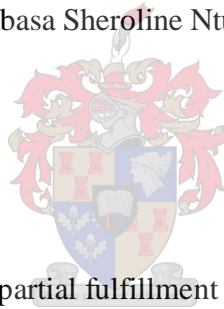


EXPLORATORY AND INFERENTIAL MULTIVARIATE STATISTICAL
TECHNIQUES FOR MULTIDIMENSIONAL
COUNT AND BINARY DATA WITH APPLICATIONS IN R

by

Nombasa Sheroline Ntushelo



Assignment presented in partial fulfillment of the requirements for the
degree of Master of Commerce at Stellenbosch University
department of Statistics and Actuarial Science

Supervisor:

Dr. M.M.C. Lamont

December 2011

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 owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Name: Nombasa Sheroline Ntushelo

Date: December 2011

Copyright © 2011 Stellenbosch University
All rights reserved

OPSOMMING

Die analise van meerdimensionele (meerveranderlike) datastelle is 'n belangrike area van navorsing in toegepaste statistiek. Oor die afgelope dekades is daar verskeie tegnieke ontwikkel om sulke data te ontleed. Die meerveranderlike tegnieke wat ontwikkel is sluit in inferensie analise, regressie analise, diskriminant analise, tros analise en vele meer verkennende data analise tegnieke. Die meerderheid van hierdie metodes hanteer gevalle waar die data numeriese veranderlikes bevat. Daar bestaan ook kragtige metodes in die literatuur vir die analise van meerdimensionele binêre en telling data.

Die primêre doel van hierdie tesis is om tegnieke vir verkennende en inferensiële analise van binêre en telling data te bespreek. In Hoofstuk 2 van hierdie tesis bespreek ons ooreenkoms analise en kanoniese ooreenkoms analise. Hierdie metodes word gebruik om data in gebeurlikheidstabelle te analiseer. Hoofstuk 3 bevat tegnieke vir tros analise. In hierdie hoofstuk verduidelik ons vier gewilde tros analise metodes. Ons bespreek ook die afstand maatstawwe wat beskikbaar is in die literatuur vir binêre en telling data. Hoofstuk 4 bevat 'n verduideliking van metriese en nie-metriese meerdimensionele skalering. Hierdie metodes kan gebruik word om binêre of telling data in 'n lae dimensionele Euclidiese ruimte voor te stel. In Hoofstuk 5 beskryf ons 'n inferensie metode wat bekend staan as die analise van afstande. Hierdie metode gebruik 'n soortgelyke redenasie as die analise van variansie. Die inferensie hier is gebaseer op 'n pseudo F -toetsstatistiek en die p -waardes word verkry deur gebruik te maak van permutasies van die data. Hoofstuk 6 bevat toepassings van bogenoemde tegnieke op werklike datastelle wat bekend staan as die Biolog data en die Barents Fish data.

Die sekondêre doel van die tesis is om te demonstreer hoe hierdie tegnieke uitgevoer word in the R sagteware. Verskeie R pakette en funksies word deurgaans bespreek in die tesis. Die gebruik van die funksies word gedemonstreer met toepaslike voorbeelde. Aandag word ook gegee aan die interpretasie van die afvoer en die grafieke. Die tesis sluit af met algemene gevolgtrekkings en voorstelle vir verdere navorsing.

SUMMARY

The analysis of multidimensional (multivariate) data sets is a very important area of research in applied statistics. Over the decades many techniques have been developed to deal with such datasets. The multivariate techniques that have been developed include inferential analysis, regression analysis, discriminant analysis, cluster analysis and many more exploratory methods. Most of these methods deal with cases where the data contain numerical variables. However, there are powerful methods in the literature that also deal with multidimensional binary and count data.

The primary purpose of this thesis is to discuss the exploratory and inferential techniques that can be used for binary and count data. In Chapter 2 of this thesis we give the detail of correspondence analysis and canonical correspondence analysis. These methods are used to analyze the data in contingency tables. Chapter 3 is devoted to cluster analysis. In this chapter we explain four well-known clustering methods and we also discuss the distance (dissimilarity) measures available in the literature for binary and count data. Chapter 4 contains an explanation of metric and non-metric multidimensional scaling. These methods can be used to represent binary or count data in a lower dimensional Euclidean space. In Chapter 5 we give a method for inferential analysis called the analysis of distance. This method use a similar reasoning as the analysis of variance, but the inference is based on a pseudo F -statistic with the p -value obtained using permutations of the data. Chapter 6 contains real-world applications of these above methods on two special data sets called the Biolog data and Barents Fish data.

The secondary purpose of the thesis is to demonstrate how the above techniques can be performed in the software package R. Several R packages and functions are discussed throughout this thesis. The usage of these functions is also demonstrated with appropriate examples. Attention is also given to the interpretation of the output and graphics. The thesis ends with some general conclusions and ideas for further research.

DEDICATED TO MY MOTHER NOSIPHO CYNTHIA NTUSHELO,
MY FATHER MLULEKI MORGAN NTUSHELO,
MY BROTHERS SIYABULELA, MUSA, SABELO AND
MY YOUNGER SISTER BUSISIWE.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to all the people who supported me throughout the course of this study.

First I would like to thank my family for being so patient and kind to let me spend so much time away from them, my mother Nosipho Cynthia Ntushelo, my father Mluleki Morgan Ntushelo, my brothers Siyabulela Mark Ntushelo, Musa Romario Ntushelo, Sabelo Gladman Ntushelo and my younger sister Busisiwe Bertha Ntushelo. I would also like to thank Khaya Ntushelo for looking out for me and always believing in me.

Secondly, I would like to thank the Agricultural Research Council (ARC) for giving me this great opportunity to further my studies at the University of Stellenbosch. I would like to thank the following people from the ARC: Mr. Frikkie Calitz, my first mentor, for being so kind, patient and encouraging me to further my studies. Mrs. Marie Smith for giving me the opportunity to become a Professional Development Programme student at the ARC Biometry Unit in Stellenbosch. Ina Clark for your love and your motivation. Rita Saunders for your love and for being such a good listener. Marieta van der Rijst, my second mentor, for your guidance, advice and motivation. Mardé Booysse, for being so kind and always willing to help. Dr. Keith du Plessis for allowing me to use your data and your motivation. Zongamele Spelmandla for being always there to help. Mzabalazo Ngwenya for being so encouraging.

Thirdly, I would also like to thank Professor T. de Wet, the head of the Statistics department, for believing in me and giving me this opportunity. Professor Nelmarie Louw for her comments in improving this manuscript. Dr. Morné Lamont, my supervisor, for your guidance, patience, putting faith in me, and motivation. Tchilabalo Kpanzou for your support and advice. My friends who supported me through this work Colette Taylor, Thamsin Taylor, Ivy Noubissi, Thulani Qwabe for being such an awesome young brother, Zukiswa Nameka, Nandipha Ntushelo and Mzwanele Mti.

Finally, I would like to thank the Most High, Jehovah God, for giving me this opportunity and supplying me with all the support I needed.

CONTENTS

CHAPTER 1

Introduction	1
1.1 Background and motivation for study	1
1.2 The Biolog data	2
1.3 The Barents Fish data	6
1.4 The aim of the thesis	10
1.5 Layout of the thesis	10

CHAPTER 2

Simple and Canonical correspondence analysis	12
2.1 Introduction	12
2.2 Simple correspondence analysis	13
2.3 Inertia and Benzécri distances	15
2.4 Performing a correspondence analysis in R	17
2.4.1 The anacor package	18
2.4.2 The ca package	23
2.5 Canonical correspondence analysis (CCA)	26
2.6 Performing a canonical correspondence analysis in R	28
2.6.1 The anacor package	29
2.6.2 The vegan package	32
2.7 Permutation tests in CCA	42
2.8 Summary	44

CHAPTER 3

Cluster analysis	46
3.1 Introduction	46
3.2 The data for cluster analysis	46
3.3 The distance and the dissimilarity matrix	48
3.3.1 Distance measures for numerical data	48

3.3.2 A dissimilarity and distance measure for count data	49
3.3.3 Dissimilarity measures for binary data	50
3.4 Agglomerative hierarchical clustering methods	52
3.4.1 Single linkage	53
3.4.2 Average linkage	54
3.4.3 Complete linkage	54
3.4.4 Ward's method	55
3.5 Performing a cluster analysis in R	56
3.6 Interpreting the cluster analysis results	61
3.7 Summary	61

CHAPTER 4

Metric and Nonmetric multidimensional scaling	62
4.1 Introduction	62
4.2 Metric multidimensional scaling (MMDS)	63
4.3 Nonmetric multidimensional scaling (NMDS)	65
4.4 Performing MMDS and NMDS in R	67
4.5 Interpreting the MDS results	78
4.6 Summary	78

CHAPTER 5

Inference using distance matrices	80
5.1 Introduction	80
5.2 The one-way analysis of variance	80
5.3 The one-way multivariate analysis of variance	83
5.4 The analysis of distance	87
5.5 Performing an analysis of variance in R	90
5.5.1 A multivariate analysis of variance in R	90
5.5.2 An analysis of distance in R	92
5.6 Summary	93

CHAPTER 6

Real-world applications	94
6.1 Introduction	94
6.2 Exploratory analysis of the Biolog data	94
6.2.1 Cluster analysis	94
6.2.2 Nonmetric multidimensional scaling	95
6.2.3 Correspondence analysis	95
6.3 Discussion of the Biolog data results	108
6.3.1 Comparing the results of the Jaccard and Bray-Curtis dissimilarity	108
6.3.2 Comparing the results of the three exploratory analysis methods	109
6.3.3 The goodness-of-fit for the multidimensional scaling and the correspondence analysis	109
6.3.4 Overall conclusion about the treatments	110
6.3.5 Overall conclusion about the carbons	110
6.4 Analysis of distance using the Biolog data	111
6.5 Canonical correspondence analysis of the Barents Fish data	112
6.6 Summary	117

CHAPTER 7

General conclusion	118
---------------------------------	------------

BIBLIOGRAPHY	121
---------------------------	------------

Chapter 1

Introduction

1.1 Background and motivation for study

Multivariate statistical techniques play a very important role in understanding data that are multidimensional in nature. Such data sets are often very complex to understand and very difficult to analyze. Over the last decades the literature on multivariate techniques in the areas of regression analysis, cluster analysis, ordination analysis, discriminant analysis and multivariate inference have been vastly expanded (see for example Mardia *et al.* (1979); Ter Braak (1986); Legendre and Legendre (1998); Cox and Cox (1994); Anderson (2001a); Anderson (2001b); Cox and Cox (2001); Quinn and Keough (2001); Greenacre (2007); Johnson and Wichern (2007); Nenadic and Greenacre (2007); de Leeuw and Mair (2009); etc.). Many of these techniques are original and very sophisticated, while others are extensions of the univariate methods. These techniques have been applied in a variety of fields such as Biology, Ecology, Medicine, Marketing, Agriculture, Psychology, Economics, and many more. The great success with which it has been applied, is instrumental in the popularity of the techniques among statisticians and researchers in other fields.

The number of techniques used for analyzing multivariate numerical data is much more than those for other types of data. Analyzing numerical data is usually easier than analyzing multivariate count, categorical and binary data sets. In this thesis we will look specifically at the analysis of multidimensional count and binary data. Researchers often make observations that involve counts or the presence (absence) of some phenomenon. How to analyze such data is often unfamiliar to them. A variety of techniques for the analysis of such data exists and in this thesis we will review many of them and also apply the techniques to data sets.

A large part of this thesis will be devoted to develop an understanding of how to analyse multidimensional count and binary data. A detailed explanation of popular techniques such as correspondence analysis, canonical correspondence analysis, cluster analysis, multidimensional scaling and analysis of distance will be given in subsequent chapters. The explanations are accompanied by practical applications in the R software. A detailed illustration of how these methods are performed in R is given using examples. A discussion of the available R packages and corresponding functions will also be given.

Another important contribution of the thesis is the analysis of two data sets. The first data set is multidimensional binary data set from the South African Agricultural Research Council (ARC). This data set, referred to as the Biolog data, will be described in more detail in the Section 1.2. The second data set, referred to as the Barents Fish data, is a multidimensional data set containing count and numerical data. This data set will be used to perform a canonical correspondence analysis and was obtained from a multivariate statistics workshop by professors M. Greenacre and R. Primicerio presented at Stellenbosch University. A description of the data is given in Section 1.3.

Throughout the thesis the advantages/ disadvantages of the methods and R functions will be highlighted. Emphasis is placed on the analysis of the data, graphical illustrations and the interpretation of the output. Many of the techniques make use of a distance or dissimilarity matrix. Choosing the appropriate distance or dissimilarity measure for the data (numerical, count or binary) also receives attention in this thesis.

1.2 The Biolog data

The Biolog data refers to an experiment that was conducted by researchers at the Nietvoorbij institute of the Agricultural Research Council (ARC) in Stellenbosch. The analysis of this data set, which will be discussed in Chapter 6, forms an important part of the thesis. The following is a description of how the experiment was conducted and how the data was obtained. See Figure 1.3 for an extraction of the Biolog data.

Chapter 1: Introduction

The experiment is about differently treated soil being used to study the activities of micro organisms in the soil. The soil was treated with 12 treatments using a randomized experimental design layout on a piece of land. Samples of the soil were collected at two depths (0-75mm and 150 – 300mm) to study the microbial activities at different layers in the ground. Samples were also collected for three months (February, September and December) to study the microbial activity over time. This experiment continued over the period 2006 to 2009. However, for the purpose of this thesis we will only analyze the data for 2006. Once the soil samples (for the 12 treatments, 2 depths and 3 month) were collected, it was dissolved in water. If dissolved in water, the soil will sink to the bottom and the micro organism in the soil will rise to the top. A sample of this water was then put in a Biolog EcoPlate to observe the microbial activity. The following is a description of the Biolog EcoPlate^a.

The Biolog EcoPlate is a tool that is used a lot for community analysis and ecological studies. A picture of the plate is shown in Figure 1.1. This EcoPlate contains 31 of the most useful carbon sources (see Figure 1.2 for a description) for soil community analysis. It should be noted that water is included in the EcoPlate as the 32nd component of the EcoPlate. These 32 components of the EcoPlate are repeated 3 times in order to give more replicates of the data.

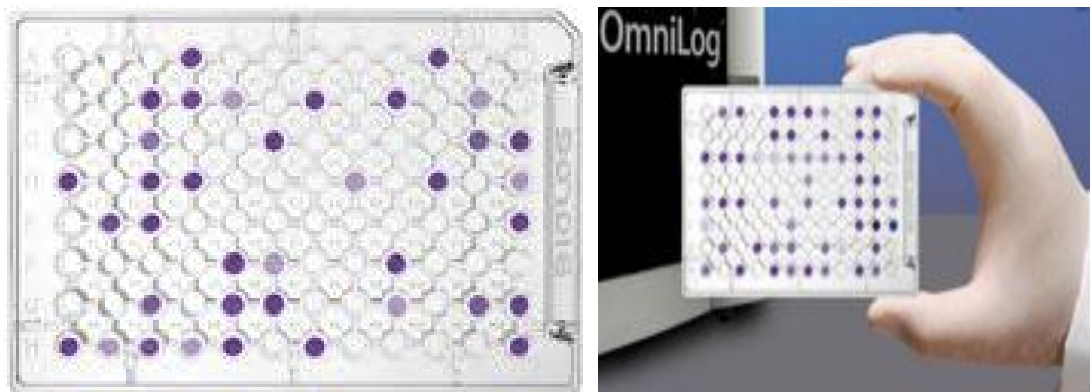


Figure 1.1: The picture of a physical Biolog EcoPlate during an experiment. The purple wells contain carbon sources that were used by the microbial community. The intensity of the purple coloration indicates the degree of carbon source usage by the community. There are 96 wells in the plate comprising the 32 carbons which are each replicated 3 times.

^a for more information visit the websites: www.biolog.com and <http://sites.google.com/site/cellbiosciencesau/services/biolog-1>

BiOLOG

EcoPlate™

Microbial Community Analysis

A1 Water	A2 β-Methyl-D-Glucoside	A3 D-Galactonic Acid γ-Lactone	A4 L-Arginine	A1 Water	A2 β-Methyl-D-Glucoside	A3 D-Galactonic Acid γ-Lactone	A4 L-Arginine	A1 Water	A2 β-Methyl-D-Glucoside	A3 D-Galactonic Acid γ-Lactone	A4 L-Arginine
B1 Pyruvic Acid Methyl Ester	B2 D-Xylose	B3 D-Galacturonic Acid	B4 L-Asparagine	B1 Pyruvic Acid Methyl Ester	B2 D-Xylose	B3 D-Galacturonic Acid	B4 L-Asparagine	B1 Pyruvic Acid Methyl Ester	B2 D-Xylose	B3 D-Galacturonic Acid	B4 L-Asparagine
C1 Tween 40	C2 i-Erythritol	C3 2-Hydroxy Benzoic Acid	C4 L-Phenylalanine	C1 Tween 40	C2 i-Erythritol	C3 2-Hydroxy Benzoic Acid	C4 L-Phenylalanine	C1 Tween 40	C2 i-Erythritol	C3 2-Hydroxy Benzoic Acid	C4 L-Phenylalanine
D1 Tween 80	D2 D-Mannitol	D3 4-Hydroxy Benzoic Acid	D4 L-Serine	D1 Tween 80	D2 D-Mannitol	D3 4-Hydroxy Benzoic Acid	D4 L-Serine	D1 Tween 80	D2 D-Mannitol	D3 4-Hydroxy Benzoic Acid	D4 L-Serine
E1 α-Cyclodextrin	E2 N-Acetyl-D-Glucosamine	E3 γ-Hydroxybutyric Acid	E4 L-Threonine	E1 α-Cyclodextrin	E2 N-Acetyl-D-Glucosamine	E3 γ-Hydroxybutyric Acid	E4 L-Threonine	E1 α-Cyclodextrin	E2 N-Acetyl-D-Glucosamine	E3 γ-Hydroxybutyric Acid	E4 L-Threonine
F1 Glycogen	F2 D-Glucosaminic Acid	F3 Itaconic Acid	F4 Glycyl-L-Glutamic Acid	F1 Glycogen	F2 D-Glucosaminic Acid	F3 Itaconic Acid	F4 Glycyl-L-Glutamic Acid	F1 Glycogen	F2 D-Glucosaminic Acid	F3 Itaconic Acid	F4 Glycyl-L-Glutamic Acid
G1 D-Cellobiose	G2 Glucose-1-Phosphate	G3 α-Ketobutyric Acid	G4 Phenylethylamine	G1 D-Cellobiose	G2 Glucose-1-Phosphate	G3 α-Ketobutyric Acid	G4 Phenylethylamine	G1 D-Cellobiose	G2 Glucose-1-Phosphate	G3 α-Ketobutyric Acid	G4 Phenylethylamine
H1 α-D-Lactose	H2 D,L-α-Glycerol Phosphate	H3 D-Malic Acid	H4 Putrescine	H1 α-D-Lactose	H2 D,L-α-Glycerol Phosphate	H3 D-Malic Acid	H4 Putrescine	H1 α-D-Lactose	H2 D,L-α-Glycerol Phosphate	H3 D-Malic Acid	H4 Putrescine

Figure 1.2: The Biolog EcoPlate with a description of the 32 carbon sources.

Experimental design				Carbon sources									
Month	Depth	Treat	Rep	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
2	1	1	1	0	1	1	0	1	0	1	1	1	0
2	1	1	2	0	1	1	1	1	1	0	1	1	0
2	1	1	3	0	1	1	1	0	0	0	0	1	0
2	1	2	1	0	1	1	1	0	1	0	1	1	0
2	1	2	2	0	1	1	1	1	1	0	1	1	0
2	1	2	3	0	1	1	1	0	1	0	1	1	0
2	1	3	1	0	1	1	1	0	1	0	1	1	0
2	1	3	2	0	1	1	1	0	0	0	1	1	0
2	1	3	3	0	1	1	1	1	1	0	1	1	0
2	1	4	1	0	1	1	1	0	1	0	1	1	0
2	1	4	2	0	1	1	1	1	1	0	1	1	0
2	1	4	3	0	1	1	1	0	1	0	1	1	0
2	1	5	1	0	1	1	1	0	1	0	1	1	0
2	1	5	2	0	1	1	1	0	0	0	1	1	0
2	1	5	3	0	1	1	1	0	0	0	1	1	0
2	1	6	1	0	1	1	1	0	1	0	1	0	0
2	1	6	2	0	0	1	1	1	1	1	0	1	0
2	1	6	3	0	1	1	1	0	0	0	1	0	0
2	1	7	1	0	1	1	1	0	1	0	1	1	0
2	1	7	2	0	1	1	1	0	1	0	1	1	0
2	1	7	3	0	1	1	1	1	0	0	1	1	0
2	1	8	1	0	1	1	1	0	1	1	1	1	0
2	1	8	2	0	1	1	1	0	0	0	1	1	0
2	1	8	3	0	1	1	1	0	1	0	1	1	0
2	1	9	1	0	1	1	1	0	0	0	1	1	0
2	1	9	2	0	1	1	1	1	1	0	1	1	0
2	1	9	3	0	1	1	1	0	0	0	1	1	0
2	1	10	1	0	1	1	1	0	0	0	1	1	0
2	1	10	2	0	1	1	1	0	0	0	1	1	0
2	1	10	3	0	1	1	1	0	0	0	1	1	0
2	1	11	1	0	1	1	1	0	1	0	1	1	0
2	1	11	2	0	1	1	1	0	1	0	1	1	0
2	1	11	3	0	1	1	1	0	0	0	1	1	0
2	1	12	1	0	1	1	1	0	0	0	1	1	0
2	1	12	2	0	1	1	1	0	1	0	1	1	0
2	1	12	3	0	1	1	1	0	0	0	1	1	0

Figure 1.3: An Excel extraction of the binary measurements in the Biolog data.

The micro organisms found in the soil digests the carbon in the EcoPlate. If digested, the organism releases a substance that turns the chemicals in the EcoPlate into a purple colour. Thus, the purple colours in Figure 1.1 are indications that there was microbial activity. The data captured in this experiment are binary (presence or absence of microbial activity). The value 1 in the data indicates that the colour in the EcoPlate turned purple and the value 0 indicates that there was no activity at all.

The data set from this experiment is multidimensional. The binary nature of the measurement makes it almost impossible to analyze the data with conventional multivariate statistical methods. Part of this thesis is to analyze the Biolog data using appropriate methods that have been developed for such data. In Chapter 6 we will perform an exploratory analysis on the data as well as an inferential analysis.

1.3 The Barents Fish data

The Barents Fish data was obtained from an observational study in the Barents Sea, north of Russia and Norway. A picture of the region is given in Figure 1.4. The grey shaded area is the region in which the data was observed. The area was divided into 89 sub-regions (stations) and each station was documented as an observation in the data. At each of the stations the following two sets of data were recorded.

The first set consists of 4 numerical variables, which are Latitude, Longitude, Depth (in metres) and Temperature ($^{\circ}\text{C}$). These environmental variables will be called the explanatory variables. The second set consists of count data. Different fish species, a total of 32, were observed at each of the 89 stations. The number of species observed at each station was counted. A list of these species is given in Table 1.1 together with the abbreviations that will be used for the data. An extraction of the Barents Fish data is given in Figure 1.5. This data were recorded in April-May 1997 over a 3 week period.

The purpose of this study is to examine the relationship among the different fish species and the environmental variables. In Chapter 6 we will perform a canonical correspondence analysis on this data to obtain the necessary answers.

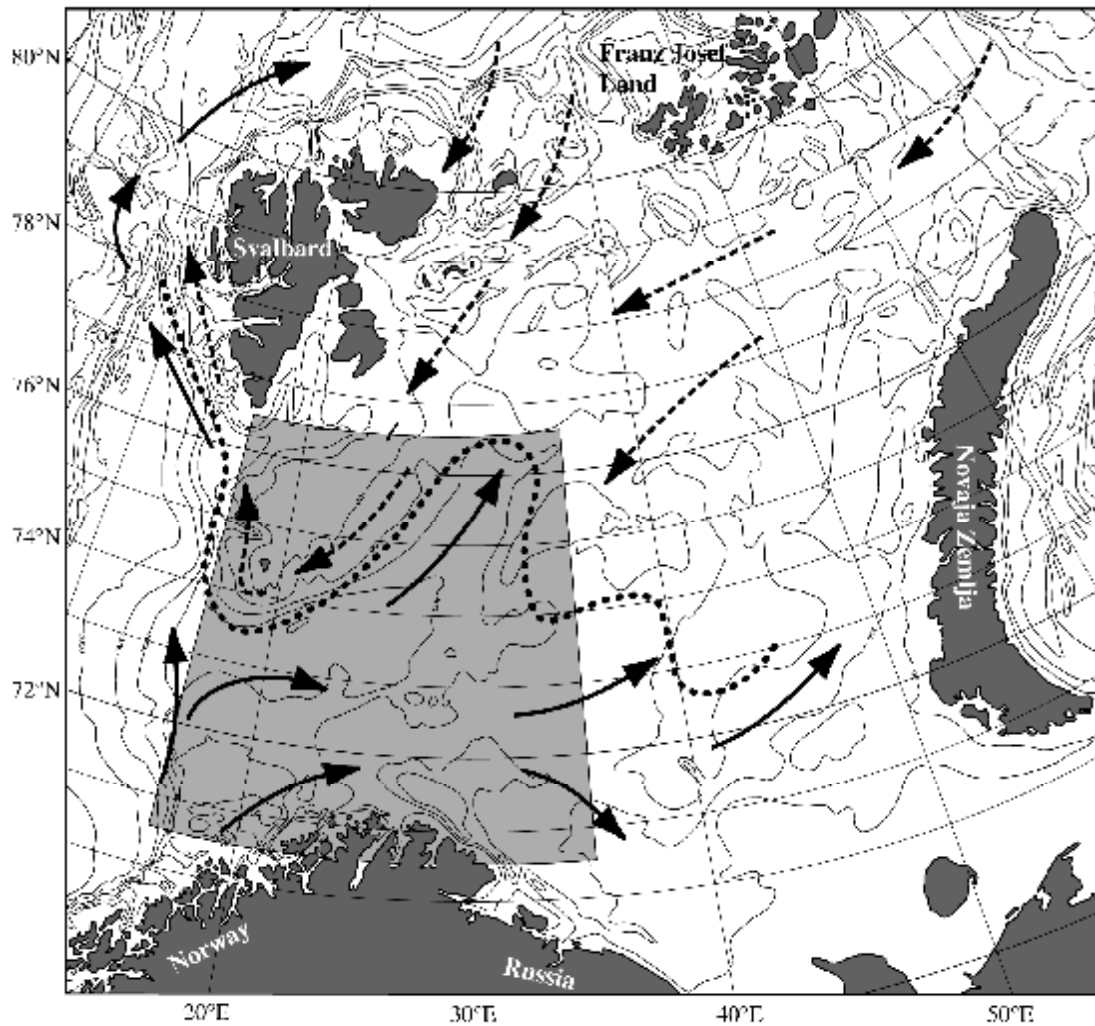


Figure 1.4: The map showing the sampling area in the Barents Sea. The site is north of Norway and Russia.

Table 1.1: List of species and their abbreviations in the data.

Abb.	Scientific name	Family	Common name
<i>An de</i>	<i>Anarhichas denticulatus</i>	Anarhichadidae	Jelly wolffish/Arctic catfish
<i>An lu</i>	<i>Anarhichas lupus</i>	Anarhichadidae	Wolffish/Atlantic catfish
<i>An mi</i>	<i>Anarhichas minor</i>	Anarhichadidae	Spotted wolffish/catfish
<i>Le de</i>	<i>Leptagonus decagonus</i>	Agonidae	Atlantic poacher
<i>Cl ha</i>	<i>Clupea harengus</i>	Clupeidae	Herring
<i>Ar at</i>	<i>Artediellus atlanticus</i>	Cottidae	Atlantic hookear sculpin
<i>Tr spp</i>	<i>Triglops murrayi</i>	Cottidae	Moustache/mailed sculpin
<i>Tr spp</i>	<i>Triglops pingelii</i>	Cottidae	Ribbed sculpin
<i>Ca re</i>	<i>Careproctus reinhardti</i>	Cyclopteridae	Longfin seasnail
<i>Cy lu</i>	<i>Cyclopterus lumpus</i>	Cyclopteridae	Lumpsucker
<i>Bo sa</i>	<i>Boreogadus saida</i>	Gadidae	Polar cod
<i>Ga mo</i>	<i>Gadus morhua</i>	Gadidae	Cod
<i>Me ae</i>	<i>Melanogrammus aeglefinus</i>	Gadidae	Haddock
<i>Mi po</i>	<i>Micromesistius poutassou</i>	Gadidae	Blue whiting
<i>Tr es</i>	<i>Trisopterus esmarkii</i>	Gadidae	Norway pout
<i>Be gl</i>	<i>Benthoosema glaciale</i>	Myctophidae	Glacier lanternfish
<i>Ma vi</i>	<i>Mallotus villosus</i>	Osmeridae	Capelin
<i>Pa bo</i>	<i>Pandalus borealis</i>	Pandalidae	Shrimp
<i>No rk</i>	<i>Notolepis rissoi krøyeri</i>	Paralepididae	White barracudina
<i>Hi pl</i>	<i>Hippoglossoides platessoides</i>	Pleuronectidae	Long rough dab
<i>Re hi</i>	<i>Reinhardtius hippoglossoides</i>	Pleuronectidae	Greenland halibut
<i>Ra ra</i>	<i>Raja radiata</i>	Rajidae	Starry ray
<i>Se ma</i>	<i>Sebastes marinus</i>	Scorpaenidae	Golden redfish
<i>Se me</i>	<i>Sebastes mentella</i>	Scorpaenidae	Deepwater redfish
<i>Le ma</i>	<i>Leptoclinus maculatus</i>	Stichaeidae	Spotted snake blenny
<i>Lu la</i>	<i>Lumpenus lampraetaeformis</i>	Stichaeidae	Snake blenny
<i>Ly es</i>	<i>Lycodes esmarkii</i>	Zoarcidae	Esmark's eelpout
<i>Ly eu</i>	<i>Lycodes eudipleurostictus</i>	Zoarcidae	Eelpout (ncn)
<i>Ly pa</i>	<i>Lycodes pallidus</i>	Zoarcidae	Pale eelpout
<i>Ly re</i>	<i>Lycodes reticulatus</i>	Zoarcidae	Arctic eelpout
<i>Ly se</i>	<i>Lycodes seminudus</i>	Zoarcidae	Eelpout (ncn)
<i>Ly va</i>	<i>Lycodes vahlii</i>	Zoarcidae	Vahl's eelpout

Station	Environmental characteristics				Station	Species abundance							
	Latitude	Longitude	Depth	Temperature		Pa_bo	Re_hi	An_de	An_mi	Hi_pl	An_lu	Me_ae	Ra_ra
356	71.10	22.43	349	3.95	356	604	0	0	0	31	0	108	0
357	71.32	23.68	382	3.75	357	3 607	0	0	0	4	0	110	0
358	71.60	24.90	294	3.45	358	1 699	0	0	0	27	0	788	0
359	71.27	25.88	304	3.65	359	2 246	0	0	1	13	0	295	0
363	71.52	28.12	384	3.35	363	9 504	0	0	0	23	0	13	2
364	71.48	29.10	344	3.65	364	4 690	1	0	0	20	0	97	0
365	71.10	29.92	347	3.55	365	7 573	0	0	0	21	0	220	0
366	71.03	30.87	300	3.85	366	2 874	0	0	0	35	0	373	0
367	71.32	31.20	260	2.95	367	700	0	2	1	74	0	118	0
368	71.30	32.15	256	3.35	368	1 023	0	3	0	39	0	336	1
369	71.22	33.15	254	2.55	369	204	3	1	0	29	0	36	9
370	71.58	32.37	297	2.65	370	2 322	1	4	4	70	1	9	4
371	71.68	31.25	332	2.85	371	5 774	1	0	4	57	0	40	2
372	71.72	30.77	358	1.95	372	1 998	5	0	1	50	0	23	6
373	72.02	31.67	320	1.65	373	17 886	7	6	1	98	0	2	5
375	72.25	32.93	285	1.25	375	8 291	1	1	1	92	0	0	9
376	72.45	34.32	285	0.15	376	27 962	2	2	0	288	0	0	4
377	72.72	35.60	234	0.65	377	7 465	0	0	0	61	0	0	3
378	72.83	34.58	228	0.55	378	2 627	0	0	0	51	0	2	0
379	72.90	33.33	227	0.35	379	3 346	0	0	3	61	1	0	0
380	72.62	32.05	268	0.95	380	9 783	1	0	2	152	0	4	6
381	72.35	30.78	295	2.85	381	4 459	1	5	0	100	0	127	5
382	72.08	29.43	290	3.05	382	5 902	0	0	0	48	0	243	4
383	71.82	28.17	308	3.25	383	3 660	0	0	0	16	0	122	0
384	71.55	27.00	350	3.35	384	13 337	0	0	0	13	0	231	0
385	71.93	26.07	280	3.35	385	2 225	0	2	0	46	0	119	0
386	72.22	27.25	234	3.15	386	231	0	0	0	34	0	1 179	0

Figure 1.5: An Excel extraction of the numerical variables and fish counts in the Barents Fish data.

1.4 The aim of the thesis

The aim of this thesis can be summarized by the following points:

- To explain various popular multivariate techniques that can be used for the exploratory analysis of count and binary data.
- To discuss a technique for inference when using count and binary data. This technique is equivalent to the analysis of variance for numerical data.
- To illustrate how these techniques can be applied using the R software package (<http://www.r-project.org/>). Clear demonstrations of the functions, analysis and graphical features will be given.
- These exploratory techniques will be employed to analyze the Biolog data. An inference method is also used to analyze and understand this data.
- The Barents Fish data is used to demonstrate how to analyze two sets of data (numerical and count data). The aim here is to study the relationship between these two sets of data.

1.5 Layout of the thesis

Chapter 2 introduces correspondence analysis as well as canonical correspondence analysis. The algebraic development for these techniques is given in detail. This chapter not only shows how correspondence analysis is constructed, but also demonstrates how it extends to canonical correspondence analysis. An example is used to illustrate how these analyses are performed using the `ca()`, `anacor()` and `cca()` functions in R. In Chapter 3 we deal with cluster analysis. This chapter starts by discussing various distance and dissimilarity measures. Different distance (dissimilarity) measures are used for different types of data and choosing the appropriate measure will be explained. Four clustering methods are discussed and also illustrated with an example in R using the `hclust()` function. A metric as well as a nonmetric multidimensional scaling technique is given in Chapter 4. Applications of these techniques are performed using the packages `cmdscale()` and `isoMDS()`. In Chapter 5 we explain a non-parametric inference technique called the analysis of distance. This technique is similar to the analysis of variance in the univariate and

Chapter 1: Introduction

multivariate cases. An example of how this techniques is applied is given using the `adonis()` function. Chapter 6 is devoted to the analysis of the Biolog and Barents Fish data. The techniques discussed in Chapters 2 to 5 are employed to perform the analysis. Attention is also given to the interpretation of the output. Chapter 7 is a general conclusion of the thesis and some recommendations for future research are also given.

Chapter 2

Simple and Canonical correspondence analysis

2.1 Introduction

Simple correspondence analysis (CA) is a multivariate statistical method which is used for exploratory data analysis. It was developed at the end of the 1960's by a French statistician Jean-Paul Benzécri for linguistic applications (Benzécri, 1973). Correspondence analysis is used to analyse simple two-way and multi-way contingency tables. The aim of the correspondence analysis is to study the relationships between the rows and columns in a contingency table.

Correspondence analysis is a nonparametric technique which makes no distributional assumptions. The type of variables used in a correspondence analysis is usually categorical variables and if continuous, the variables must be categorized into ranges. The raw data for a correspondence analysis is in the form of a contingency table with nonnegative counts (frequencies).

Canonical correspondence analysis (CCA) on the other hand, is a correspondence analysis that is performed in a restricted or constrained space^a (Greenacre, 2007). While simple correspondence analysis uses only a contingency table, canonical correspondence analysis requires an additional set of data in the form of numerical variables measured on the same observations from which the contingency table was obtained.

The aim of canonical correspondence analysis is to include these additional numerical variables (often referred to as explanatory variables) as part of the CA solution. This has been made possible by “forcing the CA solution to be a linear function of explanatory variables” (Greenacre, 2007). By taking into account the explanatory

^a For more information on CA and CCA, the constrained and unconstrained space see Greenacre (2007)

variables, CA becomes constrained and therefore the name canonical (or constrained) correspondence analysis.

The results for CA and CCA are very similar. However, CCA can give us much more information using the explanatory variables. CCA originated from the field of Ecology (ter Braak, 1986) and has been applied quite extensively by Ecologists and many other scientists. In this chapter we will discuss both CA and CCA as methods of exploratory analysis. In Section 2.2 we explain the algebra underlying simple CA. This discussion is followed by a description of measures of goodness-of-fit for CA, called the inertia and Benzécri distances. Note that these goodness-of-fit measures can also be used for CCA. We also illustrate how CA can be applied in R using two packages, namely `anacor` and `ca`. In Section 2.5 the formulation of CCA is discussed and its extension from simple CA is shown. In Section 2.6 we illustrate the application of CCA in the R packages `anacor` and `vegan`. Finally, in Section 2.7, we discuss some permutation test in CCA.

2.2 Simple correspondence analysis

Let \mathbf{X} denote an $I \times J$ contingency table with elements x_{ij} , where $I > J$. A matrix of proportions is derived from this contingency table by dividing each of the elements in \mathbf{X} by the grand total $n = \sum_{i=1}^I \sum_{j=1}^J x_{ij}$. This matrix is known as a correspondence matrix, denoted by

$$\mathbf{P} = \frac{1}{n} \mathbf{X}, \text{ with elements } p_{ij} = \frac{x_{ij}}{n}.$$

The row totals and the column totals in the correspondence matrix are known as the row masses $\left(\mathbf{r} \right)_{I \times 1}$ and column masses $\left(\mathbf{c} \right)_{J \times 1}$, respectively. These vectors are obtained from \mathbf{P} as follows:

$$\mathbf{r} = \mathbf{P} \mathbf{1}, \text{ with elements } r_i = \sum_{j=1}^J p_{ij} \text{ for } i=1,2,\dots,I \text{ and}$$

$$\mathbf{c} = \mathbf{P}' \mathbf{1}, \text{ with elements } c_j = \sum_{i=1}^I p_{ij} \text{ for } j=1,2,\dots,J .$$

Let \mathbf{D}_r and \mathbf{D}_c be diagonal matrices having \mathbf{r} and \mathbf{c} on the diagonal respectively. Thus $\mathbf{D}_r = \text{diag}(r_1, r_2, \dots, r_I)$ and $\mathbf{D}_c = \text{diag}(c_1, c_2, \dots, c_J)$. These diagonal matrices are known as row mass and column mass diagonal matrices. From these diagonal matrices we define the following square root matrices which will be used for scaling (weighting) purposes later:

$$(a) \mathbf{D}_r^{1/2} = \text{diag}(\sqrt{r_1}, \sqrt{r_2}, \dots, \sqrt{r_I}) \text{ and } \mathbf{D}_r^{-1/2} = \text{diag}\left(\frac{1}{\sqrt{r_1}}, \frac{1}{\sqrt{r_2}}, \dots, \frac{1}{\sqrt{r_I}}\right).$$

$$(b) \mathbf{D}_c^{1/2} = \text{diag}(\sqrt{c_1}, \sqrt{c_2}, \dots, \sqrt{c_J}) \text{ and } \mathbf{D}_c^{-1/2} = \text{diag}\left(\frac{1}{\sqrt{c_1}}, \frac{1}{\sqrt{c_2}}, \dots, \frac{1}{\sqrt{c_J}}\right).$$

Correspondence analysis is formulated as a weighted least squares problem (Johnson and Wichern, 2007) where we want to determine the matrix $\hat{\mathbf{P}} = \{\hat{p}_{ij}\}$ by minimizing the sum of squares

$$\sum_{i=1}^I \sum_{j=1}^J \frac{(p_{ij} - \hat{p}_{ij})^2}{r_i c_j} = \text{tr} \left[\left(\mathbf{D}_r^{-1/2} (\mathbf{P} - \hat{\mathbf{P}}) \mathbf{D}_c^{-1/2} \right) \left(\mathbf{D}_r^{-1/2} (\mathbf{P} - \hat{\mathbf{P}}) \mathbf{D}_c^{-1/2} \right)' \right].$$

To obtain $\hat{\mathbf{P}}$ that minimizes this equation a singular value decomposition based on \mathbf{P} is commonly used (for the proof see Result 12.1 in Johnson and Wichern, 2007, p.719). This result shows that $\hat{\mathbf{P}} = \mathbf{r}\mathbf{c}'$ is the best rank 1 approximation to \mathbf{P} and is often used as the estimate $\hat{\mathbf{P}}$ when performing CA. For our discussion and analyses (Section 2.4) we will use $\hat{\mathbf{P}} = \mathbf{r}\mathbf{c}'$. Define the scaled matrix of $(\mathbf{P} - \mathbf{r}\mathbf{c}')$ as

$$\mathbf{S} = \mathbf{D}_r^{-1/2} (\mathbf{P} - \mathbf{r}\mathbf{c}') \mathbf{D}_c^{-1/2}, \quad (2.1)$$

which is also known as the matrix of standardized residuals. Because of this particular scaling, a singular value decomposition (SVD) is performed on \mathbf{S} such that

$$\mathbf{S} = \sum_{k=1}^J I_k \mathbf{u}_k \mathbf{v}_k' = \mathbf{U} \mathbf{\Lambda} \mathbf{V}', \quad (2.2)$$

where I_k denote the singular values. The above matrices from the SVD are $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_I]$, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_J]$ and

$$\mathbf{\Lambda} = \begin{bmatrix} I_1 & 0 & \mathbf{L} & 0 \\ 0 & I_2 & \mathbf{L} & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 0 & 0 & \mathbf{L} & I_J \end{bmatrix}. \quad (2.3)$$

It is common in correspondence analysis to plot the first two or three columns of the following matrices:

$$\mathbf{F} = \mathbf{D}_r^{-1} (\mathbf{D}_r^{1/2} \mathbf{U}) \mathbf{\Lambda} \quad \text{and} \quad \mathbf{G} = \mathbf{D}_c^{-1} (\mathbf{D}_c^{1/2} \mathbf{V}) \mathbf{\Lambda} \quad (2.4)$$

(which can also be expressed as $I_k \mathbf{D}_r^{-1/2} \mathbf{u}_k$ and $I_k \mathbf{D}_c^{-1/2} \mathbf{v}_k$) for $k=1,2$, or maybe 3. The plot of \mathbf{F} (row coordinates) and \mathbf{G} (column coordinates) on the same graph is referred to as a joint plot, symmetric plot or a CA plot. This plot describes the relationship between the rows and the columns of the contingency matrix, \mathbf{X} . Figures 2.1 and 2.3 are examples of this CA plot.

2.3 Inertia and Benzécri distances

It is common in correspondence analysis to determine the goodness-of-fit. In other words, how well the variation in the CA plot describes the variation in the raw data. In this section we will discuss two measures of determining the goodness-of-fit in correspondence analysis. Firstly we will explain the inertia and secondly the Benzécri distances.

Chapter 2: Simple and Canonical correspondence analysis

The total inertia is a measure of the variation in the contingency table or the raw data. It is formulated as the weighted sums of squares (see Johnson and Wichern, 2007)

$$\text{tr} \left[\mathbf{D}_r^{-1/2} (\mathbf{P} - \mathbf{rc}') \mathbf{D}_c^{-1/2} (\mathbf{D}_r^{-1/2} (\mathbf{P} - \mathbf{rc}') \mathbf{D}_c^{-1/2})' \right] = \sum_{i=1}^I \sum_{j=1}^J \frac{(p_{ij} - r_i c_j)^2}{r_i c_j} = \sum_{k=1}^{J-1} I_k^2. \quad (2.5)$$

The total inertia is divided into two parts. The first part is the inertia associated with the first K dimensions and is obtained by $\sum_{k=1}^K I_k^2$. The second part is the remaining portion of the total inertia which is not accounted for by the first K dimensions. This is obtained by $\sum_{k=K+1}^{J-1} I_k^2$ and is known as the residual inertia. Thus a measure of goodness-of-fit in correspondence analysis is defined as the proportion of inertia explained by the first K dimensions relative to the total inertia and is given by

$$\frac{\sum_{k=1}^K I_k^2}{\sum_{k=1}^{J-1} I_k^2}. \quad (2.6)$$

A high value of this measure represents a good fit in simple (and canonical) correspondence analysis.

A second (graphical) measure which is used to determine the goodness-of-fit makes use of Benzécri distances (de Leeuw and Mair, 2009). The Benzécri distance between rows i and i' in the contingency table \mathbf{X} is defined as

$$d^2(i, i') = \sum_{j=1}^J \left(\frac{x_{ij}}{x_{i\bullet}} - \frac{x_{i'j}}{x_{i'\bullet}} \right)^2 / x_{\bullet j}, \quad i, i' = 1, 2, \dots, I, \quad (2.7)$$

where $x_{i\bullet}$ = total of row i

$x_{i'\bullet}$ = total of row i'

$x_{\bullet j}$ = total of column j .

Next the Euclidean distance between rows i and i' of the first K dimensions of \mathbf{F} is obtained. Plotting the Benzécri distance and the Euclidean distance for each of the row pairs gives a Benzécri plot. In a similar way the Benzécri distance is obtained on the columns of \mathbf{X} and a Euclidean distance on the first K dimensions of \mathbf{G} . Figure 2.2 contains examples of the Benzécri plot of the rows and columns. If the plot of the distances lies close to the 45° line, then the correspondence analysis has a good fit.

2.4 Performing a correspondence analysis in R

In this section we explain the application of correspondence analysis using the R software (R Development Core Team, 2009). Two R packages are discussed namely the `ca` package developed by Nenadic and Greenacre (2007) and the `anacor` package developed by de Leeuw and Mair (2009). The `ca` package allows for the computation of simple correspondence analysis based on the SVD. The `ca` package also includes the multiple and joint correspondence analysis. Both these packages provide two and three dimensional plots (see Figure 2.2 and Figure 2.5). More details about the `ca` package can be found in Nenadic and Greenacre (2007). The `anacor` package also allows for the computation of simple and canonical CA for incomplete tables (tables with missing values) based on SVD.

The `ca` package and the `anacor` package give similar output, but the `anacor` package has more features than the `ca` package. The `anacor` package performs both simple and canonical CA. It offers additional possibilities for scaling the row and column scores in simple and canonical CA (see Leeuw and Mair, 2009). Note that different scaling methods lead to different interpretations of the distances in the CA plot. It also has an additional graphical feature which includes ellipsoids and the Benzécri plots. It also allows for missing values, which are imputed using Nora's algorithm (Nora, 1975). More details about the `anacor` package can be found in de Leeuw and Mair (2009).

 Chapter 2: Simple and Canonical correspondence analysis

To illustrate correspondence analysis using the two above mentioned packages, we will make use of the smoke data set (Greenacre, 2007). This data set is part of the `ca` package and the following R commands load the data set:

```
R> library(ca) # loading the ca package
R> data(smoke) # loading the data set
R> smoke
  none light medium heavy
SM    4    2     3     2
JM    4    3     7     4
SE   25   10    12     4
JE   18   24    33    13
SC   10    6     7     2
```

This data set contains frequencies (counts) of smoking habits (none, light, medium, and heavy) for different staff groups (senior managers (SM), junior managers (JM), senior employees (SE), junior employees (JE) and secretaries (SC)) in a fictional company. The purpose of the correspondence analysis is to determine if there is any association between the smoking habits and staff groups.

2.4.1 The `anacor` package

This package contains the function also called `anacor()` which is used to perform correspondence analysis. The main arguments of the function is given below

```
R> anacor(tab, ndim = 2, row.covariates, col.covariates,
         scaling = c("Benzecri", "Benzecri"), eps = 1e-06)
```

where `tab` is the contingency table (missing values are coded as NA) and `ndim` is used to specify the number of dimensions. The following R instructions load the package and perform the correspondence analysis on the smoke data.

```
R> library("anacor") # loading the package
R> req1<-anacor(smoke, ndim=2)
```

Chapter 2: Simple and Canonical correspondence analysis

```
R> req1 # CA output/results

CA fit:
Sum of eigenvalues: 0.08477629
Benzecri RMSE rows: 2.412250e-05
Benzecri RMSE columns: 7.797221e-06

Total chi-square value: 16.442 # total inertia

Chi-Square decomposition:
      Chisq Proportion Cumulative Proportion
Component 1 14.429      0.878                0.878
Component 2  1.933      0.118                0.995
Component 3  0.080      0.005                1.000
```

The output above contains the squared singular values (2.3), the total inertia (2.5) and the proportion of variation explained by the dimensions (2.6). A total of 99.5% of the variation in the contingency table is explained by the first two dimensions. The next instruction plots the two dimensional CA plot, Figure 2.1:

```
R> plot(req1)
```

The blue labels represent the columns and the red labels represent the rows of the smoke data. It seems like the senior employees (SE) do not smoke (none). Junior managers (JM) seem to be heavy smokers while junior employees (JE) are medium type smokers. However, the senior managers (SM) and the secretaries (SE) do not seem to have any clearly identifiable smoking habits. They could be classified in any of the smoking categories.

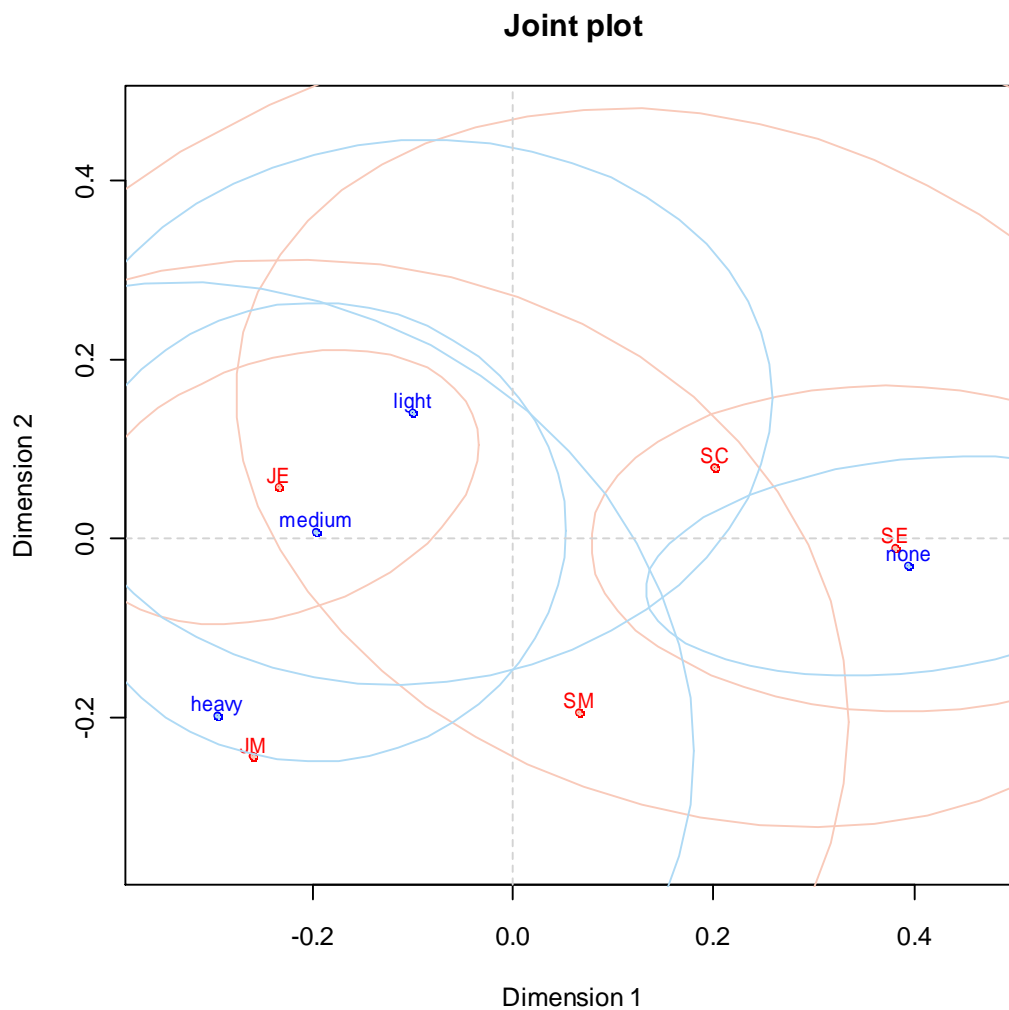


Figure 2.1: CA plot (joint plot) of the smoke data set using `anacor()`.

Additionally, the `anacor` package also allows us to create a three dimensional CA plot for correspondence analysis by using the function `plot3d()`. The main arguments for this plot function are,

```
R> plot3d(x, plot.type, plot.dim = c(1,2,3), col.r = "RED",
         col.c = "BLUE", arrows = TRUE, xlab, ylab, zlab, main, ...)
```

where `x` is a correspondence analysis object obtained from the `anacor()` function and the `plot.type` option is used to specify the type of plot required (the joint plot is the default plot type). Note that object `x` in the `plot3d()` function needs to have `ndim=3` before using `plot3d()`. The following instructions are used to create a three dimensional CA plot of the smoke data (the plot is given in Figure 2.2):

Chapter 2: Simple and Canonical correspondence analysis

```
R> req2<-anacor(smoke,ndim=3)
R> plot3d(req2)
```

The interesting property about the plot in R is that it can be rotated manually to obtain the best three dimensional view of the CA plot of the rows and column profiles.

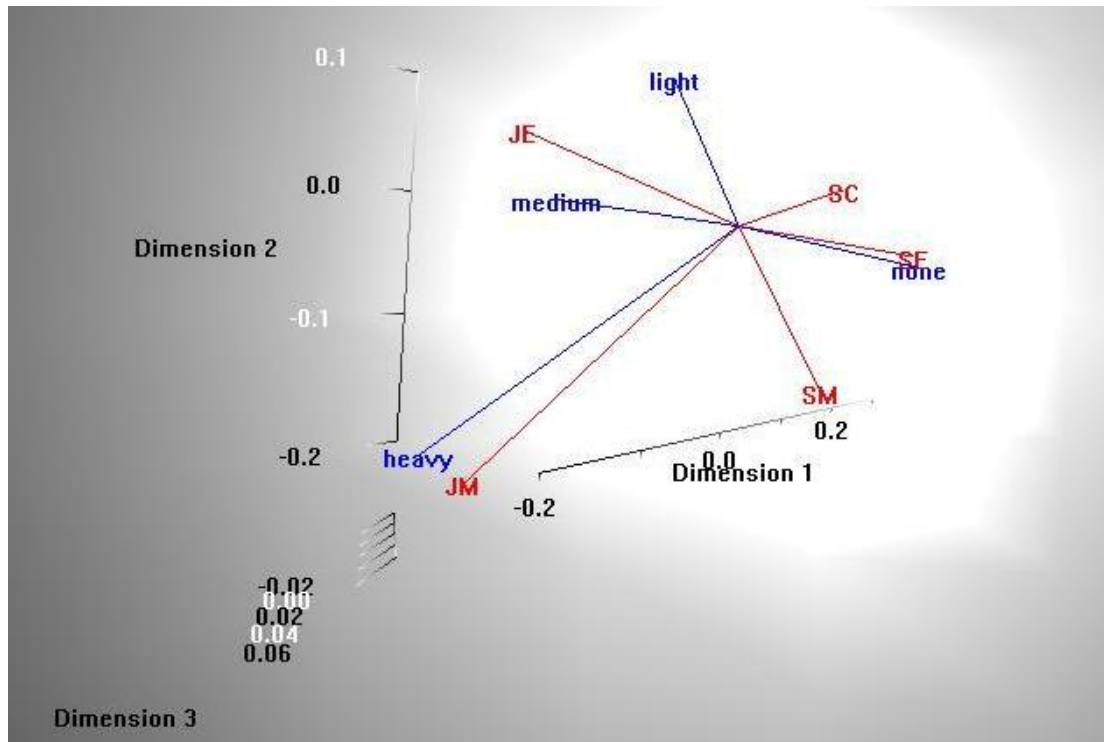


Figure 2.2: The three dimensional CA plot of the smoke data using `plot3d()`.

To obtain the Benzécri plots, we use the following instruction (based on the two dimensional correspondence analysis object `req1`):

```
R> plot(req1,plot.type="benzplot")
```

The resulting figures for the rows and columns are displayed in Figure 2.3. The fitted distances are the Euclidean distances while the observed distances represent the Benzécri distances (2.7). The plot of the fitted vs the observed distances lie close to the straight line (45° line), indicating that the two dimensional correspondence analysis is a good display of the smoke data. This is in agreement with the high inertia value *i.e.* 99.5% of the variation explained by the first two dimensions.

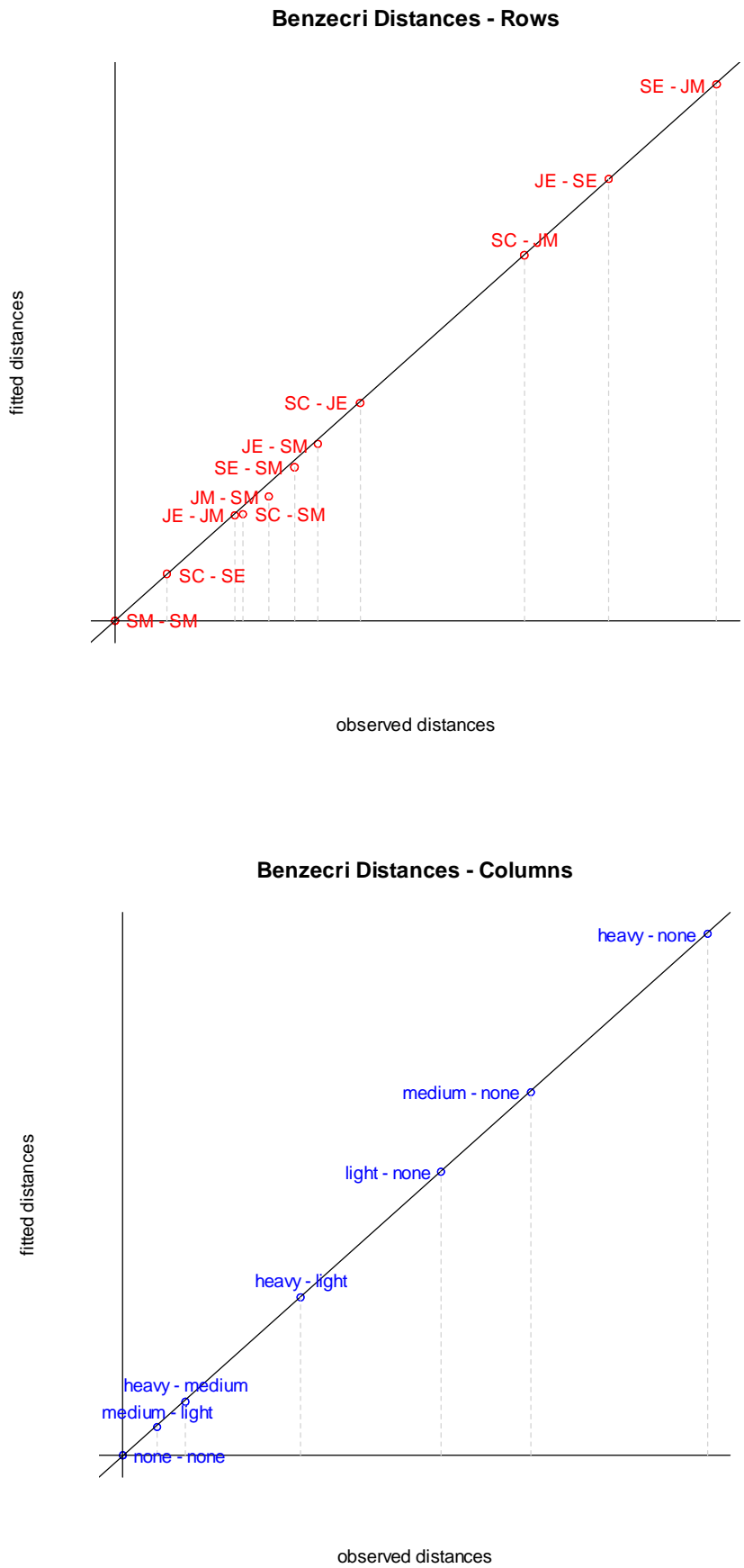


Figure 2.3: Benzécri plots of rows and columns using `anacor()`.

2.4.2 The ca package

This package uses the function `ca()`. Its main arguments are

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

In this function the argument `obj` is the contingency table and `nd` is used to specify the number of dimensions. The following R instructions load this package and perform the correspondence analysis:

```
R> library(ca) # loading the package
R> req3<-ca(smoke,nd=2) # perform CA
```

```
R> req3 # CA output

Principal inertias (eigenvalues):
      1      2      3
Value  0.074759 0.010017 0.000414
Percentage 87.76%  11.76%  0.49%

Rows:
      SM      JM      SE      JE      SC
Mass  0.056995 0.093264 0.264249 0.455959 0.129534
ChiDist 0.216559 0.356921 0.380779 0.240025 0.216169
Inertia 0.002673 0.011881 0.038314 0.026269 0.006053
Dim. 1 -0.240539 0.947105 -1.391973 0.851989 -0.735456
Dim. 2 -1.935708 -2.430958 -0.106508 0.576944 0.788435

Columns:
      none    light    medium    heavy
Mass  0.316062 0.233161 0.321244 0.129534
ChiDist 0.394490 0.173996 0.198127 0.355109
Inertia 0.049186 0.007059 0.012610 0.016335
Dim. 1 -1.438471 0.363746 0.718017 1.074445
Dim. 2 -0.304659 1.409433 0.073528 -1.975960
```


Chapter 2: Simple and Canonical correspondence analysis

The default output of `ca()` above is quite differently displayed to the output of `anacor()` showed in the previous section. However, the values of the inertia and the percentage of variation explained by the dimensions are given. Also given in the output are the row and column coordinates (labelled `Dim.1` and `Dim.2`) of the CA plot, Figure 2.4. The next instruction creates the CA plot:

```
R> plot(req3)
```

The interpretation of this figure is the same as Figure 2.1.

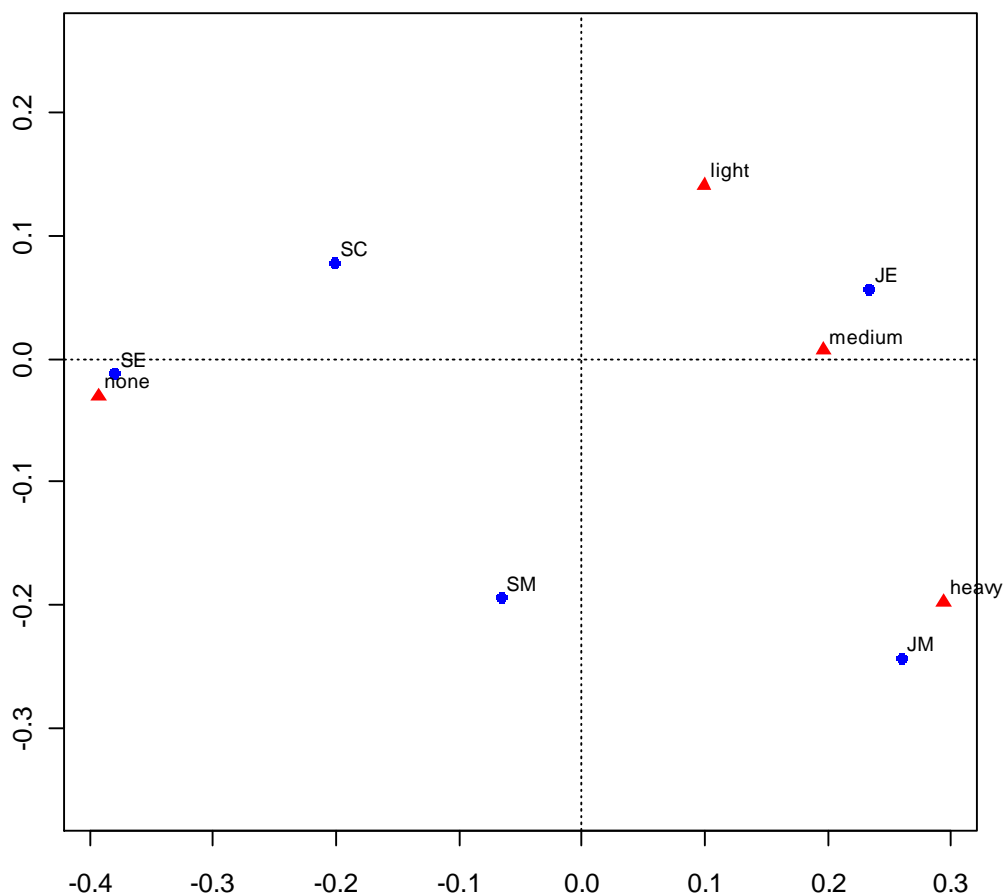


Figure 2.4: CA plot of the smoke data set using `ca()`.

Chapter 2: Simple and Canonical correspondence analysis

Similar to the `anacor` package, the `ca` package also allows us to create a three dimensional CA plot. This is done by using the function `plot3d.ca()`. The main arguments of this function are given below:

```
R> plot3d.ca(x, dim = c(1, 2, 3), map = "symmetric",
            what = c("all", "all"), contrib = c("none", "none"),
            col = c("#6666FF", "#FF6666"), labcol = c("#0000FF",
            "#FF0000"), pch = c(16, 1, 18, 9), labels = c(2, 2),
            sf = 0.00002, arrows = c(FALSE, FALSE), ...)
```

The object `x` is an object obtained from the `ca` function. To create the three dimensional CA plot with `plot3d.ca()` we need to use to a three dimensional correspondence analysis object which can be done by using the following instructions.

```
R> req4<-ca(smoke, nd=3)
R> plot3d.ca(req4)
```

Figure 2.5 is an example of the three dimensional CA plot, which can also be rotated manually in R to obtain the best view of the row and column profiles.

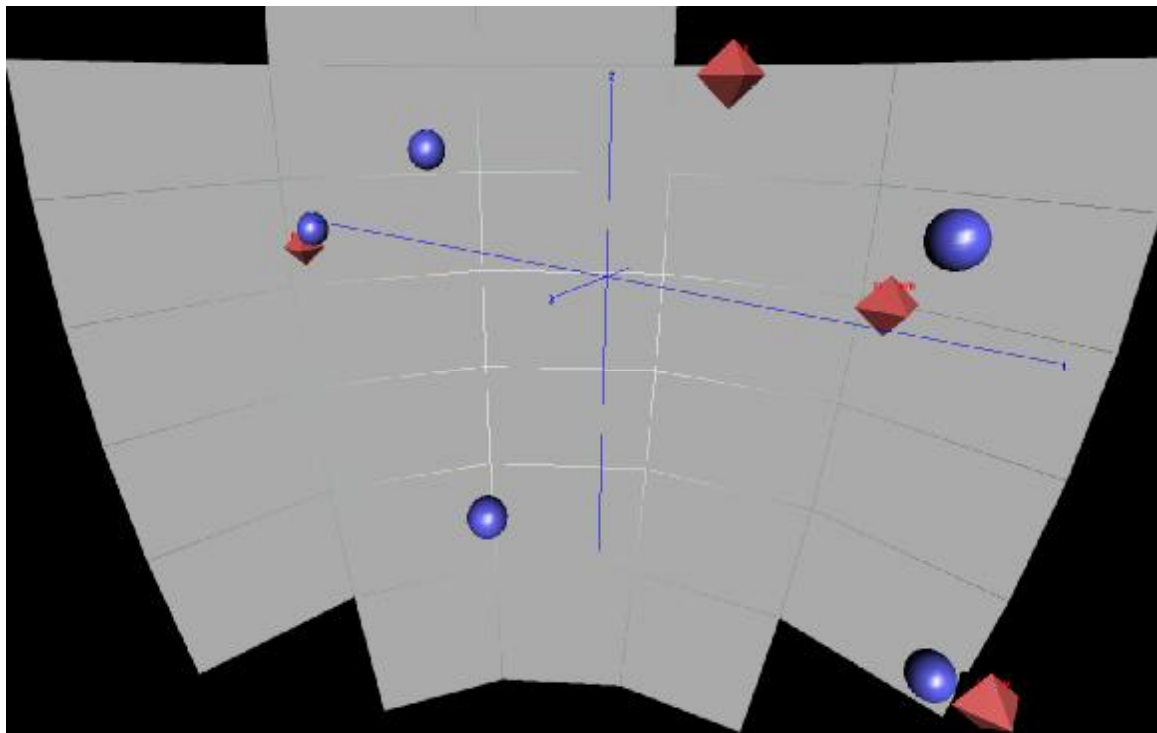


Figure 2.5: A three dimensional CA plot of the smoke data using `plot3d.ca()`.

2.5 Canonical correspondence analysis (CCA)

Canonical correspondence analysis (CCA) was introduced by Cajo ter Braak (ter Braak, 1986) for use in Ecology. Canonical (constrained) correspondence analysis is an extension of simple correspondence analysis described in Section 2.2. It has become quite useful in many applications involving two sets of data *i.e.* a frequency table and set of numerical data (recall that simple CA is based only on a frequency table). Another version of CCA was proposed by Legendre and Legendre (1998) and in this section we briefly explain this proposal.

In simple correspondence analysis, an $I \times J$ contingency table \mathbf{X} is used to obtain the correspondence matrix \mathbf{P} . The correspondence matrix \mathbf{P} is then used to define the matrix

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

Simple correspondence analysis is performed by doing a singular value decomposition on this matrix \mathbf{S} . For canonical correspondence analysis we have an additional set of numerical data (explanatory variables), which we will denote by the $I \times p$ matrix \mathbf{Y} , where p represents the number of variables in \mathbf{Y} . Performing a canonical correspondence analysis involves what is known as a weighted regression on the matrix of explanatory variables, \mathbf{Y} . The following paragraph explains how this is obtained.

Firstly \mathbf{Y} is centred by using the sums of the columns of $\mathbf{D}_r \mathbf{Y}$. Secondly the projection matrix \mathbf{Q} is obtained from the projection of \mathbf{S} onto \mathbf{Y} as follows:

$$\mathbf{Q} = \mathbf{D}_r^{1/2} \mathbf{Y} (\mathbf{Y}' \mathbf{D}_r \mathbf{Y})^{-1} \mathbf{Y}' \mathbf{D}_r^{1/2}.$$

A weighted regression of the matrix \mathbf{Q} on the matrix \mathbf{Y} is performed, which result in the following matrix of fitted values

$$\begin{aligned}\hat{\mathbf{Q}} &= \left[\mathbf{D}_r^{1/2} \mathbf{Y} (\mathbf{Y}' \mathbf{D}_r \mathbf{Y})^{-1} \mathbf{Y}' \mathbf{D}_r^{1/2} \right] \left[\mathbf{D}_r^{-1/2} (\mathbf{P} - \mathbf{r} \mathbf{c}') \mathbf{D}_c^{-1/2} \right] \\ &= \mathbf{Q} \mathbf{S}.\end{aligned}$$

CCA now entails doing a singular value decomposition on $\hat{\mathbf{Q}}$ (as apposed to \mathbf{S} for simple CA). Once the SVD of $\hat{\mathbf{Q}}$ is performed, the rest of CCA (see \mathbf{F} and \mathbf{G} in expression 2.4) is performed exactly the same as simple CA. A joint plot (or CCA plot) and the inertia for CCA are obtained in exactly the same way as for simple CA. However, it is customary to display the explanatory variables as arrows on the CCA plot in order to study the relationship between \mathbf{X} and \mathbf{Y} . See Figure 2.6 as an example. The arrows explaining this relationship are obtained as follows.

Using the row coordinates of the SVD on $\hat{\mathbf{Q}}$ (matrix \mathbf{F} in expression 2.4), we perform a regression analysis using one of the explanatory variables (as a dependent variable) and the row coordinates (as independent variables). The following illustrates how the regression works for the first two dimensions of $\mathbf{F} \Rightarrow (x_1, x_2)$ and explanatory variable y from \mathbf{Y} . Let $y = a + bx_1 + cx_2$ and $\bar{y} = a + b\bar{x}_1 + c\bar{x}_2$ then

$$\begin{aligned}y - \bar{y} &= b(x_1 - \bar{x}_1) + c(x_2 - \bar{x}_2) \\ y - \bar{y} &= b s_{x_1} \frac{(x_1 - \bar{x}_1)}{s_{x_1}} + c s_{x_2} \frac{(x_2 - \bar{x}_2)}{s_{x_2}} \\ y - \bar{y} &= b s_{x_1} x_1^* + c s_{x_2} x_2^* \\ \frac{y - \bar{y}}{s_y} &= y^* = \left(b \frac{s_{x_1}}{s_y} \right) x_1^* + \left(c \frac{s_{x_2}}{s_y} \right) x_2^*.\end{aligned}$$

The standardized regression coefficients $b s_{x_1} / s_y$ and $c s_{x_2} / s_y$ are then used as coordinates for the arrows on the CCA plot (see Figure 2.7). The CCA plot with the arrows is also referred to as a triplot (Greenacre, 2007). Note that on standardized data the intercept of the regression analysis is zero.

2.6 Performing a canonical correspondence analysis in R

CCA can be performed using one of the following two packages in R *i.e.* the `anacor` and `vegan` packages. As mentioned before, the `anacor` package was developed by Leeuw and Mair (2009) and it performs a CCA based on the method of ter Braak (1986). The `vegan` package (Oksanen *et al.*, 2009) was developed for use in Ecology and contains a vast number of statistical techniques including CCA. The paper by one of its developers Oksanen (2011) and is good reference on understanding the `vegan` package. The CCA found in `vegan` is based in the method proposed by Legendre and Legendre (1998) discussed in the previous section.

In this section we will first illustrate briefly how CCA is performed using the `anacor` package. This is followed by a more detailed CCA using the `vegan` package. The data set that will be used in our illustrations is a data set obtained from a Multivariate Statistical Modelling of Ecology data workshop. This workshop was held at the Statistics department of Stellenbosch University in December 2009 by Professors Michael Greenacre and Raul Primicerio. The Ecology data are displayed on the next page as two R objects `biodata` and `envdata`.

These data sets represent a typical setup for a CCA. The object `biodata` refers to the contingency table while the object `envdata` refers set of explanatory variables. Both sets of data represent measurements taken on 30 different sites. Five different species labelled a, b, c, d and e were counted on the 30 sites while at the same time three numerical measurements named pollution, depth and temperature was also measured on the same sites. The purpose of the CCA is now to study the relationships between species and the sites by incorporation the numerical measurements as well. Also we would like to study the relationship between the sites and the numerical measurements.

```
R> biodata
```

	a	b	c	d	e
s1	0	2	9	14	2
s2	26	4	13	11	0
s3	0	10	9	8	0
s4	0	0	15	3	0
s5	13	5	3	10	7
s6	31	21	13	16	5
s7	9	6	0	11	2
s8	2	0	0	0	1
s9	17	7	10	14	6
s10	0	5	26	9	0
s11	0	8	8	6	7
s12	14	11	13	15	0
s13	0	0	19	0	6
s14	13	0	0	9	0
s15	4	0	10	12	0
s16	42	20	0	3	6
s17	4	0	0	0	0
s18	21	15	33	20	0
s19	2	5	12	16	3
s20	0	10	14	9	0
s21	8	0	0	4	6
s22	35	10	0	9	17
s23	6	7	1	17	10
s24	18	12	20	7	0
s25	32	26	0	23	0
s26	32	21	0	10	2
s27	24	17	0	25	6
s28	16	3	12	20	2
s29	11	0	7	8	0
s30	24	37	5	18	1

```
R > envdata
```

	Pollution	Depth	Temperature
s1	4.8	72	3.5
s2	2.8	75	2.5
s3	5.4	59	2.7
s4	8.2	64	2.9
s5	3.9	61	3.1
s6	2.6	94	3.5
s7	4.6	53	2.9
s8	5.1	61	3.3
s9	3.9	68	3.4
s10	10.0	69	3.0
s11	6.5	57	3.3
s12	3.8	84	3.1
s13	9.4	53	3.0
s14	4.7	83	2.5
s15	6.7	100	2.8
s16	2.8	84	3.0
s17	6.4	96	3.1
s18	4.4	74	2.8
s19	3.1	79	3.6
s20	5.6	73	3.0
s21	4.3	59	3.4
s22	1.9	54	2.8
s23	2.4	95	2.9
s24	4.3	64	3.0
s25	2.0	97	3.0
s26	2.5	78	3.4
s27	2.1	85	3.0
s28	3.4	92	3.3
s29	6.0	51	3.0
s30	1.9	99	2.9

2.6.1 The anacor package

As described before, the `anacor()` function performs simple CA, but we will now use it to perform CCA. The `plot()` function is used to obtain the CCA plot. The main arguments of the `anacor()` and `plot()` functions are given below respectively

Chapter 2: Simple and Canonical correspondence analysis

```
R> anacor(tab, ndim = 2, row.covariates, col.covariates,
          scaling = c("Benzecri", "Benzecri"), eps = 1e-06)
```

```
R> plot(x, plot.type, plot.dim = c(1,2), legpos = "top",
        arrows = FALSE, conf = 0.95, wlines = 0, xlab, ylab,
        main, type, xlim, ylim, cex.axis2, ...)
```

CCA is performed by specifying the `row.covariates` or `col.covariates` option. The row covariates in our case refer to the numerical data. Again, `tab` is a table of frequencies (or contingency table), `ndim` is the number of dimensions and the default scaling option is the `Benzecri` scaling. More details about the scaling methods in `anacor` package can be found in de Leeuw and Mair (2009).

In the `plot()` function, `x` is an CCA object obtained from the `anacor()` function and the default `plot.type` is the joint plot. In the `anacor` package, there is a variety of types of plots to choose from for two and three dimensional plots (see de Leeuw and Mair, 2009). The following R instructions perform CCA on the Ecological data:

```
R> library(anacor)
```

```
R> req5<-anacor(biodata, ndim = 2, row.covariates = envdata)
```

```
R> req5 #CCA results
```

```
CA fit:
```

```
Sum of eigenvalues: 0.2351813
```

```
Benzecri RMSE rows: 1.157468e-05
```

```
Benzecri RMSE columns: 1.175089e-05
```

```
Total chi-square value: 319.997
```

```
Chi-Square decomposition:
```

	Chisq	Proportion	Cumulative Proportion
Component 1	266.504	0.367	0.367
Component 2	47.228	0.065	0.433
Component 3	6.266	0.009	0.441

Chapter 2: Simple and Canonical correspondence analysis

The results of the CCA are displayed above and we see that 98.04% $\left(\frac{(266.504 + 47.228)}{319.997} = 0.9804\right)$ of the variation in the contingency table is explained by the first two dimensions in the constrained space. The CCA plot can be obtained by using the instruction:

```
R> plot(req5, plot.type="orddiag", main="")
```

The resulting plot is displayed in Figure 2.6. Note that the CCA plot also goes by different names like ordination diagram or triplot. Figure 2.6 shows the CCA plot according to ter Braak (1986). The blue and red points represent ordinations of the species and sites respectively. Points lying close to each other represent a strong association, while points lying away from each other represent a weak association. For example s10, s13, s4, s15, s17 seems to be quite similar (they lie away from the rest) and is associated with specie c. The three arrows represent the direction for the three explanatory variables. A site lying in the direction of the arrow means that it is strongly associated with that particular explanatory variable. For example s10, s13, s4, s15, s17 seems to be associated with higher pollution, since they lie in the direction in which pollution increases.

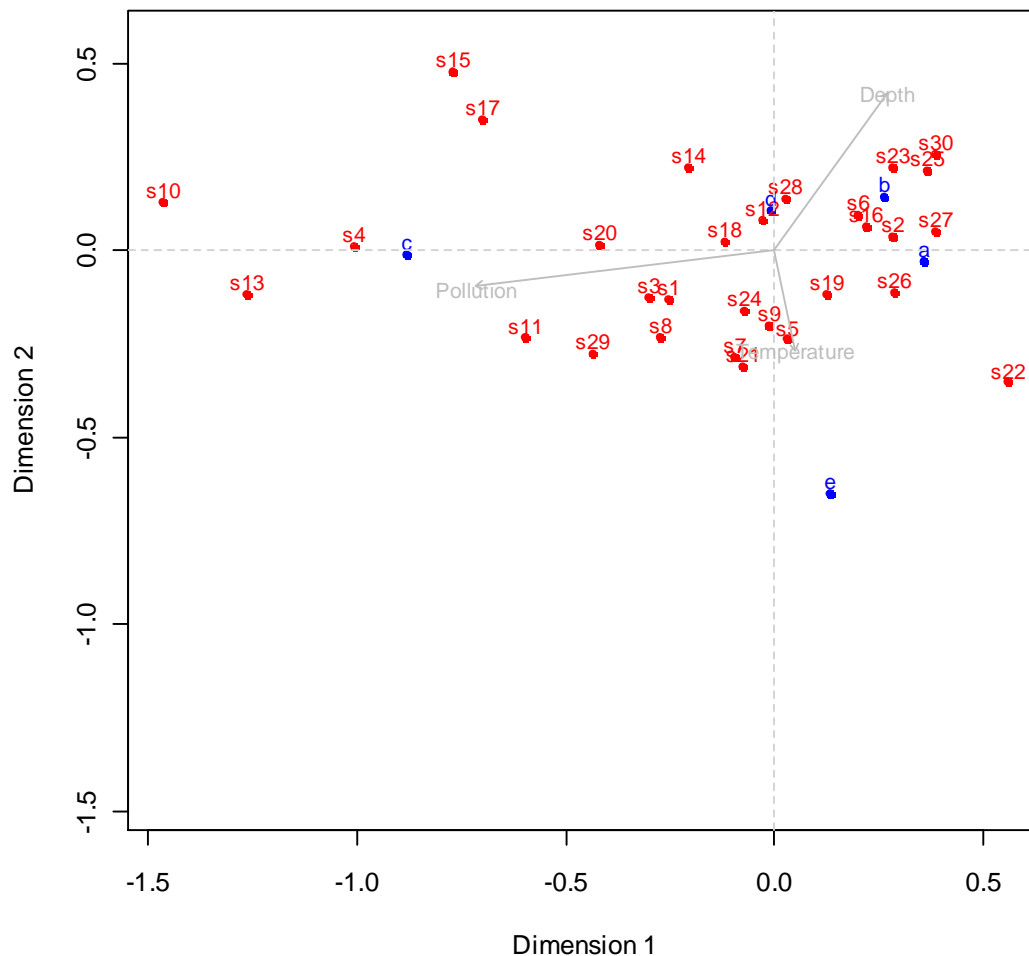


Figure 2.6: The CCA plot (or triplot) using `anacor()`.

2.6.2 The vegan package

The `vegan` package contains a function called `cca()` which can perform both simple CA and CCA. In addition it can also perform partial constrained correspondence analysis. In this section we will illustrate its usage in performing a canonical correspondence analysis. The main arguments for the `cca()` function are:

```
R> cca(X, Y, Z, ...)
```

The argument `x` is a table of frequencies (contingency table), `y` is a set of explanatory variables (usually numerical) and `z` is an argument needed to perform partial

Chapter 2: Simple and Canonical correspondence analysis

constrained correspondence analysis. Note that for a simple CA only *x* needs to be specified but for a CCA both *x* and *y* needs to be specified. The following two functions are also quite useful to obtain the appropriate CCA output:

```
R> plot(x, choices = c(1, 2), display = c("sp", "wa", "cn"),
       scaling = 2, type, xlim, ylim, const, ...)
```

```
R> summary(object, scaling = 2, axes = 6, display = c("sp", "wa",
       "lc", "bp", "cn"), digits = max(3, getOption("digits") - 3), ...)
```

The above two functions `plot()` and `summary()` produces a joint plot and a summary of the CCA results respectively. Both the arguments *x* and *object* in the above functions are objects from a CCA. In both functions `plot()` and `summary()` the default `scaling=2` option is used. For more information on the scaling options and two dimensional displays of correspondence analysis see Nenadic and Greenacre (2007) and Greenacre (2007). The following instructions load the `vegan` package and perform a CCA using the Ecology data:

```
R> library(vegan)
R> req6<-cca(X=biodata,Y=envdata)
```

The function `summary()` produces the output of CCA. The output contains the inertia, the row coordinates (site scores) and column coordinates (specie scores) for the joint plot, as well as the coordinates for the arrows (biplot scores) on the joint plot. In this output 98.04% of the variation in the contingency table is explained by the first two dimensions in the constrained space. The same was produced with `anacor()` in the previous section. Note that by using the function `scores()` on a CCA object one could also obtain the site scores and specie scores.

Chapter 2: Simple and Canonical correspondence analysis

```
R> summary(req6,scaling=3)
```

Call:
cca(X = biodata, Y = envdata)

Partitioning of mean squared contingency coefficient:

	Inertia	Proportion
Total	0.5436	1.0000
Constrained	0.2399	0.4412
Unconstrained	0.3038	0.5588

Eigenvalues, and their contribution to the mean squared contingency coefficient

Importance of components:

	CCA1	CCA2	CCA3	CA1	CA2	CA3	CA4
Eigenvalue	0.200	0.0354	0.00470	0.107	0.0865	0.0606	0.0495
Proportion Explained	0.367	0.0651	0.00864	0.197	0.1592	0.1115	0.0911
Cumulative Proportion	0.367	0.4326	0.44125	0.638	0.7975	0.9089	1.0000

Accumulated constrained eigenvalues

Importance of components:

	CCA1	CCA2	CCA3
Eigenvalue	0.200	0.0354	0.0047
Proportion Explained	0.833	0.1476	0.0196
Cumulative Proportion	0.833	0.9804	1.0000

Scaling 3 for species and site scores

* Both sites and species are scaled proportional to eigenvalues on all dimensions

Species scores

	CCA1	CCA2	CCA3	CA1	CA2	CA3
a	0.53401	0.07068	0.3300	0.4620	0.6063	-0.2926
b	0.39591	-0.32989	-0.2077	-0.2660	0.2799	0.8816
c	-1.31604	0.02584	0.1345	-0.7945	-0.1821	-0.1342
d	-0.01626	-0.25068	-0.2461	-0.1194	-0.5002	-0.4047
e	0.19655	1.49814	-0.3632	1.3745	-1.2232	0.6000

Chapter 2: Simple and Canonical correspondence analysis

Site scores (weighted averages of species scores)

	CCA1	CCA2	CCA3	CA1	CA2	CA3
s1	-0.90214	-0.18512	-1.82501	-0.5768481	-0.88834	-0.612124
s2	-0.07539	-0.18733	1.83515	-0.3617725	0.13435	-0.749307
s3	-0.66418	-0.99832	-1.53247	-1.0112038	-0.16277	0.839098
s4	-2.45972	-0.10759	1.03706	-0.8324151	-0.08200	-0.301392
s5	0.36426	1.02478	-0.51774	0.5417606	-0.27884	-0.016245
s6	0.22067	-0.05690	0.31618	-0.0005908	0.26356	0.029981
s7	0.59095	-0.20962	-0.89101	0.2138168	0.11343	-0.050405
s8	0.94307	2.90449	1.44388	2.4075204	0.43967	0.037197
s9	-0.01489	0.45570	-0.03332	0.0503420	0.02092	-0.268356
s10	-1.81132	-0.42964	0.08885	0.2026437	0.36016	0.320583
s11	-0.46928	1.20049	-2.31705	0.5084578	-0.56135	1.228181
s12	-0.23308	-0.60802	0.10803	-0.3607828	0.09498	-0.159127
s13	-2.13220	2.01530	0.21968	0.5808221	-0.43352	0.656807
s14	0.69110	-0.32305	1.37655	1.1865938	0.34144	-1.137745
s15	-0.96544	-0.50429	-0.16172	0.6270813	-0.51860	-0.732190
s16	0.99188	0.34491	1.39532	0.7478954	0.83682	0.377614
s17	1.19474	0.37566	4.81581	2.6954058	2.38567	-0.882785
s18	-0.66873	-0.45532	0.54618	-0.6181937	0.02412	-0.140524
s19	-0.73098	-0.09992	-1.45609	-0.9008934	-0.83448	-0.455599
s20	-0.99064	-0.83637	-1.06528	-0.8421333	-0.14393	0.581667
s21	0.66949	2.52494	-0.42435	1.6815855	-0.45848	-0.172920
s22	0.81439	1.67580	0.22284	0.4226111	-0.34177	0.090086
s23	0.34643	1.14856	-2.54662	0.7929498	-1.68711	0.292243
s24	-0.47383	-0.36590	1.13029	-0.7272913	0.56991	0.003995
s25	0.74599	-0.79267	-0.09015	-0.0040052	0.19590	0.111807
s26	0.88228	-0.34147	0.67600	0.0242537	0.90862	0.186185
s27	0.63140	-0.08783	-0.79904	0.0844378	-0.37140	-0.048733
s28	-0.25298	-0.15703	0.17146	-0.0093878	-0.28727	-0.857990
s29	-0.29845	-0.21403	1.46088	0.0356091	0.68496	-1.011445
s30	0.54718	-0.83749	-0.66707	-0.3018168	0.02759	0.790620

Chapter 2: Simple and Canonical correspondence analysis

Site constraints (linear combinations of constraining variables)

	CCA1	CCA2	CCA3	CA1	CA2	CA3
s1	-0.37757	0.30053	-0.404870	-0.5768481	-0.88834	-0.612124
s2	0.42230	-0.08452	0.530858	-0.3617725	0.13435	-0.749307
s3	-0.44882	0.29415	0.375189	-1.0112038	-0.16277	0.839098
s4	-1.50692	-0.02419	0.136820	-0.8324151	-0.08200	-0.301392
s5	0.04515	0.54999	0.017264	0.5417606	-0.27884	-0.016245
s6	0.29753	-0.21457	-0.461555	-0.0005908	0.26356	0.029981
s7	-0.14553	0.65292	0.223586	0.2138168	0.11343	-0.050405
s8	-0.40728	0.53618	-0.182117	2.4075204	0.43967	0.037197
s9	-0.02107	0.46358	-0.286171	0.0503420	0.02092	-0.268356
s10	-2.19421	-0.29992	0.003115	0.2026437	0.36016	0.320583
s11	-0.89150	0.54057	-0.184241	0.5084578	-0.56135	1.228181
s12	-0.04431	-0.18491	-0.069345	-0.3607828	0.09498	-0.159127
s13	-1.88995	0.27108	0.071688	0.5808221	-0.43352	0.656807
s14	-0.30820	-0.51285	0.476683	1.1865938	0.34144	-1.137745
s15	-1.15199	-1.10121	0.110191	0.6270813	-0.51860	-0.732190
s16	0.32649	-0.14230	0.035318	0.7478954	0.83682	0.377614
s17	-1.04969	-0.80510	-0.147499	2.6954058	2.38567	-0.882785
s18	-0.17859	-0.05504	0.238091	-0.6181937	0.02412	-0.140524
s19	0.18938	0.27256	-0.502708	-0.9008934	-0.83448	-0.455599
s20	-0.62556	-0.03651	0.042530	-0.8421333	-0.14393	0.581667
s21	-0.11651	0.71893	-0.256766	1.6815855	-0.45848	-0.172920
s22	0.83432	0.81494	0.345560	0.4226111	-0.34177	0.090086
s23	0.42034	-0.50899	0.090507	0.7929498	-1.68711	0.292243
s24	-0.10649	0.37061	0.093066	-0.7272913	0.56991	0.003995
s25	0.54468	-0.49131	-0.004393	-0.0040052	0.19590	0.111807
s26	0.43039	0.26515	-0.306965	0.0242537	0.90862	0.186185
s27	0.57407	-0.11215	0.040200	0.0844378	-0.37140	-0.048733
s28	0.03793	-0.31462	-0.279395	-0.0093878	-0.28727	-0.857990
s29	-0.65000	0.63928	0.121590	0.0356091	0.68496	-1.011445
s30	0.57923	-0.59372	0.081443	-0.3018168	0.02759	0.790620

Biplot scores for constraining variables

	CCA1	CCA2	CCA3	CA1	CA2	CA3
Pollution	-0.99290	0.08836	0.06858	0	0	0
Depth	0.35241	-0.88787	-0.28627	0	0	0
Temperature	0.01427	0.19071	-0.98160	0	0	0

Chapter 2: Simple and Canonical correspondence analysis

To produce the CCA plot (Figure 2.7) we may use the instruction

```
R> plot(req6, scaling = 3)
```

This instruction uses the output (site scores, specie scores and biplot scores) displayed in the summary output to construct a two dimensional plot similar to Figure 2.6. Figure 2.7 is a plot of the first canonical variates and the arrows (biplot scores) give the direction in which the explanatory variable increases. This plot should be interpreted similar to Figure 2.6 and shows the associations between the two sets of data. However, it should be remembered that Figure 2.6 and Figure 2.7 use two different CCA procedures. The CCA produced by the `vegan` package follows the discussion of CCA outlined in Section 2.5.

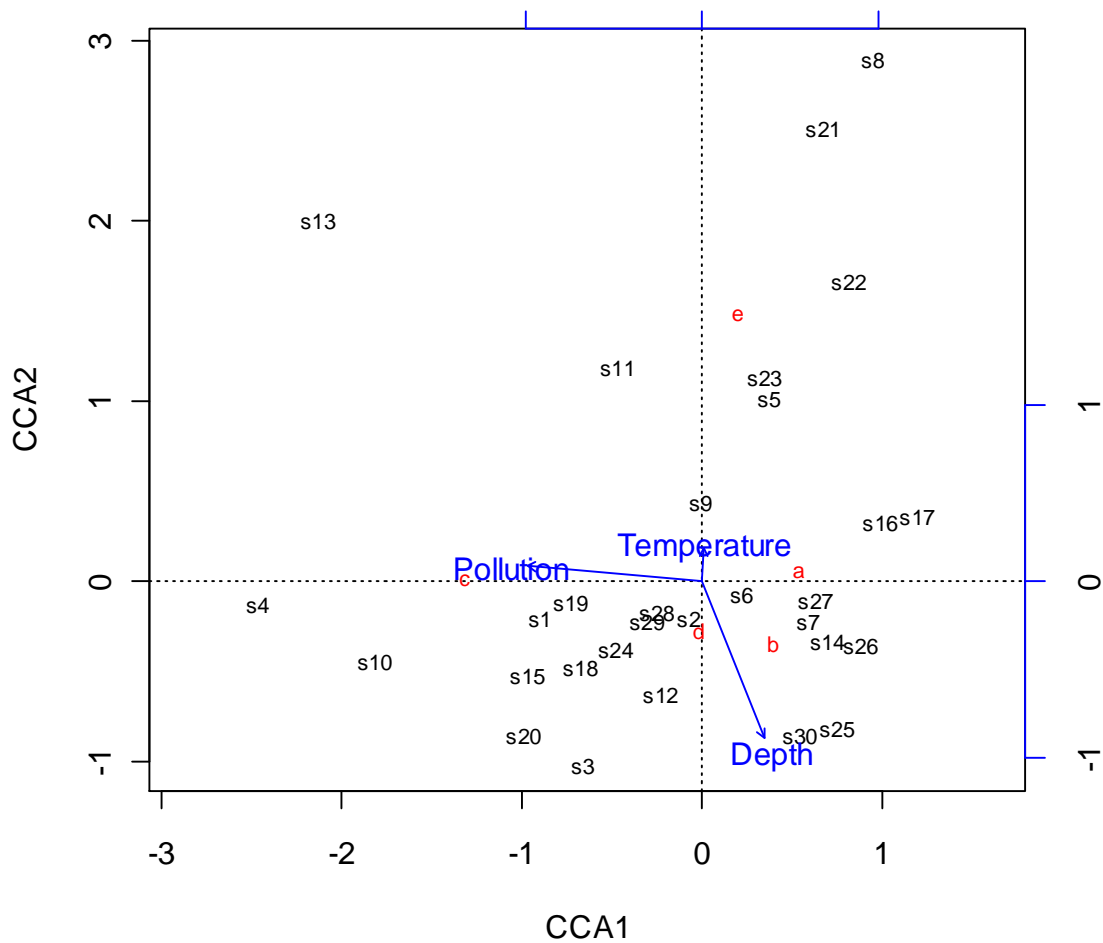


Figure 2.7: The CCA plot using `cca()`.

Chapter 2: Simple and Canonical correspondence analysis

The `vegan` package offers much more advantages than the `anacor` package in doing simple CA or CCA. Two very attractive graphical features are captured in the following two functions

```
R> ordisurf(x, y, choices=c(1, 2), knots=10, family="gaussian",
           col="red", thinplate = TRUE, add = FALSE,
           display = "sites", w = weights(x), main, nlevels = 10,
           levels, labcex = 0.6, bubble = FALSE, cex = 1, ...)
```

```
R> ordirgl(object, display = "sites", choices = 1:3, type = "p",
          ax.col = "red", arr.col = "yellow", text, envfit, ...)
```

The function `ordisurf()` allows us to create contours on the existing CCA plot. The contours basically represent the relationship between an explanatory variable and the sites. The object `x` is an CCA object produced by the `cca()` function, while the object `y` is the explanatory variable of interest. The object `knots` allows to create a simple (`knots=1`) or a more complicated (`knots>1`) contour plot.

The function `ordirgl()` uses a CCA object from `cca()` to produce a three dimensional CCA plot. This plot can be rotated manually to obtain the best view of the joint plot. We applied the above two functions to the Ecology data. The instructions to create the contours and the three dimensional plot are given next and the resulting graphs are displayed in Figure 2.8 and 2.9 respectively.

```
R> ordisurf(plot(req6, scaling=3), envdata[, 1], add=T, knots=1,
           col="green")
R> ordisurf(plot(req6, scaling=3), envdata[, 1], add=T, knots=2,
           col="green")
R> ordirgl(req6, type="t")
```

Note that knots 1 and 2 produce a linear and a quadratic contour plot respectively. The contours in Figure 2.8 increase in the direction of the pollution variable. These contours allow us to study the relationship between pollution and sites more carefully. Figure 2.9 was rotated to obtain the best view of the three dimensional plot. One can clearly see in this plot that the third dimension shows an interesting separation of sites `s23`, `s11` and `s17` from the rest. This was not visible in the two dimensional CCA plot in Figure 2.7.

Chapter 2: Simple and Canonical correspondence analysis

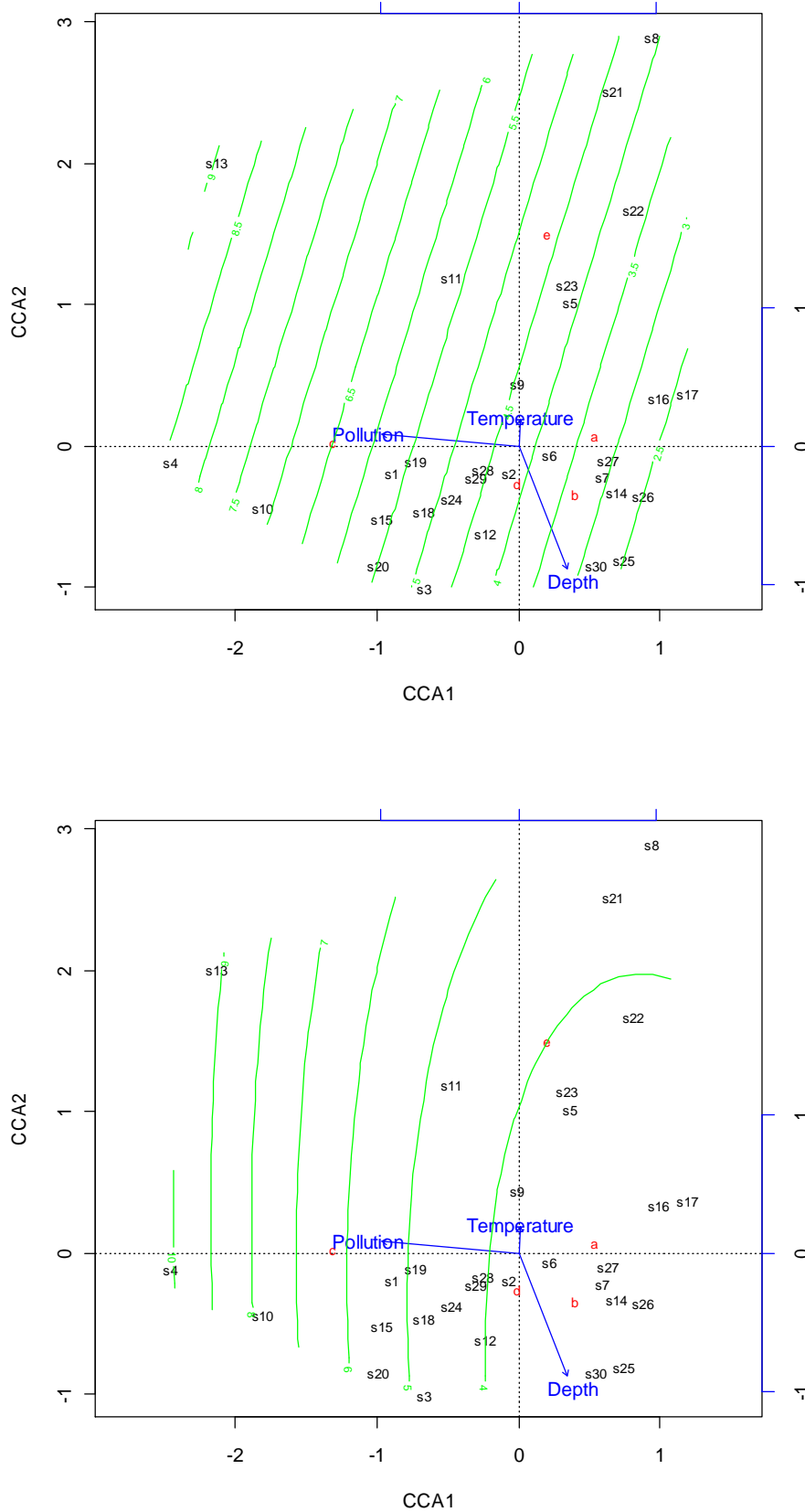


Figure 2.8: The contours (knots=1 and 2) for the pollution variable showing the direction in which the pollution is increasing using `ordisurf()`.

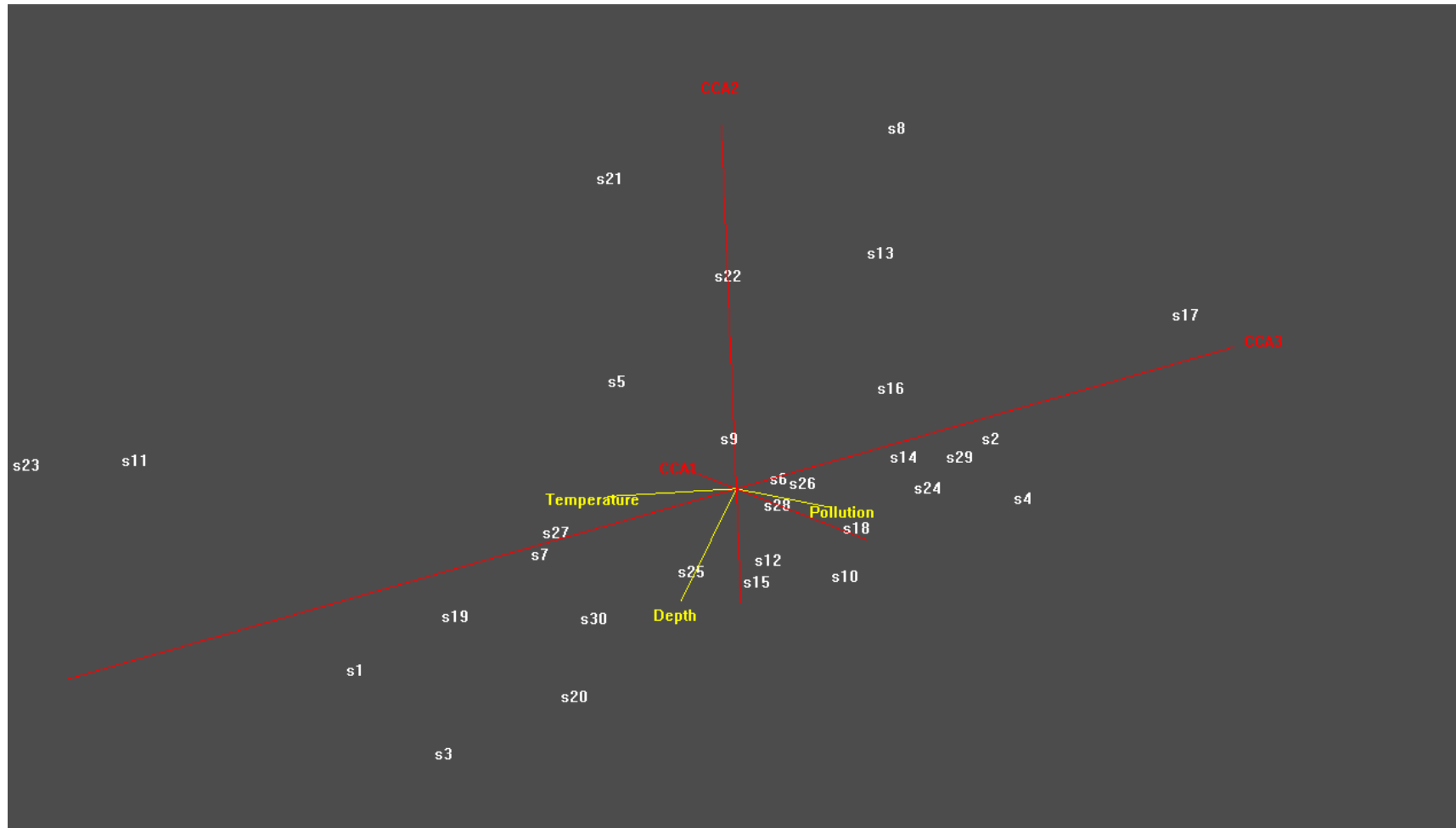


Figure 2.9: The three dimensional CCA plot using `ordigr1()`.

Chapter 2: Simple and Canonical correspondence analysis

As an alternative to the function `ordiplot3d()` one could also use the function `ordiplot3d()` to produce three dimensional CCA plots. However, the latter function is less flexible. Figure 2.10 below is an example of a plot produced using the following instruction:

```
R> ordiplot3d(req6)
```

The three dimensional plots discussed in this section for CCA could also be used for simple CA using the function `cca()`.

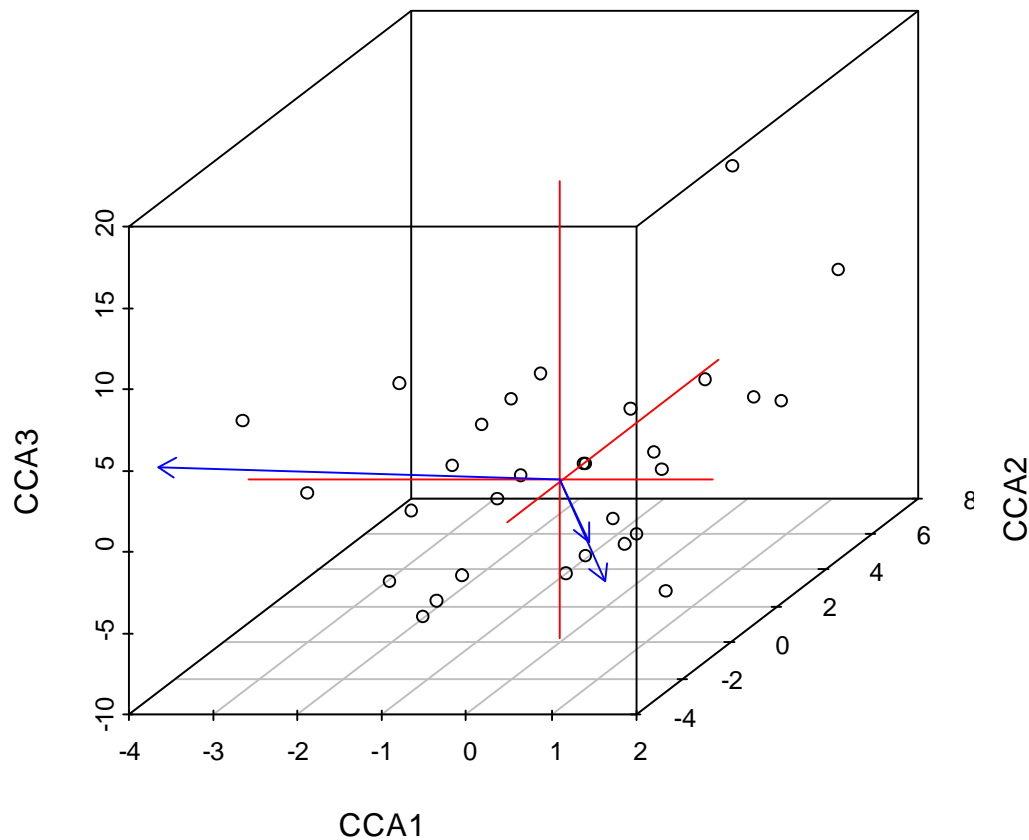


Figure 2.10: The three dimensional CCA plot using `ordiplot3d()`.

2.7 Permutation tests in CCA

In this section we give a brief illustration of the use of permutation tests in CCA. The purpose of the permutation test here is to find out which of the explanatory variables are significant in the constrained space (Greenacre, 2007). The permutation test employs the coefficient of determination, r^2 , to determine the significance of the explanatory variables.

The permutation tests in CCA can be performed using the two functions `anova()` and `envfit()` in R. The `anova()` function uses a `cca()` object to perform a global test on the explanatory variables *i.e.* it tests whether all the explanatory variables are significant in the CCA model. The `envfit()` function (from the `vegan` package) is used to test which of the individual explanatory variables are significant. Both functions are displayed below.

```
R> anova(object, alpha=0.05, beta=0.01, step=100, perm.max=9999,
        by = NULL, ...)
```

```
R> envfit(X, P, permutations = 0, strata, choices=c(1,2), ...)
```

In the `envfit()` function the argument `x` is the `cca()` object and `P` is the matrix containing the explanatory variables. The number of permutations required can be specified in the argument `perm.max`. The following are the results of the permutation tests for significant explanatory variables.

```
R> req6<-cca(biodata,envdata)
```

```
R> anova(req6)
```

```
Permutation test for cca under reduced model
```

```
Model: cca(X = biodata, Y = envdata)
```

	Df	Chisq	F	N.Perm	Pr(>F)
Model	3	0.2399	6.8441	199	0.005 **
Residual	26	0.3038			

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 2: Simple and Canonical correspondence analysis

The results of the ANOVA give a significant p -value (0.005). Thus the explanatory variables have significance in the analysis. To test which of the individual explanatory variables are significant, the following instructions can be used.

```
R> fit<-envfit(req6,envdata, perm = 999)
```

```
R> fit
```

```
***VECTORS
```

	CCA1	CCA2	r2	Pr(>r)	
Pollution	-0.993267	0.115849	0.7119	0.001	***
Depth	0.725832	-0.687873	0.3621	0.004	**
Temperature	-0.011026	0.999939	0.0110	0.875	

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
P values based on 999 permutations.
```

As we can see from the results above, only two of the explanatory variables are actually significant. These variables are pollution (p -value=0.001) and depth (p -value=0.004). Temperature is not significant (p -value=0.875). Thus it can be concluded that the two environmental variables (pollution and depth) play a more important role than temperature. The significant explanatory variables can also be displayed graphically on the CCA plot. The following instructions create the CCA plot with only the significant variables. The results are given in Figure 2.11.

```
R> plot(req6,display = c("sp", "wa") ) #without the arrows
```

```
R> plot(fit,p.max = 0.05, col = "red") #only significant variables
```

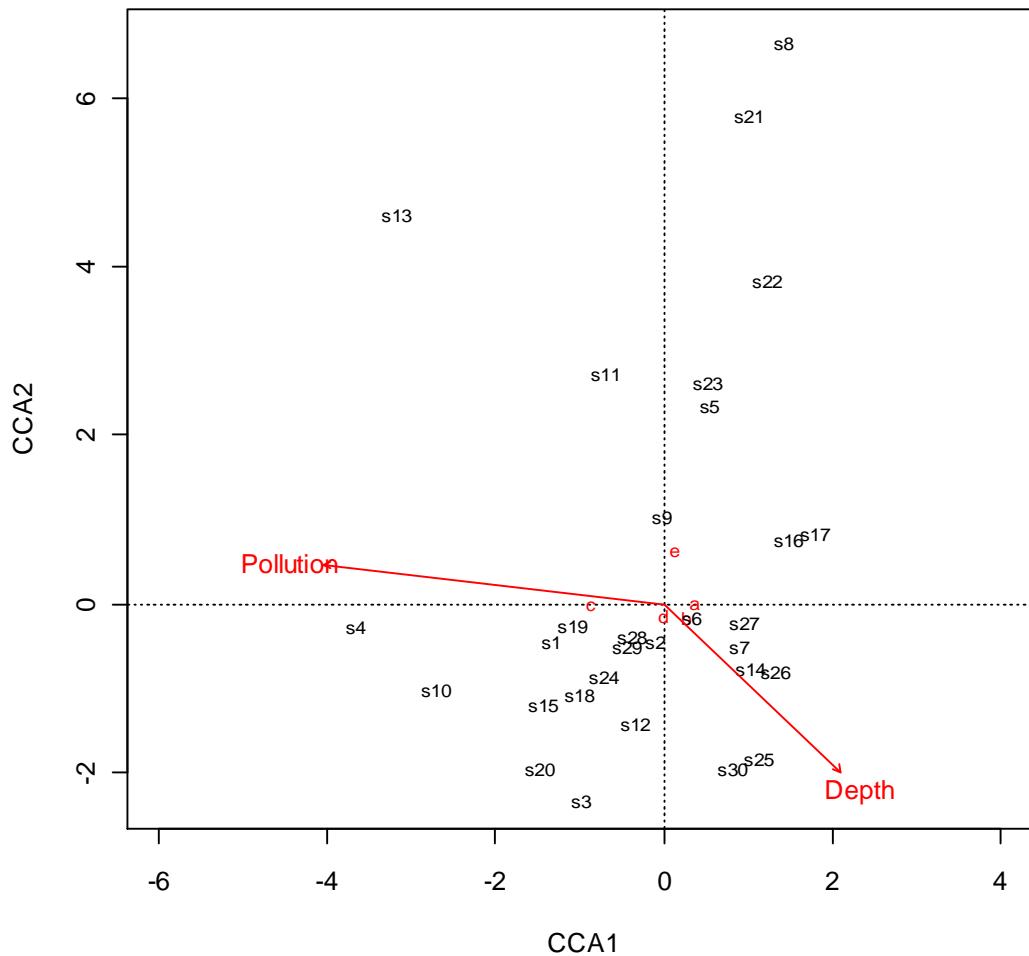


Figure 2.11: CCA plot of only the significant explanatory variables.

2.8 Summary

As mentioned in this chapter, the aim of the correspondence analysis is to study the relationships between the rows and columns of a contingency table. We have also explained the algebra behind correspondence analysis. In this chapter we have also shown how the correspondence analysis is extended to canonical correspondence analysis. Canonical correspondence analysis incorporates an additional set of numerical variables and the aim with this analysis was to study the relationship among the count data (contingency table) and the numerical variables (often called the environmental variables in Ecology). Goodness-of-fit measures like the inertia and the Benzécri plot were also discussed. These measures allow us to assess how well the variation in the original data is explained in these analyses.

Chapter 2: Simple and Canonical correspondence analysis

The different R packages *i.e.* `anacor`, `ca` and `vegan` are useful packages to perform a correspondence analysis. The `ca` package is restricted to correspondence analysis only, while `anacor` and `vegan` offers much more possibilities and advantages. For example, the `anacor` package allows us to create Benzécri plots and it can also perform a canonical correspondence analysis. The `vegan` package, besides correspondence analysis, offers canonical correspondence analysis, permutation tests and a host of other techniques (see Oksanen, 2008).

Chapter 3

Cluster analysis

3.1 Introduction

Cluster analysis is a multivariate statistical method which focuses on searching the data for group structures or other interesting patterns. It is a very useful tool in exploratory data analysis, which can provide an informal means for assessing dimensionality, identifying outliers and suggesting interesting hypotheses concerning relationships among observations or variables. Cluster analysis makes use of certain distance measures and employs step-by-step rules for grouping objects (observations or variables), which will be discussed in this chapter. Cluster analysis can be applied to different types of data such as numerical, count and binary data. For each data type an appropriate dissimilarity or distance measure is needed.

In this chapter we start by describing the distance (dissimilarity) matrices that are required to perform cluster analysis. In Section 3.3 we also define different types of distance (dissimilarity) measures, which are used to obtain the above mentioned matrices. The choice of these measures usually depends on the type of data that is used. In Section 3.4 we explain four well-known clustering algorithms. Then finally we conclude this chapter by giving an illustration of these clustering methods by using different R functions.

3.2 The data for cluster analysis

The data can be obtained in two ways. One way is that the data can be collected directly from an experiment as proximities and the other way is that the data can be transformed into proximities. Most of the time, the data is transformed into a proximity matrix by taking into consideration the objects that we want to cluster and

also the type of data. Note that the objects can be the observations or variables (the usual dimensions of a multivariate data set).

Proximity is defined as the nearness (closeness) of objects in space. There are two types of proximity measures which are dissimilarities and similarities (Cox and Cox, 1994). The data for cluster analysis is most often a dissimilarity or distance matrix (see Section 3.3). When a similarity matrix, such as the correlation matrix, is available, it is first transformed into a dissimilarity matrix before clustering is performed. Similarity between objects i and i' ($s_{ii'}$) measures how similar the two objects are, whereas the dissimilarity between objects i and i' ($d_{ii'}$) measures how dissimilar the two objects are. Thus, similarity measures the degree of resemblance, whereas dissimilarity measures the degree of difference. Similarity usually ranges between -1 and 1, or can be normalized to range from 0 to 1.

Distances also measure dissimilarity (Teknomo, 2006). The following are the properties of a true distance measure (Johnson and Wichern, 2007). Any distance measure $d(i, i')$ between two objects i and i' , is valid provided that it satisfies,

1. $d(i, i') = d(i', i)$
2. $d(i, i') > 0$ if $i \neq i'$
3. $d(i, i') = 0$ if $i = i'$
4. $d(i, i') \leq d(i, j) + d(j, i')$, (called the triangle inequality)

where j is any other intermediate point. Some of the distance measures found in the literature do not obey the fourth property. Besides the fact that they are not true distances, they are still good measures of differences between possible pairs of objects and are known as dissimilarities (Greenacre, 2007).

To conclude, the distance or dissimilarity matrices are used as input for cluster analysis. To obtain such matrices we need to define some distance or dissimilarity measure. The next section discusses examples of such measures.

3.3 The distance and dissimilarity matrix

Let \mathbf{D} be an $I \times I$ distance matrix (or dissimilarity matrix) with elements $d(i, i') \geq 0$ being the distance (or dissimilarity) between object i and object i' for $i, i' = 1, 2, \dots, I$. Such a distance (or dissimilarity) matrix can be obtained from the raw $I \times J$ data matrix \mathbf{X} as illustrated below:

$$\mathbf{X}_{I \times J} = \begin{bmatrix} x_{11} & x_{12} & \mathbf{L} & x_{1J} \\ x_{21} & x_{22} & \mathbf{L} & x_{2J} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ x_{I1} & x_{I2} & \mathbf{L} & x_{IJ} \end{bmatrix} \rightarrow \mathbf{D}_{I \times I} = \begin{bmatrix} d_{11} & d_{12} & \mathbf{L} & d_{1I} \\ d_{21} & d_{22} & \mathbf{L} & d_{2I} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ d_{I1} & d_{I2} & \mathbf{L} & d_{II} \end{bmatrix} \quad (3.1)$$

Note that there are $M = \frac{1}{2}[I(I-1)]$ distinct distances in matrix \mathbf{D} . The question that remains is how do we obtain these distances (dissimilarities). It is important to note at this point that calculating $d(i, i')$ or $d_{i'}$ depends on the type of data in matrix \mathbf{X} . In the next section we elaborate more on the distance and dissimilarity measures for numerical, count and binary data separately.

3.3.1 Distance measures for numerical data

Let \mathbf{X} be an $I \times J$ data matrix with (numerical) elements x_{ij} for $i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$. Then the following are distance measures between object i and i' in matrix \mathbf{X} (see Johnson and Wichern, 2007).

- (a) Euclidean distance: This is the most commonly used distance measure. It is defined as the straight line distance between object i and i' . This distance measure is defined as

$$d(i, i') = \sqrt{\sum_{j=1}^J (x_{ij} - x_{i'j})^2} \quad i, i' = 1, 2, \dots, I. \quad (3.2)$$

(b) Minkowski distance: This is a generalized metric distance measure defined by

$$d(i, i') = \left[\sum_{j=1}^J |x_{ij} - x_{i'j}|^m \right]^{\frac{1}{m}}.$$

When $m = 1$, it becomes what is known as the “city-block” or Manhattan distance and when $m = 2$, it becomes the Euclidean distance (3.2).

(c) Canberra metric: This is a popular measure of distance or dissimilarity for nonnegative variables only. It is defined as follows:

$$d(i, i') = \sum_{j=1}^J \frac{|x_{ij} - x_{i'j}|}{(x_{ij} + x_{i'j})}.$$

(d) Czekanowski coefficient: This measure of distance or dissimilarity for nonnegative variables is defined as

$$d(i, i') = 1 - \frac{2 \sum_{j=1}^J \min(x_{ij}, x_{i'j})}{\sum_{j=1}^J (x_{ij} + x_{i'j})}.$$

The Euclidean distance (3.2) will be used to obtain the distance matrix in Section 3.5 and in Chapters 4 and 5, where two numerical data sets are analyzed in a cluster analysis, multidimensional scaling and analysis of distance separately.

3.3.2 A dissimilarity and distance measure for count data

Let \mathbf{X} be an $I \times J$ data matrix with elements x_{ij} (frequencies or counts) for $i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$. Then the following are a distance and dissimilarity measure, respectively, between observation i and observation i' in \mathbf{X} .

(a) Bray-Curtis dissimilarity: This is the most commonly used dissimilarity for count data, especially in Ecology (Bray and Curtis, 1957; Greenacre, 2007). It is often called the Sorenson dissimilarity or the Canberra metric. It is defined as

$$d(i, i') = \frac{\sum_{j=1}^J |x_{ij} - x_{i'j}|}{\sum_{j=1}^J (x_{ij} + x_{i'j})}. \quad (3.3)$$

(b) Chi-square distance: Is also a popular distance measure for count data (Greenacre, 2007) and it is defined as

$$d(i, i') = \sqrt{\sum_{j=1}^J \frac{(x_{ij} - x_{i'j})^2}{\bar{x}_j}}, \text{ with } \bar{x}_j = \frac{1}{I} \sum_{i=1}^I x_{ij} \text{ (the average of column } j\text{).}$$

3.3.3 Dissimilarity measures for binary data

Let \mathbf{X} be an $I \times J$ data matrix with (binary data) elements $x_{ij} \in \{0,1\}$ for $i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$. Let $s_{ii'}$ represent a similarity coefficient between objects i and i' of the binary data set \mathbf{X} . The measures of similarity between objects i and i' described in this section is based on the following table:

		object i'		
		1	0	Total
object i	1	a	b	$a+b$
	0	c	d	$c+d$
	Total	$a+c$	$b+d$	$p=a+b+c+d$

where

a = the number of times when both objects have the value 0

b = the number of times when object i has value 0 and object i' has value 1

c = the number of times when object i has value 1 and object i' has value 0

d = the number of times when both objects have value 1.

Chapter 3: Cluster analysis

Based on this table, the following similarity measures between object i and i' in \mathbf{X} for binary data are defined.

(a) Jaccard similarity coefficient:

$$s_{ii'} = \frac{a}{a+b+c}.$$

(b) Bray Curtis similarity coefficient:

$$s_{ii'} = \frac{2a}{2a+b+c}.$$

Note that similarity measures can be converted to dissimilarity measures by using the transformation, $d_{ii'} = 1 - s_{ii'}$. For the measures above, this transformation results in the following dissimilarity measures.

(a) Jaccard dissimilarity coefficient:

$$d(i, i') = \frac{b+c}{a+b+c}. \quad (3.4)$$

(b) Bray-Curtis dissimilarity coefficient:

$$d(i, i') = \frac{b+c}{2a+b+c}. \quad (3.5)$$

Note that (3.3) and (3.5) are the same measures. The Jaccard and Bay-Curtis dissimilarity measures will be used to obtain the dissimilarity matrix in the analysis of the Biolog data in Chapter 6.

3.4 Agglomerative hierarchical clustering methods

Once the distance matrix for the objects has been obtained, the next step in cluster analysis is to group / cluster the objects based on these distances. There are several ways to perform cluster analysis. There are hierarchical clustering methods and non-hierarchical clustering methods. For non-hierarchical clustering methods, the number of clusters has to be specified before hand, whereas hierarchical clustering methods do not require prior knowledge of the number of clusters. Two general methods of hierarchical clustering methods are agglomerative hierarchical methods and divisive hierarchical methods (see Johnson and Wichern, 2007). The agglomerative techniques start with the individual objects. Initially, there are as many clusters as objects. Firstly, the most similar objects are grouped and these initial groups are merged according to their similarities, until only one group remains. Thus, the agglomerative technique cluster objects from the bottom to the top and the results is usually displayed a dendrogram. A dendrogram is a tree-like structure (see Figure 3.1 as an example). The divisive techniques start from a single group, partitioning that group into subgroups, partitioning these subgroups further into subgroups and so on until each object forms its own subgroup. Thus, the divisive technique starts from the top to the bottom when constructing the dendrogram. In this chapter we will study only agglomerative hierarchical methods and we briefly describe four such algorithms in the next few sections.

Johnson and Wichern (2007) give us the following general agglomerative hierarchical clustering algorithm for grouping N objects (observations / variables):

1. Start with N clusters, each containing a single entity and an $N \times N$ symmetric matrix of distances (or dissimilarities) $\mathbf{D} = \{d_{ii'}\}$.
2. Find the minimum entry in $\mathbf{D} = \{d_{ii'}\}$ and merge objects, U and V to get the first cluster (UV) .
3. The distance between cluster (UV) and any other cluster (or object) W is computed as

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\} \quad \text{for the single linkage method (Section 3.4.1)}$$

$$d_{(UV)W} = \max\{d_{UW}, d_{VW}\} \quad \text{for the complete linkage method (Section 3.4.2)}$$

$$d_{(UV)W} = \frac{1}{N_{(UV)}N_W} \sum_{i \in (UV)} \sum_{i' \in W} d_{ii'} \quad \text{for the average linkage method (Section 3.4.3)}$$

4. Update the entries in the distance matrix by first deleting the rows and columns corresponding to clusters U and V . Secondly, adding a row and column giving the distances between cluster (UV) and the remaining clusters.
5. Repeat Steps 3 and 4 until all objects are in one cluster. At this stage the algorithm stops. At each step, record the clustered objects and the distance when it is merged.

3.4.1 Single linkage

The single linkage method, which is also known as the nearest neighbour or shortest distance method, computes the distance between the two clusters (or objects) as the minimum distance between any two clusters (or objects). Using the general agglomerative algorithm above, we start by finding the minimum entry in $\mathbf{D} = \{d_{ii'}\}$ and merging the corresponding objects, say U and V , to get the first cluster (UV) . For step 3 of the general agglomerative algorithm, the distance between (UV) and any other cluster W are computed by

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\},$$

where d_{UW} and d_{VW} is the distance between the nearest neighbours of clusters U and W and clusters V and W , respectively.

The results of the single linkage method are displayed in a dendrogram containing the clusters as well as the distances at which the clusters were formed (see Figure 3.1). A disadvantage of the single linkage method is known as the chaining phenomenon. The chaining phenomenon occurs when clusters are formed in a long stringlike pattern.

Chaining occurs when the first cluster forms and then grows progressively larger by adding lone objects that have not been clustered yet. The chaining phenomenon appears in Figure 3.1 where there is no clear clustering of objects.

3.4.2 Average linkage

The average linkage method calculates the distance between two clusters (or objects) as the average distance between all pairs of objects where one object of a pair belongs to a cluster. Using the general agglomerative algorithm, we start by finding the minimum entry in $\mathbf{D} = \{d_{ii'}\}$ and merging the corresponding objects, say U and V , to get the first cluster (UV) . For step 3 of the algorithm, the distances between cluster (UV) and any other cluster (or object) W are computed by

$$d_{(UV)W} = \frac{1}{N_{(UV)}N_W} \sum_{i \in (UV)} \sum_{i' \in W} d_{ii'},$$

where $d_{ii'}$ is the distance between object i in the cluster (UV) and object i' in the cluster W . $N_{(UV)}$ and N_W are the number of objects in the clusters (UV) and W , respectively. The results of the average linkage method are also displayed in a dendrogram (see Figure 3.2 as an example).

3.4.3 Complete linkage

The complete linkage method, which is also known as the farthest neighbour method, computes the distance between clusters (or objects) in each step as the maximum distance between any two different objects in a distance matrix. Again, using the general agglomerative algorithm, we start by finding the minimum entry in $\mathbf{D} = \{d_{ii'}\}$ and merging the corresponding objects, such as U and V , to get cluster (UV) . For step 3 of the clustering algorithm, the distances between the cluster (UV) and any other cluster W are computed using

$$d_{(UV)W} = \max\{d_{UW}, d_{VW}\},$$

where d_{UW} and d_{VW} are the distances between the most distant members of clusters U and W and clusters V and W , respectively.

The results of the complete linkage method are displayed in a dendrogram as can be seen in Figure 3.3. This method of agglomerative hierarchical clustering is commonly used, since it produces clear clusters in the dendrogram and it is not affected by the chaining phenomenon. In Figure 3.1 the chaining occurred and the results do not show clear clusters being formed, while in Figure 3.3 there are clear clusters and no chaining present.

3.4.4 Ward's method

Ward's method is an alternative way of performing hierarchical cluster analysis. It uses an analysis of variance approach on the raw data (\mathbf{X}), instead of the distance (or dissimilarity) matrix (\mathbf{D}). Let the error sum of squares (ESS) for cluster k be defined by

$$ESS_k = \sum_{i \in k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)' (\mathbf{x}_i - \bar{\mathbf{x}}_k), \quad k = 1, 2, \dots, K,$$

where $\bar{\mathbf{x}}_k$ is the mean vector of the k -th cluster.

For Ward's method we start out with each observation forming a cluster, thus K equals the number of observations (rows) in \mathbf{X} . Note that at this stage $ESS_k = 0$. Step 1 in Ward's algorithm is to merge the two observations that minimizes ESS_k , thus creating cluster 1. Step 2 is to find the next two objects (where one of these objects maybe cluster 1) which minimizes ESS_k , thus forming the next cluster (or expanding cluster 1). At each step that follows, ESS_k will be evaluated, until all the observations

are grouped one big cluster. The algorithm stops when all the observations are one cluster ($K=1$).

Ward's method is most appropriate for numerical data. The results of Ward's method can also be displayed in a dendrogram, as can be seen in Figure 3.4. This is also a commonly used method which produces clear clustering results.

3.5 Performing a cluster analysis in R

In this section we will show the application of the four clustering methods discussed in Section 3.4 on a real-world data set. We will use the R functions `dist()`, `as.dendrogram()` and `hclust()`, which form part of the `stats` package.

A data set of 25 U.S. universities is used to illustrate the cluster analysis (data is taken from Johnson and Wichern, 2007, *p.729*). This is a multivariate data set with six variables:

- average SAT score of entering freshmen,
- percentage of freshmen in top 10 % of high school class,
- percentage of applicants accepted,
- student-faculty ratio,
- estimated annual expense and
- graduation rate (%).

The data of the 25 universities are displayed below as an R object.

Chapter 3: Cluster analysis

```
R> universities
      SAT Top10 Accept SFRatio Expenses Grad
Harvard    14.00    91    14     11   39.525    97
Princeton  13.75    91    14     8   30.220    95
Yale       13.75    95    19    11   43.514    96
Stanford   13.60    90    20    12   36.450    93
MIT        13.80    94    30    10   34.870    91
Duke       13.15    90    30    12   31.585    95
CalTech    14.15   100    25     6   63.575    81
Dartmouth  13.40    89    23    10   32.162    95
Brown      13.10    89    22    13   22.704    94
JohnsHopkins 13.05    75    44     7   58.691    87
UChicago   12.90    75    50    13   38.380    87
UPenn      12.85    80    36    11   27.553    90
Cornell     12.80    83    33    13   21.864    90
Northwestern 12.60    85    39    11   28.052    89
Columbia   13.10    76    24    12   31.510    88
NotreDame  12.55    81    42    13   15.122    94
UVir       12.25    77    44    14   13.349    92
Georgetown 12.55    74    24    12   20.126    92
CarnegieMellon 12.60    62    59     9   25.026    72
UMichigan  11.80    65    68    16   15.470    85
UCBerkeley 12.40    95    40    17   15.140    78
UWisconsin 10.85    40    69    15   11.857    71
PennState  10.81    38    54    18   10.185    80
Purdue     10.05    28    90    19    9.066    69
TexasA&M   10.75    49    67    25    8.704    67
```

To obtain the Euclidean distance matrix from the above data, we use the following instruction in R:

```
R> Distance<- dist(universities, method = "euclidean")
```

To perform the single linkage cluster analysis, we use the following R instruction:

```
R> plot(as.dendrogram(hclust(Distance, method="single")), ylim=c(0,30),
      main="Single linkage dendrogram", ylab="Euclidean distance")
```

The resulting dendrogram is displayed in Figure 3.1

Chapter 3: Cluster analysis

The average linkage cluster analysis is performed by changing argument `method=` in `hclust()` to "average":

```
R> plot(as.dendrogram(hclust(Distance,method="average")),  
       main="Average linkage dendrogram",ylab="Euclidean distance")
```

The output of this instruction are displayed in the dedrogram in Figure 3.2

Similarly, we can perform cluster analysis using complete linkage and Ward's method by changing the `method=` argument as follows:

For complete linkage we use

```
R> plot(as.dendrogram(hclust(Distance,method="complete")),  
       ylim=c(0,120), main=" Complete linkage dendrogram",  
       ylab="Euclidean distance")
```

and for Ward's method we use

```
R> plot(as.dendrogram(hclust(Distance,method="ward")),  
       main="Ward linkage dendrogram",ylab="Euclidean distance")
```

The dendrogram for complete linkage and Ward's method are displayed in Figures 3.3 and 3.4 respectively.

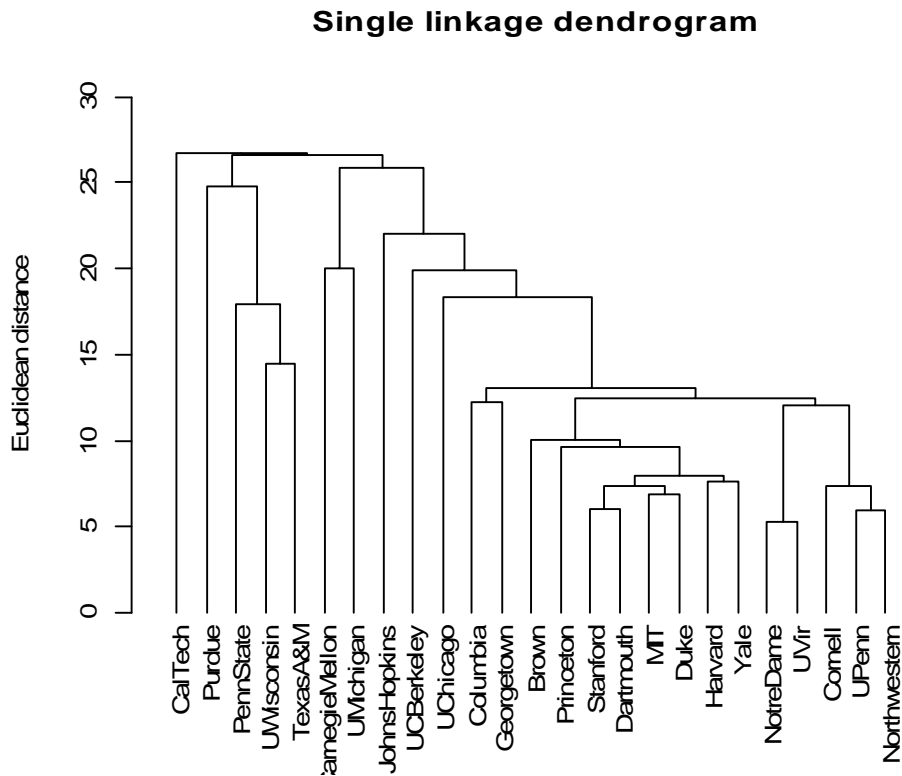


Figure 3.1: The single linkage dendrogram of the 25 U.S. universities.

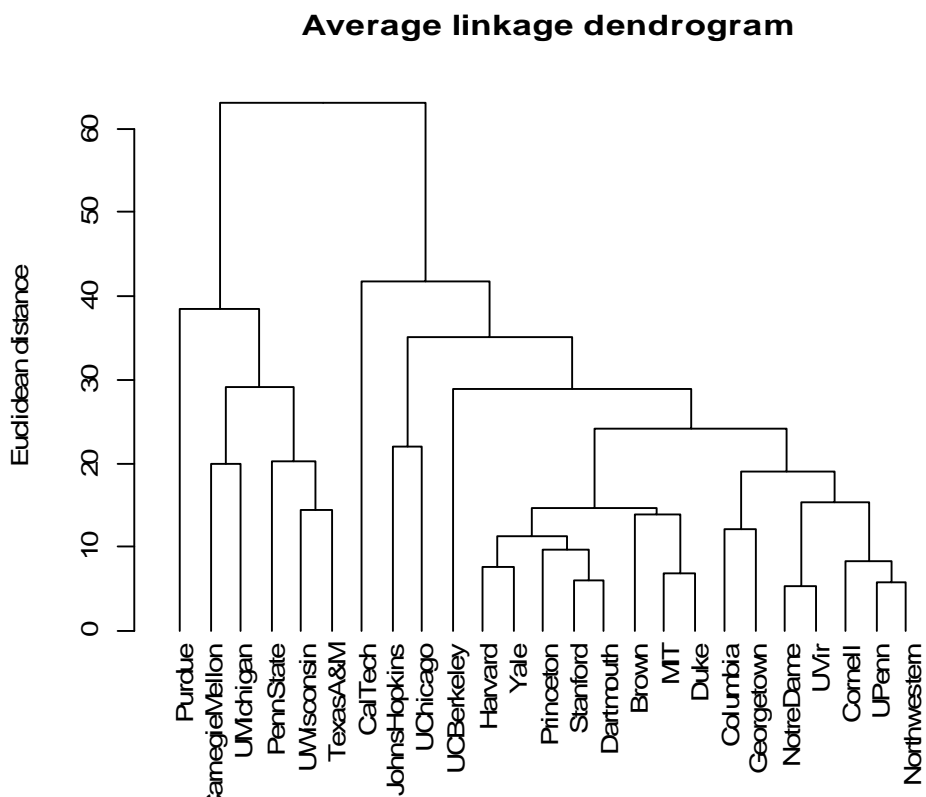


Figure 3.2: The average linkage dendrogram of the 25 U.S. universities.

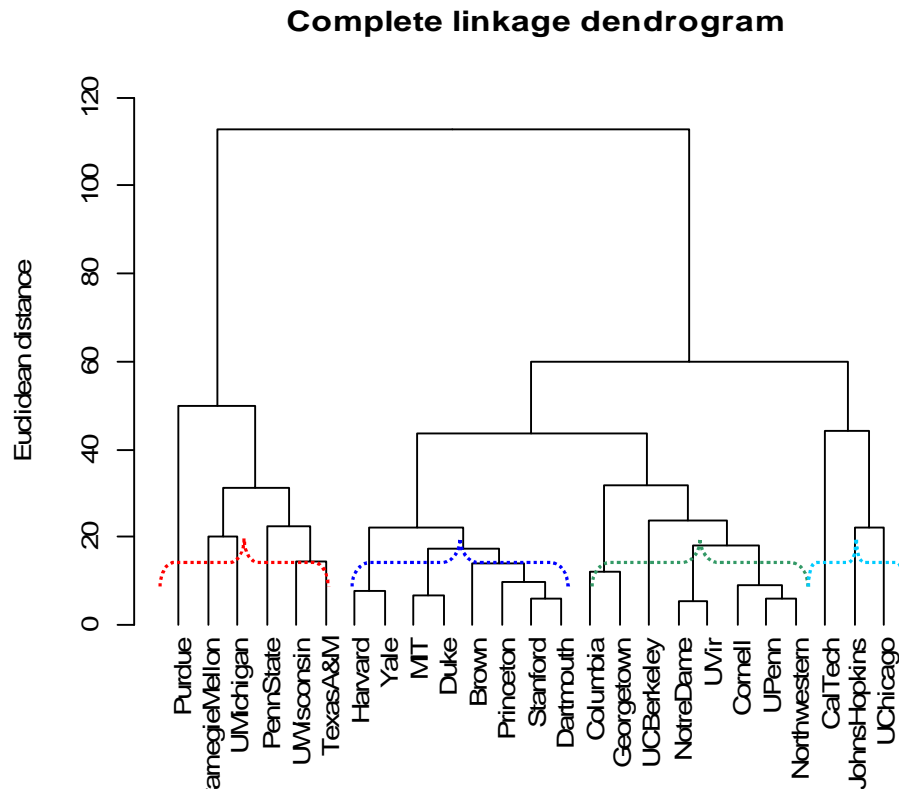


Figure 3.3: The complete linkage dendrogram of the 25 U.S. universities.

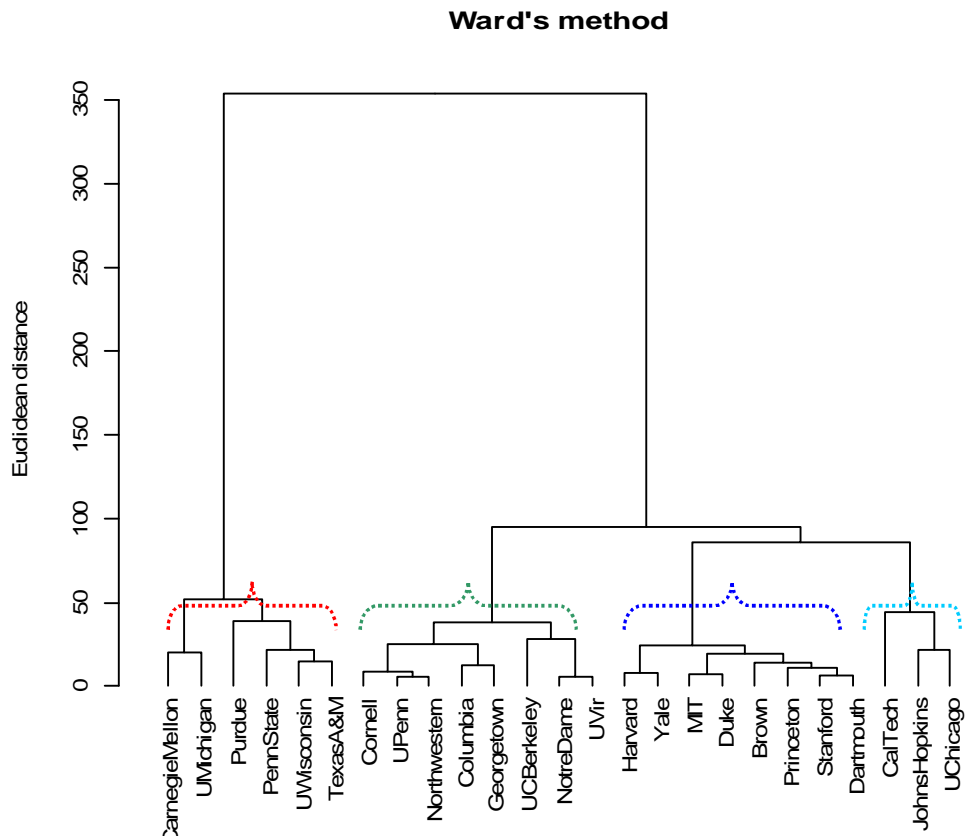


Figure 3.4: The dendrogram of Ward's method of the 25 U.S. universities.

3.6 Interpreting the cluster analysis results

By studying the dendrograms in Figures 3.1 to 3.4, one clearly sees some interesting cluster patterns for the universities. Figure 3.1, which represents the single linkage method, does not show any clear clusters. It almost seem like the whole data set is clustered as one group. In the case of the average linkage method (Figure 3.2), there appears to be two clusters. The first cluster contains 6 universities (Purdue, CarnegieMellon, UMichigan, PennState, UWisconsin and TexasA&M), while the rest of the universities form one large cluster. For the complete linkage and Ward's method (Figures 3.3 and 3.4) it appears if there are 4 distinct clusters which are

- Cluster 1: Purdue, CarnegieMellon, UMichigan, PennState, Uwisconsin and TexasA&M.
- Cluster 2: Cornell, UPenn, NorthWestern, Columbia, Georgetown, UC Berkely, NotreDame and UVir.
- Cluster 3: Harvard, Yale, MIT, Duke, Brown, Princeton, Stanford and Dartmouth.
- Cluster 4: CalTech, JohnsHopkins and UChicago.

These clusters are indicated by brackets on the above mentioned figures.

3.7 Summary

As mentioned before, the single linkage method has a drawback called the chaining phenomenon. For the single linkage and average linkage methods the clustering was not very effective for the universities. In the case of complete linkage and Ward's method, the clusters are similar and these seem to be much more effective methods than the single and average linkage methods. Stuetzle (1995) argues that some statisticians prefer complete linkage because a clearer interpretable dendrogram is often produced. Ward's method is limited to numerical data with an elliptical distribution. Complete linkage can be used for numerical and other types of data. For the rest of this thesis we will make use of complete linkage method when ever a cluster analysis is performed. In Chapter 6 we will make use the complete linkage method together with the Bray-Curtis and Jaccard dissimilarity measures in the analysis of a multidimensional binary data set.

Chapter 4

Metric and Nonmetric multidimensional scaling

4.1 Introduction

Multidimensional scaling (MDS) is a multivariate statistical technique, based on a distance or dissimilarity matrix, which allows us to visualise all the objects in a data set as points in a low dimensional space (or map). Note that the distance and dissimilarity matrices (\mathbf{D}) mentioned here are the same as in Section 3.3. The points in this space represent the objects such that the distances between the points in this space correspond as closely as possible to the original distance, $d_{ii'}$, between objects (Cox and Cox, 2001). Similar to cluster analysis, MDS is also an exploratory data analytic technique, but with MDS originating in the field of Psychometrics.

MDS can essentially be classified into two categories *i.e.* metric and nonmetric multidimensional scaling. Metric MDS (sometimes referred to as the classical MDS solution) will be discussed in Section 4.2. This approach makes use of the spectral decomposition to obtain the low dimensional space (see Mardia *et al.*, 1979). In Section 4.3 we will discuss the nonmetric MDS approach, which uses the metric MDS solution as a starting point in an optimization procedure. The idea with nonmetric MDS is to minimize the so-called stress function, proposed by Kruskal (1964), in order to obtain the low dimensional space.

Section 4.4 contains an illustration of both these approaches by using the different R functions of the `stats` and `MASS` packages. The function `cmdscale()` will be used to perform metric MDS, while the function `isoMDS()` is used to perform nonmetric MDS.

4.2 Metric multidimensional scaling (MMDS)

For metric MDS we use the data matrix $\mathbf{X}_{I \times J}$ to calculate the distance matrix $(\mathbf{D} = \{d_{i'j'}\})$ as described in Section 3.3. Each element in this matrix is then squared to obtain matrix of squared distances, $\mathbf{D}^* = \{d_{i'j'}^2\}$, which also satisfy the distance properties stated in Section 3.2. To obtain the MMDS solution we first construct matrix $\mathbf{A}_{I \times I}$ from this distance matrix:

$$\mathbf{A} = -\frac{1}{2}\mathbf{D}^*, \text{ with elements } a_{i'j'} = -\frac{1}{2}d_{i'j'}^2.$$

This matrix can be centred as follows:

$$\mathbf{B} = \mathbf{A} - I^{-1}\mathbf{A}\mathbf{J} - I^{-1}\mathbf{J}\mathbf{A} + I^{-2}\mathbf{J}\mathbf{A}\mathbf{J} = \mathbf{H}\mathbf{A}\mathbf{H}, \quad (4.1)$$

where $\mathbf{H} = \mathbf{I} - I^{-1}\mathbf{J}$ is the $(I \times I)$ centring matrix and

$$\mathbf{J}_{I \times I} = \begin{bmatrix} 1 & 1 & \mathbf{L} & 1 \\ 1 & 1 & \mathbf{L} & 1 \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 1 & 1 & \mathbf{L} & 1 \end{bmatrix}.$$

Consider the following results which can be found in Mardia *et al.* (1979):

- (a) Given that \mathbf{D} is a distance matrix and $\mathbf{B} = \mathbf{H}\mathbf{A}\mathbf{H}$ as defined in (4.1), then \mathbf{D} is Euclidean if and only if \mathbf{B} is positive semidefinite.
- (b) If \mathbf{B} is positive semidefinite, then a configuration of points in a Euclidean space can be obtained, using the spectral decomposition of \mathbf{B} . The spectral decomposition is defined as

$$\mathbf{B} = \sum_{i=1}^I I_i \mathbf{e}_i \mathbf{e}_i' = \mathbf{\Gamma} \mathbf{\Lambda} \mathbf{\Gamma}',$$

where $\mathbf{\Lambda} = \text{diag}(I_1, I_2, \dots, I_I)$ is the diagonal matrix of eigenvalues and

$\mathbf{\Gamma} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_I]$ is the matrix of corresponding eigenvectors.

(c) If \mathbf{D} is a matrix of similarities or dissimilarities, then $\mathbf{B} = \mathbf{H}\mathbf{A}\mathbf{H}$ would still be positive semidefinite under certain conditions (see Mardia *et al.*, 1979, p.402).

Note that the distances, similarities and dissimilarities defined in Chapter 3 results in \mathbf{B} being a positive semidefinite matrix, making them applicable to MMDS.

Once the spectral decomposition is applied to \mathbf{B} , a scatterplot of the first q (usually two or three) eigenvectors ($\mathbf{e}_i, i = 1, 2, 3$) is used to obtain a MDS map (see Figure 4.1 as an example). The plot reveals how close or far the objects lie in space. This plot can be helpful in identifying group structures or outliers in the data.

To establish the goodness-of-fit of MMDS, we make use of the eigenvalues to obtain a screeplot and the proportion of variation explain by the first q dimensions. The screeplot is a plot of the eigenvalues against the number of dimensions, q (see Figure 4.2). The purpose of the screeplot is to determine which number of dimensions is sufficient to represent the MDS map. The cut-off point for the number of dimensions is usually obtained where this graph makes the elbow shape. The proportion of variation explained by the first q dimensions,

$$\sum_{i=1}^q I_i / \sum_{i=1}^I I_i, \tag{4.2}$$

gives a measure of the goodness-of-fit. A small value represents a bad fit and a large value a good fit.

4.3 Nonmetric multidimensional scaling (NMDS)

The nonmetric MDS described in this section is an extension of the metric MDS given in Section 4.2. Let $\mathbf{D} = \{d_{ii'}\}$ be an $I \times I$ matrix of distances between the rows i and i' in \mathbf{X} . For the I objects in \mathbf{X} there are $M = \frac{1}{2}I(I-1)$ distances between pairs of different objects which are ranked as follows,

$$d_{i_1 i_1'}^{(q)} > d_{i_2 i_2'}^{(q)} > \dots > d_{i_M i_M'}^{(q)}, \quad (4.3)$$

where

$(i_1, i_1'), \dots, (i_M, i_M')$ = all pairs of objects of i and i' , $i < i'$

q = the number of dimensions in the low dimensional space.

The following outlines the NMDS proposed by Kruskal (1964). Kruskal defines the stress function as

$$Stress(q) = \left\{ \frac{\sum_{\substack{i, i'=1 \\ i < i'}}^M (d_{ii'}^{(q)} - \hat{d}_{ii'}^{(q)})^2}{\sum_{\substack{i, i'=1 \\ i < i'}}^M [d_{ii'}^{(q)}]^2} \right\}^{1/2}, \quad (4.4)$$

where

$d_{ii'}^{(q)}$ is the original distances in (4.3) and

$\hat{d}_{ii'}^{(q)}$ is the estimate of $d_{ii'}^{(q)}$ obtained from the low dimensional space.

Firstly we need to find $\hat{d}_{ii'}^{(q)}$ where,

$$\hat{d}_{i_1 i_1'}^{(q)} > \hat{d}_{i_2 i_2'}^{(q)} > \dots > \hat{d}_{i_M i_M'}^{(q)}, \quad (4.5)$$

such that $Stress(q)$ is as small as possible. Thus the NMDS problem is an optimization problem, quite different to MMDS. There is no algebraic solution to obtaining $\hat{d}_{ii}^{(q)}$ and therefore these values are obtained using an iterative procedure. Important to note here is that the starting values of (4.5) are the Euclidean distance obtained from the MMDS eigenvectors $(\mathbf{e}_i, i = 1, 2, \dots, q)$. These values are then updated in each step as (4.4) is being minimized, while keeping the same ranked order as (4.3). Figure 4.5 contains the iteration plot showing the optimization process of NMDS. Note how the $Stress(q)$ in this figure is high initially and then decreases (eventually reaching a minimum) as the number of iterations increase.

Once the iteration process ends, a low dimensional space is obtained for NMDS based on the chosen number of dimensions, q . Basically, these are the configuration of points in MMDS $(\mathbf{e}_i, i = 1, 2, \dots, q)$ that has moved around in space as the stress function (4.4) was being minimized. Figure 4.3 is an example of the low dimensional map of NMDS.

Similar to MMDS we can define measures of goodness-of-fit for NMDS. A plot of $Stress(q)$ against the number of dimensions, q , gives us a screeplot similar to Figure 4.2 in MMDS. An example of this NMDS screeplot are displayed in Figure 4.6. The following guidelines (Johnson and Wichern, 2007) can be used to determine the goodness-of-fit using $Stress(q)$:

Table 4.1: Guidelines for NMDS goodness-of-fit.

Stress	Goodness of fit
20%	Poor
10%	Fair
5%	Good
2.5%	Excellent
0%	Perfect

Another useful graph that is used in NMDS to determine the goodness-of-fit is called the Shepard diagram (Shepard, 1980; Groenen and van de Velden, 2004). This graph contains a plot of the distances $\hat{d}_{ii}^{(q)}$ vs $d_{ii}^{(q)}$ defined in (4.3) and (4.5). A monotone

regression line, which is a step function, is usually fitted on this plot to show the relationship between the distances. If the plot resembles a straight line, the NMDS is considered a good fit in the q -dimensional space. An example of a Shepard diagram is given in Figure 4.7.

4.4 Performing MMDS and NMDS in R

This section is aimed at demonstrating the MMDS and NMDS using the R software. To perform the MMDS we will use the function `cmdscale()` which is part of the `stats` package. The main arguments of this function is

```
R> cmdscale(d, k = 2, eig = FALSE, add = FALSE, x.ret = FALSE)
```

with object `d` being the distance matrix and `k` being the chosen number of dimensions for the MDS map. The object `eig` allows us to obtain the eigenvalues for the screeplot.

The function `isoMDS()`, which is part of the `MASS` package, will be used to perform NMDS. The following are its main arguments:

```
R> library(MASS)
R> isoMDS(d, y = cmdscale(d, k), k = 2, maxit = 50, trace = TRUE,
  tol = 1e-3, p = 2)
```

with objects `d` and `k`, the distance matrix and number of dimensions, respectively. The object `y` is a MMDS object containing initial values for $\hat{d}_{ii}^{(q)}$ in (4.5). The object `maxit` control the number of the iterations we want to use. The default number is 50 iterations.

Other important functions that is needed is the function `dist()` to obtain the distance matrix and the function `Shepard()` to obtain the Shepard diagram. The latter function is part of the `MASS` package and is applied using the instruction

```
R> Shepard(d, x, p = 2)
```

Chapter 4: Multidimensional scaling

The same data set of the 25 U.S. universities used in Chapter 3 will be used here to illustrate the use of the functions `cmdscale()` and `isoMDS()`.

The following function was written to perform metric MDS on the universities data. The output of the function is a two dimensional metric MDS plot (Figure 4.1) and a screeplot (Figure 4.2)

```
R> fix(MMDS) # R instructions to perform metric MDS
```

```
function (data)
{
  library(MASS)

  # Obtaining the Euclidean distances
  Distance<-dist(data)

  #Performing a Metric MDS
  fit1<-cmdscale(Distance,eig=TRUE,k=2)
  plot(fit1$points,type="n",xlab="Dimension 1", ylab="Dimension 2",
       main="Metric MDS")
  par(col="black",font=4,cex=0.50)
  chs<-substring(rownames(data),1,6)
  text(fit1$points,chs,col="red")
  abline(v=0,lty=3,col="green")
  abline(h=0,lty=3,col="green")

  windows()

  # Creating the Screeplot for MMDS
  plot(cmdscale(Distance,eig=TRUE,k=5)$eig,col="red",type="o",
       ylab="Eigenvalue",xlab="number of dimensions, q",
       main="Screeplot:Metric MDS")
}
```

```
R> MMDS(universities) # executing the MMDS function
```

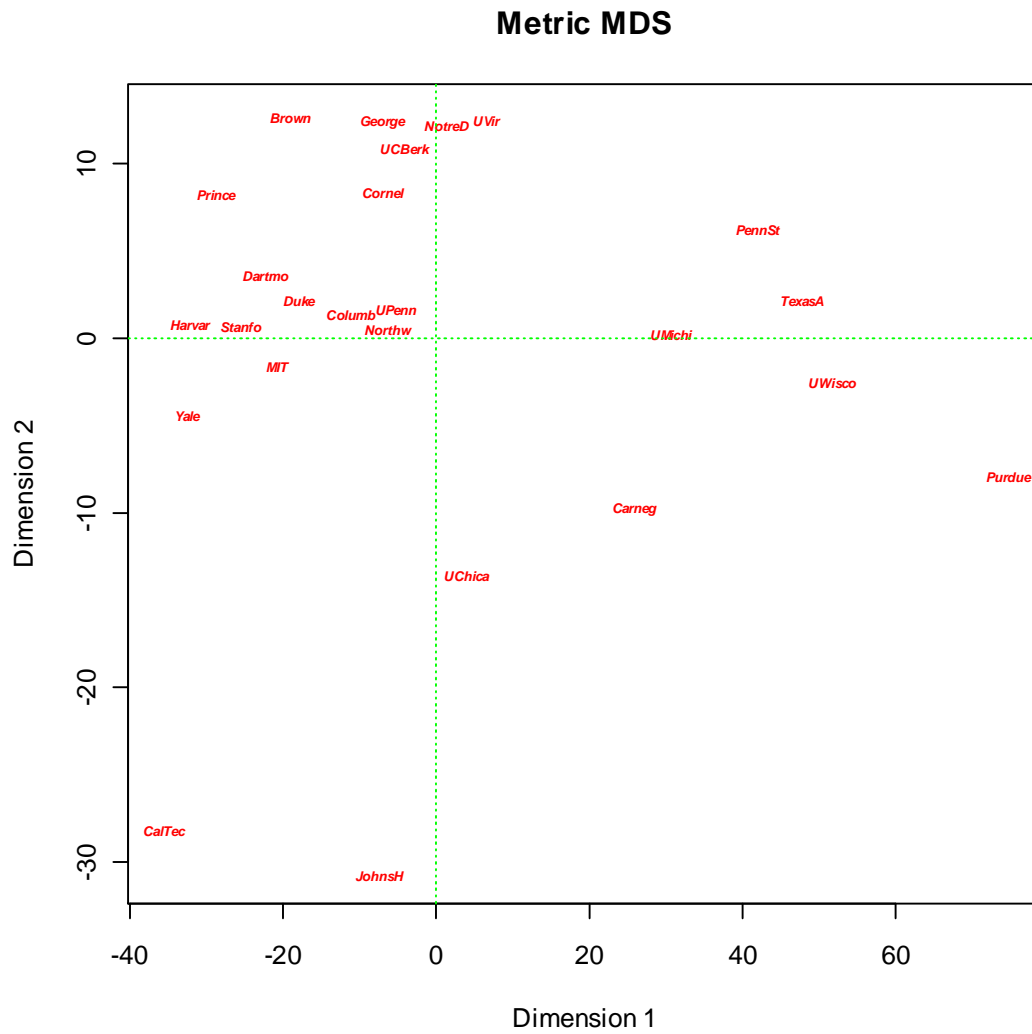


Figure 4.1: The metric MDS plot of the 25 U.S. universities.

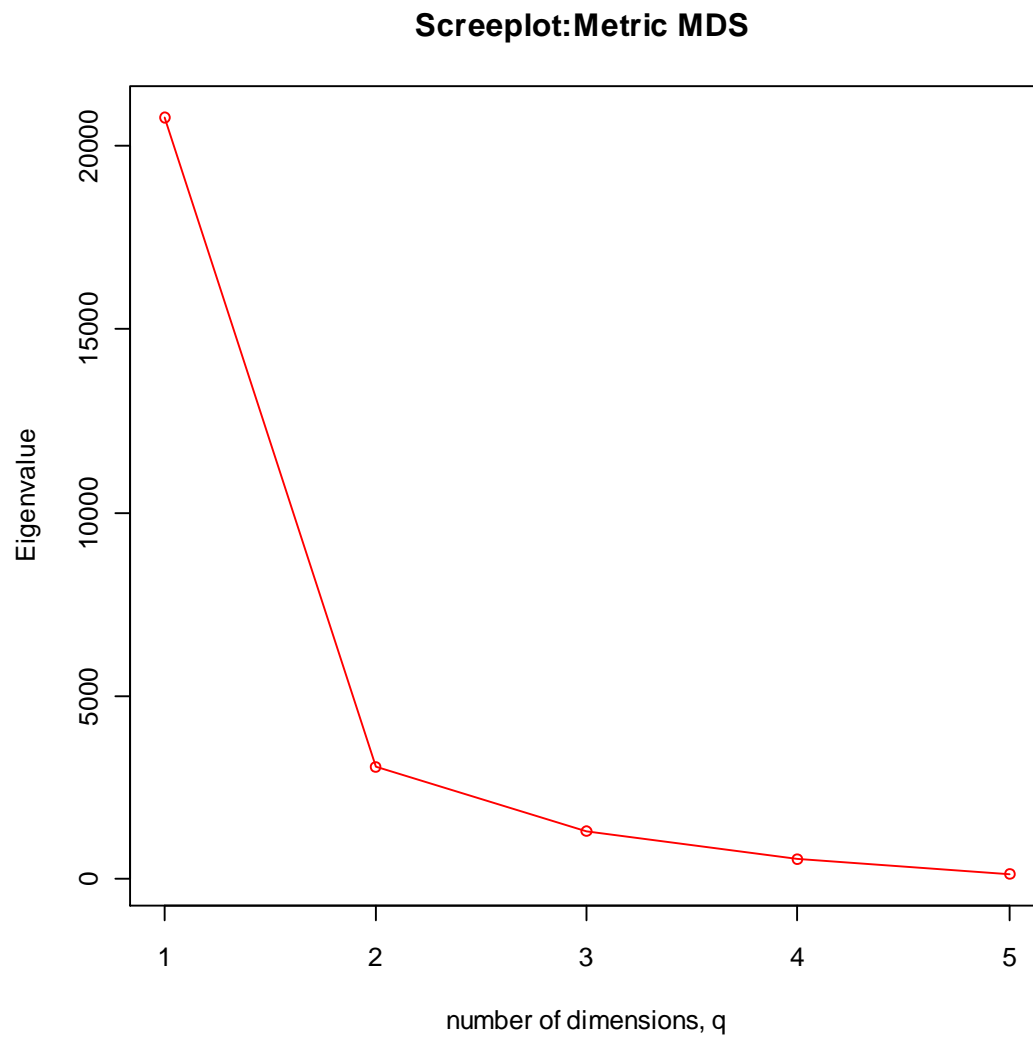


Figure 4.2: The screplot of metric MDS.

Chapter 4: Multidimensional scaling

The next function was written to perform nonmetric MDS on the universities data. The output of this function is a nonmetric MDS plot in two dimensions (Figure 4.3).

```
R> fix(NMDS) # R instructions to perform nonmetric MDS
```

```
function (data)
{
  library(MASS)

  # Obtaining the Euclidean distances

  Distance<-dist(data)

  windows()

  # Performing a Non-metric MDS

  fit2<-isoMDS(Distance, k=2)
  plot(fit2$points,type="n",xlab="Dimension 1", ylab="Dimension 2",
       main="Nonmetric MDS")
  par(col="black",font=4,cex=0.50)
  chs<-substring(rownames(data),1,6)
  text(fit2$points,chs,col="red")
  abline(v=0,lty=3,col="green")
  abline(h=0,lty=3,col="green")
}
```

```
R> NMDS(universities) # executing the NMDS function
  initial value 6.884722
  iter 5 value 5.403032
  iter 10 value 4.606783
  final value 4.440003
  converged
```

By using the following instructions one can place clusters on the existing MDS plot produced by the above function. The plot with clusters is displayed in Figure 4.4.

```
R> dis<-dist(universities,method="euclidean")
R> cluster<-hclust(dis,method="complete")
R> grps<-cutree(cluster,h=50)
R> fit2<-isoMDS(dis, k=2,trace=FALSE)
R> ordispider(fit2,grps,lty=2,col="red")
```

The function `cutree()` allows us to select the number of cluster $k=$ or the height $h=$ at which the clusters should be chosen. The function `ordispider()`, which is part of the `vegan` package, uses the results from `cutree()` and `hclust()` to display the clusters as seen in Figure 4.4. Besides `ordispider()`, we could also use `ordihull()`.

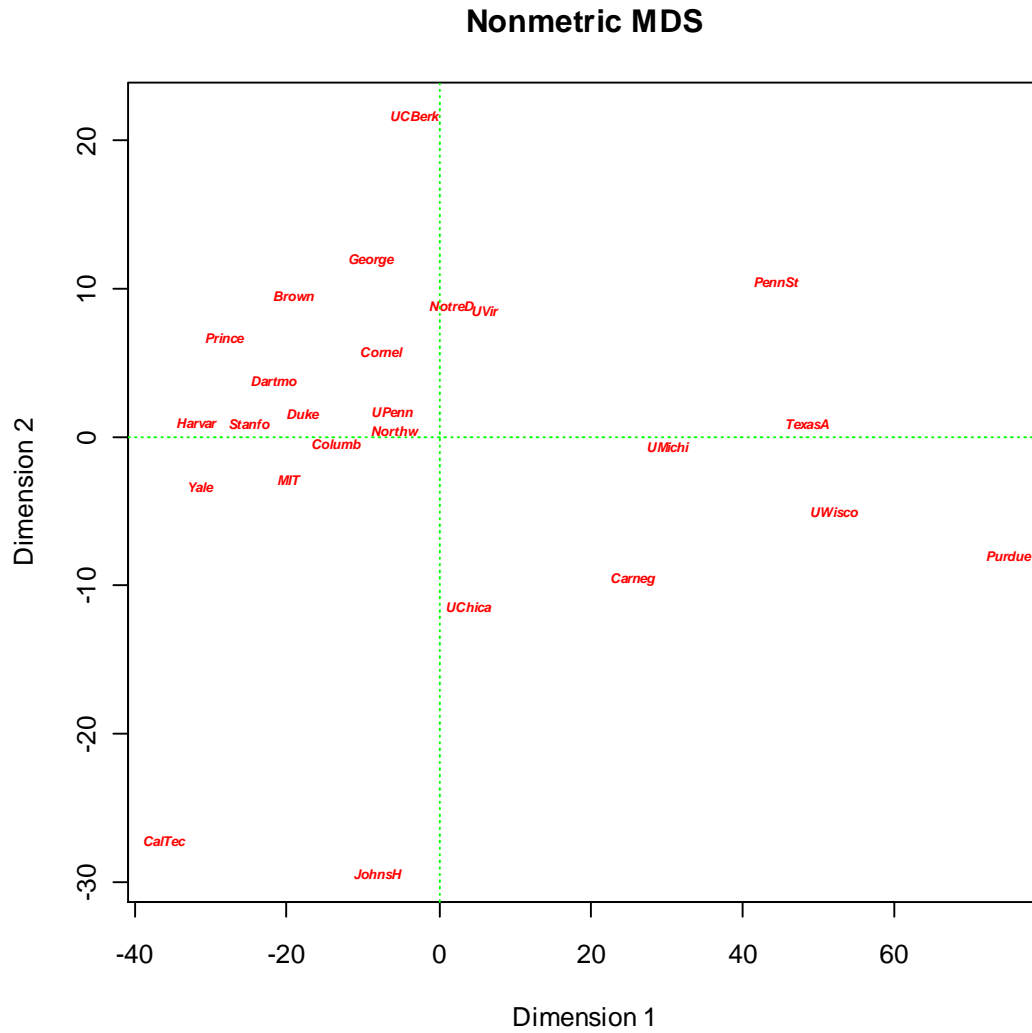


Figure 4.3: The nonmetric MDS plot of the 25 U.S. universities.

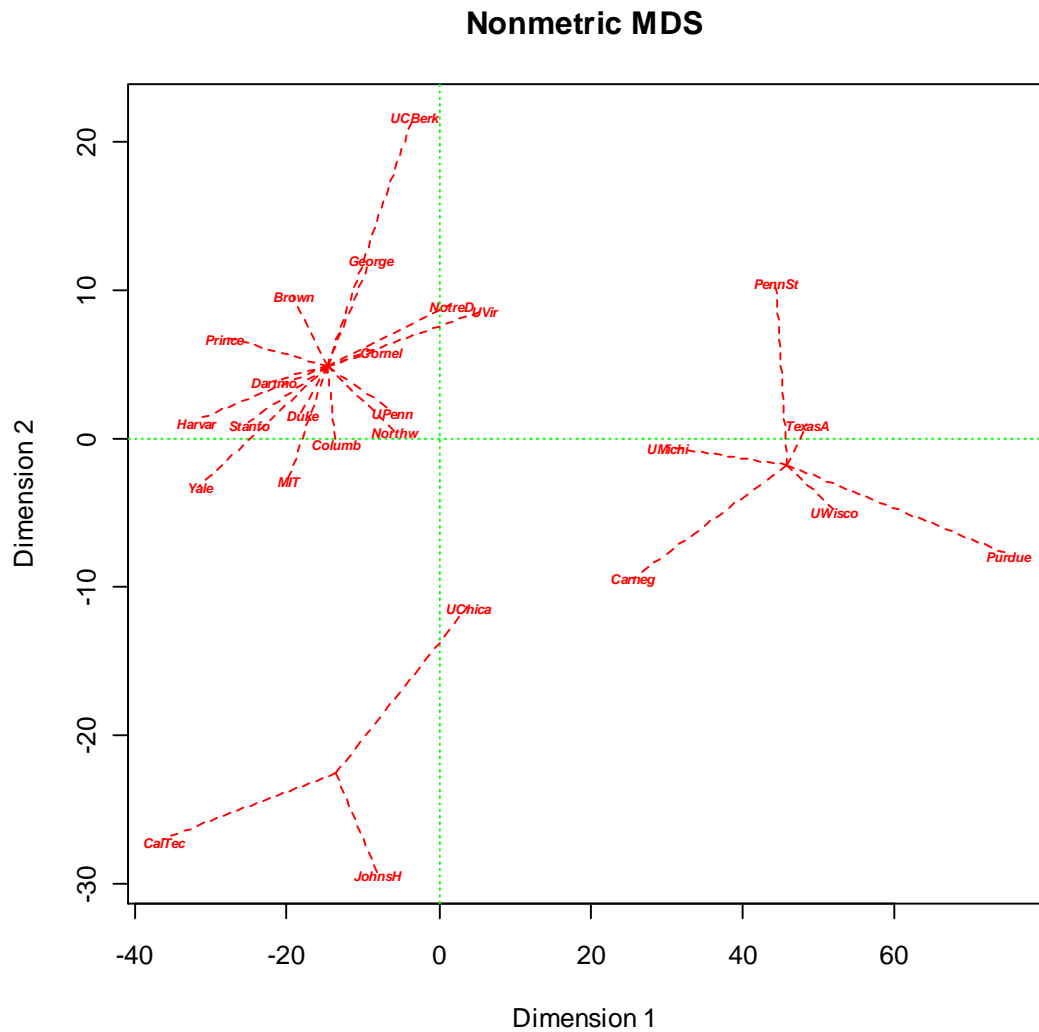


Figure 4.4: The nonmetric MDS plot of the 25 U.S. universities with clusters obtained using `ordispider()`, `hclust()` and `cutree()`.

Chapter 4: Multidimensional scaling

The function below performs nonmetric MDS on the universities data. The output of this function is the iteration plot (Figure 4.5), screeplot (Figure 4.6) and the Shepard diagram (Figure 4.7).

```
R> fix(MDS.plots) # R instructions for the diagnostic plots
```

```
function (data)
{
  library(MASS)

  Distance<-dist(data)

  # Iteration plot

  STRESS<-rep(0,100)

  for( i in 1:100){
    STRESS[i]<-isoMDS(Distance,maxit=i,trace=FALSE)$stress
  }

  plot(1:100,STRESS,type="o",ylab="STRESS (q)",xlab="Number of
    iterations",col="red",main="Iteration plot")

  # Screeplot
  STRESS2<-rep(0,5)

  for(i in 1:5){
    STRESS2[i]<-isoMDS(Distance,k=i,trace=FALSE)$stress
  }

  windows()

  plot(1:5,STRESS2,type="o",ylab="STRESS (q)",xlab="number of
    dimensions, q",col="red",main="Screeplot: Nonmetric MDS")

  # Shepard diagram
  NMDS<-isoMDS(Distance,trace=FALSE)
  Shep<-Shepard(Distance,NMDS$points)

  windows()

  plot(Shep$x,Shep$y,cex=0.75,xlab="dissimilarities",ylab="distances",
    main="Shepard diagram ")

  lines(Shep$x,Shep$yf,type="l",col="red")

}
```

```
R> MDS.plots(universities) # executing the function
```

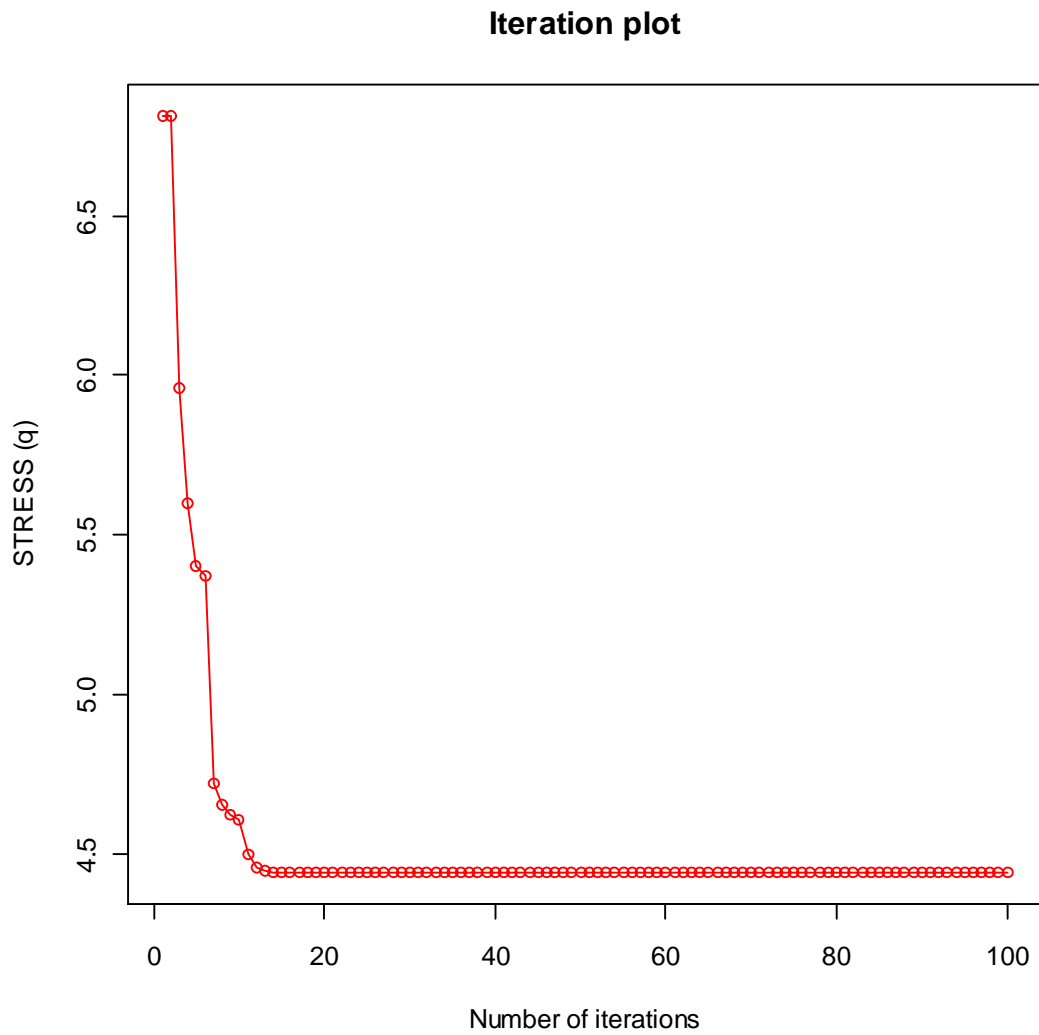


Figure 4.5: The iteration plot of nonmetric MDS.

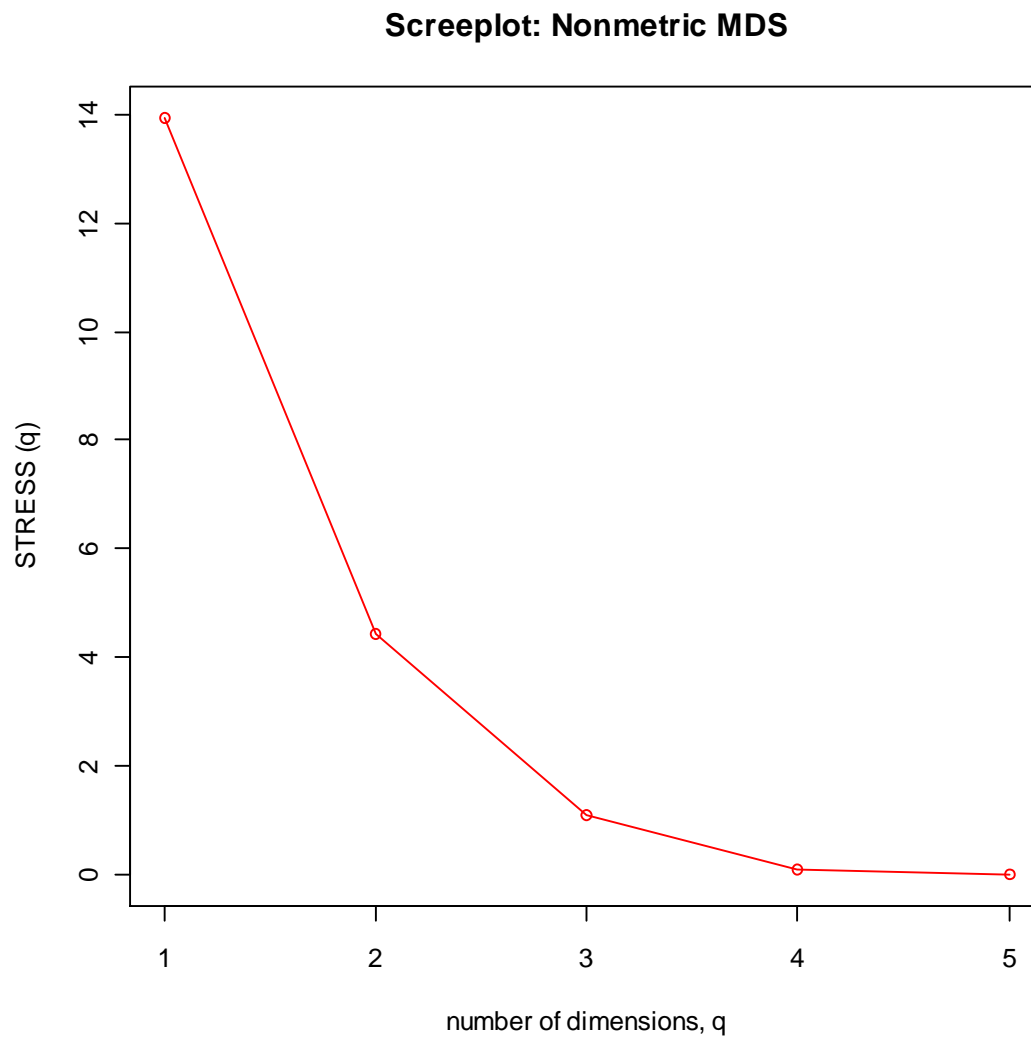


Figure 4.6: The screplot for nonmetric MDS.

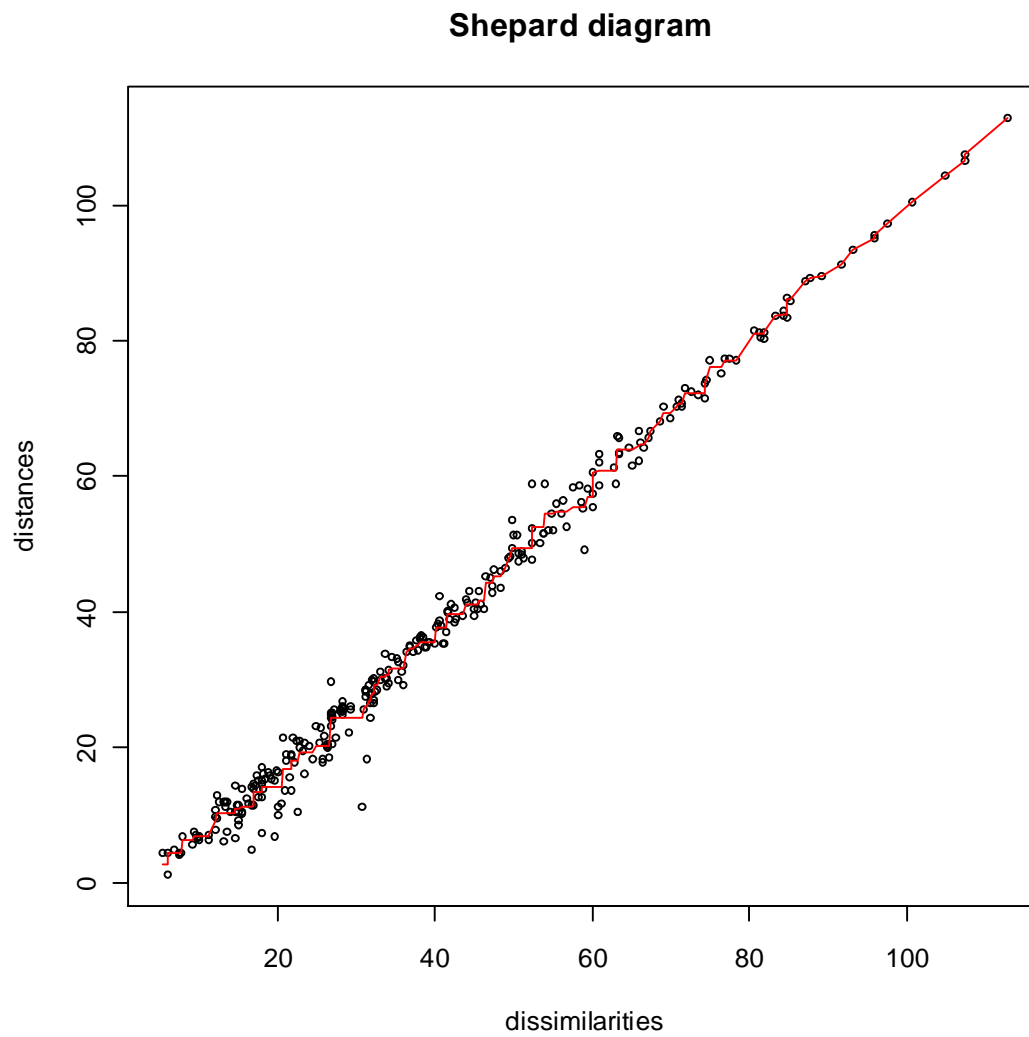


Figure 4.7: The Shepard diagram of nonmetric MDS.

4.5 Interpreting the MDS results

The metric MDS plot is displayed in Figure 4.1 in two dimensions. It is clear from this plot that there are differences among the universities. The screeplot of the eigenvalues in Figure 4.2 displays the goodness-of-fit. From this plot we can clearly see that the first two dimensions is sufficient to explain most of the variation.

The nonmetric MDS gives us quite similar results. Figure 4.3 is the nonmetric MDS plot in two dimensions. There were three major clusters in the data and this is shown in Figure 4.4 with the spider charts. The three clusters are the universities that are very similar. The screeplot in Figure 4.6 is a graph which displays the goodness-of-fit for the nonmetric MDS. The stress is used here as a measure of goodness-of-fit and Figure 4.6 shows that most of the variation is again explained by the first two dimensions. The final stress value was obtained as 4.44, which is a good fit according to Table 4.1. The iteration plot in Figure 4.5 shows the initial stress value of 6.8847 at the first iteration. The stress value dropped dramatically in the first 10 iterations and it reached a minimum at 4.44 after about 11 iterations. The Shepard diagram in Figure 4.7 also shows how well the MDS fit. Since the points lie close to the step regression function, we can conclude that the two dimensional nonmetric MDS is a good representation of the original data in the lower space.

The metric and non-metric MDS plots are very similar. They both contain the same groupings of the 25 universities and for both the two dimensional representation is sufficient. The groups identified in Figure 4.3 also agree with the clusters obtained for the complete linkage and Ward's method in Figures 3.3 and 3.4. These clusters are depicted in Figure 4.4.

4.6 Summary

In this chapter we have demonstrated both metric and nonmetric multidimensional scaling. The nonmetric MDS can use metric MDS output (eigenvectors) as starting values for the distances in the stress function (expression 4.4). Thus, nonmetric MDS allows for a much better configuration of the raw data in a low dimensional space.

Both metric and nonmetric MDS use a distance or dissimilarity matrix as input. The output for both methods is a plot in a low dimensional space. For the metric MDS this plot is obtained by using the eigenvector solution. In the case of nonmetric MDS, this plot is obtained by using an iterative process in which the stress function is minimized. We have also discussed measures for assessing the goodness-of-fit of these methods. Metric MDS uses the eigenvalues to obtain a measure of the goodness-of-fit. For nonmetric MDS we explained the stress value and the Shepard diagram as tools for assessing the goodness-of-fit.

We also discussed the functions `cmdscale()` and `isoMDS()` which can be used for metric and nonmetric MDS respectively. Another R function which performs nonmetric MDS is the `metaMDS()` function in the `vegan` package.

Chapter 5

Inference using distance matrices

5.1 Introduction

In the previous four chapters we have focused entirely on the exploratory analysis of multidimensional data. In this chapter we turn our focus to statistical inference with multiple populations. The aim of this chapter is to explain and understand three techniques which can be used to test for significant differences among several groups. The first technique that we will discuss is the well-known analysis of variance (ANOVA), which is used to test for differences among group means in the univariate case. The second inference technique is called multivariate analysis of variance (MANOVA), which is a direct extension of ANOVA to the multivariate case. Both the above mentioned methods are parametric techniques and are based on strict assumptions, which will be discussed in the sections to follow. For more detail on ANOVA and MANOVA see Johnson and Wichern (2007). In practice the assumptions for ANOVA and MANOVA are not always met. For reasons such as this an alternative to ANOVA and MANOVA is required. In this chapter we will discuss a third inferential technique called the analysis of distance (AOD). The AOD was proposed by Anderson (2001) and it offers an alternative to ANOVA and MANOVA. AOD is a non-parametric technique and is not based on any assumptions. We will also illustrate how the three techniques can be applied in R using the well-known Iris data set as an example.

5.2 The one-way analysis of variance

The univariate analysis of variance (ANOVA) is a very common and widely used method for statistical tests of factor effects and their interaction effects. This method is most often used to analyze the outcomes of designed experiments such as completely randomized designs, randomized block designs, Latin square designs and

Chapter 5: Inference using distance matrices

factorial designs. Consider a single factor experiment (*eg.* randomized design) involving g factor levels (or treatments) and a single numerical response measured on n observations in each level. The data set for such an experiment are described in Table 5.1. For our discussion in this chapter we will assume that all the groups are of size n .

Table 5.1: The data set for a single factor experiment

Treatment (group)	Observations	Total	Average
1	$y_{11} \ y_{12} \ y_{13} \ \mathbf{L} \ y_{1n}$	y_{1g}	\bar{y}_{1g}
2	$y_{21} \ y_{22} \ y_{23} \ \mathbf{L} \ y_{2n}$	y_{2g}	\bar{y}_{2g}
\mathbf{M}	\mathbf{M}	\mathbf{M}	\mathbf{M}
g	$y_{g1} \ y_{g2} \ y_{g3} \ \mathbf{L} \ y_{gn}$	y_{gg}	\bar{y}_{gg}
	Total	y_{gg}	\bar{y}_{gg}

In this table, y_{ij} is the j^{th} observation from the i^{th} group, y_{ig} is the total of the i^{th} treatment, \bar{y}_{ig} is the average of the i^{th} treatment, y_{gg} is the grand total and \bar{y}_{gg} is the grand average. The hypothesis test of interest here is usually given by,

$$\begin{aligned}
 H_0 : m_1 = m_2 = \mathbf{L} = m_g \\
 H_1 : \text{at least one of the } m\text{'s are different,}
 \end{aligned}
 \tag{5.1}$$

where we test for the equality of the treatment means. The one-way ANOVA is used to perform this hypothesis test. The ANOVA is based on the following assumptions:

- observations in each group are from a normally distributed population,
- observations are drawn independently,
- groups have equal population variances.

If these assumptions are not fulfilled, the results of the ANOVA may be questionable.

Chapter 5: Inference using distance matrices

The ANOVA partitions the total variability in the data into different components (Montgomery, 2005). The total sum of squares, $SS_{Total} = \sum_{i=1}^g \sum_{j=1}^n (y_{ij} - \bar{y}_{\mathbf{g}})^2$, contains this overall variability in the data. In one-way ANOVA the total sum of squares is decomposed into the sum of squares due to treatments ($SS_{Treatments}$) and the sum of squares due to error (residual) (SS_{Error}), *i.e.*

$$SS_{Total} = SS_{Treatments} + SS_{Error},$$

which is formulated as

$$\sum_{i=1}^g \sum_{j=1}^n (y_{ij} - \bar{y}_{\mathbf{g}})^2 = n \sum_{i=1}^g (\bar{y}_{i\mathbf{g}} - \bar{y}_{\mathbf{g}})^2 + \sum_{i=1}^g \sum_{j=1}^n (y_{ij} - \bar{y}_{i\mathbf{g}})^2. \quad (5.2)$$

The test statistic for this hypothesis test is derived from these components as the F -ratio

$$F_0 = \frac{SS_{Treatments} / (g - 1)}{SS_{Error} / (ng - g)} = \frac{MS_{Treatments}}{MS_{Error}} \quad (5.3)$$

and the corresponding critical value is obtained from the F -distribution with degrees of freedom $df_1 = g - 1$ and $df_2 = ng - g$. The output of the ANOVA is usually displayed in a table, see Table 5.2. If the data are normally distributed, the quantity (5.3) follows an F -distribution and therefore the associated p -value can be obtained from this distribution as: $p\text{-value} = P(F > F_0)$. If the p -value is less than the specified level of significance, the null-hypothesis in (5.1) is rejected. Otherwise it is not rejected.

Table 5.2: One-way ANOVA table

Source of variation	Sum of squares	Degrees of freedom	Mean square	F -value
Treatments	$SS_{Treatments}$	$g - 1$	$MS_{Treatments}$	$F_0 = \frac{MS_{Treatments}}{MS_{Error}}$
Error (Residual)	SS_{Error}	$ng - g$	MS_{Error}	
Total	SS_{Total}	$ng - 1$		

5.3 The one-way multivariate analysis of variance

Next we consider the multivariate analysis of variance (MANOVA), which is a generalization of the univariate ANOVA described above. For a single factor experiment with g treatments, we now measure multiple numerical responses (p variables) on the n observations in a treatment group. Since there are p variables for each observation in each group, we have a multivariate setup and the hypotheses of interest are formulated as follows,

$$H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \dots = \boldsymbol{\mu}_g$$

$$H_1 : \text{at least one of the } \boldsymbol{\mu}'s \text{ are different.} \quad (5.4)$$

In this instance the one-way MANOVA is used to test for the equality of the mean vectors. Similar to the ANOVA, the MANOVA is based on the following assumptions:

- observation vectors in each group are from a multivariate normal population,
- observations vectors are drawn independently from each population,
- groups have equal population covariance matrices.

Chapter 5: Inference using distance matrices

The data for a one-way MANOVA are described in Table 5.3, where \mathbf{y}_{ij} is the vector of p variables for the j^{th} observation in the i^{th} treatment, \mathbf{y}_{ig} is a vector of totals for the i^{th} treatment, $\bar{\mathbf{y}}_{ig}$ is the average vector for the i^{th} treatment, $\mathbf{y}_{\mathbf{g}}$ is the grand total and $\bar{\mathbf{y}}_{\mathbf{g}}$ is the vector of overall averages.

Table 5.3: The data set for a single factor experiment with multivariate responses

Treatment (group)	Observations	Total	Average
1	$\mathbf{y}_{11} \mathbf{y}_{12} \mathbf{y}_{13} \quad \mathbf{L} \quad \mathbf{y}_{1n}$	\mathbf{y}_{1g}	$\bar{\mathbf{y}}_{1g}$
2	$\mathbf{y}_{21} \mathbf{y}_{22} \mathbf{y}_{23} \quad \mathbf{L} \quad \mathbf{y}_{2n}$	\mathbf{y}_{2g}	$\bar{\mathbf{y}}_{2g}$
\mathbf{M}	\mathbf{M}	\mathbf{M}	\mathbf{M}
g	$\mathbf{y}_{g1} \mathbf{y}_{g2} \mathbf{y}_{g3} \quad \mathbf{L} \quad \mathbf{y}_{gn}$	\mathbf{y}_{gg}	$\bar{\mathbf{y}}_{gg}$
	Total	$\mathbf{y}_{\mathbf{g}}$	$\bar{\mathbf{y}}_{\mathbf{g}}$

The overall variation in the data can be summarized by the matrix total sum of squares and cross products, $\sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})'$. Similar to the decomposition in (5.2), we construct the decomposition of the matrix total sum of squares and cross products into two components: the treatment sum of squares and cross products and the error sum of squares and cross products *i.e.*

$$\sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})' = n \sum_{i=1}^g (\bar{\mathbf{y}}_{ig} - \bar{\mathbf{y}}_{\mathbf{g}})(\bar{\mathbf{y}}_{ig} - \bar{\mathbf{y}}_{\mathbf{g}})' + \sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{ig})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{ig})'. \quad (5.5)$$

The above matrices are summarized in Table 5.4.

Table 5.4: One-way MANOVA table

Source of variation	Matrix of sum of squares and cross products
Treatments	$\mathbf{B} = n \sum_{i=1}^g (\bar{\mathbf{y}}_{i\mathbf{g}} - \bar{\mathbf{y}}_{\mathbf{g}})(\bar{\mathbf{y}}_{i\mathbf{g}} - \bar{\mathbf{y}}_{\mathbf{g}})'$
Error (Residual)	$\mathbf{W} = \sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\mathbf{g}})'$
Total	$\mathbf{B} + \mathbf{W} = \sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})'$

The next step in the MANOVA is to obtain the appropriate test statistic. For this we first need to obtain Wilks' lambda (Wilks, 1932), which is defined by the following ratio of generalized variances

$$\Lambda^* = \frac{|\mathbf{W}|}{|\mathbf{B} + \mathbf{W}|} = \frac{\left| \sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{i\mathbf{g}})' \right|}{\left| \sum_{i=1}^g \sum_{j=1}^n (\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})(\mathbf{y}_{ij} - \bar{\mathbf{y}}_{\mathbf{g}})' \right|}. \quad (5.6)$$

To perform the hypothesis test in (5.4) we need to find the distribution of Λ^* , which can be derived for the cases given in Table 5.5 (Johnson and Wichern, 2007). Using Table 5.5 the test statistic and critical value for the hypothesis test can be obtained for any given number of groups and variables. Similar to the ANOVA, the corresponding p -value for the MANOVA is also based on the F -distribution. In the case of MANOVA, the degrees of freedom, df_1 and df_2 , are dependent on the number of groups (g), the number of variables (p) and the sample sizes (n) (see Table 5.5 for more detail).

Table 5.5: The distribution of Wilks' lambda, assuming that the sample sizes in each group are the same

Number of variables	Number of groups	Distribution
$p = 1$	$g \geq 2$	$\left(\frac{ng - g}{g - 1}\right) \left(\frac{1 - \Lambda^*}{\Lambda^*}\right) \sim F_{df_1=(g-1); df_2=(ng-g)}$
$p = 2$	$g \geq 2$	$\left(\frac{ng - g - 1}{g - 1}\right) \left(\frac{1 - \sqrt{\Lambda^*}}{\sqrt{\Lambda^*}}\right) \sim F_{df_1=2(g-1); df_2=(ng-g-1)}$
$p \geq 1$	$g = 2$	$\left(\frac{ng - p - 1}{p}\right) \left(\frac{1 - \Lambda^*}{\Lambda^*}\right) \sim F_{df_1=p; df_2=ng-p-1}$
$p \geq 1$	$g = 3$	$\left(\frac{ng - p - 2}{p}\right) \left(\frac{1 - \sqrt{\Lambda^*}}{\sqrt{\Lambda^*}}\right) \sim F_{df_1=2p; df_2=2(ng-p-2)}$

5.4 The analysis of distance

Anderson (2001) proposed a non-parametric hypothesis test using similar reasoning as in the cases of ANOVA and MANOVA. This approach to hypothesis testing of multiple groups involves a distance matrix (see Chapter 3). The idea here is to decompose the distance matrix rather than the variance. In Anderson (2001) both the one-way and two-way analysis of distance (AOD) is explained. In this section we will discuss only the one-way AOD. Similarities between AOD and ANOVA and MANOVA are also highlighted.

As mentioned in the previous sections, ANOVA and MANOVA are based on several assumptions in order for the analysis to be applicable. These assumptions are not important when we perform the AOD. The AOD allows us to compare treatments with different types of measurements (*eg.* numerical, count and binary). When we perform the AOD we test the hypotheses:

H_0 : the locations of groups are the same

H_1 : the locations of groups are different. (5.7)

To obtain the test statistic and p -value we make use of a distance matrix. Given the setup of the data in Tables 5.1 or 5.3, obtain the $N \times N$ matrix of distances (or dissimilarities) \mathbf{D} with $N = ng$. The distance and dissimilarity measures discussed in Chapter 3 can be used to obtain this matrix. Within the matrix \mathbf{D} we can obtain the $n \times n$ sub-matrices corresponding to each group. Define the following sum of squares based on the elements in \mathbf{D} :

- Total sum of squared distances

$$SS_{Total} = \frac{1}{N} \sum_{i < i'}^N d_{ii'}^2.$$

Chapter 5: Inference using distance matrices

- Treatment sum of squared distances

$$SS_{Treatments} = \frac{1}{n} \sum_{i < i'}^N d_{ii'}^2 I_{ii'} ,$$

where I is an indicator function having 1 if objects i and i' are in the same group and 0 otherwise.

- Error sum of squared distances

$$SS_{Error} = SS_{Total} - SS_{Treatment} .$$

Thus we can decompose the total sum of squared distances similar to (5.2) and (5.5). The output of the AOD can also be summarized in a table. This summary table is displayed in Table 5.6.

For the AOD a pseudo F -test statistic is used which is analogous to the F -test statistic (5.3). The pseudo F -value is obtained as the ratio

$$F = \frac{SS_{Treatments} / (g - 1)}{SS_{Error} / (ng - g)} = \frac{MS_{Treatments}}{MS_{Error}} . \quad (5.8)$$

Note that the AOD follows the same idea as the ANOVA. However, it should be remembered that the AOD can be performed on a univariate as well as a multivariate data set having several groups.

Table 5.6: One-way AOD table

Source of variation	Sum of Squares	Degrees of Freedom	Mean Square	Pseudo F -value
Treatments	$SS_{Treatments}$	$g - 1$	$MS_{Treatments}$	$F = \frac{MS_{Treatments}}{MS_{Error}}$
Error (Residual)	SS_{Error}	$ng - g$	MS_{Error}	
Total	SS_{Total}	$ng - 1$		

The next step in the AOD is to obtain the p -value for the hypothesis test. Since we do not make any distributional assumptions when performing AOD, the p -value cannot be obtained from a known distribution function. In this case we make use of permutations to obtain the distribution. The following steps explain the process:

1. Let F in (5.8) be the F -ratio from the original data.
2. Perform a large number of permutations (say P) on the group label and each time calculate the distance matrix $(\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_P)$ and the pseudo F -ratio $(F_1^P, F_2^P, \dots, F_P^P)$.
3. The p -value is obtain as:

$$p\text{-value} = \frac{\text{number of } F^P \geq F}{P}. \quad (5.9)$$

According to Anderson (2001) at least $P=1000$ permutations should be done when a 0.05 level of significance is used and at least $P=5000$ permutations should be performed for a 0.01 level of significance.

5.5 Performing an analysis of variance in R

To perform the analysis of variance we will make use of the Iris data set (Anderson, 1935; Fisher, 1936). The data set consists of 3 groups (Iris species) and 4 numerical variables measured on 50 observations in each group. The analysis will be performed using the MANOVA and AOD techniques discussed in the previous sections. In both cases we test the hypotheses

H_0 : the locations of the 3 groups are the same

H_1 : the locations of the 3 groups are different.

To perform the MANOVA we will make use of the `manova()` function in the R package `stats` and to perform the AOD we will use the `adonis()` function in the `vegan` package.

5.5.1 A multivariate analysis of variance in R

A MANOVA is performed in R using the function `manova()` and the arguments of this function is a formula (as can be seen below). The function `summary()` is used together with the `manova()` function to obtain the MANOVA output.

```
R> manova(formula,data,...)
R> summary(object,
            test = c("Pillai", "Wilks", "Hotelling-Lawley", "Roy"),
            intercept = FALSE, tol = 1e-7, ...)
```

The `summary()` function uses the `manova()` object and one can also specify which test should be used. In Section 5.3 Wilks' lambda was discussed and we will use this test in our illustration of the `manova()` function. The following R instructions are used to perform the one-way MANOVA. The object `Y` is the vector containing the labels of the three groups and `X` is a data matrix with the four numerical variables.

Chapter 5: Inference using distance matrices

```
R> Y<-as.factor(iris[,5])
R> X<-as.matrix(iris[,1:4])
```

The output for the MANOVA is summarized below.

```
R> manova(X~Y)
Call:
  manova(X ~ Y)

Terms:
              Y Residuals
resp 1          63.2121   38.9562
resp 2          11.3449   16.9620
resp 3         437.1028   27.2226
resp 4           80.4133    6.1566
Deg. of Freedom      2      147

Residual standard error: 0.5147894 0.3396877 0.4303345 0.2046500
Estimated effects may be unbalanced
```

```
> summary(manova(X~Y),test="Wilks")
              Df  Wilks approx F num Df den Df    Pr(>F)
Y              2  0.023  199.145      8   288 < 2.2e-16 ***
Residuals 147
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

The p -value for this analysis is <0.001 , indicating that we reject the null hypothesis. Thus, there is enough evidence to conclude that the three groups (species) are significantly different.

5.5.2 An analysis of distance in R

To perform the AOD in R we will use the function `adonis()`, which is part of the `vegan` package. Below are the arguments of this function

```
R> adonis(formula, data, permutations = 999, method = "bray",
          strata = NULL, contr.unordered = "contr.sum",
          contr.ordered = "contr.poly", ...)
```

The `formula` object is similar to the one used in `manova()`. The number of permutations can be specified in the `permutations` argument. This function performs an analysis of distance and therefore requires the calculation of a distance matrix on the data. To obtain the distance matrix the function `vegdist()`, which is also part of the `vegan` package, is used as default. The argument `method` calls the `vegdist()` function. The function `vegdist()`, given below, works similar to the function `dist()` used in Chapter 3.

```
R> vegdist(x, method="bray", binary=FALSE, diag=FALSE, upper=FALSE,
          na.rm = FALSE, ...)
```

The next instruction loads the `vegan` package and performs the AOD. The Euclidean distance is used since the data consists of numerical variables. For any other type of variable (eg. count or binary) the Bray-Curtis or Jaccard dissimilarity measures can be used.

```
R> library(vegan)

R> adonis(X~Y,permutations=999,method="euclidean")

Call:
adonis(formula = X ~ Y, permutations = 999, method = "euclidean")

          Df SumsOfSqs  MeanSqs  F.Model    R2 Pr(>F)
Y           2.00000  592.07320  296.03660  487.33088 0.8689 0.001 ***
Residuals 147.00000   89.29740   0.60747          0.1311
Total     149.00000  681.37060          1.0000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the AOD results we can conclude that the null hypothesis is rejected, since the p -value is <0.001 . Thus, there are significant differences among the groups, which we also found with the MANOVA in the previous section.

5.6 Summary

This chapter illustrated the analysis of distance as an alternative to the conventional ANOVA and MANOVA. Even though ANOVA and MANOVA are very popular methods for comparing multiple groups, they only work well for numerical data from normal distributions. The AOD on the other hand, does not make assumptions about the underlying distribution of the data and is completely non-parametric. AOD can be performed even if the number of variables exceeds the number of observations. AOD is not only applicable to numerical data, but can be used with count and binary data by choosing the appropriate measure of dissimilarity. Finally, it should be mentioned that AOD can also be used for other designed experiments, *eg.* factorial designs, randomized block designs and Latin square designs.

The function `adonis()` was the only R function discussed in this chapter to perform AOD. Another function used to perform AOD is the function `anosim()` which is also part of the `vegan` package.

Chapter 6

Real-world applications

6.1 Introduction

In this chapter we will apply the techniques, discussed Chapters 2 to 5, on the Biolog and the Barents Fish data described in Chapter 1. The Biolog data were obtained from an experiment involving 32 carbon sources and 12 treatments. The measurements for this experiment are binary (the presence or absence of microbial activity in soil; see Figure 1.3). The soil samples were taken for three months (February, September and December) and at two depths (0-75 mm and 150-300 mm). The Biolog data will be subjected to an exploratory analysis using the following methods: cluster analysis, multidimensional scaling and correspondence analysis. This data set will also be subjected to the analysis of distance method to test for differences among the treatments. The Jaccard and Bray-Curtis dissimilarities will be used for the clustering, multidimensional scaling and the analysis of distance. The Barents Fish data were obtained from an observational study (see Figure 1.5). This data set contains two numerical measurements of interest (temperature and depth). Furthermore, it also contains a set of count data for 32 fish species. Both the numerical variables and count data were measured at 89 sites in the Barents Sea. This data set will be subjected to a canonical correspondence analysis to study the relationship between the numerical variables and the count data.

6.2 Exploratory analysis of the Biolog data

6.2.1 Cluster analysis

In this section we perform a cluster analysis using the complete linkage method described in Chapter 3. Firstly, the carbons are clustered using the Jaccard and the Bray-Curtis dissimilarities respectively. The clustering was done for the three months

and the two depths separately. The results of this cluster analysis are displayed as dendrograms in Figures 6.1 and 6.2. Secondly, the treatments were clustered using the same configuration described above. The results of this cluster analysis are displayed in Figures 6.3 and 6.4.

6.2.2 Nonmetric multidimensional scaling

We also performed a nonmetric multidimensional scaling, as discussed in Chapter 4. Again we will make use of both the Jaccard and the Bray-Curtis dissimilarities in our analysis. The multidimensional scaling was performed for the three months and the two depths separately. The results are displayed in Figures 6.5 and 6.6 for the carbons as well as Figures 6.7 and 6.8 for the treatments. Tables 6.1 and 6.2 contain the final stress values (see Chapter 4, Section 4.3) for the multidimensional scaling on the carbons and treatments respectively. These values will be used as a measure of the goodness-of-fit for the two dimensional plots. Note that in cases where the distance (dissimilarity) matrix contains zero values, nonmetric multidimensional scaling can not be performed. In such a case a small value (*e.g.* 0.0001) was added to each of the distances.

6.2.3 Correspondence analysis

Finally, the Biolog data was subjected to a simple correspondence analysis (see Chapter 2). The purpose of this analysis was to study the relationships/ associations between the carbons and the treatments. The results of the correspondence analysis are given in Figure 6.9 for the months and the depths separately. We also report the inertias for the two dimensional configurations in Table 6.3 as measures of the goodness-of-fit. Note that for cases where the row or column totals of the contingency table are zero, correspondence analysis can not be performed. For such a case the particular row or column can be removed from the contingency table and the analysis can be performed on the remaining data.

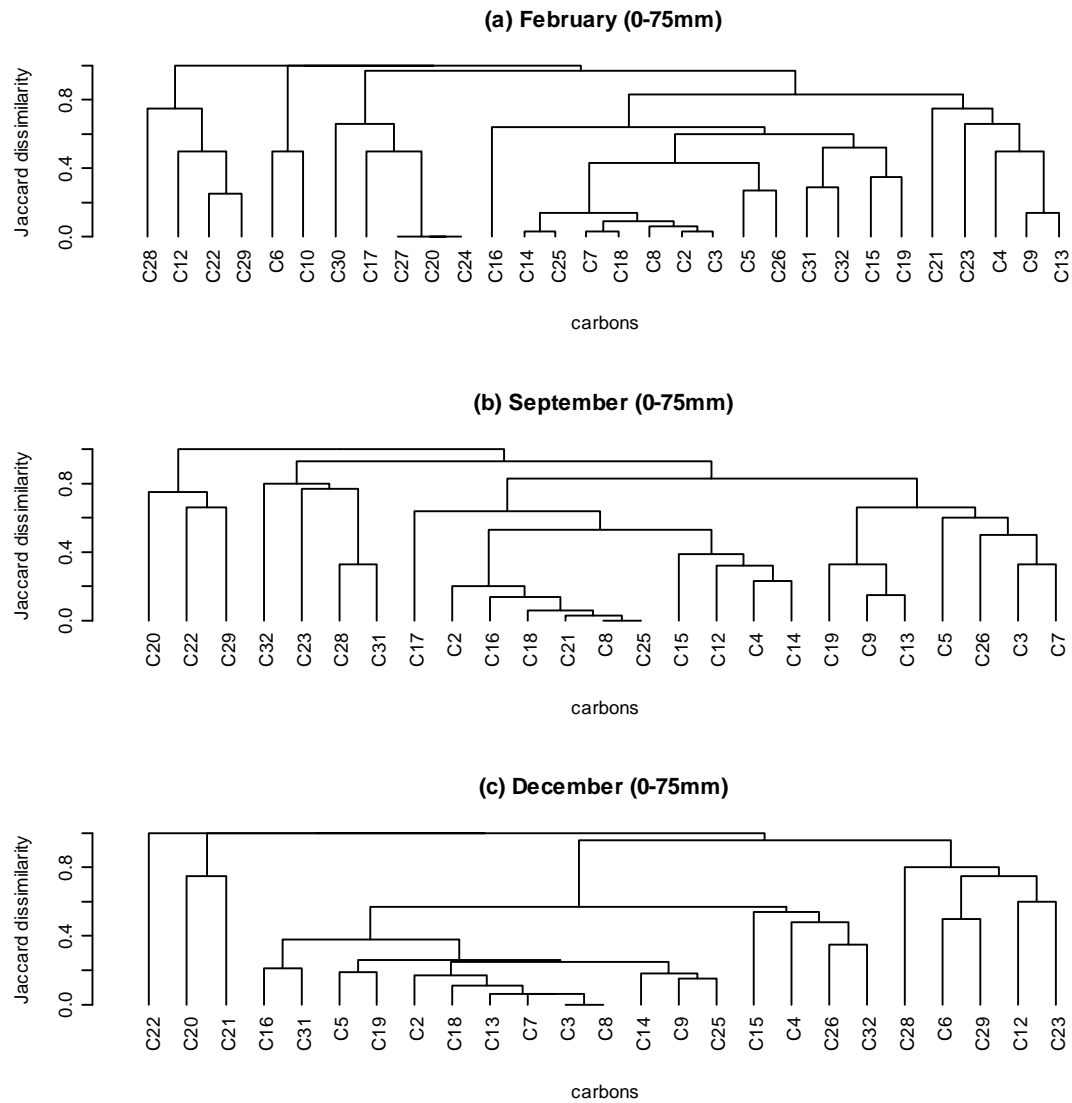


Figure 6.1: The complete linkage dendrograms of the 32 carbons per month and depth using the Jaccard dissimilarity.

Chapter 6: Real –world applications

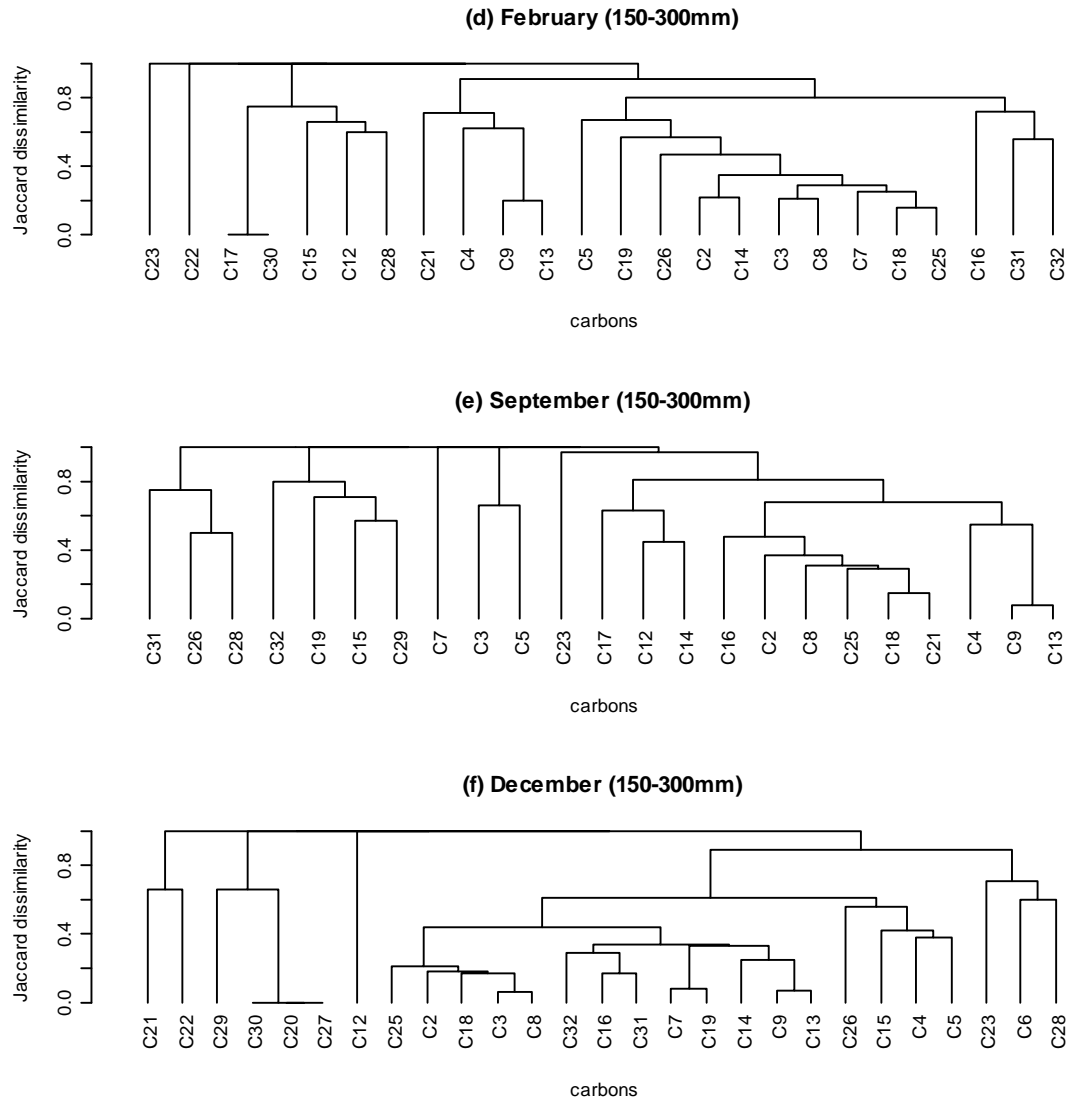


Figure 6.1: Continued.

Chapter 6: Real –world applications

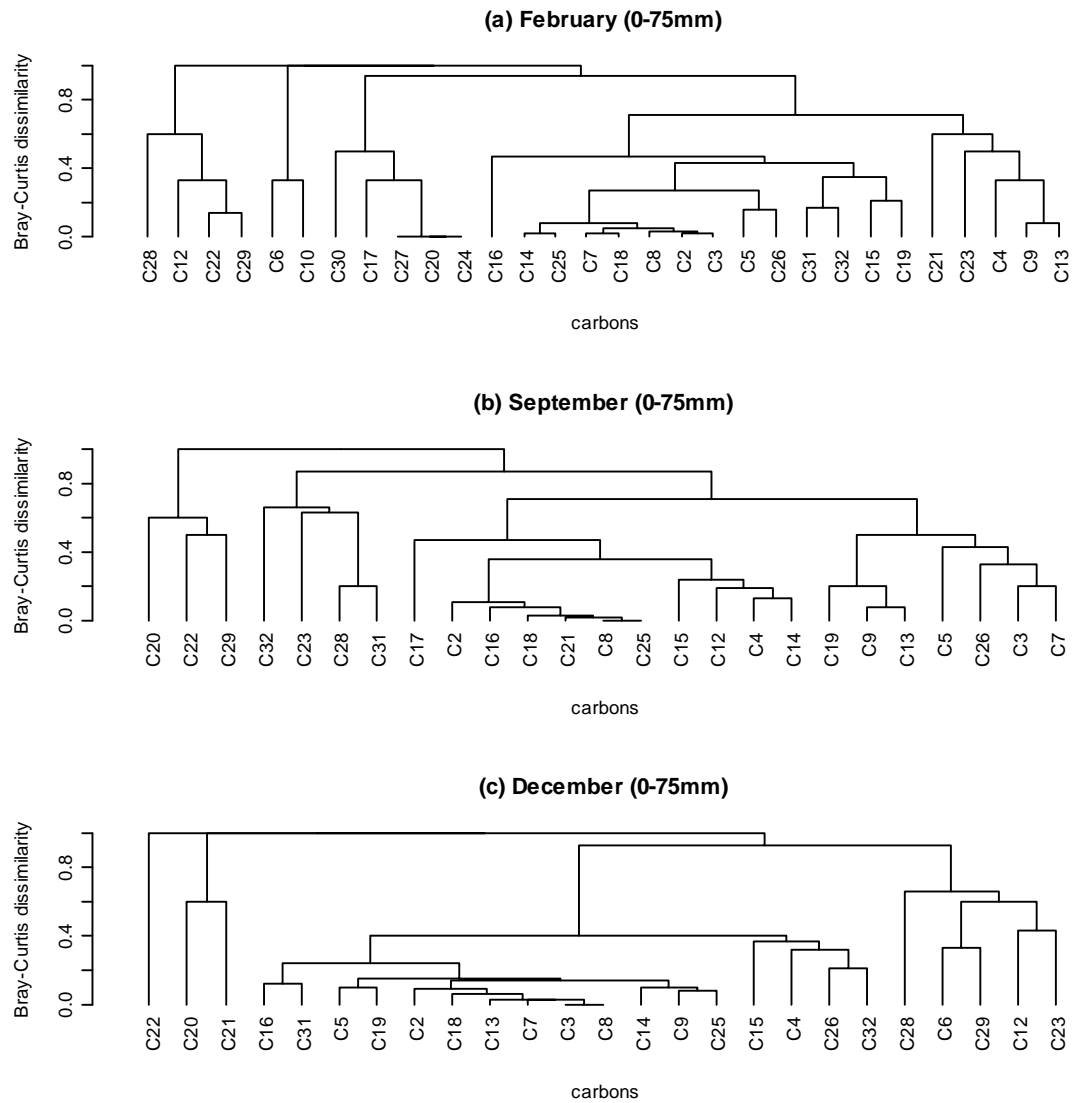


Figure 6.2: The complete linkage dendrograms of the 32 carbons per month and depth using the Bray-Curtis dissimilarity.

Chapter 6: Real –world applications

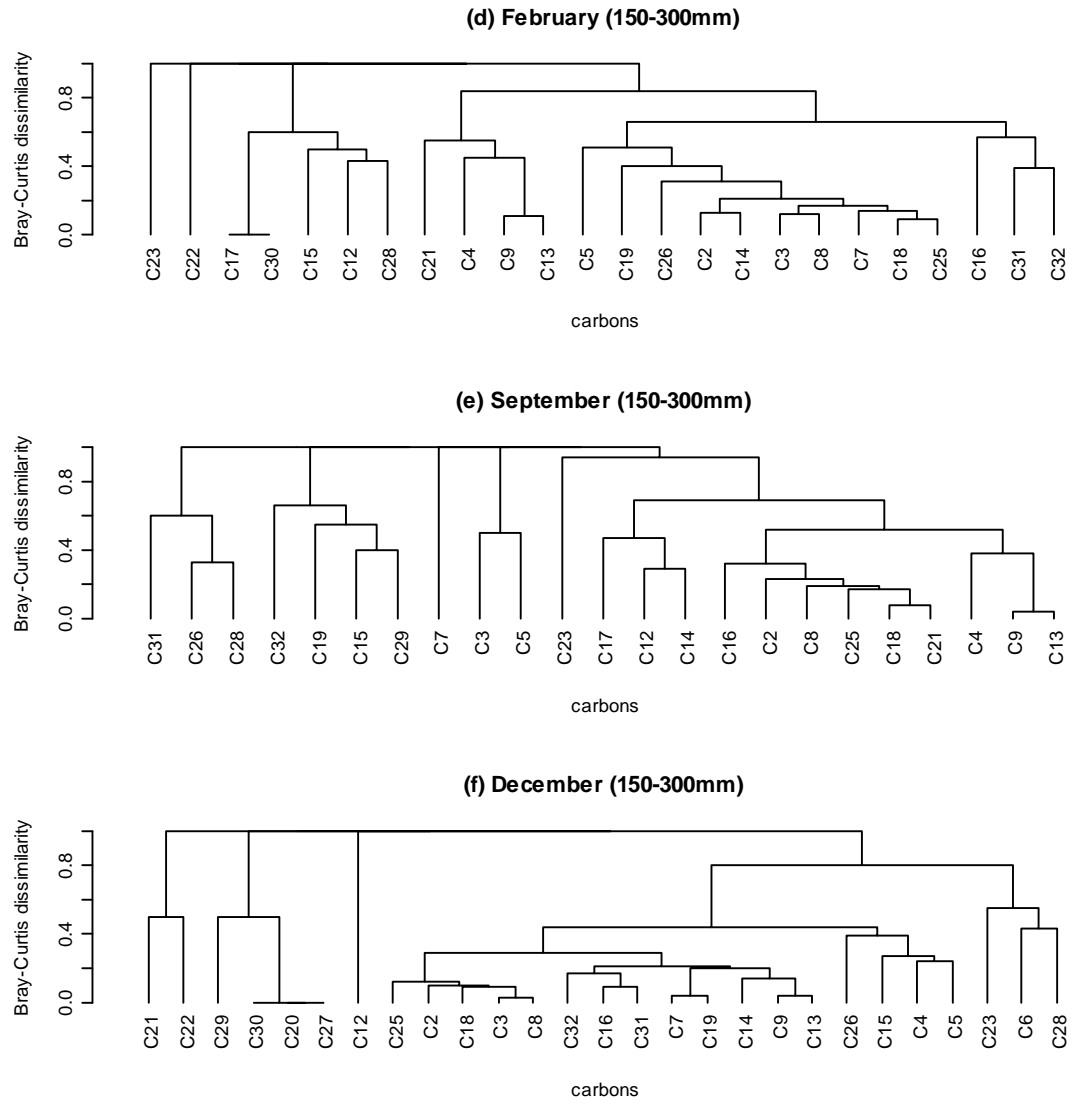


Figure 6.2: Continued.

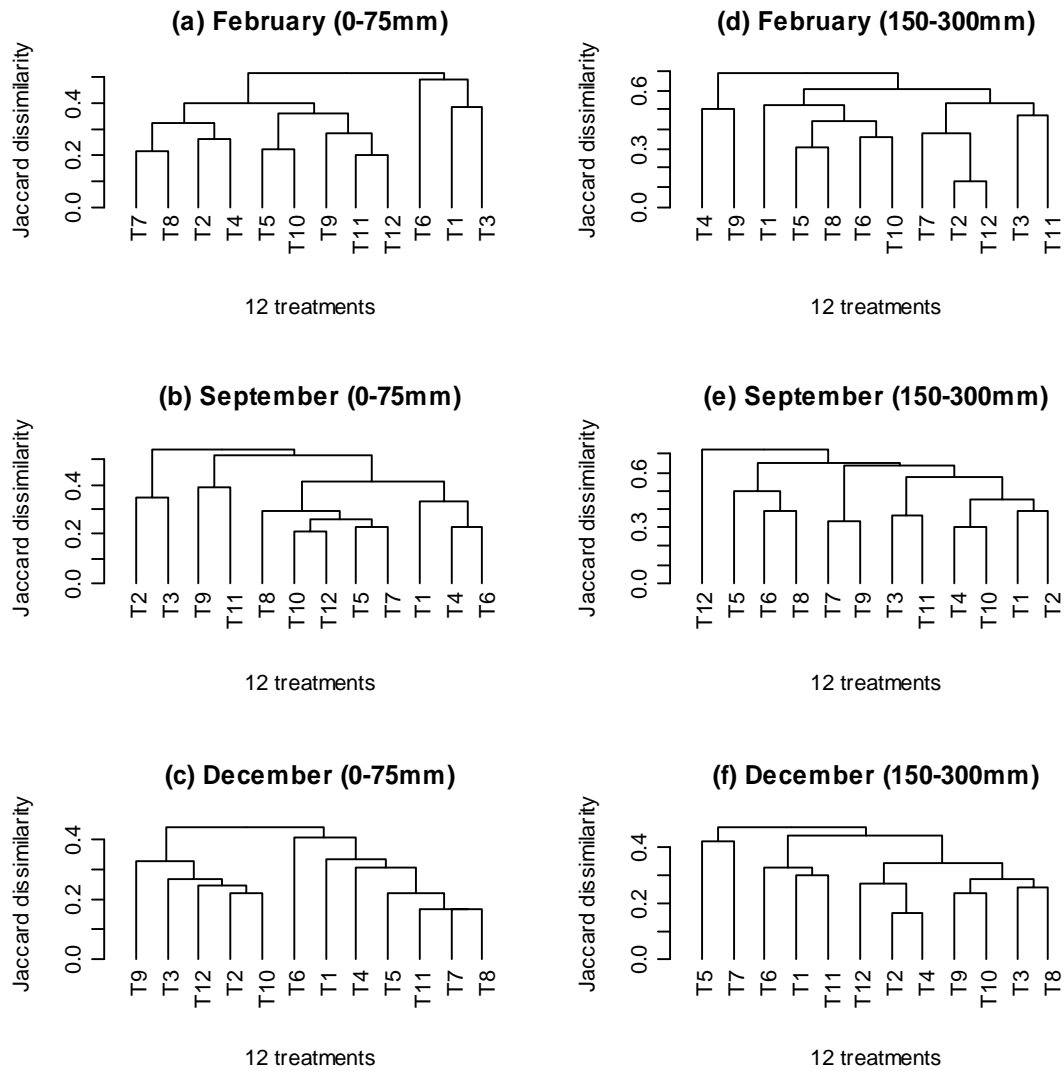


Figure 6.3: The complete linkage dendrograms of the 12 treatments per month and depth using the Jaccard dissimilarity.

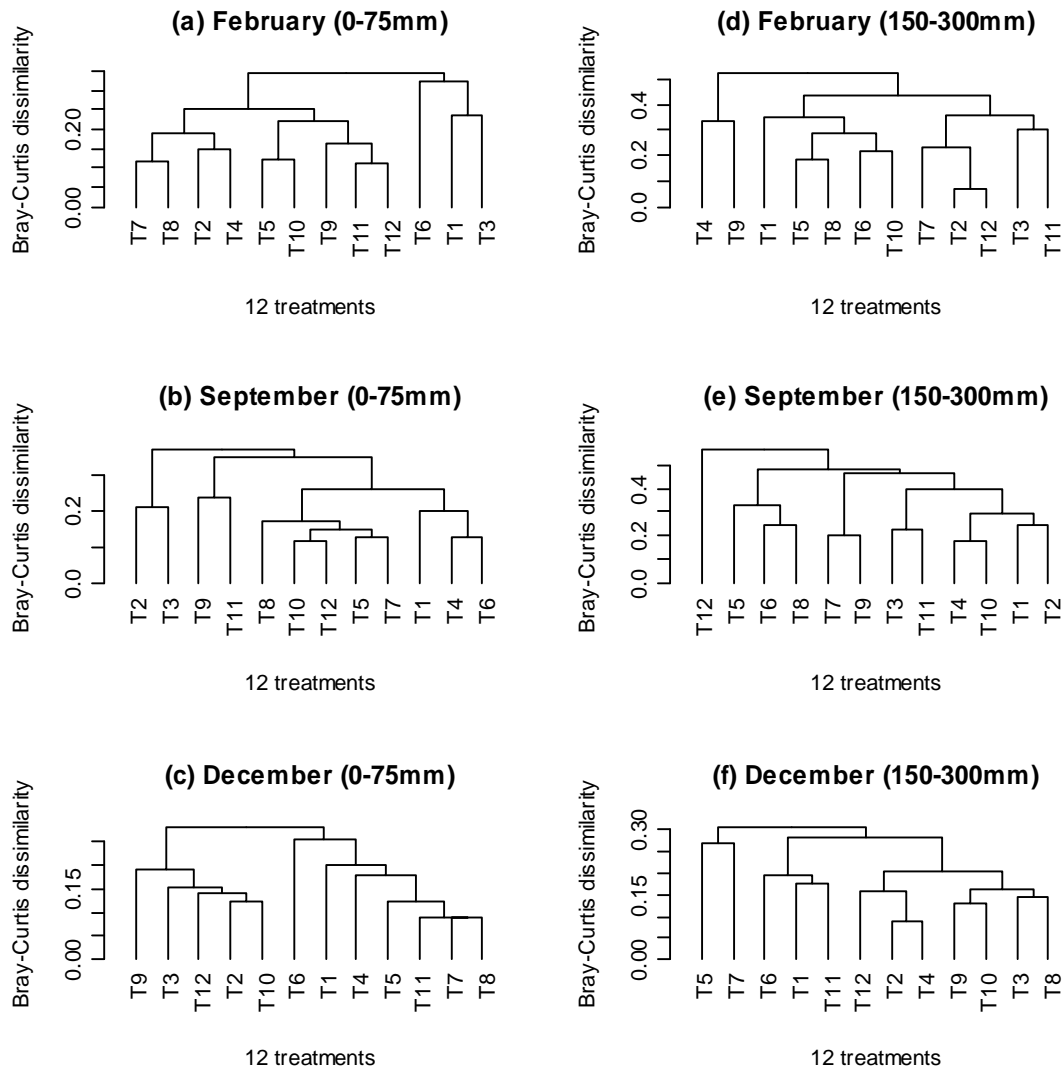


Figure 6.4: The complete linkage dendrograms of the 12 treatments per month and depth using the Bray-Curtis dissimilarity.

Chapter 6: Real –world applications

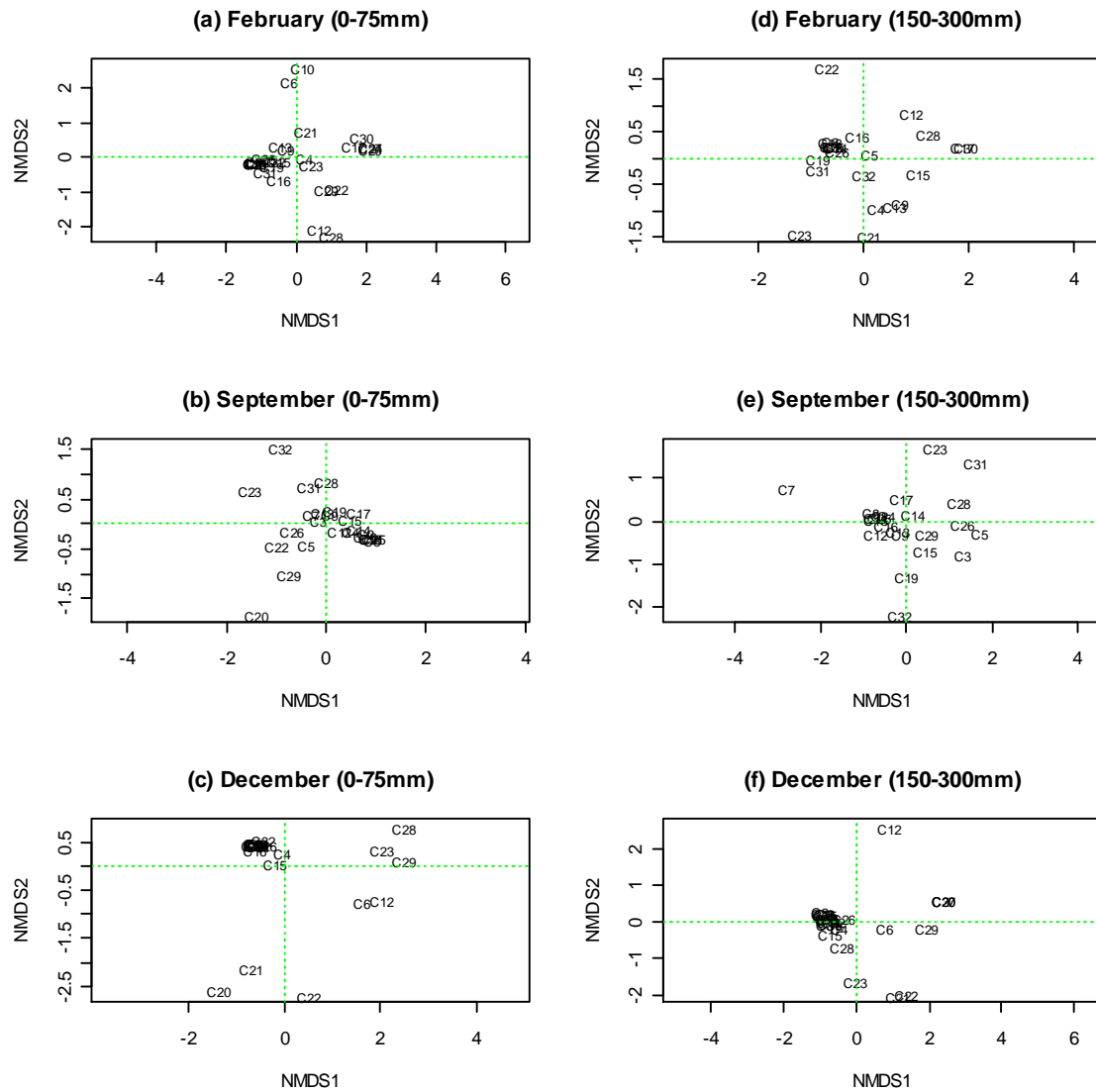


Figure 6.5: The nonmetric multidimensional scaling of the 32 carbons per month and depth using the Jaccard dissimilarity.

Chapter 6: Real –world applications

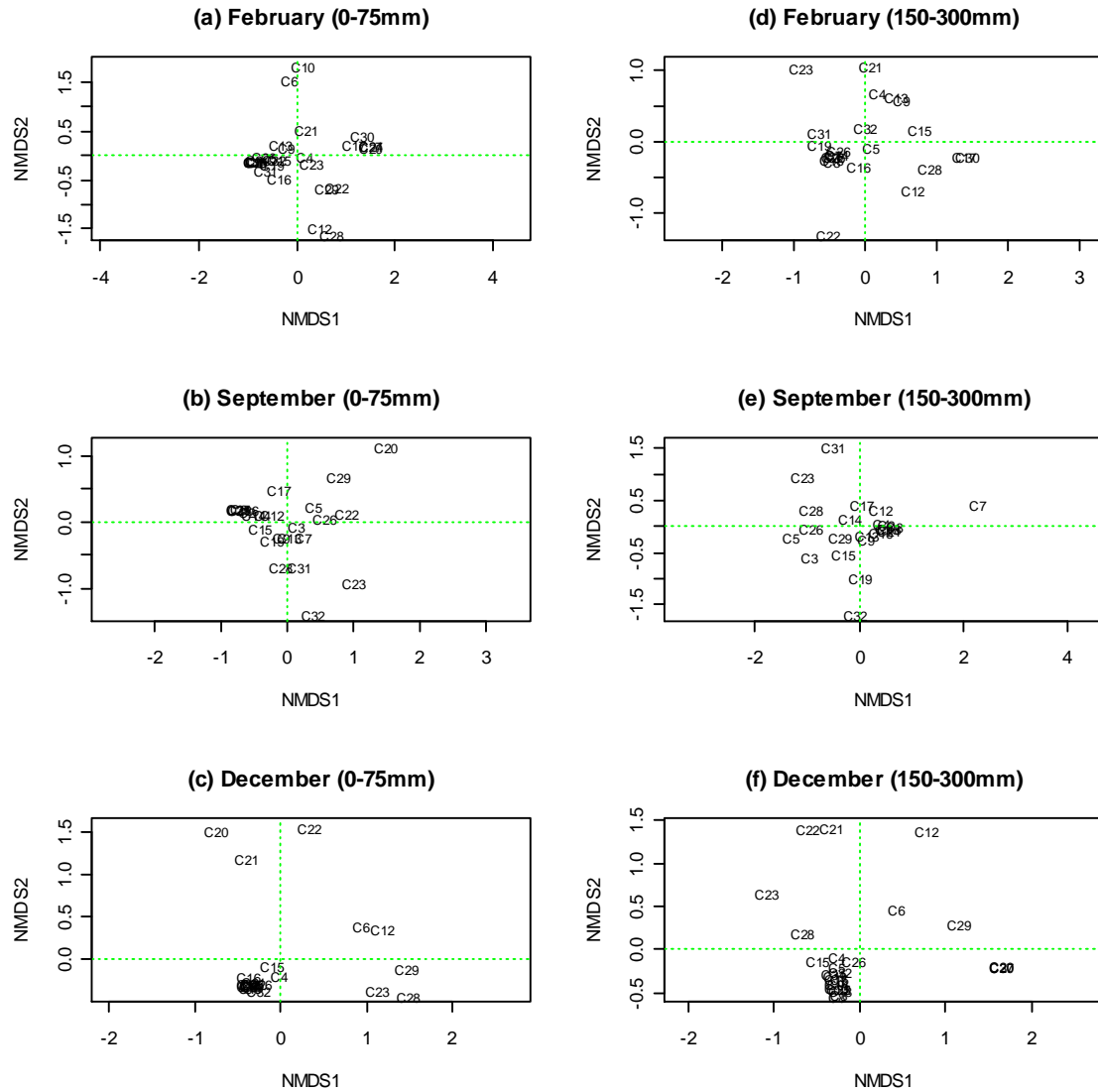


Figure 6.6: The nonmetric multidimensional scaling of the 32 carbons per month and depth using the Bray-Curtis dissimilarity.

Chapter 6: Real –world applications

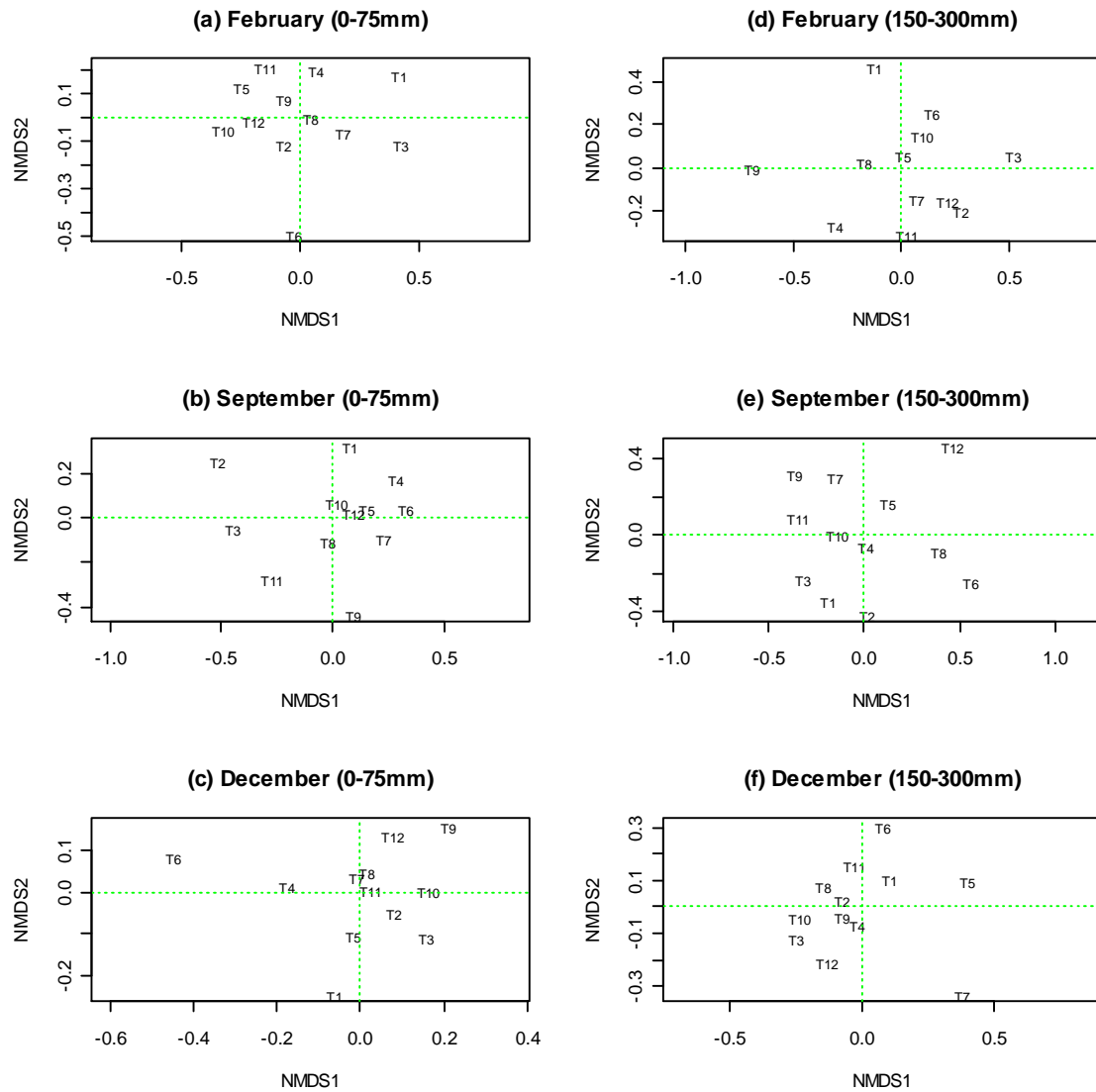


Figure 6.7: The nonmetric multidimensional scaling of the 12 treatments per month and depth using the Jaccard dissimilarity.

Chapter 6: Real –world applications

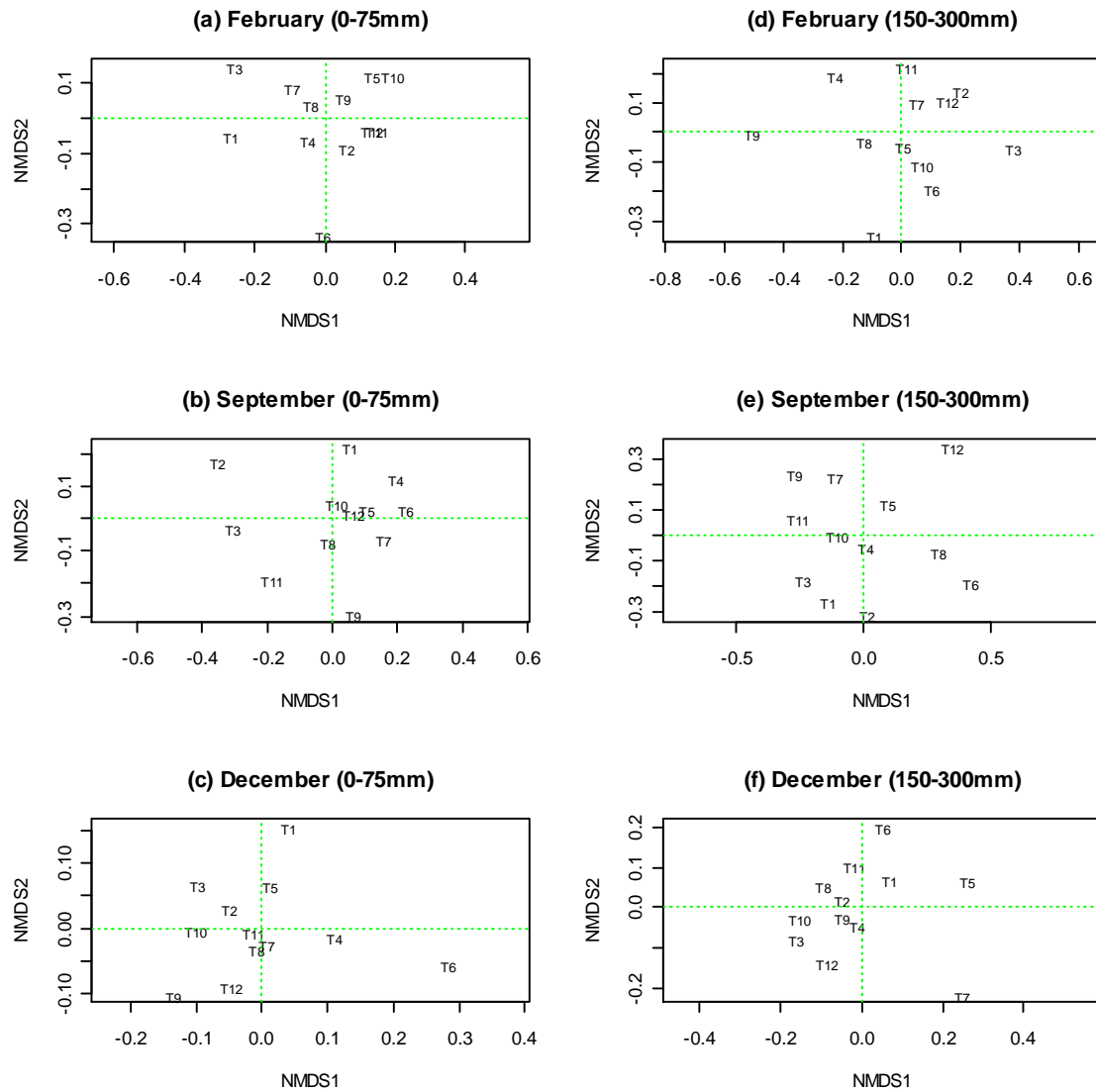


Figure 6.8: The non-metric multidimensional scaling of the 12 treatments per month and depth using the Bray-Curtis dissimilarity.

Table 6.1: The final stress values for the nonmetric multidimensional scaling on the carbons for both Jaccard and Bray-Curtis dissimilarity.

	Month (depth)	Stress value (Jaccard)	Stress value (Bray-Curtis)
(a)	February (0-75 mm)	11.805	11.805
(b)	September (0-75 mm)	9.985	9.985
(c)	December (0-75 mm)	8.621	8.621
(d)	February (150-300 mm)	15.332	15.332
(e)	September (150-300 mm)	13.699	13.411
(f)	December (150-300 mm)	10.928	10.928

Table 6.2: The final stress values of the nonmetric multidimensional scaling on the treatments for both Jaccard and Bray-Curtis dissimilarity.

	Month (depth)	Stress value (Jaccard)	Stress value (Bray-Curtis)
(a)	February (0-75 mm)	14.029	14.029
(b)	September (0-75 mm)	10.652	10.652
(c)	December (0-75 mm)	12.197	12.197
(d)	February (150-300 mm)	13.245	13.245
(e)	September (150-300 mm)	15.561	15.561
(f)	December (150-300 mm)	11.772	11.772

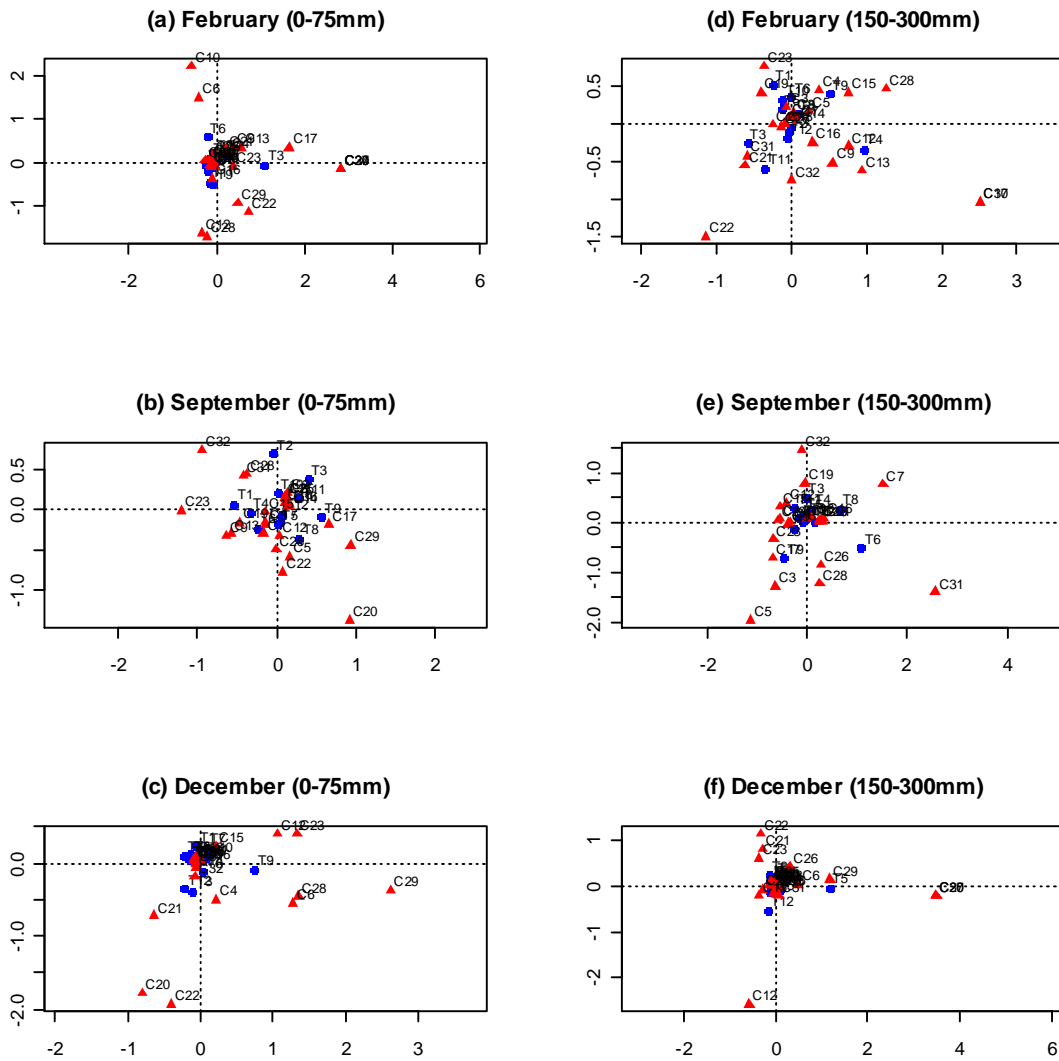


Figure 6.9: The correspondence analysis plots per month and depth.

Table 6.3: The inertia of the first two dimensions for the correspondence analysis.

	Month (depth)	Inertia (%)
(a)	February (0-75 mm)	51.4
(b)	September (0-75 mm)	45.6
(c)	December (0-75 mm)	45.7
(d)	February (150-300 mm)	42.2
(e)	September (150-300 mm)	40.8
(f)	December (150-300 mm)	48.5

6.3 Discussion of the Biolog data results

For the discussion of the results given in Section 6.2, we will focus on the following five points.

6.3.1 Comparing the results of the Jaccard and the Bray-Curtis dissimilarity

These two measures were used for both the cluster analysis and the multidimensional scaling. Comparing the dendrograms for the cluster analysis in Figures 6.1 and 6.2 (the 32 carbons), we observe that the Jaccard and the Bray-Curtis give almost identical answers. Similar conclusions are made when comparing Figures 6.3 and 6.4 (the 12 treatments). The two dimensional configurations for the multidimensional scaling in Figure 6.5 and 6.6 represent the Jaccard and Bray-Curtis dissimilarity measures respectively for the 32 carbons. The configuration of the points here are very different for the two measures. Similar conclusions can be drawn for the 32 carbons. The same can be said about the configurations of points in Figures 6.7 and 6.8 for the 12 treatments. Overall it seems as if the Jaccard and Bray-Curtis dissimilarity measures allow us to make quite similar conclusions whether we are working with the cluster analysis or the multidimensional scaling results.

6.3.2 Comparing the results of the three exploratory analysis methods

Since the cluster analysis, multidimensional scaling and correspondence analysis are performed on the same data, we expect some agreement among the three exploratory methods concerning the carbons and treatments. For example, consider the dendrogram in Figure 6.1 (c) where cases C22, C20 and C21 are lying in a cluster, which are separated further from the rest. The same cases lie close to each other in Figure 6.5 (c) and Figure 6.9 (c). Consider Figure 6.1 (c) where cases C16, C31, C5, C19, C2, C18, C13, C7, C3, C8, C14, C9, C25 C15, C4, C26 and C32 form one large cluster. The same cases are clustered together in Figure 6.5 (c) and Figure 6.9 (c). Thus overall the three methods give us the same picture of relationships among the carbons. These methods also give similar grouping structures for the treatments.

6.3.3 The goodness-of-fit for the multidimensional scaling and the correspondence analysis

The final stress values for the multidimensional scaling are given in Table 6.1 and 6.2 for the carbons and treatments respectively. The values for the Jaccard and the Bray-Curtis dissimilarity measures are given for each month and depth separately. To determine the goodness-of-fit for the two dimensional configurations, we can compare these values to the guidelines in Table 4.1. The lowest and highest stress values in Tables 6.1 and 6.2 are 8.621 and 15.332 respectively. According to Table 4.1 this means that the multidimensional scaling represent a fair to poor fit in reproducing the original distances.

The goodness-of-fit for the correspondence analysis are determined by using the inertias given in Table 6.3. As seen in Table 6.3, the proportion of inertia explained the first two dimensions for the correspondence analysis on the Biolog data ranges between 40% and 51%. These values show the percentage of variation in the raw data explained by the first two dimensions. The inertia is quite low, indicating that two dimensions may not be sufficient to study the relationship between the treatments and carbons.

6.3.4 Overall conclusion about the treatments

There are no real clustering patterns among the treatments. An exception to this maybe Figure 6.3 (c) and 6.4(c) for December (0-75 mm). However, Figures 6.7 and 6.8 show a random display of the treatments with no observable structure among them. Thus it seems like the microbial activity for the three months and the two depths are not different for the 12 treatments. The micro organisms display a similar activity in all the treatments.

6.3.5 Overall conclusion about the carbons

The clustering for the carbons seems to have more structure. There are definitely some clusters that can be identified from Figures 6.1 or 6.2, especially for December (0-75 mm). In Figure 6.5 or 6.6 we also observe a small group of carbons clustering, while the rest are scattered. Thus we can conclude the microbial activity for the 32 carbons are definitely showing a difference for the months and depths.

Considering the correspondence analysis depicted in Figure 6.9, we can also conclude that there are no associations among the treatments and carbons. The cases (carbons and treatments) are clustered around the origin of the graphs. There are no real visible relational patterns among the treatments and carbons.

6.4 Analysis of distance using the Biolog data

The aim of this section is to test for differences among the 12 treatments by using the analysis of distance method discussed in Chapter 5. As mentioned earlier, this method is similar to a MANOVA, but does not make the same assumptions. In fact, there are no assumptions when performing an analysis of distance. Since this analysis makes use of a distance (dissimilarity) matrix, we will again use the Jaccard and Bray-Curtis measures. The analysis of distance tests the following hypotheses in the Biolog data:

H_0 : the locations of the treatments are the same

H_1 : the locations of the treatments are different ,

for the three months and two depths separately. The p -values resulting from the analysis of distance are displayed in Table 6.4.

Table 6.4: The p -values obtained from the analysis of distance of the 12 treatments for the Jaccard and Bray-Curtis dissimilarities.

	Month (depth)	p -value (Jaccard)	p -value (Bray-Curtis)
(a)	February (0-75 mm)	0.6533	0.7433
(b)	September(0-75 mm)	0.4885	0.5964
(c)	December (0-75 mm)	0.02098*	0.02897*
(d)	February (150-300 mm)	0.01898*	0.01998*
(e)	September (150-300 mm)	0.1369	0.3097
(f)	December (150-300 mm)	0.961	0.962

* Significant p -values at a 5% level of significance.

As can be seen from Table 6.4, there are only significant differences among the 12 treatments for December (0-75 mm) and February (150-300 mm) at a 5% level of significance. Thus, for these cases there were discrepancies among the microbial activities. There were no significant differences among the 12 treatments for the rest of the cases. This means there were no discrepancies among the microbial activities for these cases. Both the Jaccard and the Bray-Curtis dissimilarity measures produce similar conclusions.

6.5 Canonical correspondence analysis of the Barents Fish data

In this section we will analyze the Barents Fish data by using canonical correspondence analysis which was discussed in Chapter 2. There are two environmental variables (depth and temperature) and the count data of 32 different fish species from 89 different stations. The canonical correspondence analysis is designed to analyze such data in order to study the relationships or patterns among the environmental variables and the fish species. The CCA plot of the canonical correspondence analysis is given in Figure 6.10. The black three-digit numbers on the graph are the station numbers. The red abbreviations refer to the 32 fish species (see Table 1.1). The blue arrows show the direction in which those variables increase.

It is clear from Figure 6.10 that there are seven sites that stand out among the 89 sites (sites number 356, 462, 386, 399, 459, 458, 465). Most of the other sites are scattered around the origin of the graph. Focusing our attention on the seven sites, we can see that three of these sites (356, 462 and 386) are associated with high temperatures. The other four sites (386, 399, 458, 459 and 465) are associated with lower depths. The contour lines in Figure 6.11 were created to identify the levels of the temperature at each site. The contour lines in Figure 6.12 on the other hand allow us to identify the levels of the depth.

If we turn our attention to the fish species, it can be seen that the species labeled as An_lu (the Atlantic catfish – see Table 1.1) lies further away from the other species. This fish species seems to be associated especially with a lower depth level. Thus, the species occurs in the shallow part of the Barents Sea. On the other hand, some species

Chapter 6: Real –world applications

for example No_rk (the white barracudina), seem to be associated with a higher depth level, implying that it occurs more frequently in the deeper part of sea. In a similar way we can interpret the relationship among the other species and the depth variable. Studying the relationship of the species with the temperature level, one can see for example that species like Mi_po (Blue whiting), Tr_es (Norway pout) and Cl_ha (Herring) occur more frequently where the temperature levels are higher. Again the contour lines in Figure 6.11 and 6.12 are useful in determining the level of temperature and depth at which the fish species occur the most.

If we study the association among the sites and species, we can see in Figure 6.10 that species An_lu seems to occur most at sites 386, 399, 458, 459 and 465. Overall it seems that most of the sites and species are clustered around the origin. This means that most sites and species, respectively, have on average the same profile. This also means that all the different fish species seems to occur at most of the sites in the Barents Sea.

Judging the equivalent lengths of the two arrows, it seems as if the variables temperature and depth carries the same weight in the analysis. The following results show that both variables are highly significant in the analysis.

```
> envfit(cca(X,Y),Y,perm=999)
***VECTORS
          CCA1      CCA2      r2 Pr(>r)
Depth      0.37128  0.92852  0.4287  0.001 ***
Temperature -0.83355  0.55244  0.3675  0.001 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
P values based on 999 permutations.
```

Both p -values are 0.001 and therefore highly significance.

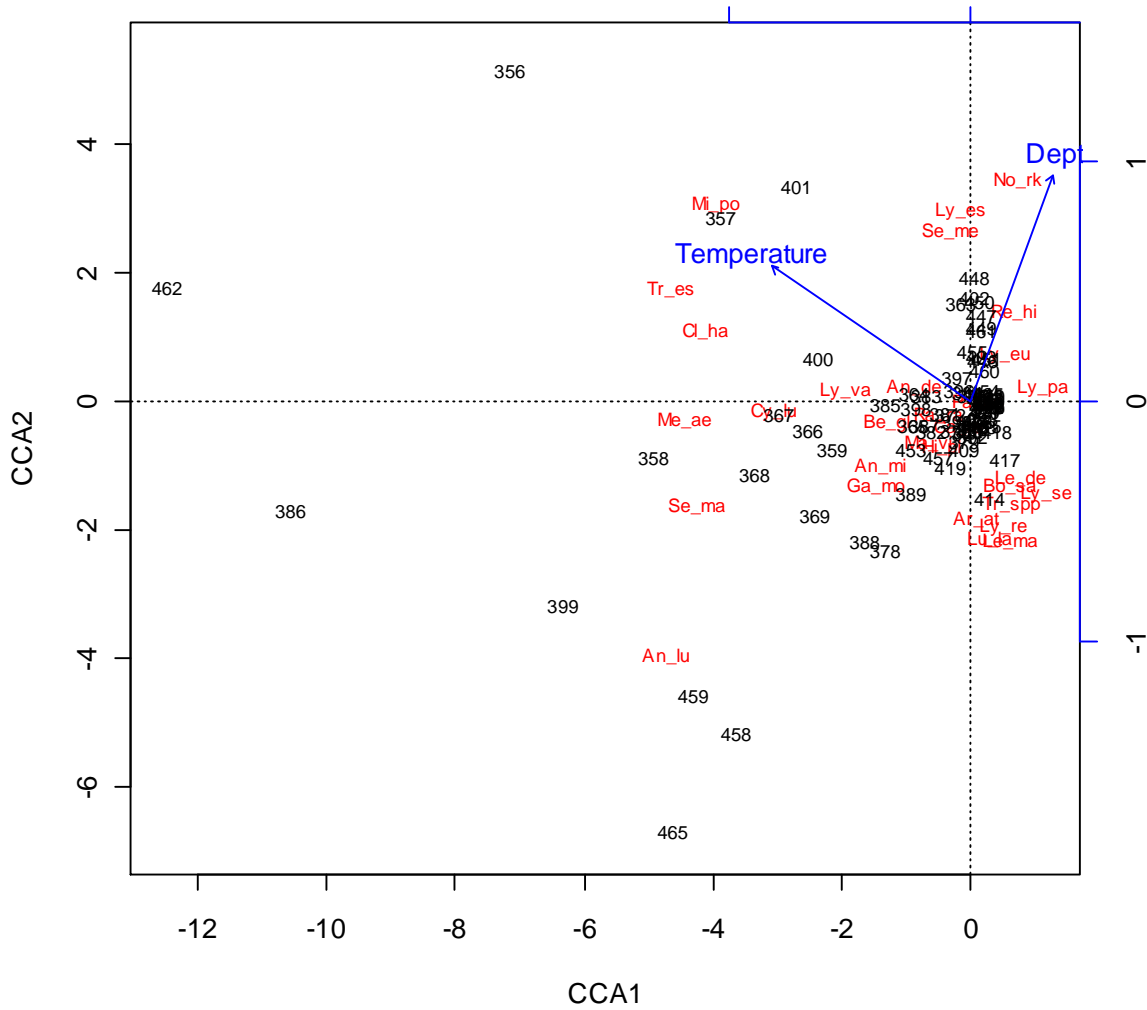


Figure 6.10: CCA plot from the canonical correspondence analysis of the Barents Fish data.

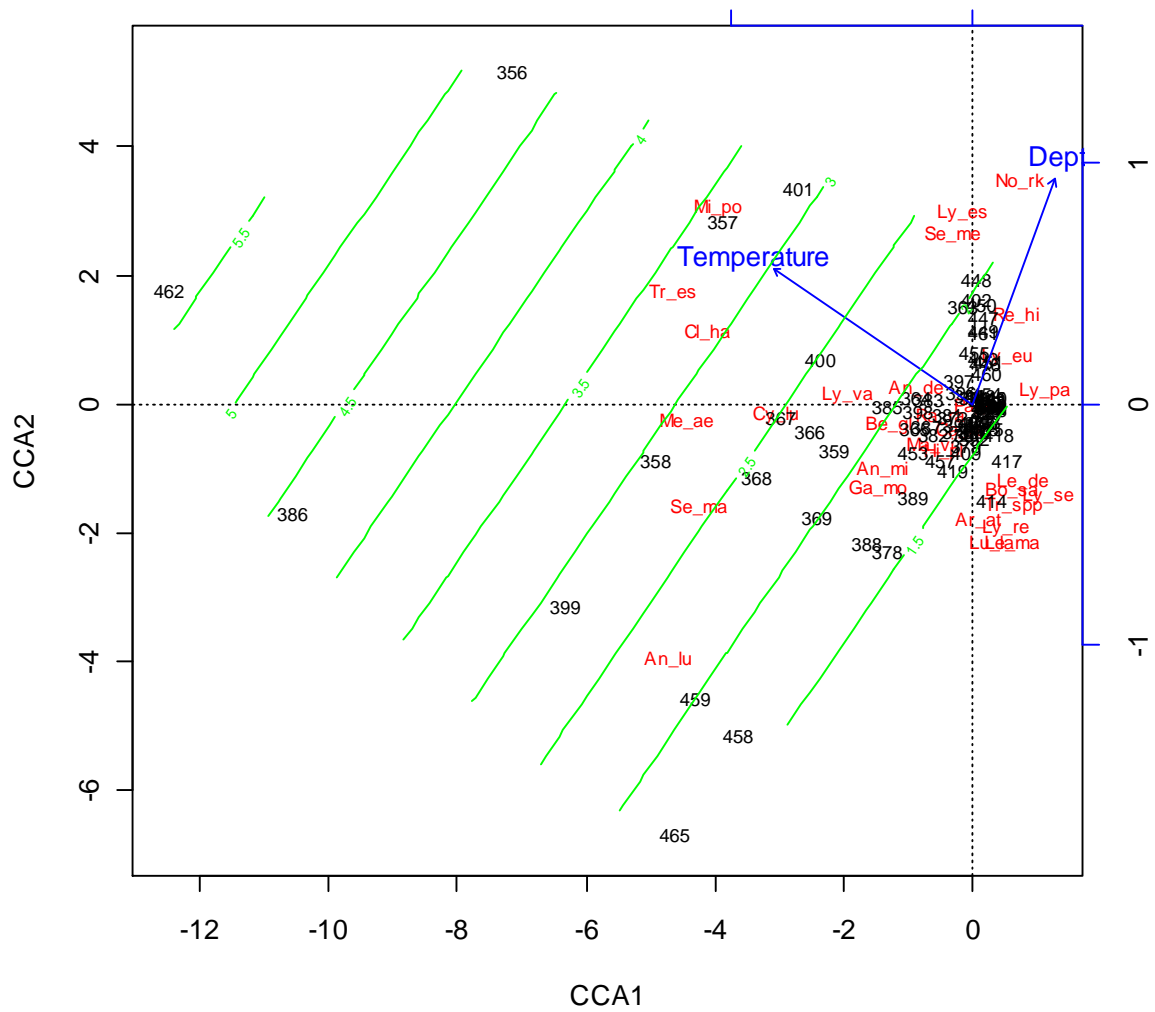


Figure 6.12: Contour plot of the temperature levels. The contours in the plot increase in the direction of the temperature variable.

6.6 Summary

The aim of the chapter can be summarized in the following points:

- Firstly, the Biolog data was analyzed as an investigation into the data for interesting patterns (microbial activity) among the carbons and treatments based on the binary measurements. This was done by using the exploratory methods: clustering analysis, correspondence analysis and nonmetric multidimensional scaling.
- Secondly we tested for significant differences among the treatments in the Biolog data.
- Thirdly, we compared the results for the Jaccard and Bray-Curtis dissimilarity measures. These measures are popular measures for binary and count data.
- Fourthly, the Barents Fish data was analyzed to demonstrate the usefulness of canonical correspondence analysis when we want to study the relationship among two sets of data.

Overall it did not seem as if there were any clear patterns among the carbons and treatments in the Biolog data. However, there were a few cases where there were significant differences among the treatments (see Table 6.4). Both the Jaccard and Bray-Curtis dissimilarities gave similar results which lead to the same conclusions. The canonical correspondence analysis shows that most of the sites and species are clustered around the origin. This shows that all the different fish species seem to occur in most of the sites in the Barents Sea.

Chapter 7

General conclusion

The aim of this thesis was to explore and understand ways of analyzing multidimensional count or binary data. This task was accomplished by using two approaches, namely exploratory analysis and inferential analysis of the data. The methods used for exploratory data analyses were correspondence analysis, canonical correspondence analysis, cluster analysis and multidimensional scaling. These methods have been successfully applied to the Biolog data and the Barents Fish data. An analysis of distance method was used to perform an inferential analysis on the multidimensional Biolog data. This method by Anderson (2001a) is a quite powerful method and unlike the analysis of variance, this method makes no distributional assumptions.

Correspondence analysis is an exploratory technique that studies the relationship between the rows and columns of a contingency table. The goodness-of-fit in correspondence analysis is determined by the proportion of inertia explained by the first two dimensions. The screeplot and the Benzécri plot can be used to identify the appropriate number of dimensions to obtain a good fit. Canonical correspondence analysis is a correspondence analysis in a restricted space. CCA is a very useful technique in investigating the relationship between the count data and the explanatory variables. The goodness-of-fit is also determined by the proportion of inertia explained in the constrained space. Correspondence analysis is quite an active area of research. Partial constrained correspondence analysis, joint and multiple correspondence analyses are more techniques which could be used for analyzing data in contingency tables (see Greenacre, 2007).

Cluster analysis was used as an exploratory technique for identifying groups in the data. Cluster analysis uses a distance or dissimilarity matrix obtained from the data. In this way cluster analysis can be applied to any type of data by choosing the appropriate distance or dissimilarity measure. Several distance and dissimilarity

Chapter 7: Conclusion

measures were given in this thesis. The Jaccard and Bray-Curtis were specifically used for binary or count data. Four agglomerative hierarchical clustering methods were discussed. In Chapter 6 only complete linkage clustering was used in the analysis of the Biolog data, since it tends to produce clearer dendrograms when compared to the other agglomerative hierarchical clustering methods. However, the literature on cluster analysis is very large. Other non-hierarchical methods for cluster analysis are also available, such as the *K*-means cluster method. Model-based clustering methods are also very powerful in finding clusters in the data by using statistical distributions (see Johnson and Wichern, 2007).

A very attractive non-parametric technique is the analysis of distance discussed in Chapter 5. The conventional parametric analysis of variance approach is based on some assumptions which are: (1) the data in each group are from a normal population, (2) the observations in each group are independent and (3) the population variances in the groups are equal. Therefore, the data are assumed to be numerical data as well. However, the analysis of distance is not based on any such assumptions. This method can be applied to any type of data by employing the appropriate distance or dissimilarity measure.

The statistical software R is available on the internet (<http://www.r-project.org/>) and can be downloaded free of charge. All the methods discussed in this thesis have been programmed in R and are readily available for usage. Any person with a basic knowledge of R will be able to apply the functions for the corresponding methods. Some functions are standard in R, while other functions can be obtained from the packages available on the R website. These packages can also be freely downloaded. The following is a summary of the methods and the functions used (with package name in brackets {}):

- Correspondence analysis: `ca()` {ca}; `anacor()` {anacor}; `cca()` {vegan}
- Canonical correspondence analysis: `anacor()` {anacor}; `cca()` {vegan}
- Cluster analysis: `dist()`, `hclust()` {stats}
- Multidimensional scaling: `cmdscale()` {stats}; `isoMDS()` {MASS}; `metaMDS()` {vegan}

- Analysis of distance: `adonis()` {vegan}.

There are many other packages and functions to perform the above mentioned methods in R. This software is a powerful tool for many statistical applications. It contains the most recently developed techniques in statistics. The graphics are quite impressive and the option to write your own programs gives the user much freedom to explore his/ her own ideas.

Open research questions:

- The use of permutation tests in statistics has become quite popular with the development of computing software. In this thesis we have used permutation tests in canonical correspondence analysis to identify significant environmental variables (Chapter 2). In Chapter 5 we used permutation tests in the analysis of distance to obtain p -values for a hypothesis test. Using permutation tests in multidimensional scaling and cluster analysis should also be investigated. Permutation tests may for example be useful as a mechanism to determine the goodness-of-fit for clustering analysis or how many clusters are sufficient.
- If the permutation test is applicable in these methods, bootstrap techniques may also be explored in future research concerning these methods.
- Since there are various options to perform MDS (for example metric and non-metric with different distance or dissimilarity measures), techniques like Procrustes analysis can be employed in a study to compare the performance of the different MDS options.
- With count data we often have the case where the counts can be very high and very low in some cells. This causes large variation in the data. Clarke and Warwick (1994) argued that the 4th root transformations should be applied to count data to reduce the influence of very abundant species. How to transform the count data should also receive further attention.

Bibliography

1. Anderson, E. (1935) The irises of the Gaspé peninsula. *Bulletin of the American Iris Society*, **59**, 2–5.
2. Anderson, M. J. (2001a) A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, **26**, 32-46.
3. Anderson, M. J. (2001b) Permutation tests for univariate or multivariate analysis of variance and regression. *Canadian Journal of Fisheries and Aquatic Sciences*, **58**, 626-639.
4. Benzecri, J.P. (1973) L'Analyse des Donne'es. Volume II. L'Analyse des Correspondences. Paris, France: Dunod.
5. Borg, I and Groenen, P.J.F. (2005) *Multidimensional scaling: Theory and Applications*, 2nd edition. New York: Springer Press.
6. Bray, J. R. and Curtis, J.T. (1957) An ordination of upland forest communities of Southern Wisconsin. *Ecological Monographs*, **27**, 325-349.
7. Cox, T.F. and Cox, M.A.A. (2001) *Multidimensional Scaling*, 2nd edition. Chapman and Hall.
8. Clarke K.R. and Warwick R.M. (1994) Relearning the ABC: taxonomic changes and abundance/ biomass relationships in disturbed benthic communities. *Mar Biol*, **118**, 739-744.
9. de Leeuw, J. and Mair, P. (2009) Simple and canonical correspondence analysis using the R package anacor. *Journal of Statistical Software*, **31**(5), 1-18. URL: <http://www.jstatsoft.org/v31/i05/>.
10. Fisher, R.A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**:179-188.
11. Greenacre, M. (2007) *Correspondence analysis in practice*, 2nd edition. Chapman & Hall / CRC, London.
12. Groenen, P.J.F. and van de Velden, M. (2004) Multidimensional scaling. *Econometric Institute Report EI 2004-15*. Erasmus University Rotterdam, Econometric Institute.
13. Johnson, R.A. and Wichern, D.W. (2007) *Applied Multivariate Statistical Analysis*, 6th edition. Pearson Education, Inc.

Bibliography

14. Kruskal, J.B. (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, **29**, 1-27,115-129.
15. Legendre, P. and Legendre, L. (1998) *Numerical Ecology*, 2nd English edition. Amsterdam: Elsevier B.V.
16. Mardia, K. V., Kent, J. T. and Bibby, J.M. (1979) *Multivariate analysis*. London: Academic Press.
17. Montgomery, D.C. (2005) *Design and Analysis of Experiments*, 6th edition. John Wiley & Sons, Inc.
18. Nenadic, O. and Greenacre, M. (2007) Correspondence analysis in R, with two- and three-dimensional graphics: The ca package. *Journal of Statistical Software*, **20** (3). URL: <http://www.jstatsoft.org/v20/i03/>.
19. Nora, C. (1975) *Une méthode de reconstitution et d'analyse de données incomplètes* [A method for reconstruction and for the analysis of incomplete data]. Master's thesis, Unpublished Thèse d'Etat, Université P.et M. Curie, Paris VI.
20. Oksanen, J (2011) Multivariate analysis of ecological communities in R: vegan tutorial. URL: <http://cc.oulu.fi/~jarioksa/opetus/metodi/vegantutor.pdf>
21. Oksanen, J., Kindt, R., Legendre, P., O'Hara, B., Henry, M. and Stevens, H. (2009) vegan: *Community Ecology Package*. R package version 1.15-3. URL: <http://www.r-project.org/>.
22. Quinn, G. P. and Keough, M. J. (2002) *Experimental Design and Data Analysis for Biologists*. Cambridge University Press, UK.
23. R Development Core Team (2007) *R: A language and Environment for statistical computing*. R Foundation for statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. URL: <http://www.r-project.org/>.
24. Steyvers, M. (2000) Multidimensional Scaling. *Encyclopedia of cognitive science*. Macmillan Reference Ltd. Stanford University, US.
25. Stuetzle, W. (1995) Data Visualization and Interactive Cluster Analysis. ICPSR, Ann Arbor, MI.
26. Shepard, R.N. (1980) Multidimensional scaling, tree-fitting, and clustering. *Science*, **210**, 390-398.
27. Ter Braak, C. J. F. (1986) Canonical Correspondence Analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology*, **67**, 1167-1179.
28. Wilks, S. S. (1932) Certain generalizations in the analysis of variance. *Biometrika* **24**, 471-94.