selected visualisations as noted in the relevant chapters of the dissertation.

Appendix A Simulation Code

A.1 Creating complete categorical data sets

```
mySim <- function (vars=vars.vec, samps=samps.vec, type=c("uniform", "dirich",
"normalBl"),cat.min=2,cat.max=5, seed=seed.vec, lvl=c("letter","numb"))
{
###################################################################################
###################################################################################
#Information
#This function generates a multivariate categorical data set according to various
#simulation parameters
###################################################################################
#Arguments
#"vars" number of variables
#"samps" number of samples
#"type" simulation distribution (uniform, Dirichlet or normal Block correlation)
#"cat.min" and "cat.max" number of category levels can be specified
#"seed" to ensure that the results can be reproduced, the seed is fixed
#"lvl" provides category levels in numerical or alphabetical format
###################################################################################
#Value
#"mat.cat" is a multivariate categorical data sets
###################################################################################
#Required packages
#MCMCpack and Matrix
###################################################################################
#Required functions
#Auxiliary function: upTrIt()
###################################################################################
###################################################################################
require(MCMCpack)   #Dirichlet distribution
require(Matrix)     #triangular matrix functions

vars <- vars
samps <- samps

set.seed(seed)
#generating the number of categories for each variable
cats <- floor(runif(vars, cat.min,cat.max))

listlev <-  vector("list",vars)
for(i in 1:vars)
{
       if(lvl=="letter")
       {
              listlev[[i]] <- as.factor(c(LETTERS[1:cats[i]]))
       }else
       {
            listlev[[i]]<- as.factor(c(1:cats[i]))
        }
}
if (type=="uniform")
{
mat <- matrix(0,samps,vars)
      for (i in 1:vars)
            {
            set.seed(seed)
            mat[,i] <- runif(n=samps,min=0,max=1)
            }
      mat <- mat
} else

if (type=="dirich")
{
set.seed(seed)
mat <- rdirichlet(n=samps, alpha=c(rep(1,vars))) #alpha – shape parameter set to 1
} else

if (type=="normalBl")
{
sigma.0 <- matrix(0,vars,nrow=vars)
#upTrIt function calculates the number of elements in the upper triangle of a given
#matrix
totIt <- upTrIt(vars)

sigPatIt <- upTrIt(round((2/3)*vars,0))
sigPat1 <- rep(0.4, sigPatIt)
sigPat2 <- rep(0.8,totIt- sigPatIt)
sig.seq <- c(sigPat1, sigPat2)
sigma.0[upper.tri(sigma.0, diag=F)] <- sig.seq
diag(sigma.0) <- c(rep(1,vars))
sigma.0 <- forceSymmetric(sigma.0, uplo="U")

Sigma <- t(sigma.0)%*% sigma.0 #positive definite matrix
mu <- rep(0,vars)
set.seed(seed)
mat <- mvrnorm(n=samps,mu=mu,Sigma=Sigma)
} else {return("no method provided")}

#creating CLs
#The cut function is used to allocate the levels. Therefore the distance between
#the maximum and minimum values in an array determines the break points

mat.cat <- sapply(1:ncol(mat), function(i) cut(mat[,i], cats[i],
labels=listlev[[i]]))#, dig.lab=5))
mat.cat <- as.data.frame(mat.cat)
return(mat.cat=mat.cat)
}
##################################################################################
```

A.2 Creating a MAR MDM

```
myMAR <-function (mat.cat=mat.cat, pctmiss=pctmiss, seed=seed)
{
##################################################################################
##################################################################################
#Information
#This function generates missing values based on observed responses
#Currently six conditions are set, but can be increased to relax the conditions
##################################################################################
#Arguments
#"mat.cat" a complete categorical data set
#"pctmiss" specifies the percentage of missing data (10% is given as 0.1)
#"seed" to ensure that the results can be reproduced, the seed is fixed
##################################################################################
```

```
#Value
#"mat.miss" is a multivariate categorical data set with missing values
#"TRpct" is the percentage of missing values in the data set (mat.miss)
####################################################################################
####################################################################################
compdat <- mat.cat
compdatIn <- mat.cat      #keep initial compdat to replace with deleted sample
after possibilities have been selected

vars <- ncol(compdat)
samps <- nrow(compdat)
#conditions
set1 <- compdat[,(vars/vars)]==levels(compdat[,(vars/vars)])[1:2]

set2find <- length(levels(compdat[,(vars/2)]))
set2     <-     compdat[,round((vars/2),0)]==levels(compdat[,(vars/2)])[(set2find-
1):set2find]
set3 <-
compdat[,((vars/vars)+1)]==levels(compdat[,2])[round(length(levels(compdat[,2]))/2,
0)]
set4 <- compdat[,vars]==max(levels(compdat[,vars]))
set5 <-
compdat[,((vars/vars)+2)]==levels(compdat[,((vars/vars)+2)])[round(length(levels(co
mpdat[,2]))/2,0)]
set6 <- compdat[,(vars/vars)]==levels(compdat[,vars])

#possibilities for NA based on conditions
compdat[set1,seq(from=2,to=(vars-2),by=2)]<- NA
compdat[set2,(round(((vars/2)+1),0))]<- NA
compdat[set3,seq(from=(vars-2),to=vars,by=1)]<- NA
compdat[set4,2]<- NA
compdat[set5,seq(from=2,to=vars,by=2)]<- NA
compdat[set6,vars] <- NA

miss <- is.na(compdat)
numbNA <- sum(is.na(compdat))

sampleSp <- which(miss, arr.ind = FALSE, useNames = TRUE)
size <- samps*vars*pctmiss

if (size > numbNA)
       {#if the possible missing values is less than the specified percentage,
       delete all possibilities as identified by conditions
       size <- numbNA
       }
set.seed(seed)

delSamp <- sample(sampleSp,size,replace=F)

for(i in 1:length(delSamp))
{
       if(delSamp[i]%%samps==0 && delSamp[i]%/%samps==1)
              {
              compdatIn[delSamp[i]%/%samps,1] <- NA
              }
              if     (delSamp[i]%/%samps==vars)
                     {
                     compdatIn[delSamp[i]%%samps,vars] <- NA
                     }
              else
              {
              compdatIn[delSamp[i]%%samps,(delSamp[i]%/%samps)+1] <- NA
              }
compdatIn
}
missdat <- compdatIn

#True percentage of missingness
```

```
TRpct <- sum(is.na(missdat))/(samps*vars)*100
return(list(mat.miss=missdat,TRpct=TRpct))
}
###################################################################################
```

A.3 Creating an MCAR MDM

```
myMCAR <- function (mat.cat=mat.cat, pctmiss=pctmiss, seed=seed)
{
###################################################################################
###################################################################################
#Information
#This function generates missing values based on the MCAR MDM by selecting a random
#sample of values to delete
###################################################################################
#Arguments
#"mat.cat" a complete categorical data set
#"pctmiss" specifies the percentage of missing data (10% is given as 0.1)
#"seed" to ensure that the results can be reproduced, the seed is fixed
###################################################################################
#Value
#"mat.miss" is a multivariate categorical data set with missing values
#"TRpct" is the percentage of missing values in the data set (mat.miss)
###################################################################################
###################################################################################
compdat <- mat.cat
vars <- ncol(compdat)
samps <- nrow(compdat)

set.seed(seed)
missInd <- sample(samps*vars, size = pctmiss*samps*vars)
#ALTERNATIVE using runif()
#missInd <- round(runif(pctmiss*samps*vars,1,samps*vars))

for(i in 1:length(missInd))
{
        if(missInd[i]%%samps==0)
                {
                is.na(compdat[[missInd[i]%/%samps]]) <- samps
                }
        else
                {
                is.na(compdat[[missInd[i]%/%samps +1]]) <- missInd[i]%%samps
                }
}
missdat <- compdat

TRpct <- sum(is.na(missdat))/(samps*vars)*100
return(list(mat.miss=missdat,TRpct=TRpct))
}
###################################################################################
```

Appendix B OPA function

```
myOPA <- function(Y.in = comp.CLP, X.in = NULL, centring = TRUE)
{
###################################################################################
###################################################################################
#Information
#This function performs Orthogonal Procrustes Analysis on centred data
###################################################################################
#Arguments
#"Y.in" target configuration (complete simulated configuration when available)
#"X.in" testee configuration
#"centring" by centring the data before OPA, translation step is redundant
###################################################################################
#Value
#"X.new" is the updated testee configuration
#"ProStat" is the Procrustes Statistic
#"Res.SS" is the residual sum of squares (Gower & Dijksterhuis 2004)
#"Tot.SS" is the total sum of squares (Gower & Dijksterhuis 2004)
#"Fitted.SS" is the fitted sum of squares (Gower & Dijksterhuis 2004)
###################################################################################
###################################################################################
n.Y <- nrow(Y.in)
p.Y <- ncol(Y.in)
n.X <- nrow(X.in)
p.X <- ncol(X.in)

X.in <- as.matrix(X.in)
Y.in <- as.matrix(Y.in)

if(!centring)
       {
       X.in <- X.in
       Y.in <- Y.in
       }
else
       {
       X.in <- scale(X.in,T,F)
       #centre=T, scale=F results are similar to Cox and Cox, Gower and
       #Dijkersthuis, Borg and Groenen
       Y.in <- scale(Y.in,T,F)
       }
#transformations
C.mat <- t(Y.in)%*%X.in
svd.C <- svd(C.mat)
A.mat <- svd.C[[3]]%*%t(svd.C[[2]])
s.fact <- sum(diag(t(Y.in)%*%X.in%*%A.mat))/sum(diag(t(X.in)%*%X.in))
#Gower and Dijksterhuis P32
b.fact <- as.vector(1/n.Y * t(Y.in - s.fact * X.in %*% A.mat)%*%rep(1,n.Y))

X.new <- b.fact + s.fact*X.in%*%A.mat

Res.SS <- sum(diag(t(((s.fact*X.in%*%A.mat)-Y.in))%*%((s.fact*X.in%*%A.mat)-Y.in)))
Tot.SS <- s.fact^2*sum(diag(t(X.in)%*%X.in))+sum(diag(t(Y.in)%*%Y.in))
Fitted.SS <- 2*s.fact*sum(diag(svd.C[[1]]))
PS <- Res.SS/sum(diag(t(Y.in)%*%Y.in))

return(list(X.new=X.new, ProStat=PS, Res.SS=Res.SS, Tot.SS=Tot.SS,
Fitted.SS=Fitted.SS))
}
###################################################################################
```

Appendix C Fit measures

C.1 Measures of comparison

```
fitMeas <- function (Target=comp , Testee=imp, dim=c("All", "2D"))
{
###################################################################################
###################################################################################
#Information
#This function calculates the following measures of comparison between two
#configurations
#PS, CC, RMSB, MB, AMB
###################################################################################
#Arguments
#"Target" is the target configuration
#"Testee" is the testee configuration
#"dim" to compare the configurations in 2D or the maximum available ("All") dims
###################################################################################
#Value
#Returns a data frame with the measures of comparison as stated above.
###################################################################################
###################################################################################
counter <- 0
Target <- as.matrix(Target)
Testee <- as.matrix(Testee)

nCLTar <- nrow(Target)
nCLTes <- nrow(Testee)

Tarnam <- rownames(Target)
Tesnam <- rownames(Testee)

#finding the CLs that occur in both Target and Testee and deleting the CLs that do
#not appear in both in order to obtain a one-to-one comparison
rem <- which(is.na(match(Tarnam,Tesnam)))
if(is.integer0(rem))
{
Target[[i]] <- Target[[i]]
counter <- counter+1       #counts the matched cases
} else {Target[[i]]<- Target[[i]][-rem,]}

if(dim=="All")
       {
       pY <- ncol(Target)
       pX <- ncol(Testee)
       #the maximum number of common columns to use
       colUse <- min(pY,pX)
       Target <- Target[,1:colUse]
       Testee <- Testee[,1:colUse]
       } else
if (dim=="2D")
       {
       Target <- Target[,1:2]
       Testee <- Testee[,1:2]
       }

       OPA <- myOPA (Y.in=Target,X.in=Testee,centring=TRUE)
       PS <- OPA[[1]]
       CC <- sum(dist(Testee) * dist(Target))/(sqrt(sum(dist(Testee)^2)) *
       sqrt(sum(dist(Target)^2)))
       RMSB <- ((sum(sum((Target-Testee)^2)))/length(Testee))^(0.5)
       MB <- (sum(sum((Target-Testee)^1)))/length(Testee)
       AMB <- (sum(sum(abs(Target-Testee))))/length(Testee)

       FitSS <- OPA[[4]]
       ResSS <- OPA[[2]]
       TotSS <- OPA[[3]]
       #presenting measures of comparison in a table
       REStable <- data.frame(c(FitSS, ResSS, TotSS, PS, CC, RMSB, MB, AMB))
       colnames(REStable)<- c("Measure of fit")
       rownames(REStable)<- c("Fitted", "Residual", "Total", "PS", "CC", "RMSB",
       "MB", "AMB")
       return(REStable)
}
###################################################################################
```

C.2 Similarity percentage

```
sim.pct <- function(comp.ls, meth.ls)
{
###################################################################################
###################################################################################
#Information
#This function calculates the similarity percentage by comparing the category
#levels of the data sets in two lists
####################################################################################
#Arguments
#"comp.ls" is the list of the complete categorical data sets
#"meth.ls" is the list of the completed categorical data set for a particular
#method (GPAbin, sMCA, RIMCA)
####################################################################################
#Value
#Returns a vector containing a similarity percentage for each list element
####################################################################################
#Required functions
#myOPA()
#Auxiliary function: is.interger0()
####################################################################################
####################################################################################
JJ <- length(meth.ls)
II <- length(meth.ls[[1]])

count.mat <- matrix(0,II,JJ)
size.mat <- matrix(0,II,JJ)
counter <- 0

for (j in 1:JJ)
{
meth <- meth.ls[[j]]
comp <- comp.ls[[j]]

for (i in 1:II)
{
compnam <- colnames(comp[[i]])
methnam <- colnames(meth[[i]])
rem <- which(is.na(match(compnam, methnam)))
if(is.integer0(rem))
{
comp[[i]] <- comp[[i]]
counter <- counter+1
} else {comp[[i]]<- comp[[i]][,-rem]
}

count.mat[i,j] <-
sum(mapply(as.character,comp[[i]])==mapply(as.character,meth[[i]]))
size.mat[i,j] <- nrow(comp[[i]])*ncol(comp[[i]])
}
}
pct.mat <- count.mat/size.mat*100
pct.overall <- colMeans(pct.mat)

return(pct.overall)
}
####################################################################################
```

Appendix D GPA function

```
GPA <- function(Xk, G.target=NULL, iter=500, eps=0.001)
{
####################################################################################
####################################################################################
#Information
#This function contains the OPA function to compare two configurations and the GPA
#function for multiple configuration comparisons
#The original function was published as supplementary material to:
#(Arnold, Gower, Gardner-Lubbe & Le Roux, 2007)
####################################################################################
#Arguments
#"Xk" argument is a list containing the testee configurations which is updated on
#each iteration
#"G.target" argument is the target configuration. If not specified the centroid
#configuration will be used as the target
#"iter" is the number of iterations allowed before convergence
#"eps" is the threshold value for convergence of the alogrithm
######################################################################################
#Value
#"Xk.F" is a list containing the updated testee configurations
#"sk.F" is a vector containing the final scaling factors
#"Qk.F" is a list containing the final rotation matrices
#"Gmat" is the final target configuration
#"sum.sq" is the final minimised sum of squared distance
######################################################################################
######################################################################################
OPA <- function(X.mat, Z.mat)
{
svd.zx <- svd(t(Z.mat) %*% X.mat)
svd.zx[[3]] %*% t(svd.zx[[2]])
}
K <- length(Xk)
n <- nrow(Xk[[1]])
p <- ncol(Xk[[1]])
means <- t(sapply(Xk, function(X)apply(X, 2, mean)))
Xk.scale <- sapply (Xk, scale, simplify=F)
Xk.F <- sapply (1:K, function(k, Xk) as.matrix (Xk[[k]]), Xk=Xk.scale, simplify=F)
Qk <- sapply(1:K, function(k, p) return(diag(p)), p=p, simplify = F)
sk <- rep(1, K)
sk.F <- sk
Qk.F <- sapply(1:K, function(k, Qk) Qk[[k]], Qk = Qk, simplify = F)
tel <- 0
sum.sq.old <- Inf

repeat
       {tel <- tel + 1
       if(tel > iter)
       stop(paste("Maximum number of specified iterations reached! Increase iter
\n",II))
       Xk.F <- sapply(1:K, function(k, Xk, sk, Qk) sk[k] * Xk[[k]] %*% Qk[[k]], Xk
= Xk.F, sk = sk, Qk = Qk, simplify = F)
if (is.null(G.target))
{Gmat <- Xk.F[[1]]
for(k in 2:K)  Gmat <- Gmat + Xk.F[[k]]
Gmat <- Gmat/K
}
else Gmat <- G.target
Qk <- sapply(1:K, function(k, Xind, Gmat) OPA(Xind[[k]], Gmat), Xind = Xk.F, Gmat =
Gmat, simplify = F)
Qk.F <- sapply(1:K, function(k, Qk, QF) QF[[k]] %*% Qk[[k]], Qk = Qk, QF = Qk.F,
simplify = F)
Smat <- matrix(0, ncol = K, nrow = K)
for(i in 1:K)
for(j in i:K) {
Smat[i, j] <- sum(diag(t(Qk[[i]]) %*% t(Xk.F[[i]]) %*% Xk.F[[j]] %*% Qk[[j]]))
if(i != j) Smat[j, i] <- Smat[i, j]
}
Smat.min.half <- diag(1/sqrt(diag(Smat)))
swd <- svd(Smat.min.half %*% Smat %*% Smat.min.half)
sk <- Smat.min.half %*% swd[[2]][, 1] * sqrt(K)
sk.factor <- sum(sk)/K
sk <- sk / sk.factor
if(sk[1] < 0) sk <- -1 * sk

sum.sq <- sum(sapply(1:K, function(k, Xk, Gmat)sum(diag((Xk[[k]] - Gmat) %*%
t(Xk[[k]] - Gmat))), Xk = Xk.F, Gmat = Gmat))
cat("iter", tel, "sum.sq: ", sum.sq, "\n")
if((sum.sq.old - sum.sq) < eps) break
sum.sq.old <- sum.sq
}
list(Xk.F=Xk.F, sk.F=sk.F, Qk.F=Qk.F, Gmat=Gmat, sum.sq=sum.sq)
}
######################################################################################
```

Appendix E GPAbin

```
GPAbin <- function(CLP.list,Z.list,G.target=NULL)
{
####################################################################################
####################################################################################
#Information
#This function combines multiple configurations obtained from the output of the
#MI.impute() function
####################################################################################
#Arguments
#"CLP.list" argument is the list contains the CLPs of the multiple imputations
#"Z.list" argument is the list contains the sample points of the multiple
#imputations
#"G.target" argument is the target configuration. If not specified the centroid
#configuration will be used as the target
####################################################################################
#Value
#"Z.GPAbin" is the sample coordinates for the GPAbin biplot
#"CLP.GPAbin" is the CLPs for the GPAbin biplot
#"Z.GPA.list" is a list containing the sample coordinates for each MI after GPA
#"CLP.GPA.list" is a list containing the CLPs for each MI after GPA
####################################################################################
#Required functions
#GPA()
####################################################################################
####################################################################################
M <- length(CLP.list)
GPA.out <- GPA(CLP.list,G.target=NULL)
G.target <- GPA.out[[4]]
Q.list <- GPA.out[[3]]
s.list <- GPA.out[[2]]
CLP.GPA.list <- GPA.out[[1]]

Z.scal <- vector("list",M)
for (m in 1:M)
{
Z.scal[[m]] <- Z.list[[m]]*s.list[[m]]
}

Z.GPA.list <- vector("list",M)
for (m in 1:M)
{
Z.GPA.list[[m]] <- Z.scal[[m]]%*%Q.list[[m]]
}

Z.GPAbin <- Reduce("+",Z.GPA.list)/length(Z.GPA.list)
CLP.GPAbin <- Reduce("+",CLP.GPA.list)/length(CLP.GPA.list)

return(list(Z.GPAbin=Z.GPAbin, CLP.GPAbin=CLP.GPAbin, CLP.GPA.list=CLP.GPA.list,
Z.GPA.list=Z.GPA.list))
}
####################################################################################
```

Appendix F Functions related to RIMCA

F.1 Single imputation call functions

```
SIimpute <- function(datNA=NULL, seed=123)
{
####################################################################################
####################################################################################
#Information
#This function contains the function calls for SI using the RIMCA algorithm
####################################################################################
#Arguments
#"datNA" is a categorical data set with missing data entries
#"seed" fixes the random seed in order to replicate results
###################################################################################
#Value
#"SIset" is the categorical data set after SI
#"CLPs" is the CLPs of the SI MCA biplot
#"Zs" is the sample coordinates of the SI MCA biplot
#"lvls" is the names of the CLs
###################################################################################
#Required packages
#ca
###################################################################################
#Required functions
#myestim_ncpMCA() and myimputeMCA()
###################################################################################
###################################################################################
require(ca)

datNA <- as.data.frame(datNA)
datNA <- FormatDat(datNA)        #formatting row and column names
set.seed(seed)
colZscal <- ncol(indmat(datNA))  #determining the number of columns in the
                                 #indicator matrix
pvar <- ncol(datNA)              #number of variables in missing data set
ncp.max <- colZscal-pvar-1       #maximum number of dimensions available for MCA
                                 #solution

#error handling of myestim_npcMCA()
#stop() has been updated in myimputeMCA() to return the proposed number of
#dimensions to retain and is used to run the function again without manual input

nd <-try(myestim_ncpMCA(datNA,method="Regularized",method.cv="Kfold",ncp.min=0,
ncp.max=ncp.max,verbose=FALSE, nbsim = 100,seed=seed)[[1]],silent=F)
if(inherits(nd,"try-error")) nd<-as.numeric(conditionMessage(attr(nd,"condition")))

set.seed(seed)

SI.output <- myimputeMCA(don=datNA,ncp=nd,method="Regularized")[[2]]

SI.out <- mjca(SI.output, lambda="indicator")
SI.CLP <- scale(SI.out[[23]])
SI.sampl <- scale(SI.out[[16]])
SI.lnames <- SI.out[[6]]

return(list(SIset=SI.output,CLPs=SI.CLP,Zs=SI.sampl,lvls=SI.lnames))
}
###################################################################################
```

F.2 Estimating the number of dimensions

```
myestim_ncpMCA <-function (don, ncp.min = 0, ncp.max = 5, method = c("Regularized",
"EM"), method.cv = c("Kfold", "loo"), nbsim = 100, pNA = 0.05, threshold = 1e-03,
verbose = TRUE,seed=seed)
{
###################################################################################
###################################################################################
#Information
#This function includes minor changes to the original estim_ncpMCA() function
#available in the missMDA R package
#Code starting and ending with #JNS# indicates changes made to original functions
###################################################################################
#Changes made to arguments
#"threshold" changed to 1e-0.3
#"seed" argument added
###################################################################################
#Value
#Returns the number of dimensions to be used
```

```
##################################################################################
##################################################################################
#JNS# addition of fixed seed
set.seed(seed)
#JNS#
tab.disjonctif.NA <- function(tab)
{
tab <- as.data.frame(tab)
modalite.disjonctif <- function(i)
{
       moda <- tab[, i]
       nom <- names(tab)[i]
       n <- length(moda)
       moda <- as.factor(moda)
       x <- matrix(0, n, length(levels(moda)))
       ind <- (1:n) + n * (unclass(moda) - 1)
       indNA <- which(is.na(ind))
       x[(1:n) + n * (unclass(moda) - 1)] <- 1
       x[indNA, ] <- NA

       if ((ncol(tab) != 1) & (levels(moda)[1] %in% c(1:nlevels(moda), "n", "N",
       "y", "Y")))
       dimnames(x) <- list(row.names(tab), paste(nom, levels(moda), sep = "."))
       else dimnames(x) <- list(row.names(tab), levels(moda))
       return(x)
}
       if (ncol(tab) == 1)
              res <- modalite.disjonctif(1)
       else   {
              res <- lapply(1:ncol(tab), modalite.disjonctif)
              res <- as.matrix(data.frame(res, check.names = FALSE))
               }
        return(res)
}

prodna <- function(x, noNA,seed=NULL)
{
#JNS# addition of fixed seed
       set.seed(seed)
#JNS#
       n <- nrow(x)
       p <- ncol(x)
       NAloc <- rep(FALSE, n * p)
       NAloc[sample(n * p, floor(n * p * noNA))] <- TRUE
       x[matrix(NAloc, nrow = n, ncol = p)] <- NA
       return(x)
}
       method <- match.arg(method, c("Regularized", "regularized", "EM", "em"),
       several.ok = T)[1]
       method.cv <- match.arg(method.cv, c("loo", "Kfold", "kfold", "LOO"),
       several.ok = T)[1]
       method <- tolower(method)
       method.cv <- tolower(method.cv)
       auxi = NULL
       don <- droplevels(don)

       for (j in 1:ncol(don)) if (is.numeric(don[, j]))
       auxi = c(auxi, colnames(don)[j])

       if (!is.null(auxi))
       stop(paste("\nAll variables are not categorical, the following ones are
       numeric: ", auxi))
       vrai.tab = tab.disjonctif.NA(don)

       if (method.cv == "kfold")
       {
              res = matrix(NA, ncp.max - ncp.min + 1, nbsim)
```

```
             if (verbose)
                    pb <- txtProgressBar(min = 1/nbsim * 100, max = 100, style = 3)
                    for (sim in 1:nbsim)
                    {
                           continue <- TRUE
                           while (continue) {
                           donNA <- prodna(don, pNA,seed)
                           continue <- (sum(unlist(sapply(as.data.frame(donNA),
                           nlevels))) != sum(unlist(sapply(don, nlevels))))
                           }
#JNS#
#Omit for loop, factor levels are ordered incorrectly
#for (i in 1:ncol(don)) donNA[, i] = as.factor(as.character(donNA[,i]))
#JNS#
             for (nbaxes in ncp.min:ncp.max)
             {
             tab.disj.comp <- myimputeMCA(as.data.frame(donNA), ncp = nbaxes,
             method = method, threshold = threshold)[[1]]
             if (sum(is.na(donNA)) != sum(is.na(don)))
                    {
                    res[nbaxes - ncp.min + 1, sim] <- sum((tab.disj.comp -
vrai.tab)^2, na.rm = TRUE)/(sum(is.na(tab.disjonctif.NA(donNA))) -
sum(is.na(tab.disjonctif.NA(don))))
                    }  else
                    {
#JNS#
#no alternative provided for cases where the sum of missing values are equal in don
#and donNA res is a matrix of NAs, therefore else statement was added.

donNA <- prodna(don, pNA,seed+1)
continue <- (sum(unlist(sapply(as.data.frame(donNA), nlevels))) !=
sum(unlist(sapply(don, nlevels))))
#JNS#

#JNS#
#Omit for loop, since factor levels are ordered incorrectly
#for (i in 1:ncol(don)) donNA[, i] = as.factor(as.character(donNA[,i]))
#JNS#
             for (nbaxes in ncp.min:ncp.max)
             {
             tab.disj.comp <- myimputeMCA(as.data.frame(donNA), ncp = nbaxes,
             method = method, threshold = threshold)[[1]]
             if (sum(is.na(donNA)) != sum(is.na(don)))
                    {
                    res[nbaxes - ncp.min + 1, sim] <- sum((tab.disj.comp -
vrai.tab)^2, na.rm = TRUE)/(sum(is.na(tab.disjonctif.NA(donNA))) -
sum(is.na(tab.disjonctif.NA(don))))
                    }
             }
             }
       }
            if (verbose)
            setTxtProgressBar(pb, sim/nbsim * 100)
    }

        if (verbose)
        close(pb)
        crit = apply(res, 1, mean, na.rm = TRUE)

        names(crit) <- c(ncp.min:ncp.max)
        result = list(ncp = as.integer(which.min(crit) + ncp.min - 1),
        criterion = crit)
        return(result)
    }
#JNS#
#code omitted for “loo” method in appendix, since not of interest for this study
}
##################################################################################
```

F.3 Adapted RIMCA function

```
myimputeMCA <- function (don, ncp = 2, method = c("Regularized", "EM"), row.w =
NULL, coeff.ridge = 1, threshold = 1e-03, seed = NULL, maxiter = 1000)
{
##################################################################################
##################################################################################
#Information
#This function includes minor changes to the original imputeMCA () function
#available in the missMDA R package
#Code starting and ending with #JNS# indicates changes made to original functions
##################################################################################
#Value
#"tab.disj" is the indicator matrix with fuzzy values for the missing values
#according to the proportion of response category levels per variable
#"completeObs" is the final categorical data set after SI
##################################################################################
#Changes made to arguments
#"threshold" changed to 1e-0.3
##################################################################################
#Required packages
#FactoMineR
##################################################################################
##################################################################################
moy.p <- function(V, poids)
{
res <- sum(V * poids, na.rm = TRUE)/sum(poids[!is.na(V)])
}

find.category <- function(X, tabdisj)
{
nbdummy <- rep(1, ncol(X))
is.quali <- which(!unlist(lapply(X, is.numeric)))

#JNS#
#To only consider the available observed category levels
for (i in is.quali)
{
       X[,i] <- droplevels(X[,i],exclude=NA)
}
#JNS#
nbdummy[is.quali] <- unlist(lapply(X[, is.quali, drop = FALSE], nlevels))
vec = c(0, cumsum(nbdummy))
Xres <- X

for (i in is.quali)
{
#JNS#
#Original function did not foresee variables with only one category level observed.
#This could occur when the percentage of missing values increase
       if (length((vec[i] + 1):vec[i + 1])==1)
       {
              temp <- as.factor(levels(X[,i]))
       } else
       {
#JNS#
              temp <- as.factor(levels(X[, i])[apply(tabdisj[,(vec[i] + 1):vec[i +
              1]], 1, which.max)])
       }
              Xres[, i] <- factor(temp, levels(X[, is.quali][,i]))
        }
        return(Xres)
}

tab.disjonctif.NA <- function(tab)
{
       tab <- as.data.frame(tab)
       modalite.disjonctif <- function(i)
       {
       moda <- tab[, i]
       nom <- names(tab)[i]
       n <- length(moda)
       moda <- as.factor(moda)
       x <- matrix(0, n, length(levels(moda)))
       ind <- (1:n) + n * (unclass(moda) - 1)
       indNA <- which(is.na(ind))
       x[(1:n) + n * (unclass(moda) - 1)] <- 1
       x[indNA, ] <- NA
       if ((ncol(tab) != 1) & (levels(moda)[1] %in% c(1:nlevels(moda),
       "n", "N", "y", "Y")))
       dimnames(x) <- list(row.names(tab), paste(nom, levels(moda), sep = "."))
       else dimnames(x) <- list(row.names(tab), levels(moda))
       return(x)
       }
       if (ncol(tab) == 1)
       res <- modalite.disjonctif(1)
       else  {
              res <- lapply(1:ncol(tab), modalite.disjonctif)
              res <- as.matrix(data.frame(res, check.names = FALSE))
              }
       return(res)
}
method <- match.arg(method, c("Regularized", "regularized", "EM", "em"),
several.ok = T)[1]
method <- tolower(method)
don <- droplevels(don)
if (is.null(row.w))
row.w <- rep(1/nrow(don), nrow(don))
if (ncp == 0)
return(list(tab.disj = tab.disjonctif.prop(don, NULL, row.w = row.w), completeObs =
find.category(don, tab.disjonctif.prop(don, NULL, row.w = row.w))))
tab.disj.NA <- tab.disjonctif.NA(don)
hidden <- which(is.na(tab.disj.NA))
tab.disj.comp <- tab.disjonctif.prop(don, seed, row.w = row.w)
tab.disj.rec.old <- tab.disj.comp
continue <- TRUE
nbiter <- 0
while (continue) {
       nbiter <- nbiter + 1
       M <- apply(tab.disj.comp, 2, moy.p, row.w)/ncol(don)
       if (any(M < 0))
#JNS#
#stop statement updated to only provide the proposed number of dimensions
#this enables the use of call function to run the function again with the proposed
#number of dimensions without requiring manual feedback

stop(ncp-1)

#stop(paste("The algorithm fails to converge. Choose a number of components (ncp)
#less or equal than " ,ncp - 1, " or a number of iterations (maxiter) less or equal
#than ",maxiter - 1, sep = ""))
#JNS#
       Z <- t(t(tab.disj.comp)/apply(tab.disj.comp, 2, moy.p, row.w))
       Z <- t(t(Z) - apply(Z, 2, moy.p, row.w))
       Zscale <- t(t(Z) * sqrt(M))
       svd.Zscale <- svd.triplet(Zscale, row.w = row.w, ncp = ncp)
       moyeig <- 0
       if (nrow(don) > (ncol(Zscale) - ncol(don)))
       moyeig <- mean(svd.Zscale[[1]][-c(1:ncp, (ncol(Zscale)-ncol(don) +
       1):ncol(Zscale))]^2)
       else moyeig <- mean(svd.Zscale[[1]][-c(1:ncp)]^2)
       moyeig <- min(moyeig * coeff.ridge, svd.Zscale[[1]][ncp + 1]^2)
        if (method == "em")
       moyeig <- 0
       eig.shrunk <- ((svd.Zscale[[1]][1:ncp]^2 - moyeig)/svd.Zscale[[1]][1:ncp])
```

```
        if (ncp == 1)
        rec <- tcrossprod(svd.Zscale[[2]][, 1] * eig.shrunk, svd.Zscale[[3]][, 1])
        else rec <- tcrossprod(t(t(svd.Zscale[[2]][, 1:ncp, drop = FALSE]) *
        eig.shrunk), svd.Zscale[[3]][, 1:ncp, drop = FALSE])
        tab.disj.rec <- t(t(rec)/sqrt(M)) + matrix(1, nrow(rec), ncol(rec))
        tab.disj.rec <- t(t(tab.disj.rec) * apply(tab.disj.comp, 2, moy.p, row.w))
        diff <- tab.disj.rec - tab.disj.rec.old
        diff[hidden] <- 0
        relch <- sum(diff^2 * row.w)
        tab.disj.rec.old <- tab.disj.rec
        tab.disj.comp[hidden] <- tab.disj.rec[hidden]
        continue = (relch > threshold) & (nbiter < maxiter)
    }

    tab <- find.category(don, tab.disj.comp)

    return(list(tab.disj = tab.disj.comp, completeObs = tab))
}
#################################################################################
```

Appendix G Functions related to MIMCA

G.1 Multiple imputation call functions

```
MIimpute <- function(datNA=NULL, seed=123, imps=10)
{
#################################################################################
#################################################################################
#Information
#This function contains the function calls for MI using the MIMCA algorithm and
#performing MCA on the MIs
#################################################################################
#Arguments
#"datNA" is a categorical data set with missing data entries
#"seed" fixes the random seed in order to replicate results
#"imps" specifies the number of multiple imputations
#################################################################################
#Value
#"CLP.list" is a list containing the CLPs of the MCA biplots of the MI data sets
#"Z.list" is a list containing the sample coordinates of the MCA biplots of the MI
#data sets
#"datNA" is the input missing data
#"Imp.list" is a list containing the MI categorical data sets
#################################################################################
#Required packages
#ca and FactoMineR
#################################################################################
#Required functions
#myestim_ncpMCA() and myimputeMCA()
#Auxiliary functions: FormatDat(), indmat(), indcol(), FormatImpList(), rmOneCL()
#################################################################################
#################################################################################
require(ca)
require(FactoMineR)

datNA <- as.data.frame(datNA)
datNA <- FormatDat(datNA)         #formatting row and column names
set.seed(seed)
colZscal <- ncol(indmat(datNA))   #determining the number of columns in the
                                  #indicator matrix
pvar <- ncol(datNA)               #number of variables in missing data set
ncp.max <- colZscal-pvar-1        #maximum number of dimensions available for MCA
                                  #solution
m <- imps
#error handling of myestim_npcMCA()
#stop() has been updated in myimputeMCA() to return the proposed number of
#dimensions to retain and is used to run the function again without manual input

nd <-try(myestim_ncpMCA(datNA,method="Regularized",method.cv="Kfold",ncp.min=0,
ncp.max=ncp.max,verbose=FALSE, nbsim = 100,seed=seed)[[1]],silent=T)
if(inherits(nd,"try-error")) nd<-as.numeric(conditionMessage(attr(nd,"condition")))

set.seed(seed)

if(is.na(nd))
{
#change seed if nd cannot be estimated
nd <- try(myestim_ncpMCA(datNA,method="Regularized",method.cv="Kfold",ncp.min=0,
ncp.max=ncp.max,verbose=FALSE, nbsim = 100,seed=(saad+12345))[[1]],silent=F)
if(inherits(nd,"try-error")) nd<-as.numeric(conditionMessage(attr(nd,"condition")))
}
set.seed(seed)

MI.output <- try(myMIMCA(datNA, ncp=nd, nboot=m, verbose=FALSE, seed=seed),
silent=F)
if(inherits(MI.output,"try-error")) MI.output <- myMIMCA(datNA,
ncp=(as.numeric(conditionMessage(attr(MI.output,"condition")))),nboot=m, verbose=F,
seed=seed)

set.seed(seed)

Imp.list <- MI.output[[1]]              #list of imputed data sets
Imp.list <- FormatImpList(Imp.list)     #preparing colnames and rownames
Imp.list <- rmOneCL(Imp.list)           #preparation for MCA

Z.list <- vector("list",m)
CLP.list <- vector("list",m)

for (imp in 1:m)
{
dat <- Imp.list[[imp]]
out <- mjca(dat,lambda="indicator")
nam <- out[[6]]
Z.list[[imp]] <- out[[16]]
CLPs <- out[[23]]
rownames(CLPs) <- nam
CLP.list[[imp]] <- CLPs
}
return(list(CLP.list=CLP.list,Z.list=Z.list,datNA=datNA, Imp.list=Imp.list))
}
###################################################################################
```

G.2 Adapted MIMCA function

```
myMIMCA <- function (X, nboot = 10, ncp, coeff.ridge = 1, threshold = 1e-03,
maxiter = 1000, verbose = FALSE, seed=NULL)
{
###################################################################################
###################################################################################
#Information
#This function includes minor changes to the original MIMCA () function available
#in the missMDA R package
#Code starting and ending with #JNS# indicates changes made to original functions
###################################################################################
#Changes made to arguments
#"threshold" changed to 1e-0.3
###################################################################################
#Value
#"X.imp" is a list containing the MI categorical data sets
#"tab.disj" is the indicator matrix with fuzzy values for the missing values
#according to the proportion of response category levels per variable
#A list of of the input arguments are given:
```

```
#"X", "nboot", "ncp", "coeff.ridge", "threshold", "seed", "maxiter", "tab.disj"
##################################################################################
#Required packages
#FactoMineR
##################################################################################
#Required functions
#myimputeMCA() function
##################################################################################
##################################################################################
imputeMCA.print <- function(don, ncp, method = c("Regularized", "EM"), row.w =
NULL, coeff.ridge = 1, threshold = 1e-03, seed = NULL, maxiter = 1000, verbose,
printm)
       {

if (verbose)
{
cat(paste(printm, "...", sep = ""))
}
res <- myimputeMCA(don = don, ncp = ncp, method = method, row.w = row.w,
coeff.ridge = coeff.ridge, threshold = threshold, seed = seed, maxiter = maxiter)

return(res)
}
normtdc <- function(tab.disj, data.na) {
tdc <- tab.disj
tdc[tdc < 0] <- 0
tdc[tdc > 1] <- 1
col.suppr <- cumsum(sapply(data.na, function(x) {nlevels(x)}))
tdc <- t(apply(tdc, 1, FUN = function(x, col.suppr) {
if (sum(x[1:col.suppr[1]]) != 1) {
x[1:col.suppr[1]] <- x[1:col.suppr[1]]/sum(x[1:col.suppr[1]])
}
for (i in 2:length(col.suppr))
{
x[(col.suppr[i - 1] + 1):(col.suppr[i])] <- x[(col.suppr[i-1] + 1):(col.suppr[i])]
/sum(x[(col.suppr[i-1] + 1):col.suppr[i]])
}
return(x)
}, col.suppr = col.suppr))
return(tdc)
}

draw <- function(tabdisj, Don, Don.na) {
nbdummy <- rep(1, ncol(Don))
is.quali <- which(!unlist(lapply(Don, is.numeric)))

#JNS# edit to handle only one CL
for (i in is.quali)
{
Don[,i] <- droplevels(Don[,i],exclude=NA)
}
#JNS#
nbdummy[is.quali] <- unlist(lapply(Don[, is.quali, drop = FALSE],nlevels))
vec = c(0, cumsum(nbdummy))
Donres <- Don

#JNS#
for (i in is.quali) {
if (length((vec[i] + 1):vec[i + 1])==1)
{
temp <- as.factor(levels(Don[,i]))
} else
#JNS#

{
Donres[, i] <- as.factor(levels(Don[, i])[apply(tabdisj[,(vec[i] + 1):vec[i+1]], 1,
function(x) {
sample(1:length(x), size = 1, prob = x)
}))
}
Donres[, i] <- factor(Donres[, i], levels(Don[, is.quali][, i]))
}
return(don.imp = Donres)
}

temp <- if (coeff.ridge == 1) {
"regularized"
}
else if (coeff.ridge == 0) {"EM"}
else {
paste("coeff.ridge=", coeff.ridge)
}
if (verbose)
{
cat("Multiple Imputation using", temp, "MCA using", nboot, "imputed arrays", "\n")
}
n <- nrow(X)
Boot <- matrix(sample(1:n, size = nboot * n, replace = T), n, nboot)
Weight <- matrix(1/(n * 1000), n, nboot, dimnames = list(1:n, paste("nboot=",
1:nboot, sep = "")))
Boot.table <- apply(Boot, 2, table)
for (i in 1:nboot) {
Weight[names(Boot.table[[i]]), i] <- Boot.table[[i]]
}

Weight <- sweep(Weight, 2, STATS = colSums(Weight), FUN = "/")
Weight <- as.data.frame(Weight)

   # res.imp <- mapply(Weight, FUN = imputeMCA.print, MoreArgs = list(don = X,
   #     ncp = ncp, coeff.ridge = coeff.ridge, method = "Regularized",
   #     threshold = threshold, maxiter = maxiter, verbose = verbose),
   #     printm = as.character(1:nboot), SIMPLIFY = FALSE)

#JNS# calling myimputeMCA()
res.imp <- mapply(Weight, FUN = myimputeMCA, MoreArgs = list(don = X, ncp = ncp,
coeff.ridge = coeff.ridge, method = "Regularized", threshold = threshold, maxiter =
maxiter), SIMPLIFY = FALSE)
#JNS#

tdc.imp <- lapply(res.imp, "[[", "tab.disj")
res.comp <- lapply(res.imp, "[[", "completeObs")
tdc.norm <- mapply(FUN = normtdc, tab.disj = tdc.imp, data.na = res.comp, SIMPLIFY
= F)

X.imp <- mapply(FUN = draw, tabdisj = tdc.norm, Don = res.comp, MoreArgs =
list(Don.na = X), SIMPLIFY = F)

res <- list(res.MI = X.imp, res.imputeMCA = myimputeMCA(X, ncp = ncp, coeff.ridge =
coeff.ridge, threshold = threshold, seed = NULL, maxiter = maxiter)[[1]], call =
list(X = X, nboot = nboot, ncp = ncp, coeff.ridge = coeff.ridge, threshold =
threshold, seed = NULL, maxiter = maxiter, tab.disj = array(unlist(tdc.imp), dim =
c(nrow(tdc.imp[[1]]), ncol(tdc.imp[[1]]), length(tdc.imp)))))
class(res) <- c("MIMCA", "list")
if (verbose) {
cat("\ndone!\n")
}
return(res)
}
##################################################################################
```

Appendix H Majority rule prediction

H.1 Majority rule prediction call functions

```
MajPredCall <- function(comp.lvls, datNA, Z.list, CLP.list, seed)
{
###################################################################################
###################################################################################
#Information
#This function contains the function calls for the prediction of a final
#multivariate categorical data set from MI MCA biplots
#MCA is applied to the predicted data set
###################################################################################
#Arguments
#"comp.lvls" is the number of levels per variable in the original simulated data
#set
#"datNA" is the data set containing missing observations
#"Z.list" and "CLP.list" are lists containing the coordinate matrices of the MI MCA
#biplots
#"seed" fixes the random seed in order to replicate results
###################################################################################
#Value
#"Z.Maj" is the sample coordinates of the MCA biplot constructed from the predicted
#data set
#"CLP.Maj" is the CLPs of the MCA biplot constructed from the predicted data set
###################################################################################
#Required packages
#ca
###################################################################################
#Required functions
#MajPred() and MajRule()
#Auxiliary function: CreateOutlist()
###################################################################################
###################################################################################
require(ca)

nsamples <- nrow(Z.list[[1]])     #number of samples
imp <- length(CLP.list)           #number of imputation

catPredout <- MajPred(comp.lvls=comp.lvls,datNA=datNA,Z.list=Z.list,
CLP.list=CLP.list, nsamples=nsamples, imp=imp)
catPred <- catPredout[[1]]
pvar <- catPredout[[2]]
X.pred.out <- CreateOutlist(inputlist=catPred,imp=imp,nsamples=nsamples,pvar=pvar)
Comb.pred <-
MajRule(inlist=X.pred.out,imp=imp,nsamples=nsamples,pvar=pvar,seed=saad)
MajBipl <- mjca(Comb.pred,lambda="indicator")
Z.Maj <- MajBipl[[16]]
CLP.Maj <- MajBipl[[23]]
rownames(CLP.Maj) <- rownames(CLP.list[[1]])
rownames(Z.Maj) <- rownames(Z.list[[1]])

return(list(Z.Maj=Z.Maj,CLP.Maj=CLP.Maj))
}
###################################################################################
```

H.2 Category predictions from multiple imputed configurations

```
MajPred <- function (comp.lvls, datNA, CLP.list, Z.list, nsamples, imp)
{
###################################################################################
###################################################################################
#Information
#This function predicts category levels for each MI MCA biplot using the distances
#between samples and CLPs
###################################################################################
#Arguments
#"comp.lvls" is the number of levels per variable in the original simulated data
#set
#"datNA" is the data set containing missing observations
```

```
#"Z.list" and "CLP.list" are lists containing the coordinate matrices of the MI MCA
#biplots
#"nsamples" is the number of samples (rows) in the data sets
#"imp" is the number of MIs
####################################################################################
#Value
#"Pred.list" a list of the predicted categorical data set for each imputation
#"pvarPred" the number of variables in the particular data set
####################################################################################
#Required functions
#Auxiliary functions: is.integer0(), df2fact(), FormatDimNam()
####################################################################################
I <- nsamples
M <- imp

comp.nam <- levels(comp.lvls[,1])[comp.lvls[,1]]
comp.nams <- unique(substr(comp.nam,1,1))
NA.nams <- unique(substr(rownames(CLP.list[[1]]),1,1))
finder <- which(is.na(match(comp.nams,NA.nams)))

if (is.integer0(finder))
{
datNA <- datNA
} else
{
datNA <- datNA[,-finder]
}

J <- ncol(datNA)           #number of variables
X.pred <- as.data.frame(matrix(0,I,J))
X.pred <- df2fact(X.pred)
Pred.list <- vector("list",M)

vec.lvl <- vector("numeric",J)
for (j in 1:J)
{
vec.lvl[j] <- length(levels(delCL(datNA)[,j]))
}
cumlvl <- cumsum(vec.lvl)

for (m in 1:M)
{
for (i in 1:I)
{
#distance matrix between first row of Z.list[[i]][1,] and all rows of CLP.list[[i]]
#(distances for first sample over all variables)
d <- as.matrix(as.matrix(dist(rbind(Z.list[[m]][i,],CLP.list[[m]])))[1,-1])

pluggin <- matrix(0,J,1)
inds <- matrix(0,J,1)
for(j in 1:J)
       {
              if(j==1)
              {
                     if (length(j:cumlvl[j])==1)
                     {
                     inds[j] <- 1
                     } else
              {
              pluggin[j] <- min(d[j:cumlvl[j]])
              inds[j] <- which(d[j:cumlvl[j]]==min(d[j:cumlvl[j]]),arr.ind=TRUE)
              }
              } else
              {
                     if (length((cumlvl[j-1]+1):cumlvl[j])==1)
                     {
                     inds[j] <- 1
                     }else
```

```
                    {
                    pluggin[j] <- min(d[(cumlvl[j-1]+1):cumlvl[j]])
                    inds[j] <- which(d[(cumlvl[j-1]+1):cumlvl[j]]==min(d[(cumlvl[j-
1]+1):cumlvl[j]]),arr.ind=TRUE)
                    }
              }
       levels(X.pred[,j]) <- levels(delCL(datNA)[,j])
       place <- inds[j]
       X.pred[i,j] <- levels(X.pred[,j])[place]
       }
}
inds.NNA <- which(!is.na(datNA), arr.ind=T)
X.pred[inds.NNA] <- datNA[inds.NNA]#replaces non missing with original categories
X.pred <- FormatDimNam(X.pred)

Pred.list[[m]] <- X.pred
if(M==1)
{Pred.list <- as.data.frame(Pred.list[[m]])}
}
return(list(Pred.list=Pred.list,pvarPred=J))
}
####################################################################################
```

H.3 Assigning the final category levels of multiple predicted data sets

```
MajRule <- function (inlist, imp, nsamples, pvar, seed)
{
####################################################################################
####################################################################################
#Information
#This function allocates the final category levels from the multiple predicted data
#sets obtained from the MajPred() function.
####################################################################################
#Arguments
#"inlist" is the output list of the CreateOutlist function
#"imp" is the number of MIs
#"nsamples" is the number of samples (rows) in the data sets
#"pvar" is the number of variables (columns) in the data set
#"seed" fixes the random seed in order to replicate results
####################################################################################
#Value
#Returns a multivariate categorical data set with predicted category levels
####################################################################################
#Required functions
#Auxiliary function: df2fact()
####################################################################################
####################################################################################
predFin <- as.data.frame(matrix(0,nsamples,pvar))
predFin <- df2fact(predFin)

dimlist <- length(inlist)

for(i in 1:dimlist)
{
for (j in 1:pvar)
{
useDF <- inlist[[i]][,j]
unDF <- unique(useDF)
levels(predFin[,j])<- levels(inlist[[i]][,j])

if (length(unDF)==1)
{
out <- unDF
} else
{
maxim <- tabulate(match(useDF, unDF))
#if there is more than one mode, select a category randomly
#check whether first two maxim elements are equal to determine whether there is
#more than one mode
if(all.equal(maxim[1],maxim[2])==TRUE)
{ set.seed(seed)
select <- sample(1:length(maxim),1)
} else
{
select <- which.max(maxim)
}
out <- unDF[select]
}
predFin[i,j] <- out
}
}
return(predFin)
}

delCL <- function(data)
{
for (j in 1:ncol(data))
        {
        col <- data[,j]
        nl <- levels(col)

                for (i in 1:length(nl))
                {
                col <- droplevels(col,exclude=NA)
                }
        data[,j] <- col
        }
return(data)
}
####################################################################################
```

Appendix I GPAbin prediction

I.1 GPAbin prediction call functions

```
GPAbPredCall <- function(comp.lvls, datNA, CLP.GPA, Z.GPA, seed)
{
####################################################################################
####################################################################################
#Information
#This function contains the function calls for the prediction of a final
#multivariate categorical data set from a GPAbin biplot
####################################################################################
#Arguments
#"comp.lvls" the number of levels per variable in the original simulated data set
#"datNA" is the data set containing missing observations
#"CLP.GPA" and" Z.GPA" contains the coordinate matrices of the GPAbin biplot
#"seed" fixes the random seed in order to replicate results
####################################################################################
#Value
#"Z.GPApred" the sample coordinates of the MCA biplot constructed from the
#predicted data set
#"CLP.GPApred" the CLPs of the MCA biplot constructed from the predicted data set
####################################################################################
#Required packages
#ca
####################################################################################
#Required functions
#GPAPred()
####################################################################################
####################################################################################
require(ca)
nsamples <- nrow(Z.GPA)    #number of samples
pvar <- ncol(datNA)        #number of variables

GPAbinPred <- GPAPred(comp.lvls=comp.lvls,datNA=datNA, CLP.GPA=CLP.GPA,Z.GPA=Z.GPA,
nsamples=nsamples, pvar=pvar)
GPApredBipl <- mjca(GPAbinPred,lambda="indicator")
Z.GPApred <- GPApredBipl[[16]]
CLP.GPApred <- GPApredBipl[[23]]
rownames(CLP.GPApred) <- rownames(CLP.GPA)
rownames(Z.GPApred) <- rownames(Z.GPA)

return(list(Z.GPApred=Z.GPApred,CLP.GPApred=CLP.GPApred))
}
###################################################################################
```

I.2 GPAbin prediction function

```
GPAPred <- function (comp.lvls, datNA, CLP.GPA, Z.GPA, nsamples, pvar)
{
###################################################################################
###################################################################################
#Information
#This function predicts category levels for GPAbin biplot using the distances
#between samples and CLPs
###################################################################################
#Arguments
#"comp.lvls" is the number of levels per variable in the original simulated data
#set
#"datNA" is the data set containing missing observations
#"CLP.GPA" and "Z.GPA" contains the coordinate matrices of the GPAbin biplot
#"nsamples" is the number of samples (rows) in the data sets
#"pvar" is the number of variables (columns) in the data set
###################################################################################
#Value
#"Pred" a final predicted categorical data set
###################################################################################
#Required functions
#Auxiliary functions: is.integer0(), df2fact(), delCL(), FormatDimNam()
###################################################################################
###################################################################################
I <- nsamples       #number of samples

comp.nam <-  levels(comp.lvls[,1])[comp.lvls[,1]]
comp.nams <- unique(substr(comp.nam,1,1))
NA.nams <- unique(substr(rownames(CLP.GPA),1,1))
finder <- which(is.na(match(comp.nams,NA.nams)))

if (is.integer0(finder))
{
datNA <- datNA
} else
{
datNA <- datNA[,-finder]
}

J <- ncol(datNA)
X.pred <- as.data.frame(matrix(0,I,J))
X.pred <- df2fact(X.pred)

vec.lvl <- vector("numeric",J)
for (j in 1:J)
{
vec.lvl[j] <- length(levels(delCL(datNA)[,j]))
}
cumlvl <- cumsum(vec.lvl)

for (i in 1:I)
{
d <- as.matrix(as.matrix(dist(rbind(Z.GPA[i,],CLP.GPA)))[1,-1])#distance matrix
between first row of Z.list[[i]][1,] and all rows of CLP.list[[i]] (distances for
first sample over all variables)
pluggin <- matrix(0,J,1)
inds <- matrix(0,J,1)
for(j in 1:J)
{
       if(j==1)
       {
              if(length(j:cumlvl[j])==1)
              {inds[j] <- 1
              } else
                     {
                     pluggin[j] <- min(d[j:cumlvl[j]])
                     inds[j] <-
which(d[j:cumlvl[j]]==min(d[j:cumlvl[j]]),arr.ind=TRUE)
                     }
       }else
       {
              if(length((cumlvl[j-1]+1):cumlvl[j])==1)
                     {inds[j] <- 1
              }else
                     {pluggin[j] <- min(d[(cumlvl[j-1]+1):cumlvl[j]])
                     inds[j] <- which(d[(cumlvl[j-1]+1):cumlvl[j]]==min(d[(cumlvl[j-
1]+1):cumlvl[j]]),arr.ind=TRUE)
                     }
              }
              levels(X.pred[,j]) <- levels(delCL(datNA)[,j])
              place <- inds[j]
              X.pred[i,j] <- levels(X.pred[,j])[place]
       }
}
inds.NNA <- which(!is.na(datNA), arr.ind=T)
X.pred[inds.NNA] <- datNA[inds.NNA]#replaces non missing with original categories
X.pred <- FormatDimNam(X.pred)

return(Pred=X.pred)
}
##################################################################################
```

Appendix J sMCA

J.1 sMCA call functions

```
mysMCA <- function(datNA=NULL,seed=seed.vec, lambda="indicator", method=c("single",
"multiple"))
{
##################################################################################
##################################################################################
#Information
#This function performs sMCA and returns the coordinates for the sMCA missing and
#observed subsets
#This is not a plotting function
#This function requires CLPna(), df2fact(), delCL()
##################################################################################
#Arguments
#"datNA" is the data set containing missing observations
#"seed" fixes the random seed in order to replicate results
#"lambda" specifies the MCA approach using the indicator matrix
#"method" specified whether single or multiple active handling should be applied
##################################################################################
#Value
#"emptCLPs" CLPs for the sMCA biplot of the missing susbet
#"emptZs" sample coordiates for the sMCA biplot of the missing subset
#"emptlvl" names of the category levels for the missing subset
```

```
#"obsCLPs" CLPs for the sMCA biplot of the observed susbet
#"obsZs" sample coordiates for the sMCA biplot of the observed subset
#"obslvl" names of the category levels for the observed subset
###################################################################################
#Required packages
#ca
###################################################################################
#Required functions
#adap.mjca()
#Auxiliary function: CLPna()
###################################################################################
require(ca)

set.seed(seed)

data.Xclp <- CLPna(datNA, method=method)
p <- ncol(datNA)    #number of variables

datNA <- delCL(datNA)

numb.org <- vector("numeric", ncol(datNA))

for (i in 1: ncol(datNA))
{
numb.org[i] <- length(levels(datNA[,i]))
}

mca.na.out <- mjca(data.Xclp, reti=T, lambda=lambda)
numb.na <- mca.na.out[[8]]
indiNA <- mca.na.out[[33]]
indiCol <- ncol(indiNA)

if (method=="single")
{
sub.vec <- matrix(0,1,p)   #empty row vector for number of variables
for (i in 1:p)
{
       if(numb.na[i]-numb.org[i]==1) #determine if an empty CL was created
              {
              sub.vec[i] <- cumsum(numb.na)[i]
              }
       else
              {
              sub.vec[i] <- 0
              }
}
sub.vec <- (sub.vec[sub.vec!=0])
}
#multiple
else
{
#create a list with number of items equal to number of variables
sub.vec <- vector("list", p)
       if(numb.na[1]!=numb.org[1])
              {
              sub.vec[[1]] <- c((cumsum(numb.org)[1]+1):(cumsum(numb.na)[1]))
              }
       else
              {
              sub.vec[[1]] <- NULL
              }

for (i in 2:p)
       {
       if(numb.na[i]!=numb.org[i])
              {
              sub.vec[[i]] <- c(((numb.org[i]+1)+cumsum(numb.na)[i-
              1]):cumsum(numb.na)[i])
```

```
             }
       else
             {
             sub.vec[[i]] <- NULL
             }
       }
sub.vec <- unlist(sub.vec)
}

out.empt <- adap.mjca(data.Xclp,lambda=lambda,subsetcat=(1:indiCol)[sub.vec],
reti=F)

emptCLPs <- out.empt[[23]]
emptZs <- out.empt[[16]]
emptlvl <- out.empt[[6]]

out.obs <- adap.mjca(data.Xclp,lambda=lambda,subsetcat=(1:indiCol)[-sub.vec],
reti=F)

obsCLPs <- out.obs[[23]]
obsZs <- out.obs[[16]]
obslvl <- out.obs[[6]]

return(list(emptCLPs=emptCLPs, emptZs=emptZs, emptlvl=emptlvl ,obsCLPs=obsCLPs,
obsZs=obsZs, obslvl=obslvl))
}
#################################################################################
```

J.2 Adapted mjca() function

Only the two changes made to the original `mjca()` function are shown.

Line 189 is replaced by the following:

```
evd.S <- eigen(S[subsetcol, subsetcol,drop=FALSE])
#JNS# added drop statement to maintain matrix structure
```

Additional code added after Line 192:

```
evd.S[[1]][evd.S[[1]] < 0] <- 0
#JNS# eigenvalues close to zero (very small negative values) are set to zero

subinr <- function (B, ind)
#################################################################################
#Available in the ca package, but is required when using the adapted version of
#mjca()
#This function computes the inertia of sub-matrices
#################################################################################
{
    nn <- length(ind)
    subi <- matrix(NA, nrow = nn, ncol = nn)
    ind2 <- c(0, cumsum(ind))
    for (i in 1:nn) {
        for (j in 1:nn) {
            tempmat <- B[(ind2[i] + 1):(ind2[i + 1]), (ind2[j] +
                1):(ind2[j + 1])]
            tempmat <- tempmat/sum(tempmat)
            er <- apply(tempmat, 1, sum)
            ec <- apply(tempmat, 2, sum)
            ex <- er %*% t(ec)
            subi[i, j] <- sum((tempmat - ex)^2/ex)
        }
    }
    return(subi/nn^2)
}
```

```
###################################################################################
```

Appendix K Cluster analysis

```
myClust <- function(clustdat= emptCLPs,seed=seed)
{
###################################################################################
###################################################################################
#Information
#This function obtains the average silhouette widths for 2 to (max(CLP)-1) clusters
#using the pam() functio
###################################################################################
#Arguments
#"clustdat" is the CLPs of the empty subset obtained from mysMCA()
#"seed" fixes the random seed in order to replicate results
###################################################################################
#Value
#Returns a vector of silhouette coefficients for each number of specified clusters
###################################################################################
#Required packages
#cluster
###################################################################################
###################################################################################
require(cluster)

nlvl <- nrow(clustdat)
lvl.vec <- c(2:(nlvl-1))
K <- length(lvl.vec)
silout <- vector("list", K)

for (k in 1:K)
{
       set.seed(seed)
       clustPam <- pam(clustdat[,1:2],k=lvl.vec[k],diss=FALSE)
       silInfo <- clustPam$silinfo
       silout[[k]] <- silInfo$avg.width
}
return(silout)
}
###################################################################################
```

Appendix L Auxiliary functions

L.1 Removing empty factor levels

```
delCL <- function(data)
{
###################################################################################
###################################################################################
#Information
#This function removes empty factor levels
###################################################################################
#Arguments
#"data" is an object of the factor class
###################################################################################
#Value
#"Returns the input object, now with empty category levels removed
###################################################################################
###################################################################################
for (j in 1:ncol(data))
        {
        col <- data[,j]
        nl <- levels(col)

                for (i in 1:length(nl))
                {
                col <- droplevels(col,exclude=NA)
                }
        data[,j] <- col
        }
return(data)
}
##################################################################################
```

L.2 Removing variables with one category level

```
rmOneCL <- function(inlist)
{
##################################################################################
##################################################################################
#Information
#This function removes columns with only one CL before applying MCA
##################################################################################
#Arguments
#"inlist" list of factor objects
##################################################################################
#Value
#Returns the input list, now with removed columns
##################################################################################
##################################################################################
M <- length(inlist)

for(m in 1:M)
{
        dat <- inlist[[m]]
        pvar <- ncol(dat)
        vecel <- vector("numeric",pvar)
        for (i in 1:pvar)
        {
                if(length(levels(dat[,i]))==1)
                vecel[i] <- i
        }
if(sum(vecel)==0)
{
inlist[[m]] <- dat
} else
{
        vecel <- vecel[vecel != 0]
        inlist[[m]] <- dat[,-vecel]
}
}
return(inlist)
}
##################################################################################
```

L.3 Creating missing category levels

```
CLPna <- function(data=data.fact, method=c("single","multiple"))
{
##################################################################################
##################################################################################
#Information
#This function creates additional category levels for missing values
##################################################################################
#Arguments
#"data" is a factor object
#"method" specifies whether single or multiple active handling should be applied
##################################################################################
#Value
#Returns the input object, now with additional missing category levels
##################################################################################
#Required functions
#Auxiliary functions: df2fact(), delCL()
####################################################################################
####################################################################################
data <- df2fact(data)

#creating a CLP for missing values
if (method=="single")
{
for (j in 1:ncol(data))
{
       {
       #creating new levels
       levels(data[[j]]) <- c(levels(data[[j]]),"NA")
       data[[j]][is.na(data[[j]])] <- "NA"
       }
}
}
else
{
for (i in 1:nrow(data))
       {
       for (j in 1:ncol(data))
              {
              #creating new levels
              if (is.na(data[i,j]))
              {
              levels(data[[j]]) <- c(levels(data[[j]]),paste(rownames(data)[i],
              "NA",colnames(data)[j],sep=""))
              data[i,j][is.na(data[i,j])] <- paste(rownames(data)[i],"NA",
              colnames(data)[j],sep="")
              }
              }
       }
}
data <- delCL(data)
return(data)
}
####################################################################################
```

L.4 Formatting of a dataframe to a factor

```
df2fact <- function (in.df, ordered=FALSE)
{
####################################################################################
####################################################################################
#Information
#This function transforms each column of a data.frame into unordered factor
#variables (ordered=F) or ordered factor variables (ordered=T)
####################################################################################
#Arugments
#"in.df" a data.frame containing factors
#"ordered" if TRUE (T) factor levels are ordered, if FALSE (F) factor levels are
#unordered
####################################################################################
#Value
#Returns a transformed data.frame according to "ordered" argument.
####################################################################################
####################################################################################
out.df <- factor(in.df[,1],ordered=ordered)

for(i in 2:ncol(in.df)) out.df <- data.frame(out.df, factor(in.df[,i],
ordered=ordered))

colnames(out.df) <- colnames(in.df)
rownames(out.df) <- rownames(in.df)
return(out.df)
}
###################################################################################
```

L.5 Detecting zero length integers

```
is.integer0 <- function(x)
{
###################################################################################
###################################################################################
#Information
#This function detects integers with zero length
#This function is used as a precaution to eliminate possible errors
###################################################################################
#Argument
#"x" scalar value
###################################################################################
#Value
#Returns a TRUE or FALSE
###################################################################################
###################################################################################
is.integer(x) && length(x) == 0L
}
###################################################################################
```

L.6 Constructing an indicator matrix

```
indcol <- function(col.vec)
{
###################################################################################
###################################################################################
#Information
#This function constructs the dummy variables for the columns of an indicator
#matrix per variable
#This function is used in combination with the indmat() function
###################################################################################
#Arguments
#"col.vec" is a particular column from a multivariate categorical data set
###################################################################################
#Value
#Returns the dummy coded variables for a particular column from a multivariate
#categorical data set
###################################################################################
###################################################################################
elements <- levels(factor(col.vec))
Y <- matrix(0, nrow = length(col.vec), ncol = length(elements))
dimnames(Y) <- list(NULL, paste(elements))
for(i in 1:length(elements))
       {
       Y[col.vec == elements[i], i] <- 1
       }
return(Y)
}
###################################################################################

indmat <- function(dat)
{
###################################################################################
###################################################################################
#Information
#This function constructs an indicator matrix
###################################################################################
#Arguments
#"dat" is a multivariate categorical data set
###################################################################################
#Value
#Returns an indicator matrix
###################################################################################
#Required functions
#Auxiliary function: indcol()
###################################################################################
###################################################################################
cols <- ncol(dat)
samps <- nrow(dat)
out <- matrix(1,samps,1)
for (k in 1:cols)
       {
       ncol <- indcol(dat[,k])
       out <- cbind(out,ncol)
       }
out <- out[,-1]
return(out)
}
###################################################################################
```

L.7 Determining the number of elements in the upper triangle of a matrix

```
upTrIt <- function (vars=vars)
{
###################################################################################
###################################################################################
#Information
#This function calculates the number of elements in the upper triangle of a square
#matrix
###################################################################################
#Arguments
#"vars" the number of columns (or rows) of a square matrix
###################################################################################
#Value
#Returns the number of elements in the upper triangle of a square matrix
###################################################################################
###################################################################################

items <- 0      #number of items in upper triangle

for (i in 1:vars)
{
items <- (vars-i) + items
}
return(items)
}
###################################################################################
```

L.8 Formatting data

```
FormatDat <- function (datNA)
{
###################################################################################
###################################################################################
#Information
#This function is used to change the names of rows, columns and factor levels for
#consistent display of results
###################################################################################
#Arguments
#"datNA" a multivariate categorical data set containing missing values
###################################################################################
#Value
#Returns a formatted "datNA"
###################################################################################
###################################################################################
nn <- nrow(datNA)
pp <- ncol(datNA)
let.vec <- c(LETTERS[1:8],LETTERS[10:26])      #excluding I
```

```
lvl.vec <- c("one", "two", "thr","fou", "fiv", "six", "sev","eig", "nin",
"ten","ele","twe","thi","frt","fivt")

for (i in 1:nn) {rownames(datNA) <- paste('s',1:nn,sep='')}
for (j in 1:pp) {colnames(datNA) <- let.vec[1:pp]
numb <- length(levels(datNA[,j]))
levels(datNA[,j]) <-lvl.vec[1:numb]
 }
datNA
}
###################################################################################

CreateOutlist <- function(inputlist,imp,nsamples,pvar)
{
###################################################################################
###################################################################################
#Information
#This function is used in combination with the prediction functions
###################################################################################
#Arugments
#"inputlist" is a list containing each imputation in the from of a dataframe of
#size n x p
#"imp" is the number of imputations
#"nsamples" is the number of samples (rows of a data matrix)
#"pvar" is the number of variables (columns of a data matrix)
###################################################################################
#Value
#Returns a list of n elements i.e. an element for each sample containing the
#predicted level for all p categorical variables over all imputations.
###################################################################################
###################################################################################
outlist <- vector('list',nsamples)
names(outlist) <- paste('PredSample',1:nsamples, sep='')

for(samp in 1:nsamples)
{
temp.df <- data.frame(inputlist[[1]][samp,])
#return(temp.df)
for(extract in 2:imp) temp.df <-
rbind(temp.df,data.frame(inputlist[[extract]][samp,]))

outlist[[samp]] <- temp.df
}
return(outlist)
}
###################################################################################

FormatDimNam <- function(dat)
{
###################################################################################
###################################################################################
#Information
#This function changes row- and column names, but not factor levels
###################################################################################
#Arguments
#"dat" a multivariate categorical data set
###################################################################################
#Value
#Returns formatted "dat"
###################################################################################
###################################################################################
nn <- nrow(dat)
pp <- ncol(dat)
let.vec <- c(LETTERS[1:8],LETTERS[10:26])       #excluding I

for (i in 1:nn) {rownames(dat) <- paste('s',1:nn,sep='')}
for (j in 1:pp)
{
```

```
colnames(dat) <- let.vec[1:pp]
}
return(dat)
}
####################################################################################

FormatImpList <- function (mylist)
{
####################################################################################
####################################################################################
#Information
#This function adds names to elements of the list containing the imputations
####################################################################################
#Arguments
#"mylist" is a list containing MIs
####################################################################################
#Value
#Returns formatted "mylist"
####################################################################################
####################################################################################
imp <- length(mylist)
names(mylist) <- paste('Imputation',1:imp,sep='.')
return(mylist)
}
####################################################################################
```

Appendix M Example of script file submitted to PBS

```
####################################################################################
#!/bin/bash
#PBS -N GPAbin_thirMA7
#PBS -l nodes=1:ppn=24
#PBS -l mem=48Gb
#PBS -l walltime=20:00:00
#PBS -M nienkemperj@sun.ac.za
#PBS -m abe

cd ${PBS_O_WORKDIR}

module load app/R/3.4.3

R --vanilla <<RSCRIPT
{
########################################
#required R packages
library(ca)
library(snow)
library(doSNOW)
library(parallel)
library(missMDA)
library(FactoMineR)
########################################

########################################
#data available on the HPC server
########################################
load("/home/nienkemperj/GPAthr/thiMAset.rdat")
load("/home/nienkemperj/GPAthr/saad.rdat")

miss.set <- thiMAset

II <- 1000

CLP.list <- vector("list",II)
Z.list <- vector("list",II)
Z.GPAbin <- vector("list",II)
CLP.GPAbin <- vector("list",II)
```

```
#########################################
#required functions should be added here
#########################################

#specify number of cores to use
cl <- makeSOCKcluster(24)
registerDoSNOW(cl)

temp.list <-
foreach(ii=1:II,.packages=c("missMDA","ca","FactoMineR"),.errorhandling="pass")
%dopar%
{
GPA.out <- GPAbinCall(datNA=miss.set[[7]][[ii]],imp=10,seed=saad[ii])

Z.list[[ii]]<- GPA.out[[1]]
CLP.list[[ii]]<- GPA.out[[2]]
Z.GPAbin[[ii]] <- GPA.out[[3]]
CLP.GPAbin[[ii]] <- GPA.out[[4]]

return(list(Z.list[[ii]],CLP.list[[ii]],Z.GPAbin[[ii]],CLP.GPAbin[[ii]]))
}
stopCluster(cl)

#saving multiple imputations to HPC server
mm <-10
thiMACLPmi7<- vector("list", II)
fill <- vector("list",mm)
for (ii in 1:II)
{
for (m in 1:mm)
{
fill[[m]] <- tryCatch(as.data.frame(temp.list[[ii]][[2]][[m]]), error=function(e)
print(ii))
thiMACLPmi7[[ii]]<- fill
}
}
save(thiMACLPmi7,file="/home/nienkemperj/GPAthr/thiMACLPmi7.rdat")

thiMAZmi7<- vector("list", II)
fill <- vector("list",mm)
for (ii in 1:II)
{
for (m in 1:mm)
{
fill[[m]] <- tryCatch(as.data.frame(temp.list[[ii]][[1]][[m]]),error=function(e)
print(ii))
thiMAZmi7[[ii]]<- fill
}
}
save(thiMAZmi7,file="/home/nienkemperj/GPAthr/thiMAZmi7.rdat")
#saving multiple imputations to HPC server

#saving mca output to server
thiMACLPGPb7 <- vector("list", II)
for (ii in 1:II)
{
thiMACLPGPb7[[ii]] <- tryCatch(as.data.frame(temp.list[[ii]][[4]]),
error=function(e) print(ii))
}
save(thiMACLPGPb7,file="/home/nienkemperj/GPAthr/thiMACLPGPb7.rdat")

thiMAZGPb7 <- vector("list", II)

for (ii in 1:II)
{
thiMAZGPb7[[ii]] <- tryCatch(as.data.frame(temp.list[[ii]][[3]]), error=function(e)
print(ii))
}
save(thiMAZGPb7,file="/home/nienkemperj/GPAthr/thiMAZGPb7.rdat")
}
#saving mca output to server
RSCRIPT
###################################################################################
```

Appendix N Plotting functions

N.1 MCA biplot

```
myMCAbipl <- function(data, lambda="indicator", Z.col="black", CLP.col=
"forestgreen", Z.pch=1, CLP.pch=17, main="MCA biplot")
{
###################################################################################
###################################################################################
#Information
#This function constructs an MCA biplot
###################################################################################
#Arguments
#"data" a multivariate categorical data set
#"lambda" is the type of MCA that is performed
#(e.g. "adjusted", "indicator", "Burt", "JCA")
#"Z.col" is the colour of the sample coordinates
#"CLP.col" is the colour of the CLPs
#"Z.pch" is the plotting character of the sample coordinates
#"CLP.pch" is the plotting character of the CLPs
#"main" is the title of the biplot
###################################################################################
#Required packages
#ca
###################################################################################
#Value
#Returns an MCA biplot
###################################################################################
###################################################################################
require(ca)
mca.out <- mjca(data,lambda=lambda)
#Constructing an MCA biplot

windows()
par(pty="s")
plot(rbind(mca.out$rowpcoord,mca.out$colcoord),pch="",xlab="",ylab="",xaxt="n",yaxt
="n",main=main)
points(mca.out$rowpcoord,col=Z.col,pch=Z.pch)
points(mca.out$colcoord,col=CLP.col,pch=CLP.pch)
legend("bottomright",pch=c(Z.pch,CLP.pch),col=c(Z.col,CLP.col),legend=c("Sample","C
LP"))
}
###################################################################################
```

N.2 Stacked barplots

Only the two changes made to the original `barplot()` function are shown.

Lines 121 to 128 are replaced by the following:

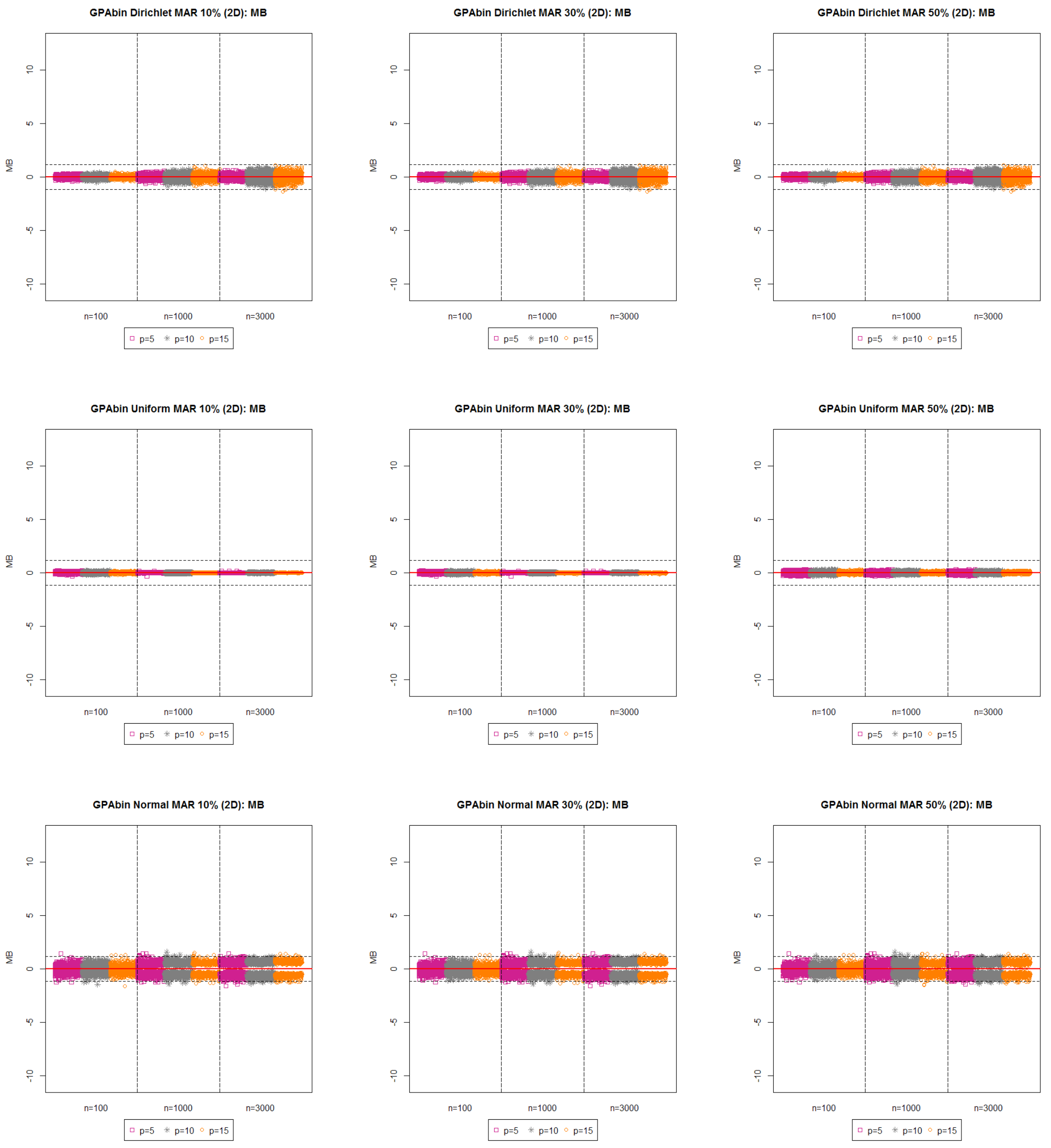
```
#JNS# changing colour and density vectors to matrices
             else {
            for (i in 1L:NC) {
                xyrect(height[1L:NR, i] + offset[i], w.l[i],
                  height[-1, i] + offset[i], w.r[i], horizontal = horiz,
                  angle = angle, density = density[1L:NR, i], col = col[1L:NR, i],
                  border = border)
                }
        }
```

Removing line 146:

```
#JNS# removes argument that woule have reversed the order of the density vector#
#                density <- rev(density)
```

Appendix O Measures of comparison: missing data approaches

O.1 Mean bias: GPAbin and RIMCA (MAR MDM)



GPAbin Dirichlet MAR 10% (2D): MB

GPAbin Dirichlet MAR 30% (2D): MB

GPAbin Dirichlet MAR 50% (2D): MB

GPAbin Uniform MAR 10% (2D): MB

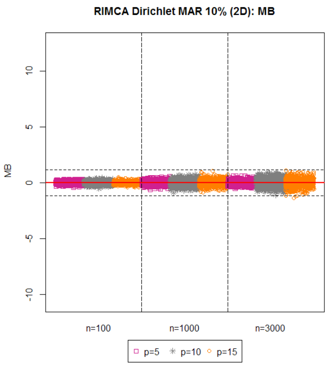
GPAbin Uniform MAR 30% (2D): MB

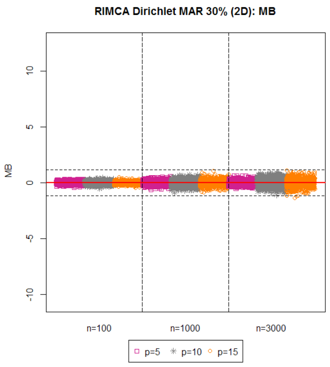
GPAbin Uniform MAR 50% (2D): MB

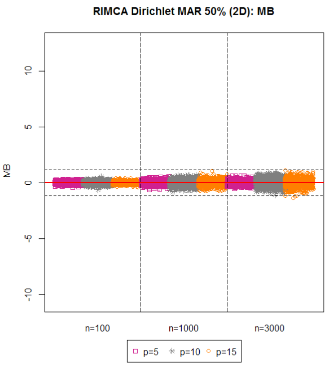
GPAbin Normal MAR 10% (2D): MB

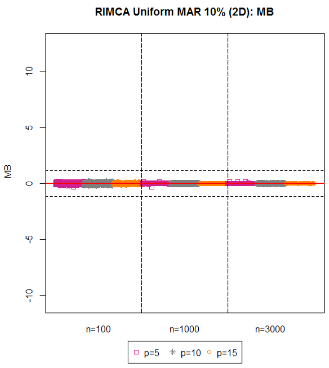
GPAbin Normal MAR 30% (2D): MB

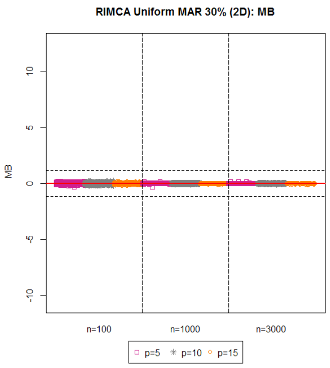
GPAbin Normal MAR 50% (2D): MB

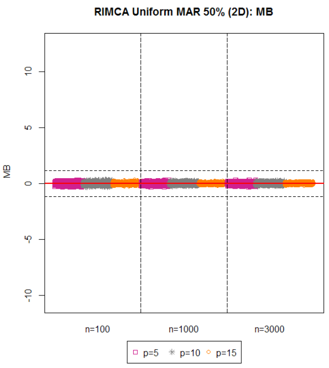


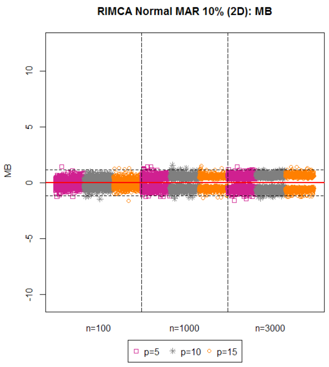


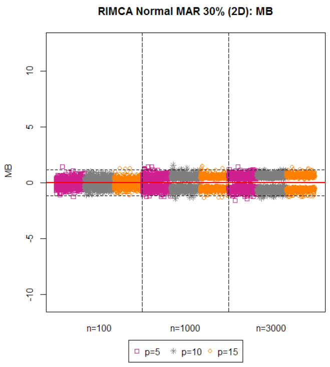


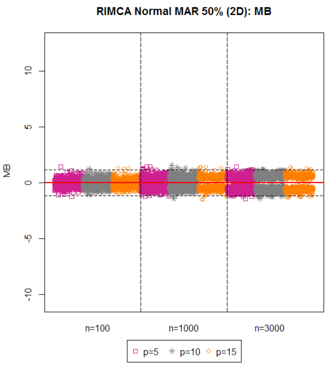




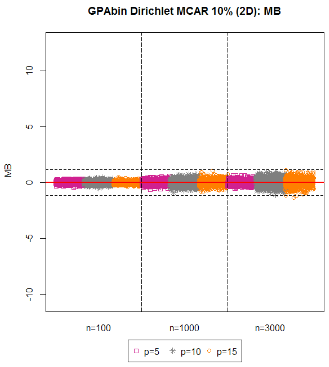


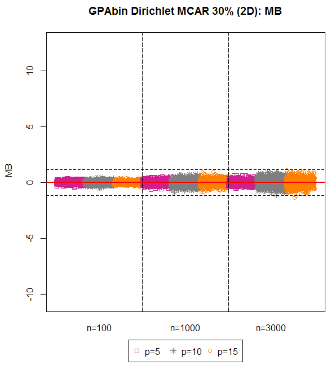


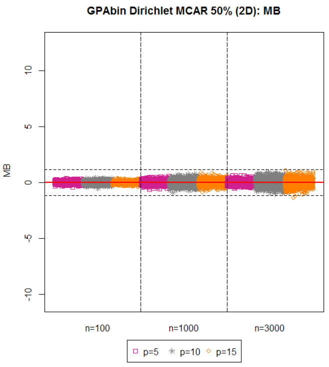


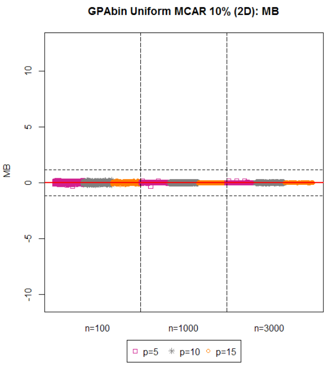


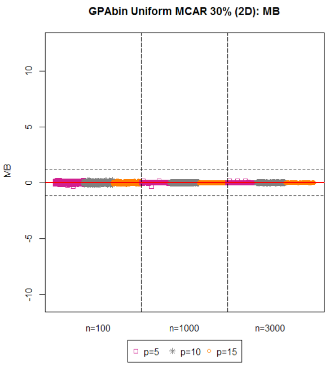
O.2 Mean bias: GPAbin and RIMCA (MCAR MDM)

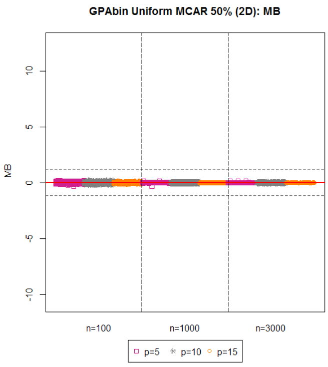


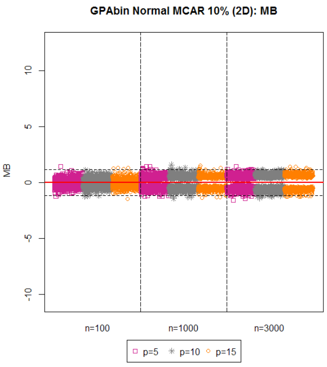


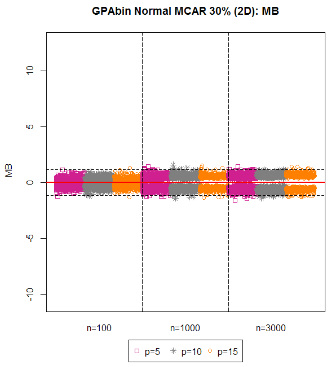


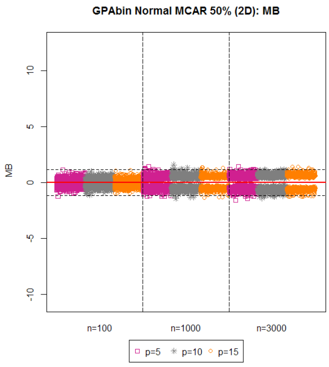

GPAbin Uniform MCAR 10% (2D): MB

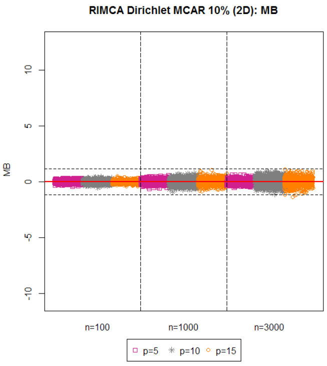

GPAbin Uniform MCAR 30% (2D): MB

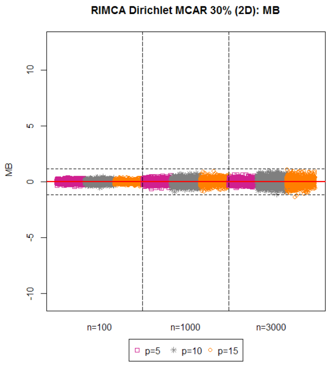

GPAbin Uniform MCAR 50% (2D): MB

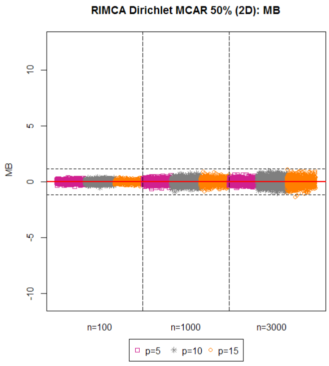

GPAbin Normal MCAR 10% (2D): MB

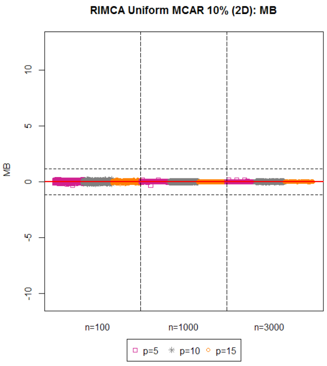

GPAbin Normal MCAR 30% (2D): MB

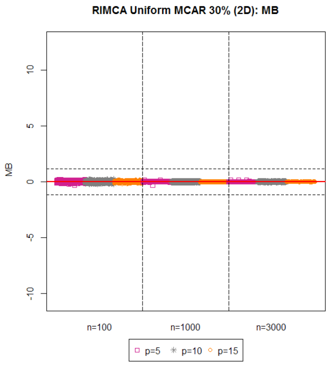

GPAbin Normal MCAR 50% (2D): MB

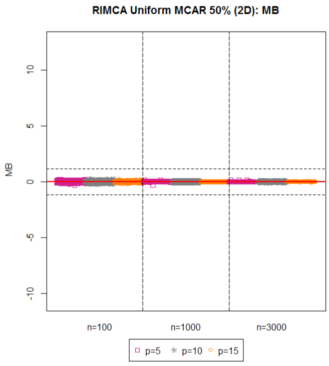

RIMCA Dirichlet MCAR 10% (2D): MB

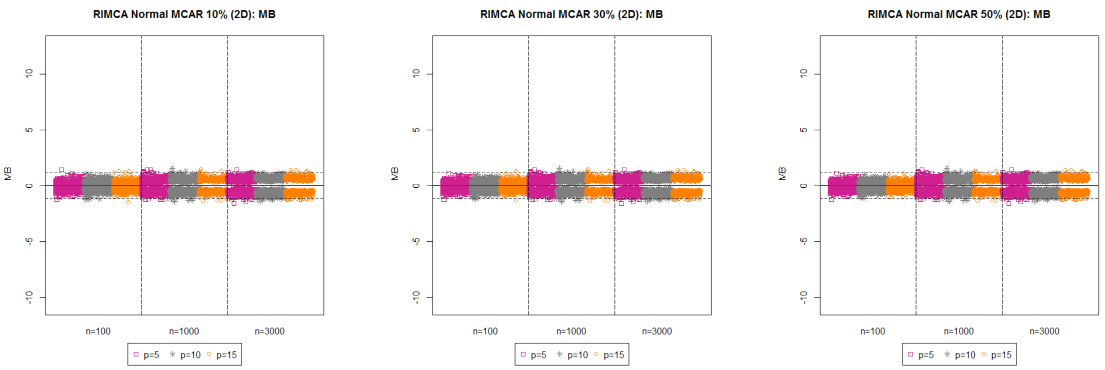

RIMCA Dirichlet MCAR 30% (2D): MB


RIMCA Dirichlet MCAR 50% (2D): MB


RIMCA Uniform MCAR 10% (2D): MB

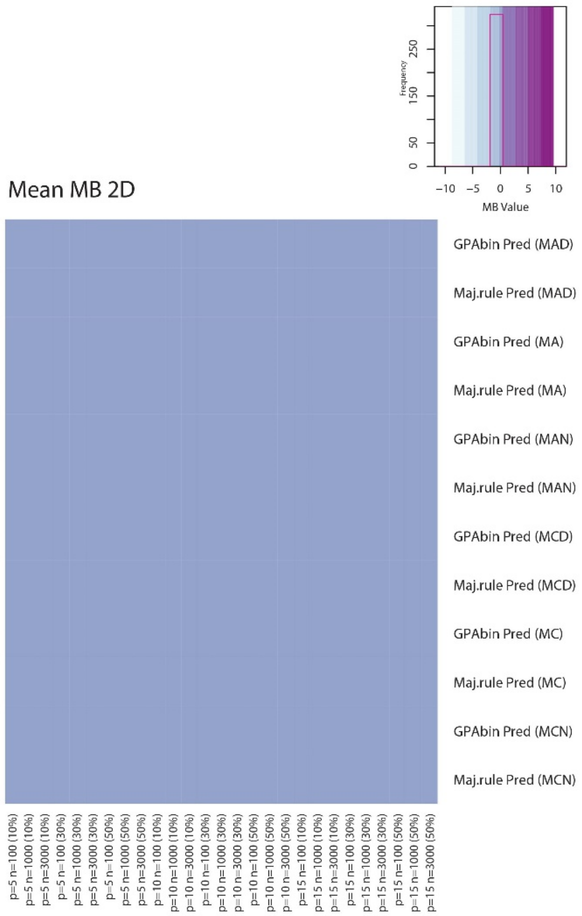

RIMCA Uniform MCAR 30% (2D): MB


RIMCA Uniform MCAR 50% (2D): MB

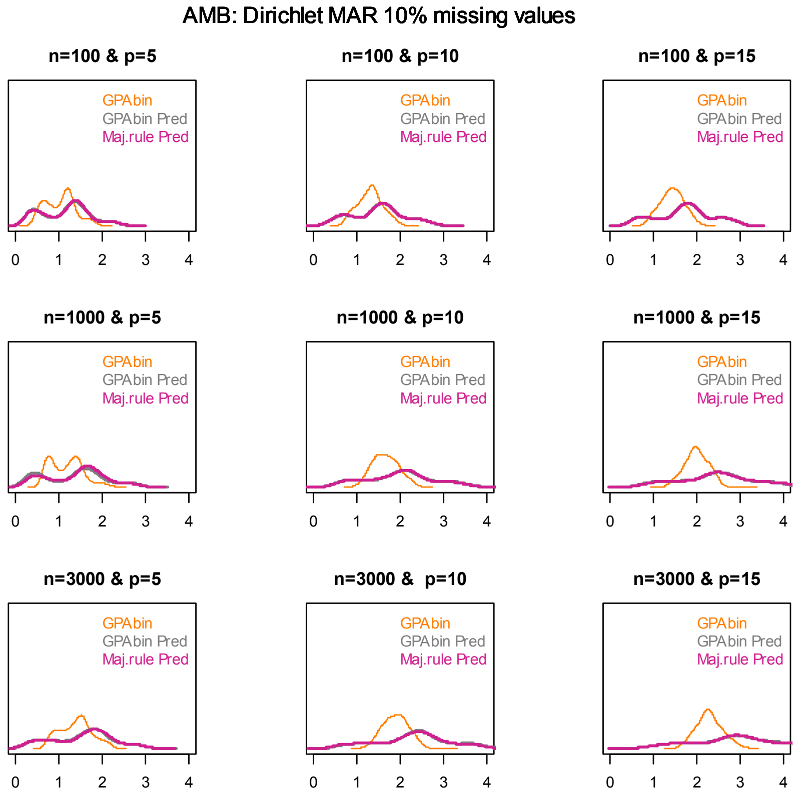


Appendix P Measures of comparison: prediction of categorical data sets

P.1 Heat map of MB values for prediction methods



P.2 Selection of density plots of prediction methods compared to GPAbin



AMB: Dirichlet MAR 10% missing values

n=100 & p=5

n=100 & p=10

n=100 & p=15

n=1000 & p=5

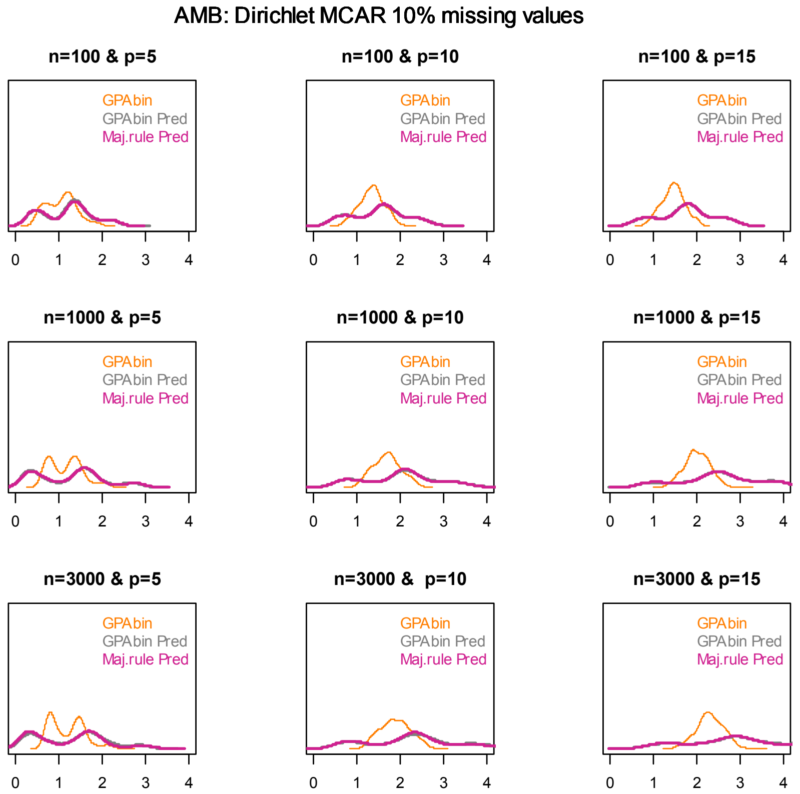
n=1000 & p=10

n=1000 & p=15

n=3000 & p=5

n=3000 & p=10

n=3000 & p=15



AMB: Dirichlet MCAR 10% missing values

n=100 & p=5

n=100 & p=10

n=100 & p=15

n=1000 & p=5

n=1000 & p=10

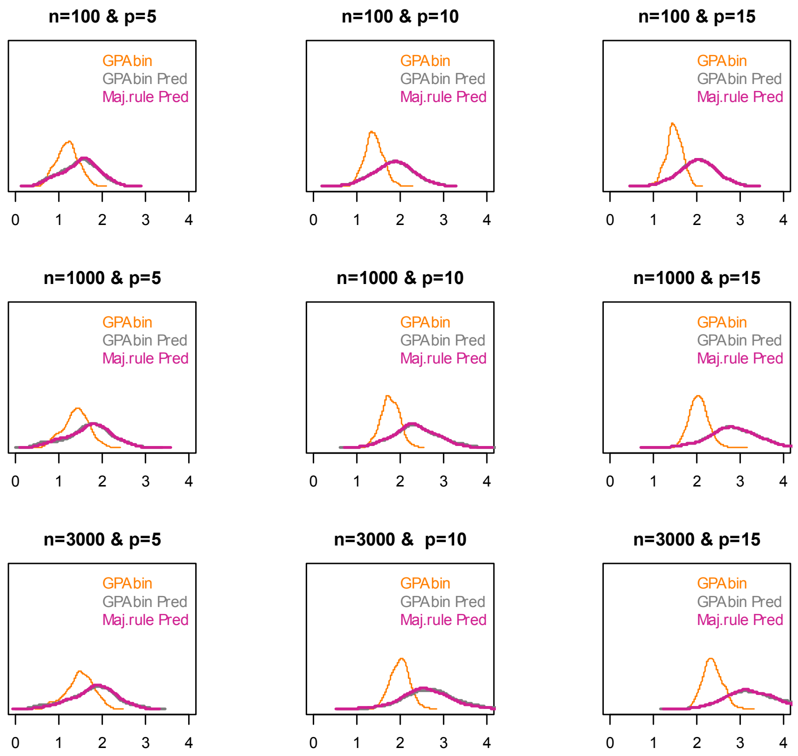
n=1000 & p=15

n=3000 & p=5

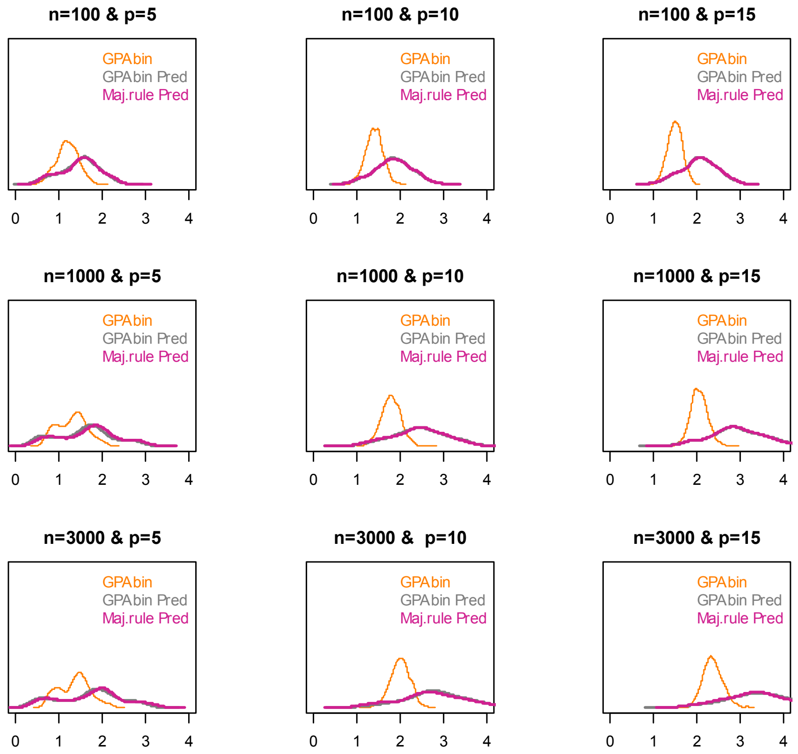
n=3000 & p=10

n=3000 & p=15

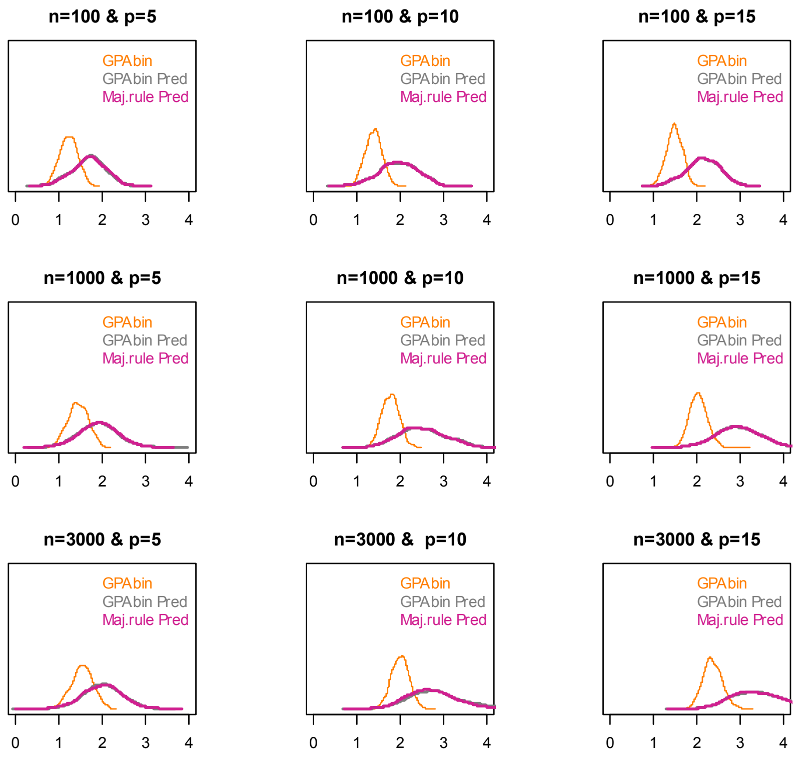
AMB: Dirichlet MAR 30% missing values



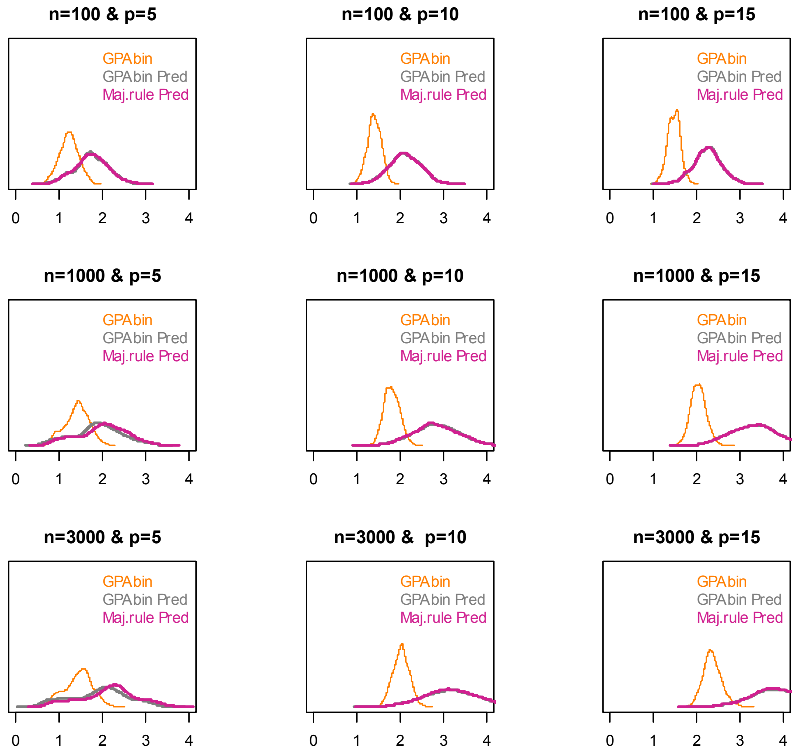
AMB: Dirichlet MCAR 30% missing values



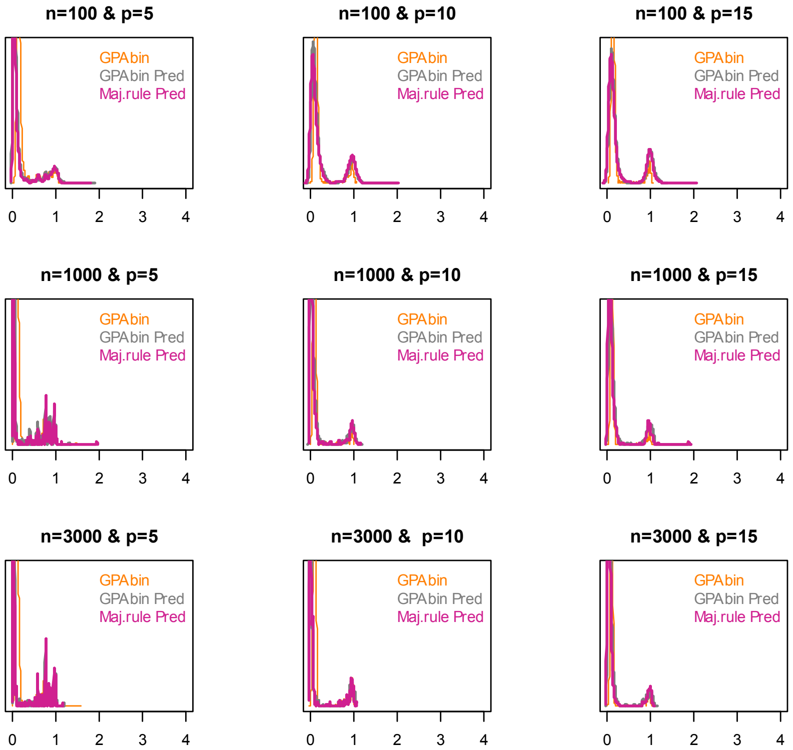
AMB: Dirichlet MAR 50% missing values



AMB: Dirichlet MCAR 50% missing values



AMB: Uniform MAR 10% missing values



n=100 & p=5

n=100 & p=10

n=100 & p=15

n=1000 & p=5

n=1000 & p=10

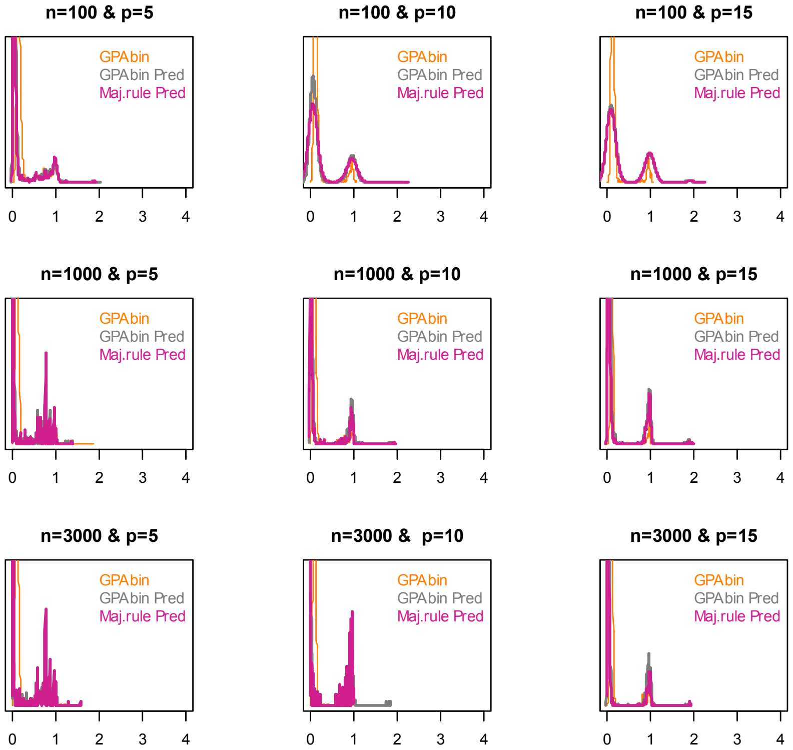
n=1000 & p=15

n=3000 & p=5

n=3000 & p=10

n=3000 & p=15

AMB: Uniform MCAR 10% missing values



n=100 & p=5

n=100 & p=10

n=100 & p=15

n=1000 & p=5

n=1000 & p=10

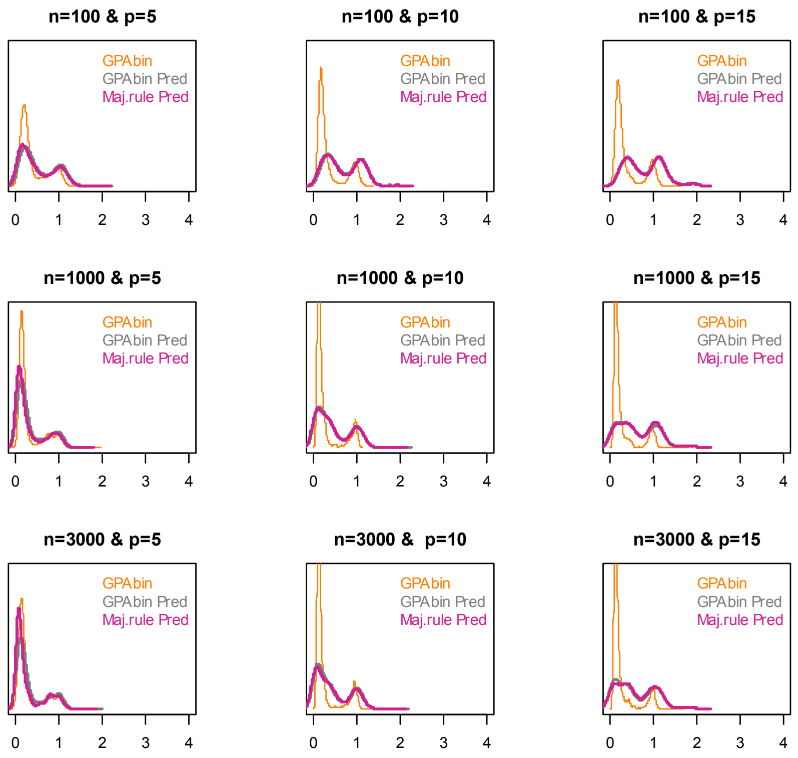
n=1000 & p=15

n=3000 & p=5

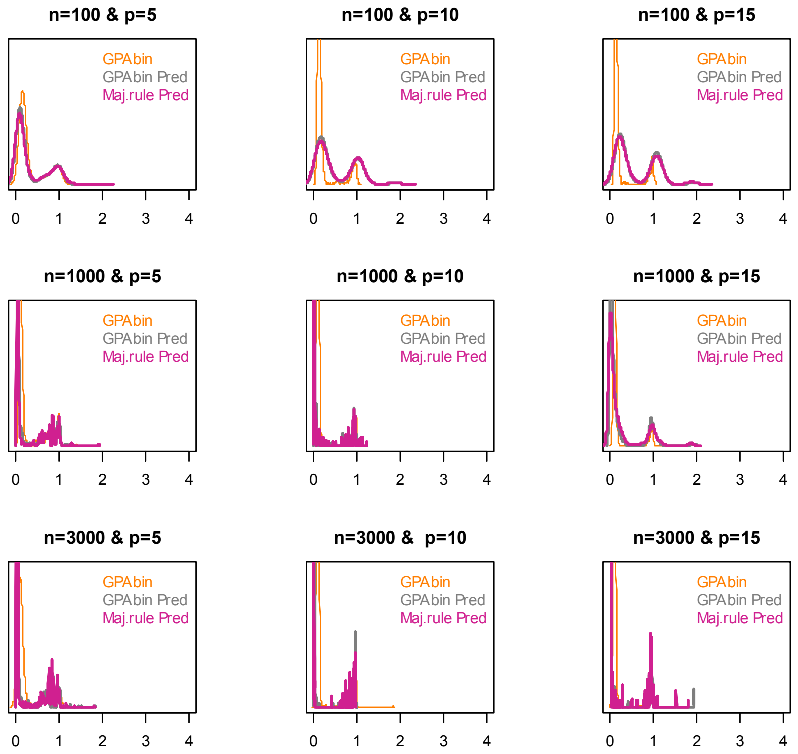
n=3000 & p=10

n=3000 & p=15

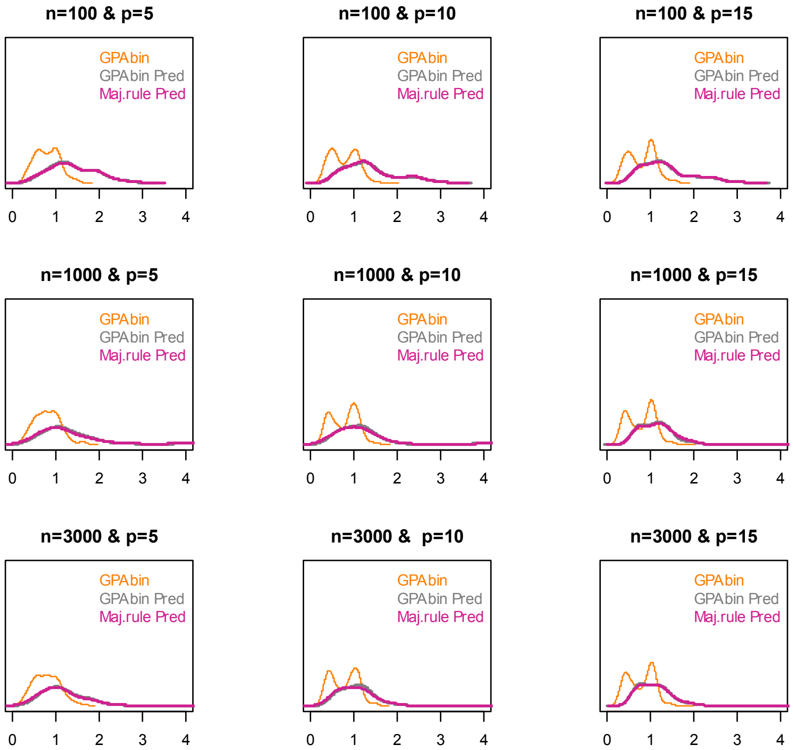
AMB: Uniform MAR 30% missing values



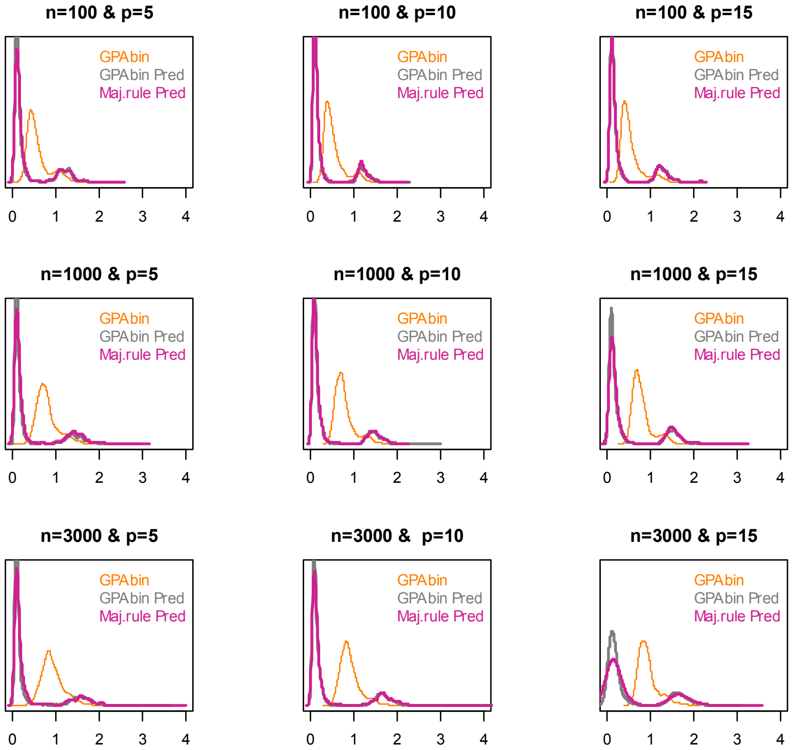
AMB: Uniform MCAR 30% missing values



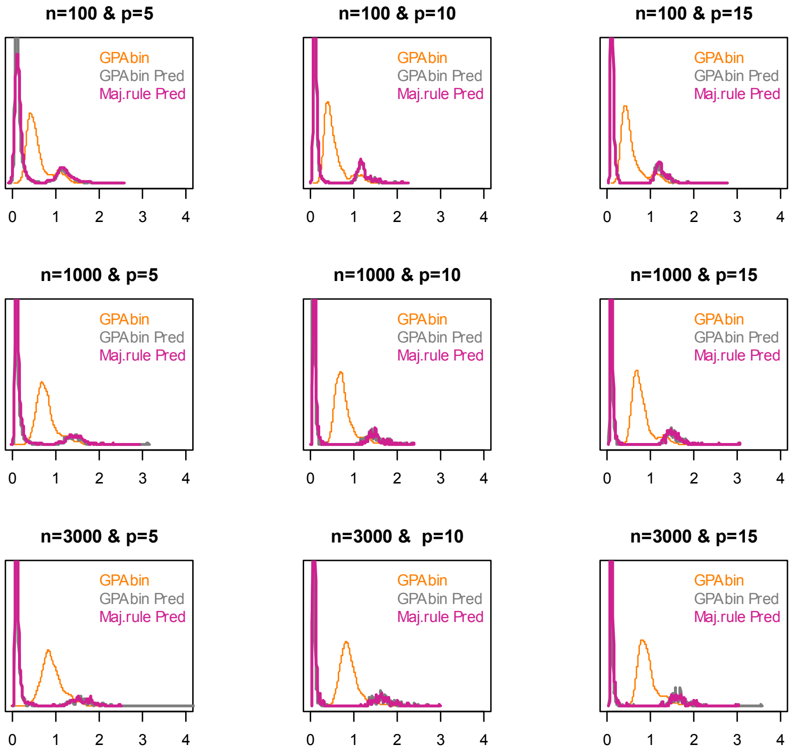
AMB: Uniform MAR 50% missing values



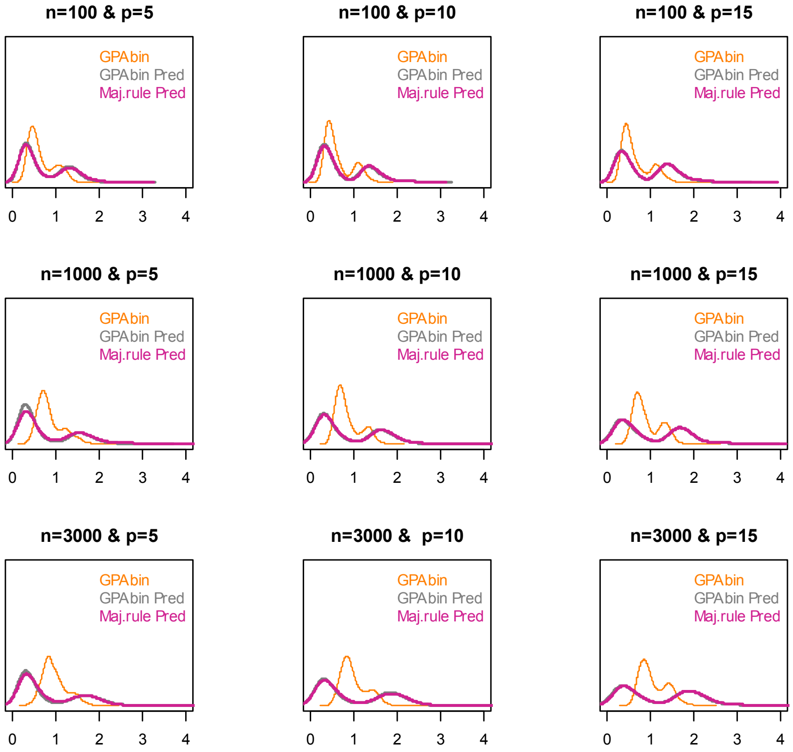
AMB: Normal MAR 10% missing values

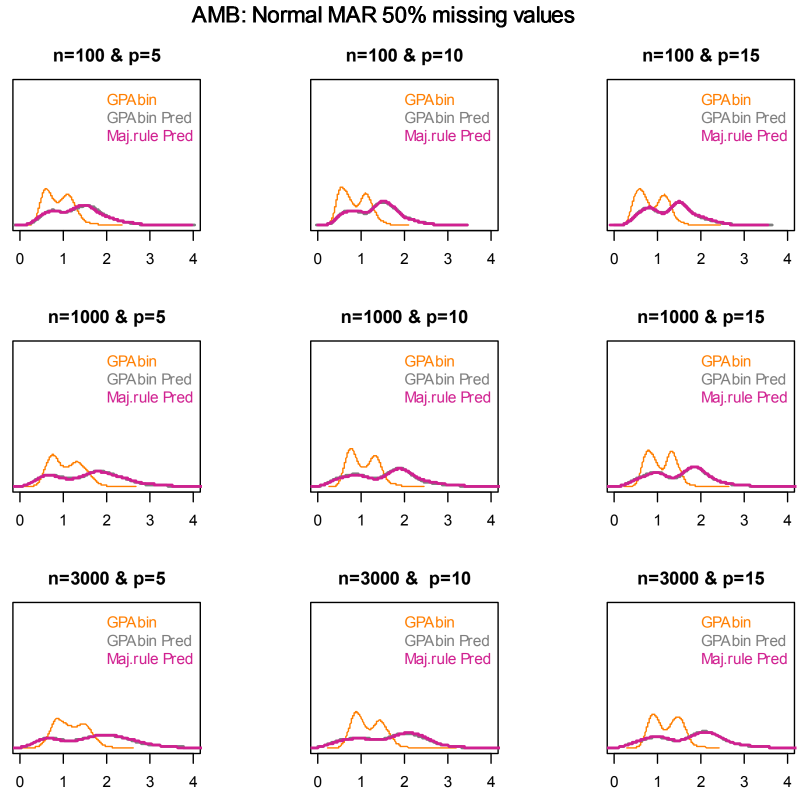


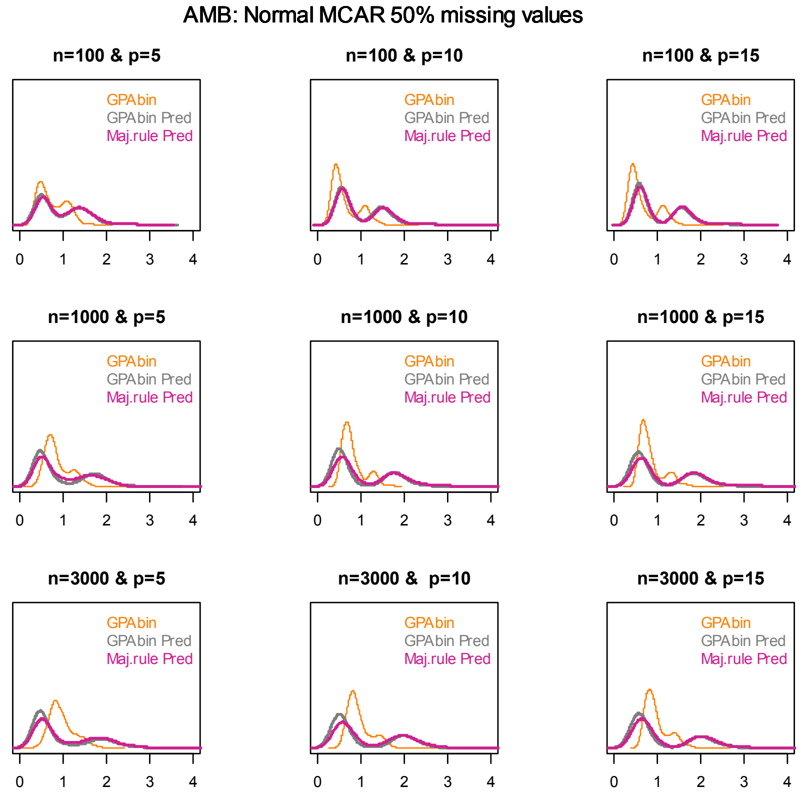
AMB: Normal MCAR 10% missing values



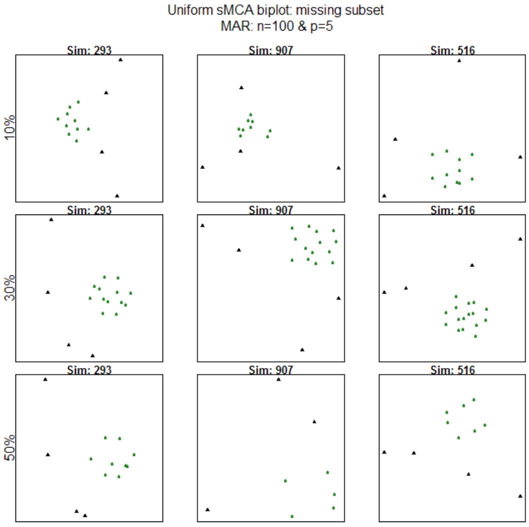
AMB: Normal MAR 30% missing values

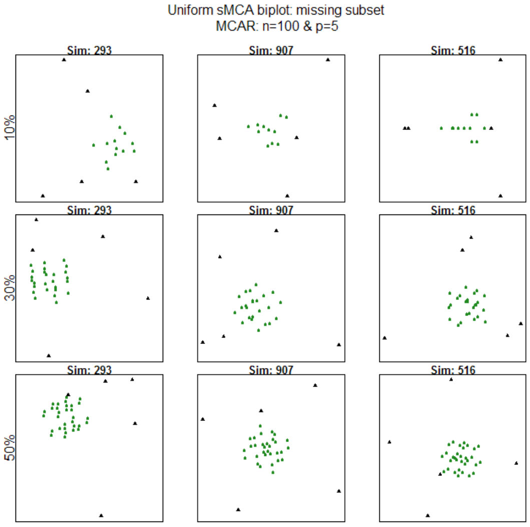


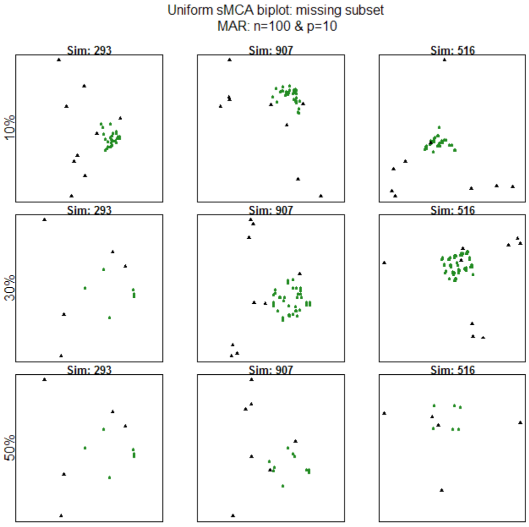

AMB: Normal MAR 50% missing values
n=100 & p=5
n=100 & p=10
n=100 & p=15
n=1000 & p=5
n=1000 & p=10
n=1000 & p=15
n=3000 & p=5
n=3000 & p=10
n=3000 & p=15
GPAbin
GPAbin Pred
Maj.rule Pred
0
1
2
3
4

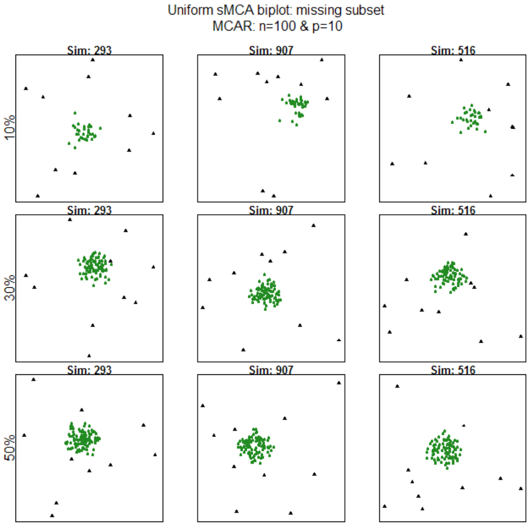

AMB: Normal MCAR 50% missing values
n=100 & p=5
n=100 & p=10
n=100 & p=15
n=1000 & p=5
n=1000 & p=10
n=1000 & p=15
n=3000 & p=5
n=3000 & p=10
n=3000 & p=15
GPAbin
GPAbin Pred
Maj.rule Pred
0
1
2
3
4

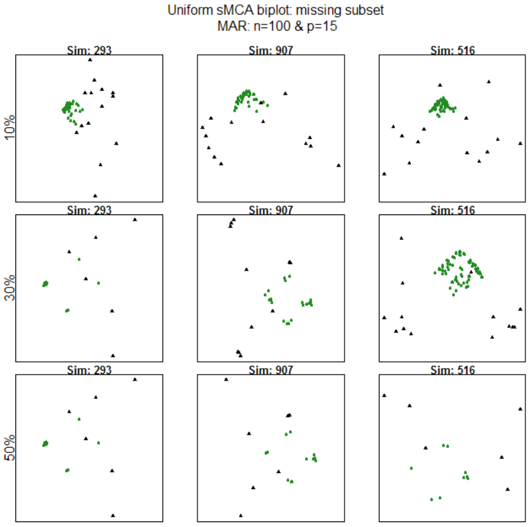
Appendix Q sMCA biplots: missing subsets

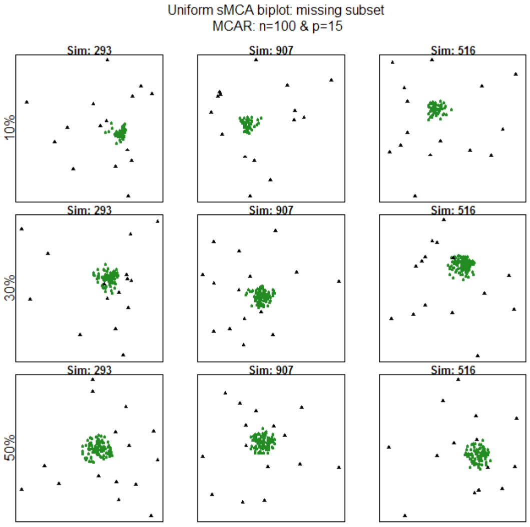


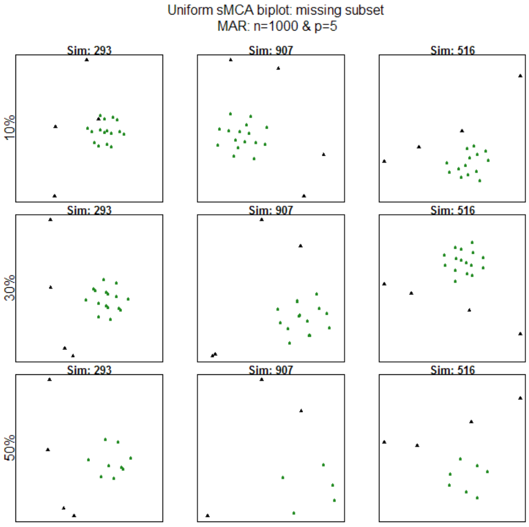


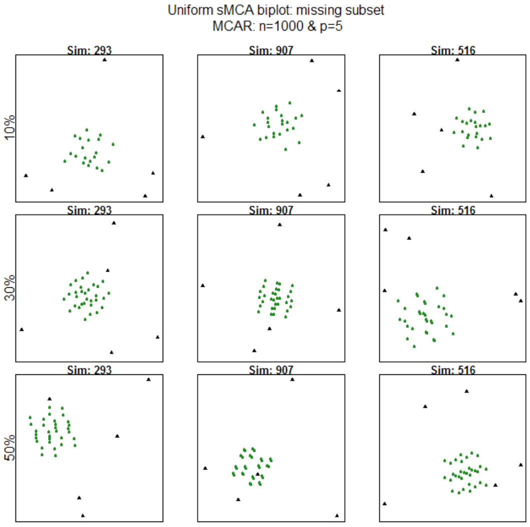


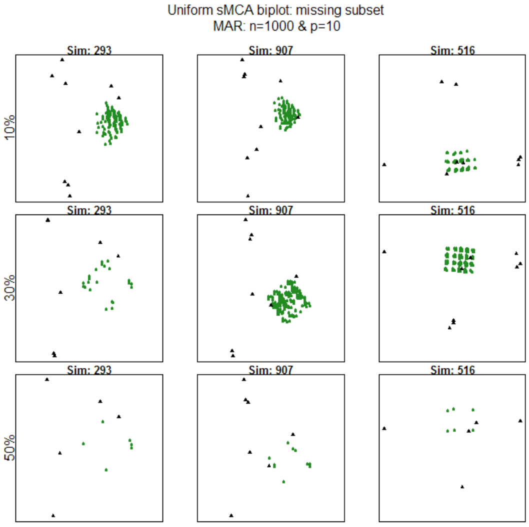


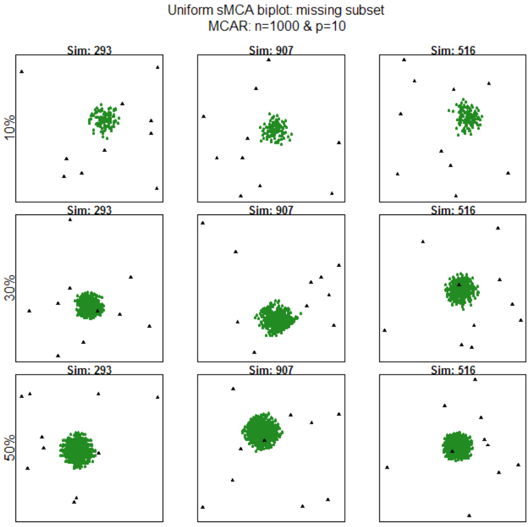


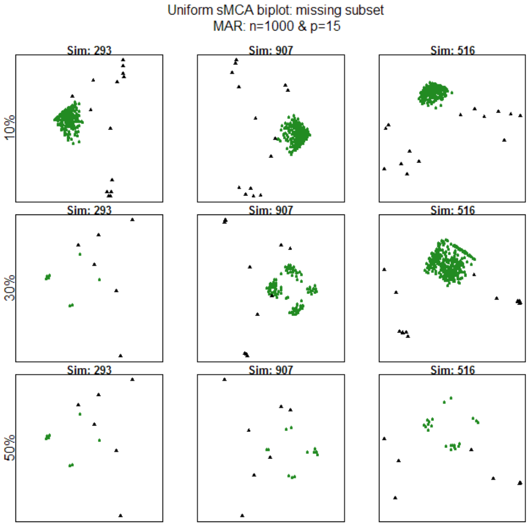


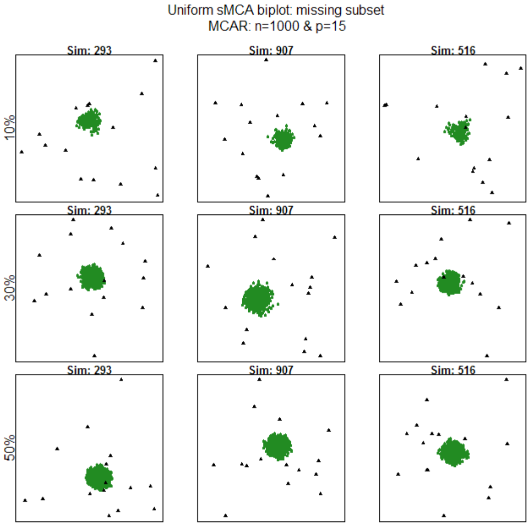

Uniform sMCA biplot: missing subset
MAR: n=1000 & p=5


Uniform sMCA biplot: missing subset
MCAR: n=1000 & p=5

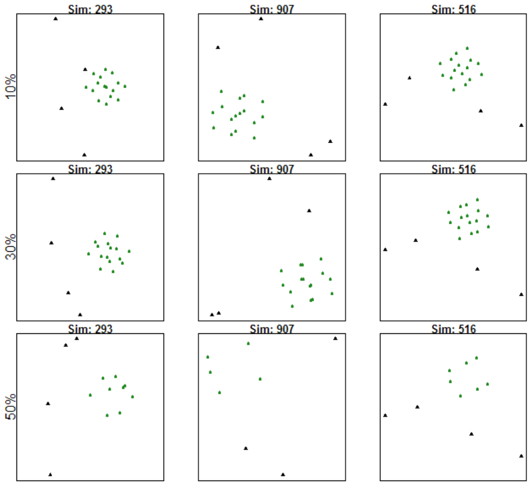

Uniform sMCA biplot: missing subset
MAR: n=1000 & p=10


Uniform sMCA biplot: missing subset
MCAR: n=1000 & p=10

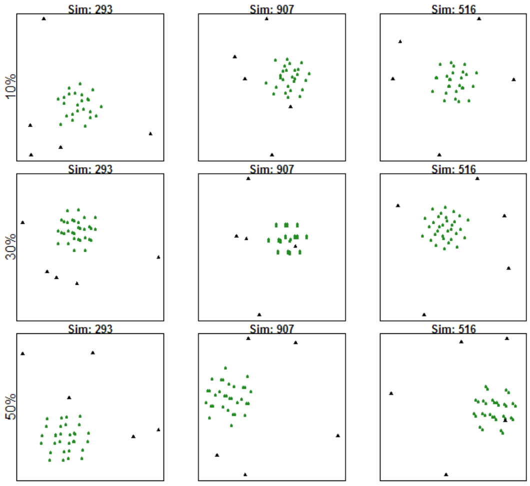

Uniform sMCA biplot: missing subset
MAR: n=1000 & p=15


Uniform sMCA biplot: missing subset
MCAR: n=1000 & p=15

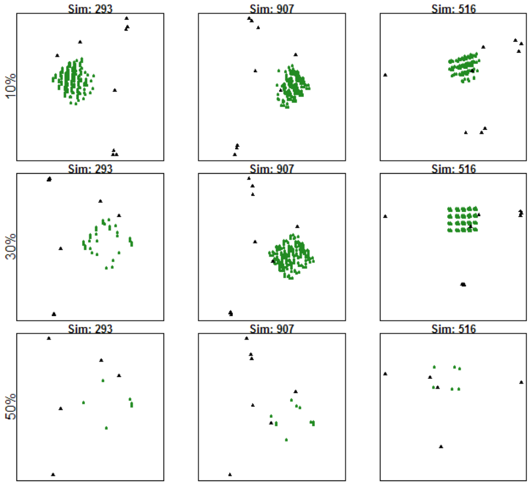
Uniform sMCA biplot: missing subset
MAR: n=3000 & p=5



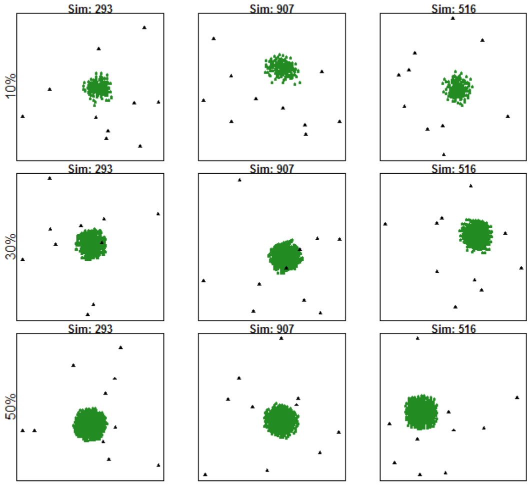
Uniform sMCA biplot: missing subset
MCAR: n=3000 & p=5



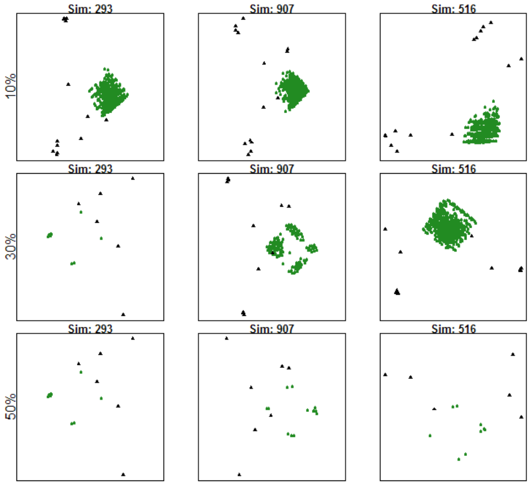
Uniform sMCA biplot: missing subset
MAR: n=3000 & p=10



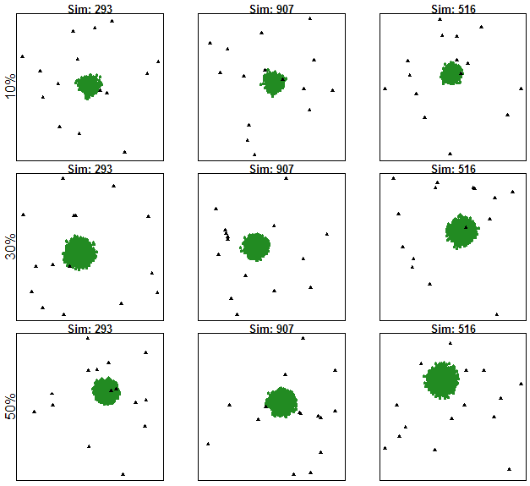
Uniform sMCA biplot: missing subset
MCAR: n=3000 & p=10

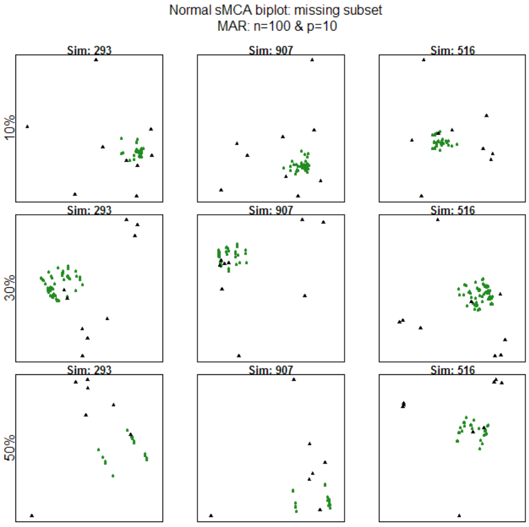


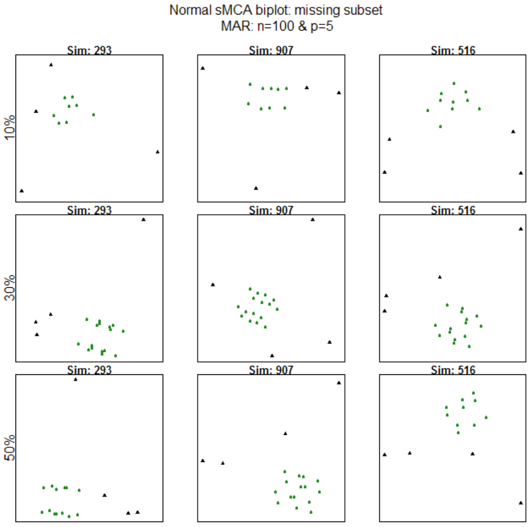
Uniform sMCA biplot: missing subset
MAR: n=3000 & p=15

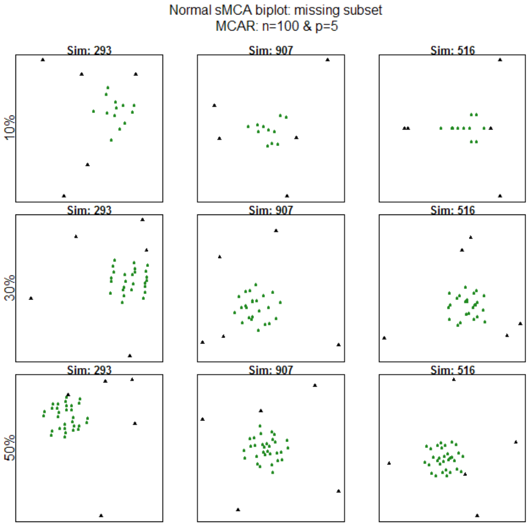


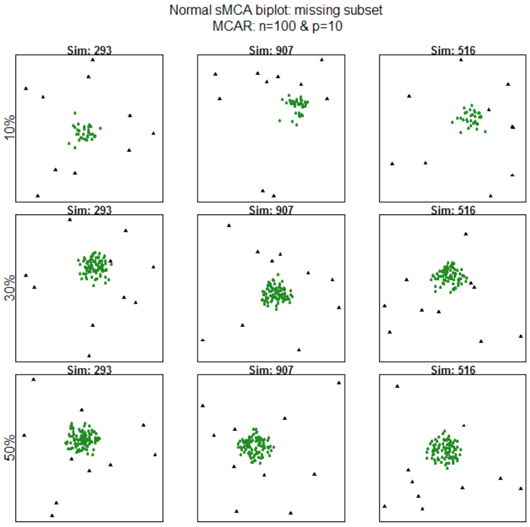
Uniform sMCA biplot: missing subset
MCAR: n=3000 & p=15

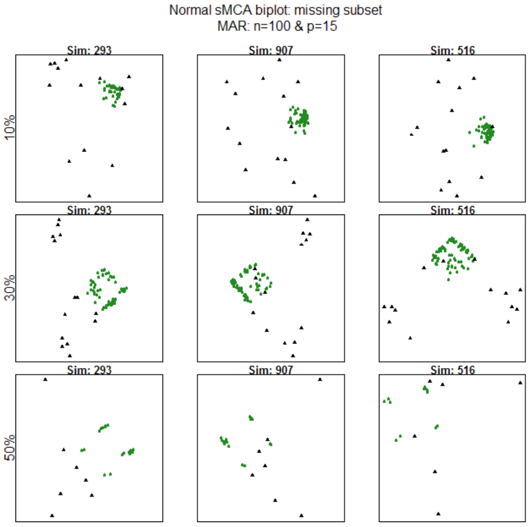


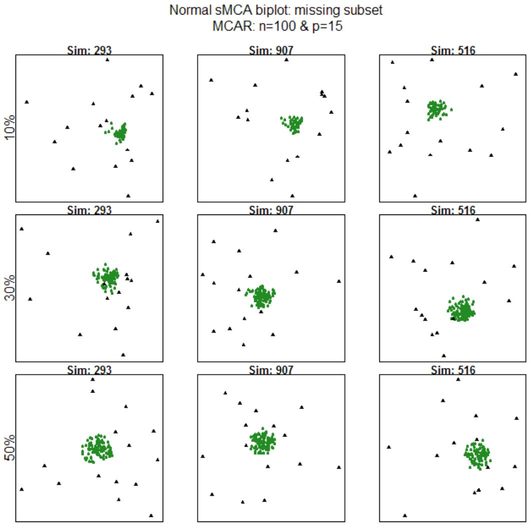

Normal sMCA biplot: missing subset
MAR: n=100 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

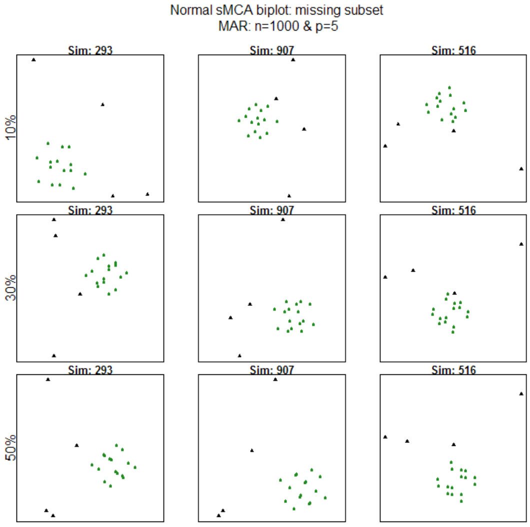

Normal sMCA biplot: missing subset
MAR: n=100 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

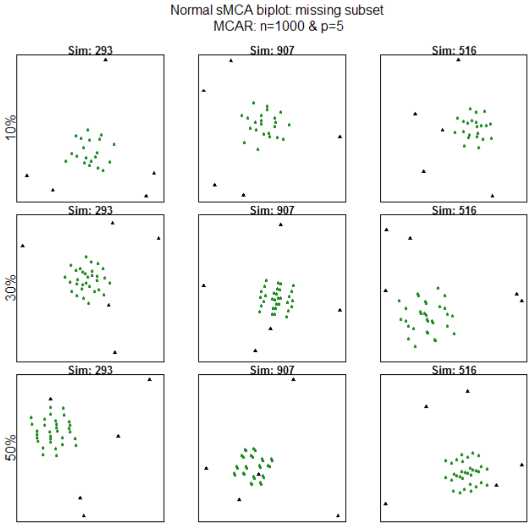

Normal sMCA biplot: missing subset
MCAR: n=100 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

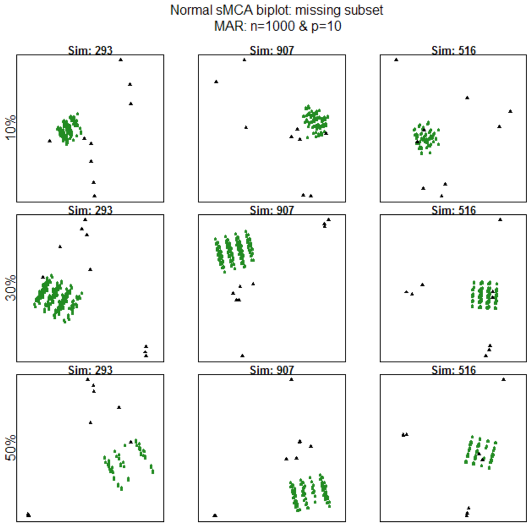

Normal sMCA biplot: missing subset
MCAR: n=100 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

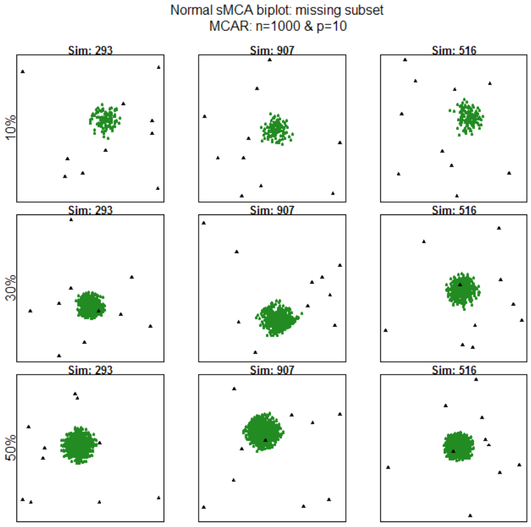

Normal sMCA biplot: missing subset
MAR: n=100 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

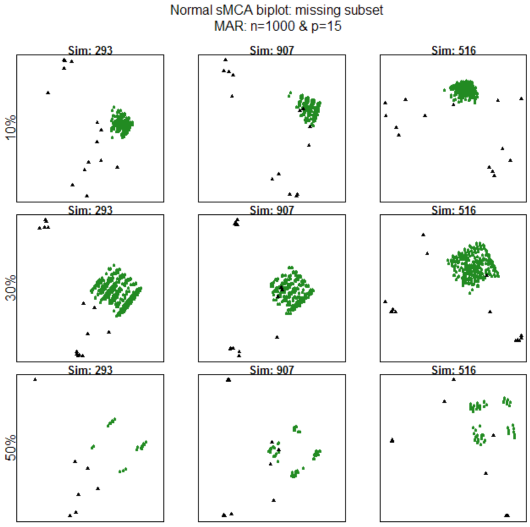

Normal sMCA biplot: missing subset
MCAR: n=100 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

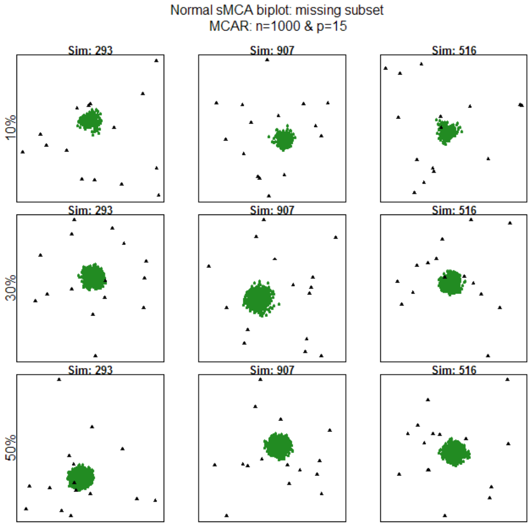

Normal sMCA biplot: missing subset
MAR: n=1000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

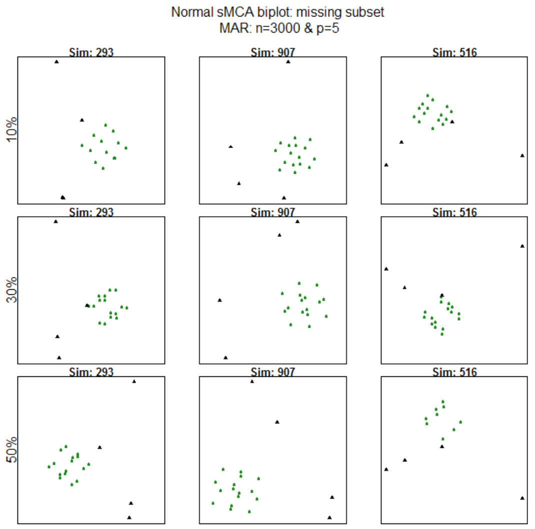

Normal sMCA biplot: missing subset
MCAR: n=1000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

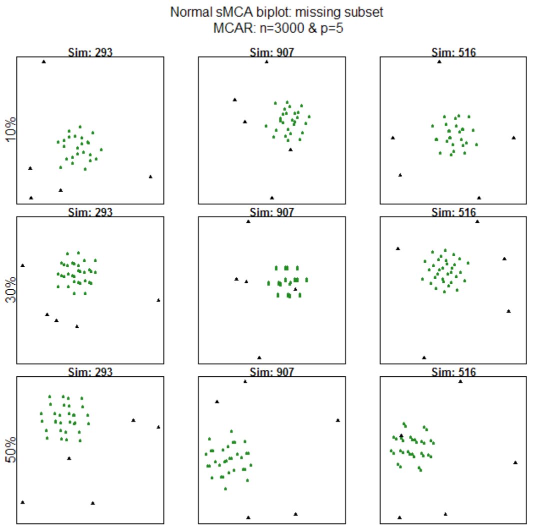

Normal sMCA biplot: missing subset
MAR: n=1000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

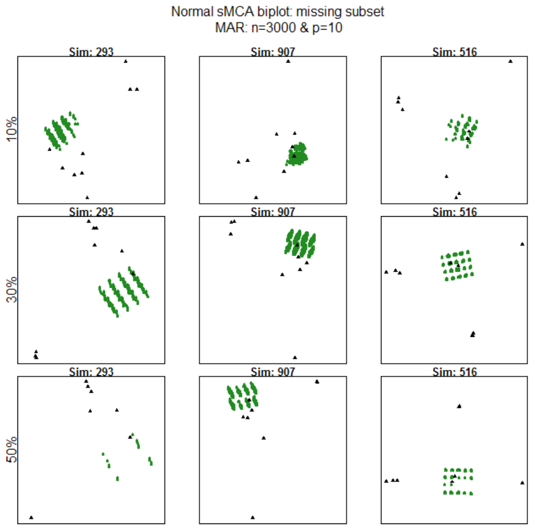

Normal sMCA biplot: missing subset
MCAR: n=1000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

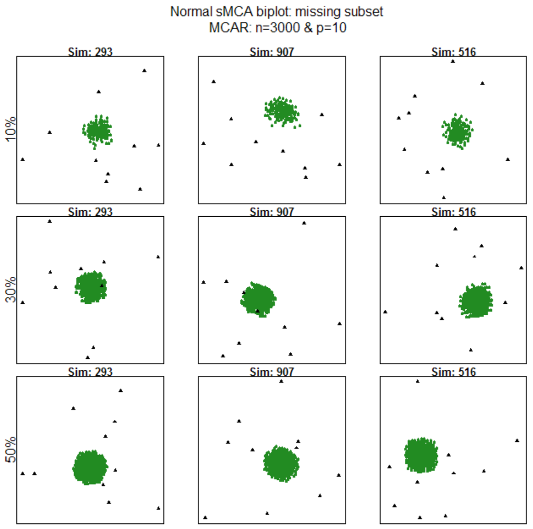

Normal sMCA biplot: missing subset
MAR: n=1000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

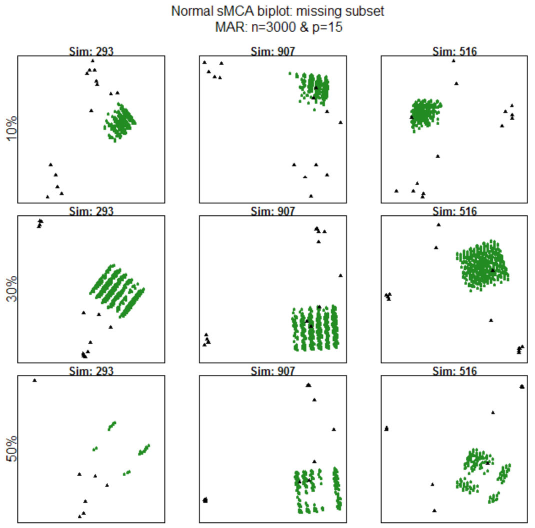

Normal sMCA biplot: missing subset
MCAR: n=1000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

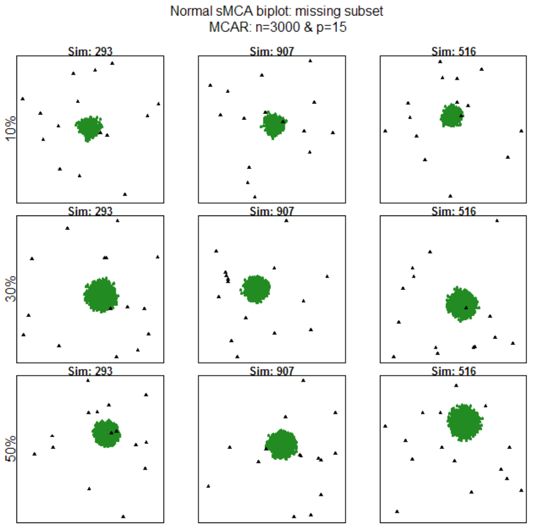

Normal sMCA biplot: missing subset
MAR: n=3000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

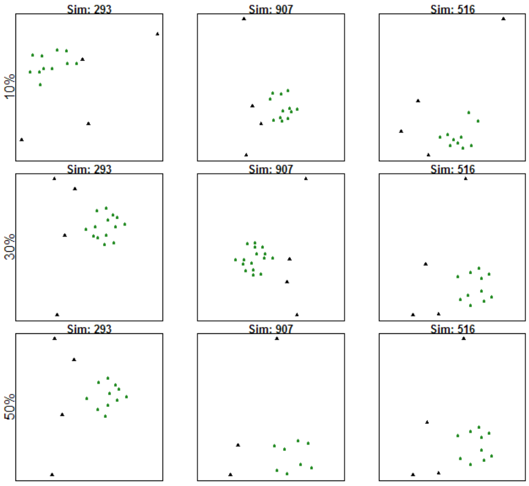

Normal sMCA biplot: missing subset
MCAR: n=3000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Normal sMCA biplot: missing subset
MAR: n=3000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

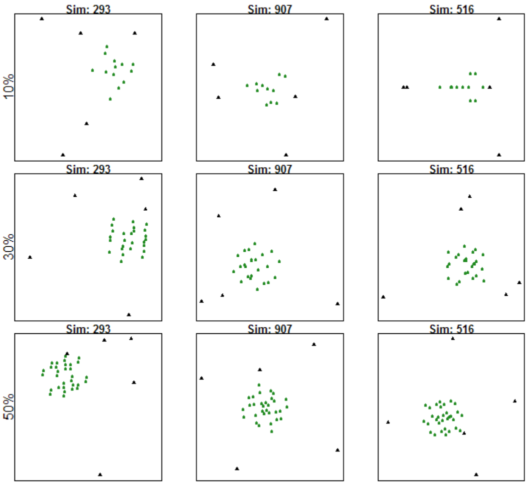

Normal sMCA biplot: missing subset
MCAR: n=3000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Normal sMCA biplot: missing subset
MAR: n=3000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

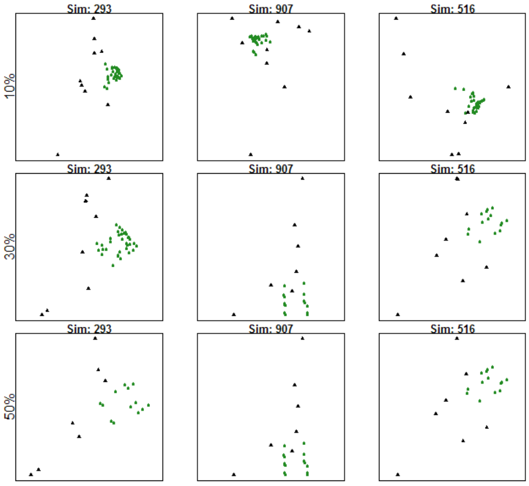

Normal sMCA biplot: missing subset
MCAR: n=3000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%



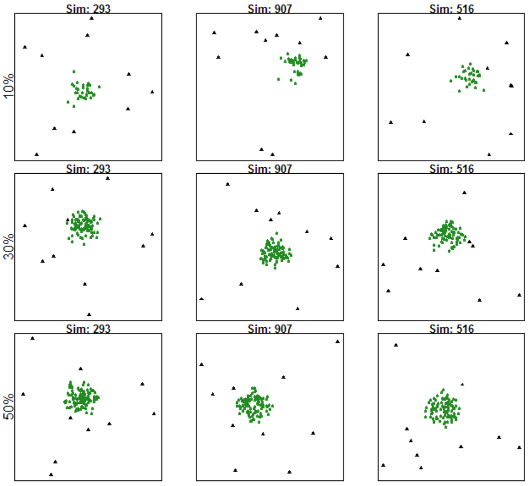
Dirichlet sMCA biplot: missing subset
MAR: n=100 & p=5



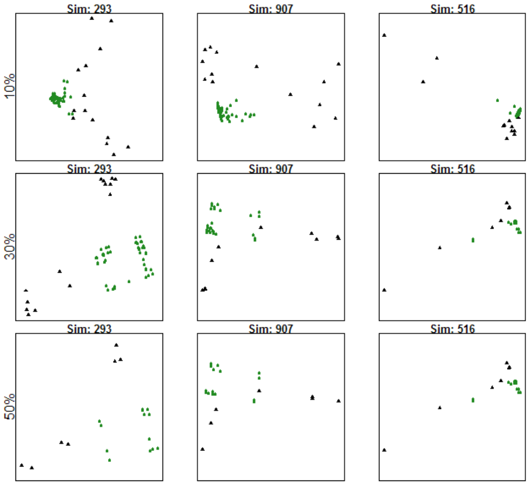
Dirichlet sMCA biplot: missing subset
MCAR: n=100 & p=5



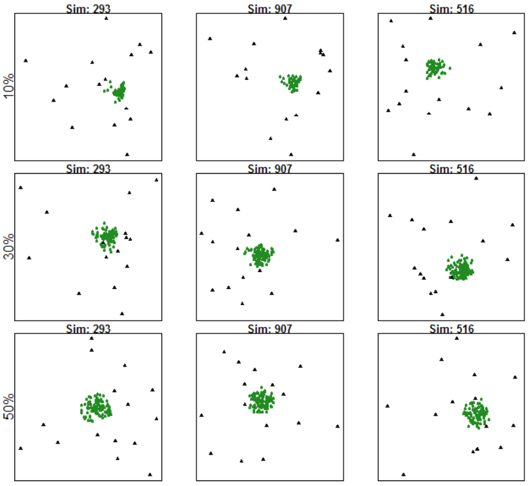
Dirichlet sMCA biplot: missing subset
MAR: n=100 & p=10



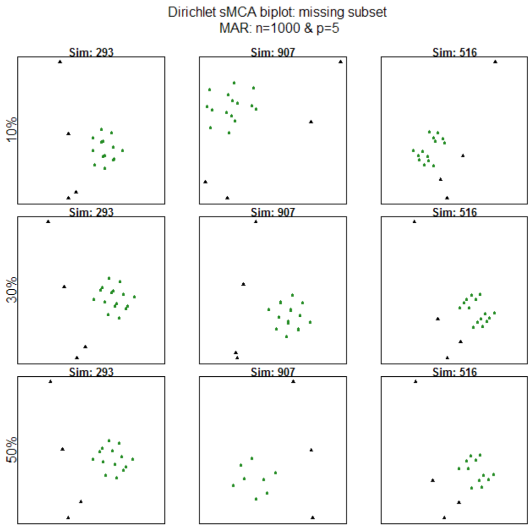
Dirichlet sMCA biplot: missing subset
MCAR: n=100 & p=10

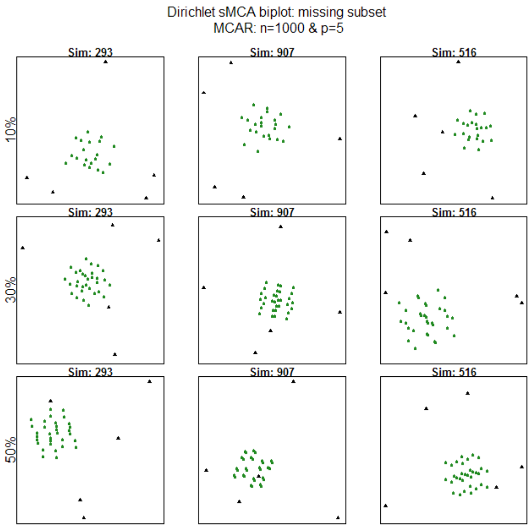


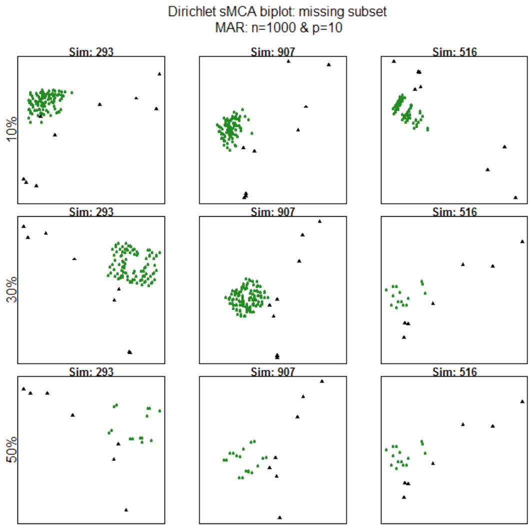
Dirichlet sMCA biplot: missing subset
MAR: n=100 & p=15

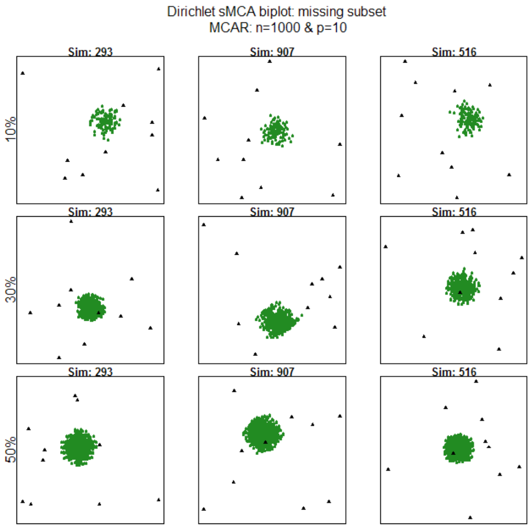


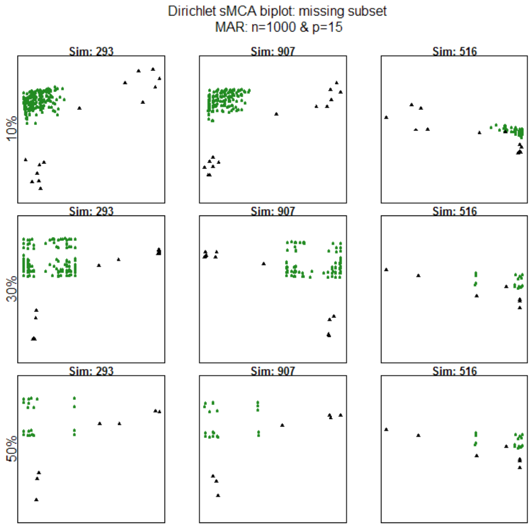
Dirichlet sMCA biplot: missing subset
MCAR: n=100 & p=15

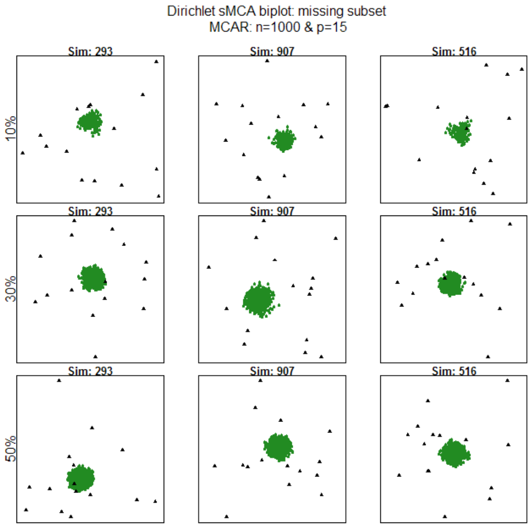

Dirichlet sMCA biplot: missing subset
MAR: n=1000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

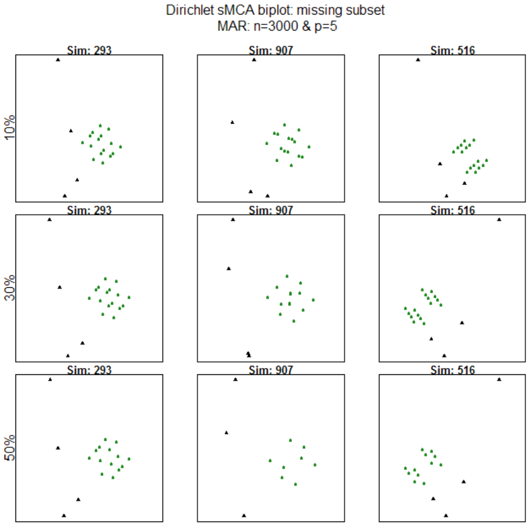

Dirichlet sMCA biplot: missing subset
MCAR: n=1000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

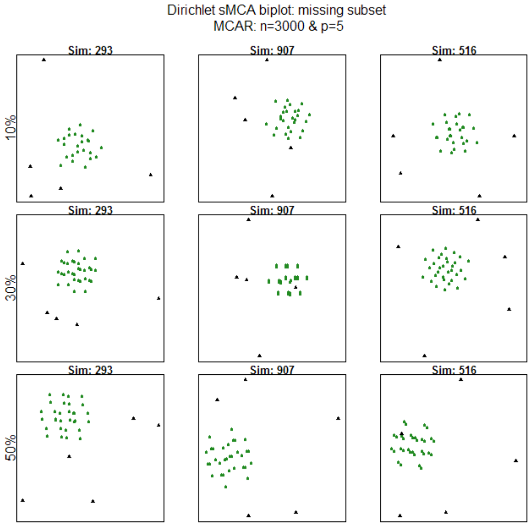

Dirichlet sMCA biplot: missing subset
MAR: n=1000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

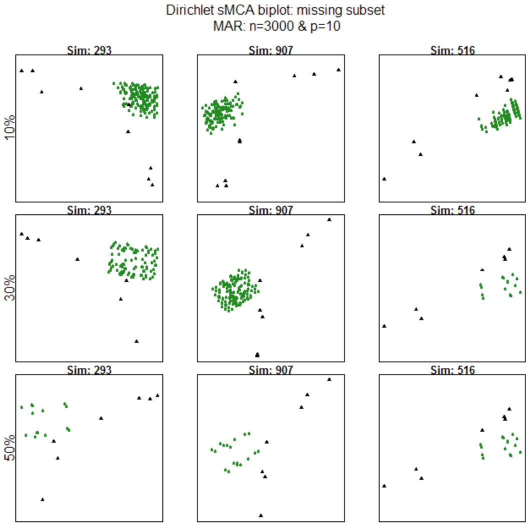

Dirichlet sMCA biplot: missing subset
MCAR: n=1000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

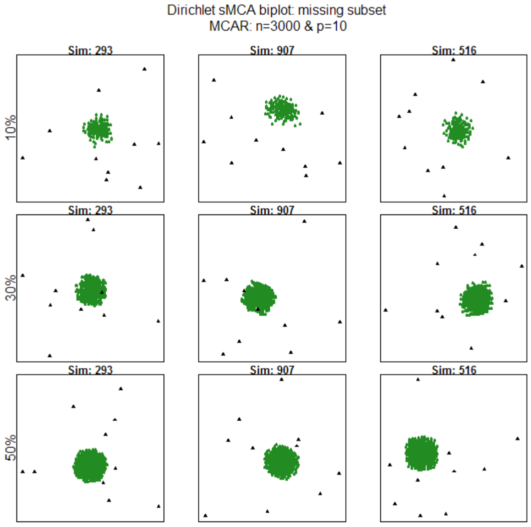

Dirichlet sMCA biplot: missing subset
MAR: n=1000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

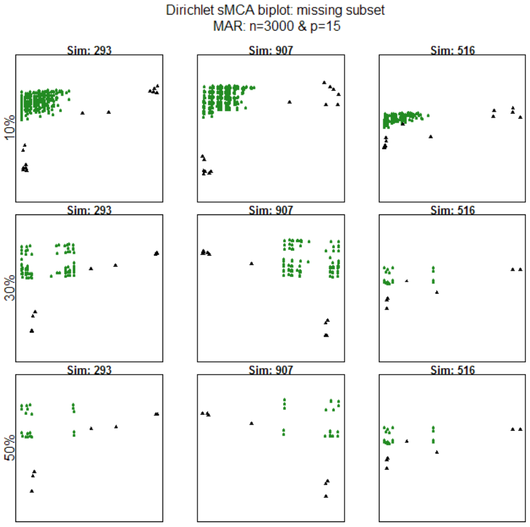

Dirichlet sMCA biplot: missing subset
MCAR: n=1000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%

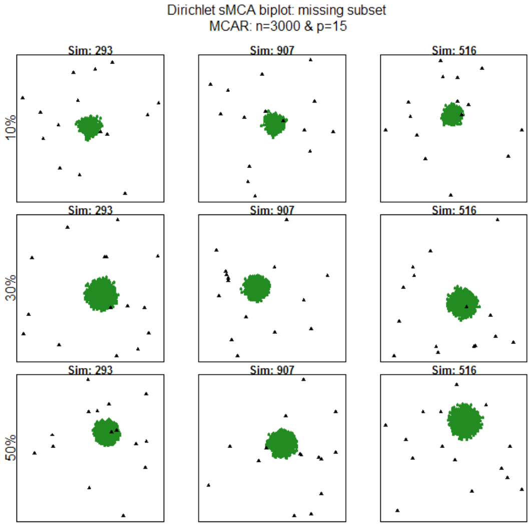

Dirichlet sMCA biplot: missing subset
MAR: n=3000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Dirichlet sMCA biplot: missing subset
MCAR: n=3000 & p=5
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Dirichlet sMCA biplot: missing subset
MAR: n=3000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Dirichlet sMCA biplot: missing subset
MCAR: n=3000 & p=10
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Dirichlet sMCA biplot: missing subset
MAR: n=3000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%


Dirichlet sMCA biplot: missing subset
MCAR: n=3000 & p=15
Sim: 293
Sim: 907
Sim: 516
10%
30%
50%