

Written examination, date: 10. December 2018
 Course name: Multivariate Statistics
 Course number: 02409
 Aids allowed: All
 Exam duration: 4 hours
 Weighting: The questions are given equal weight

Page 1 of 48 pages Enclosure: 10 pages

This exam is answered by:

 (name)

 (signature)

 (study no.)

There is a total of 30 questions for the 6 problems. The answers to the 30 questions must be written into the table below.

Problem	1	1	1	1	1	1	2	2	2	3
Question	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	3.1
Answer	5	4	1	2	2	1	2	3	5	3

Problem	3	3	3	3	3	4	4	4	4	4
Question	3.2	3.3	3.4	3.5	3.6	4.1	4.2	4.3	4.4	4.5
Answer	1	4	2	1	3	5	2	5	4	1

Problem	5	5	5	5	5	5	5	6	6	6
Question	5.1	5.2	5.3	5.4	5.5	5.6	5.7	6.1	6.2	6.3
Answer	2	2	4	1	2	4	3	3	1	2

The possible answers for each question are numbered from 1 to 6. If you enter a wrong number, you may correct it by crossing the wrong number in the table and writing the correct answer immediately below. If there is any doubt about the meaning of a correction then the question will be considered not answered.

Only the front page must be returned. The front page must be returned even if you do not answer any of the questions or if you leave the exam prematurely. Drafts and/or comments are not considered, only the numbers entered above are registered.

A correct answer gives 5 points, a wrong answer gives – 1 point. Unanswered questions or a 6 (corresponding to “don’t know”) give 0 points. The total number of points needed for a satisfactorily answered exam is determined at the final evaluation of the exam. Especially note that the grade 10 may be given even if only one answer is wrong or unanswered. Remember to write your name, signature, and study number on the front page.

Remember to write your name, signature, and study number on the front page.

Please note, that there is one and only one correct answer to each question. Furthermore, some of the possible alternative answers may not make sense. When the text refers to SAS-output, the values may be rounded to fewer decimal places than in the output itself. The enclosures do not necessarily contain all the output generated by the given SAS programs. Please check that all pages of the exam paper and the enclosures are present.

Problem 1.

You are encouraged to use statistical software to solve this problem.

In the table below (source: <http://www.statistikbanken.dk>) we present some data related to hospital treatment for the five Danish regions that are responsible for healthcare. More specifically we give corresponding values of

1. No. of ambulant (outpatient) treatments (pr. 1000 capita)
2. No. of hospital admissions (pr. 1000 capita)
3. No. of bed days (pr. 1000 capita)
4. Fraction of population aged 65 or older
5. Sex

Region	Ambulant treatments (pr. 1000 capita)	Admissions (pr. 1000 capita)	Bed days (pr. 1000 capita)	Fraction of population aged 65 or older	Sex 1=male, 0 = female
RegionH	1077	216	667	0.15080411	1
RegionS	1142	282	790	0.205731552	1
RegionSyd	1447	190	615	0.192755409	1
RegionM	1072	192	559	0.173395829	1
RegionN	1069	177	628	0.195063255	1
RegionH	1490	248	703	0.18459817	0
RegionS	1395	294	770	0.23579325	0
RegionSyd	1859	204	604	0.222110388	0
RegionM	1459	208	549	0.198799601	0
RegionN	1449	194	629	0.224970663	0

We are interested in the differences between the regions. First we consider the model

$$[Ambulant \quad Admission \quad Bed \text{ days}] = \mu + region_i + sex_j, i = 1 \dots 5, j = 1, 2$$

Question 1.1.

The usual test-statistic for no region effect has – under the null-hypothesis – the following distribution:

We identify the problem as a 2-side (2-way) MANOVA and use theorem 4.26.

We have $p=3$, $k=5$ and $m=2$

This yields:

$$U(p, k-1, (k-1)(m-1)) = U(3, 5-1, (5-1)(2-1)) = U(3, 4, 4)$$

||| Theorem 4.26

The ratio test at level α for test of H_0 against H_1 is given by the critical region

$$\{y_{11}, \dots, y_{km} \mid \frac{\det(\mathbf{q}_1)}{\det(\mathbf{q}_1 + \mathbf{q}_2)} \leq U(p, k-1, (k-1)(m-1))_\alpha\}.$$

The ratio test at level α for test of K_0 against K_1 is given by the critical region

$$\{y_{11}, \dots, y_{km} \mid \frac{\det(\mathbf{q}_1)}{\det(\mathbf{q}_1 + \mathbf{q}_3)} \leq U(p, m-1, (k-1)(m-1))_\alpha\}.$$

Question 1.2.

The usual test-statistic for no sex effect is:

In R:

```
# Create a data frame with the given data
hospitaluse <- data.frame(
  region = c("RegionH", "RegionS", "RegionSyd", "RegionM", "RegionN", "RegionH", "RegionS",
"RegionSyd", "RegionM", "RegionN"),
  amb = c(1077, 1142, 1447, 1072, 1069, 1490, 1395, 1859, 1459, 1449),
  adm = c(216, 282, 190, 192, 177, 248, 294, 204, 208, 194),
  bed = c(667, 790, 615, 559, 628, 703, 770, 604, 549, 629),
  age65 = c(0.15080411, 0.205731552, 0.192755409, 0.173395829, 0.195063255, 0.18459817, 0.23579325,
0.222110388, 0.198799601, 0.224970663),
  sex = c(1, 1, 1, 1, 1, 0, 0, 0, 0, 0)
)

# Perform MANOVA
manova <- manova(cbind(amb, adm, bed) ~ region + sex, data=hospitaluse)

# Print the MANOVA summary
manova_summary <- summary(manova)

##### wilks_lambda #####
# Extract and print values with names
wilks_lambda_test <- summary(manova, test = "Wilks")
wilks_lambda <- wilks_lambda_test$stats['sex', "Wilks"]
approx_f_Wilks_Lambda <- wilks_lambda_test$stats['sex', 'approx F']
num_df_Wilks_Lambda <- wilks_lambda_test$stats['sex', 'num Df']
den_df_Wilks_Lambda <- wilks_lambda_test$stats['sex', 'den Df']
p_value_Wilks_Lambda <- wilks_lambda_test$stats['sex', 'Pr(>F)']

cat("Wilks' Lambda:", wilks_lambda, "\n")
```

```
cat("Wilks' Lambda Approximate F-value:", approx_f_Wilks_Lambda, "\n")
cat("Wilks' Lambda Num DF:", num_df_Wilks_Lambda, "\n")
cat("Wilks' Lambda Denominator DF:", den_df_Wilks_Lambda, "\n")
cat("Wilks' Lambda p-value:", p_value_Wilks_Lambda, "\n")
```

Result in R:

```
> cat("Wilks' Lambda:", wilks_lambda, "\n")
wilks' Lambda: 0.002106193
> cat("Wilks' Lambda Approximate F-value:", approx_f_wilks_Lambda, "\n")
wilks' Lambda Approximate F-value: 315.8601
> cat("Wilks' Lambda Num DF:", num_df_wilks_Lambda, "\n")
wilks' Lambda Num DF: 3
> cat("Wilks' Lambda Denominator DF:", den_df_wilks_Lambda, "\n")
wilks' Lambda Denominator DF: 2
> cat("Wilks' Lambda p-value:", p_value_wilks_Lambda, "\n")
wilks' Lambda p-value: 0.003157626
```

Question 1.3.

We now only consider RegionH and RegionS and the following variables [Ambulant Admission Bed days]. Note that Sex is now just considered as a replicate, i.e. we have two observations for each region. The usual test statistic for mean difference between RegionH and RegionS is:

We use

||| Theorem 4.9

We use the same notation as given above. Now, let

$$T^2 = \frac{nm}{n+m} (\bar{X} - \bar{Y})^T S^{-1} (\bar{X} - \bar{Y}).$$

Then the critical region for a test of H_0 against H_1 at level α is equal to

$$C = \{x_1, \dots, x_n, y_1, \dots, y_m \mid \frac{n+m-p-1}{(n+m-2)p} t^2 > F(p, n+m-p-1)_{1-\alpha}\}$$

Here t^2 is the observed value of T^2 .

We already have the data. We now run the following script

In R:

```
library(MASS)
```

```
# Prior probabilities for the classes
# SAS by default has equal prior probabilities, so we need equal prior in R to receive the same results
different_regions <- unique(hospitaluse$region)
num_regions <- length(different_regions)
prior <- rep(1/num_regions, num_regions)
```

```
#linear discriminant analysis
```

```

# Define the classes (region)
hospitaluse$class <- as.factor(hospitaluse$region)
# Define the variables for the analysis
variables <- c("amb", "adm", "bed")
# Perform Linear Discriminant Analysis
z <- lda(class ~ ., data = hospitaluse[, c("class", variables)],prior=prior)

Class_Level_Information = data.frame("Frequency" = z$counts,"Proportion"=z$counts/z$N,"Prior"=z$prior)
print("Class Level Information:")
Class_Level_Information

hospitaluse$region = as.factor(hospitaluse$region)
library(Rfast)
pcov <- pooled.cov(as.matrix(hospitaluse[,variables]),hospitaluse$region)
Means <- as.matrix(z$means)
invCov <- solve(pcov)
# Extract unique levels from hospitaluse$region
unique_levels <- levels(hospitaluse$region)
num_col <- length(unique(hospitaluse$region))

# Create an empty matrix to store the Mahalanobis distances with the equal priors #####
maha <- matrix(c(rep(0, num_col^2)), ncol = num_col)

# Define the names for rows and columns (assuming unique_levels contains the names)
rownames(maha) <- unique_levels
colnames(maha) <- unique_levels

for (i in 1:num_col) {
  for (j in 1:num_col) {
    mu <- Means[i, ] - Means[j, ]
    maha[i, j] <- mu %*% invCov %*% mu
  }
}

# squared Mahalanobis distances between the 5 region types.
# maha : Assuming equal priors

print("Generalized Squared Distance to region (equal priors):")
maha

n <- Class_Level_Information["RegionH", "Frequency"]
m <- Class_Level_Information["RegionS", "Frequency"]
T_squared = (n*m/(n+m))*maha["RegionH", "RegionS"]

cat("The test statistic is:", T_squared, "\n")

```

Result in R:

```
[1] "Generalized Squared Distance to region (equal priors):"
> maha
      RegionH  RegionM  RegionN  Regions  RegionSyd
RegionH    0.00000    95.22985   221.896496  264.2333  274.476175
RegionM    95.22985     0.00000   307.674947  419.8111  338.891392
RegionN    221.89650   307.67495     0.000000   964.2855    5.826088
Regions    264.23327  419.81110   964.285493     0.0000  1073.134031
RegionSyd  274.47618  338.89139    5.826088  1073.1340    0.000000

> cat("The test statistic is:", T_squared, "\n")
The test statistic is: 264.2333
```

Or, Inserting

$$T^2 = \frac{nm}{n+m} 264.23327 = \frac{2 \cdot 2}{2+2} 264.23327 = 264.23327$$

We now investigate how Age, ambulant treatments, and sex affect admissions and bed days with the following model

$$[Admission \quad Bed \text{ days}] = [Age65 \quad Ambulant \quad Sex] \begin{bmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \\ \theta_{3,1} & \theta_{3,2} \end{bmatrix}$$

We test whether ambulant treatments and sex have any effect

$$\begin{bmatrix} \theta_{2,1} & \theta_{2,2} \\ \theta_{3,1} & \theta_{3,2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

with the following model

$$H_0: A \begin{bmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \\ \theta_{3,1} & \theta_{3,2} \end{bmatrix} B^T = C \quad \text{vs.} \quad H_1: A \begin{bmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \\ \theta_{3,1} & \theta_{3,2} \end{bmatrix} B^T \neq C$$

Question 1.4.

In the above model A is equal to?

We need to select the two lower rows. Thus

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Question 1.5.

The usual test-statistic for the above model has – under the null-hypothesis – the following distribution:

We use 4.21

$$A(r \times k), B(s \times p) \quad \text{and} \quad C(r \times s)$$

$$\{y \mid \frac{\det(\mathbf{e})}{\det(\mathbf{e} + \mathbf{h})} \leq U(s, r, n - k)_\alpha\}$$

$$A(r \times k) = A(2 \times 3)$$

$$C(r \times s) = C(2 \times 2)$$

Inserting

$$U(s, r, n - k) = U(2, 2, 10 - 3) = U(2, 2, 7)$$

Question 1.6.

The H matrix in the usual test statistic is? (hint: use the option **‘INVERSE’** in the PROC GLM model statement to get $(X^T X)^{-1}$)

In R:

```
# Load the necessary libraries (if not already loaded)
library(car)

Model_Q1 <- lm(cbind(adm, bed) ~ age65 + amb + sex -1, data = hospitaluse)
summary <- summary(Model_Q1)

Coefficients_adm <- summary$`Response adm`$coefficients[, "Estimate"]
Coefficients_bed <- summary$`Response bed`$coefficients[, "Estimate"]

# Create a dataframe with the two columns
theta <- as.matrix(cbind(Coefficients_adm, Coefficients_bed))

# Create a dataframe with the two columns
x <- as.matrix(cbind(hospitaluse$age65, hospitaluse$amb, hospitaluse$sex))#, hospitaluse$adm,
hospitaluse$bed))

# Define matrix invxx
invxx <- solve(t(x)%*%x)

# Define matrices A, B, C
A <- matrix(c(0, 1, 0, 0, 0, 1), nrow = 2, ncol = 3, byrow = TRUE)
B <- matrix(c(1, 0, 0, 1), nrow = 2, ncol = 2, byrow = TRUE)
C <- matrix(c(0, 0, 0, 0), nrow = 2, ncol = 2, byrow = TRUE)

# Calculate delta
delta <- A %*% theta %*% t(B) - C

# Calculate h
h <- t(delta) %*% solve(A %*% invxx %*% t(A)) %*% delta

# Print h
print(h)
```

Result in R:

```
> # Print h
> print(h)
```

	[,1]	[,2]
[1,]	823.2456	4399.122
[2,]	4399.1224	26899.649

Problem 2.

We consider a three dimensional normally distributed random variable with mean

$$E\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix}$$

And dispersion

$$D\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} 1 & \rho & \varphi \\ \rho & 1 & \rho \\ \varphi & \rho & 1 \end{bmatrix}$$

Question 2.1.

What is the expectation of Y given $\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} x \\ z \end{bmatrix}$

We use

||| Theorem 1.27

If X_2 is regularly distributed, i.e. if Σ_{22} has full rank, then the distribution of X_1 conditioned on $X_2 = x_2$ is again a normal distribution, and the following holds

$$\begin{aligned} E(X_1|X_2 = x_2) &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ D(X_1|X_2 = x_2) &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \end{aligned}$$

If Σ_{22} does not have full rank then the conditional distribution is still normal and Σ_{22}^{-1} in the above equations should be substituted by a generalised inverse Σ_{22}^- .

We reorder the matrix

$$D\begin{pmatrix} Y \\ X \\ Z \end{pmatrix} = \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \varphi \\ \rho & \varphi & 1 \end{bmatrix}$$

$$E(Y|\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} x \\ z \end{bmatrix}) = \mu_y + [\rho \quad \rho] \begin{bmatrix} 1 & \varphi \\ \varphi & 1 \end{bmatrix}^{-1} \begin{bmatrix} x - \mu_x \\ z - \mu_z \end{bmatrix}$$

Question 2.2.

What is the dispersion of $\begin{bmatrix} X \\ Y \end{bmatrix}$ given $Z=z$

We again use Theorem 1.27

$$D\left(\begin{bmatrix} X \\ Y \end{bmatrix} \middle| Z = z\right) = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} - \begin{bmatrix} \varphi \\ \rho \end{bmatrix} 1^{-1} [\varphi \quad \rho] = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} - \begin{bmatrix} \varphi^2 & \rho\varphi \\ \rho\varphi & \rho^2 \end{bmatrix} = \begin{bmatrix} 1 - \varphi^2 & \rho - \rho\varphi \\ \rho - \rho\varphi & 1 - \rho^2 \end{bmatrix}$$

Question 2.3.

What is the squared maximum correlation of Y with a linear combination of X and Z

We use

||| Theorem 1.42

We consider the situation above. Let σ_i be the i 'th column in Σ_{xy} , i.e. σ_i^T is the i 'th row in Σ_{yx} . Further, let σ_{ii} denote the i 'th diagonal element, i.e. the variance of Y_i

Then

$$\rho_{y_i|x} = \frac{\sqrt{\sigma_i^T \Sigma_{xx}^{-1} \sigma_i}}{\sqrt{\sigma_{ii}}}.$$

If we let

$$\Sigma_i = \begin{bmatrix} \sigma_{ii} & \sigma_i^T \\ \sigma_i & \Sigma_{xx} \end{bmatrix},$$

then

$$1 - \rho_{y_i|x}^2 = \frac{\det \Sigma_i}{\sigma_{ii} \det \Sigma_{xx}} = \frac{V(Y_i|X)}{V(Y_i)},$$

We have $\Sigma_i = \begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \varphi \\ \rho & \varphi & 1 \end{bmatrix}$ and $\Sigma_{xx} = \begin{bmatrix} 1 & \varphi \\ \varphi & 1 \end{bmatrix}$

$$\rho_{y|x}^2 = 1 - \frac{\det \Sigma_i}{\sigma_{ii} \det \Sigma_{xx}} = 1 - \frac{2\varphi\rho^2 - 2\rho^2 - \varphi^2 + 1}{1 - \varphi^2}$$

Problem 3.

Enclosure A with SAS program and SAS output belongs to this problem. The data was compiled in order to investigate the use of different ingredients in three types of baking goods: cookies, pastries, and pizzas. For the three types, the use of 133 different ingredients was recorded. In total, 1931 recipes were investigated.

Background: The difference between cookies, pastries and pizza can lead to heated debates. Reddit user *u/everest4ever* compiled the following dataset by scraping <http://www.foodnetwork.com/> to win an argument relating to a cookie competition in his office, where his cookies were beaten by the egg tarts of a colleague. His analysis showed that based on these data, egg tarts cannot be classified as cookies, and that the colleague should thus be disqualified. The reddit post can be found here:

https://www.reddit.com/r/dataisbeautiful/comments/7ke5a6/the_christmas_cookie_competition_at_my_office/

However, the background information is not relevant for the problem at hand.

Load the data in R

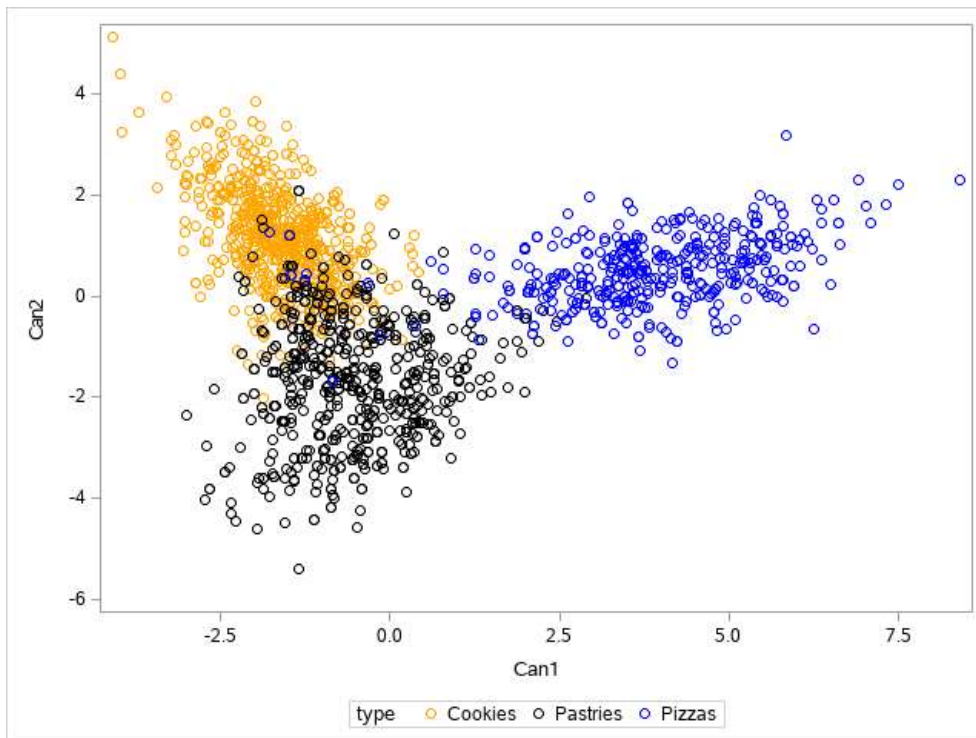
```
# Read the data from the csv file
data_P3=read.csv("home_food.csv")
```

Question 3.1.

As we can only generate $k-1$ Canonical Discriminant Functions (CDF) the two plotted are all there is. See page 360.

In this way one can continue until one gets an eigenvalue for $W^{-1}B$ which is 0 (or until $W^{-1}B$ is exhausted). Since $\text{rk}(B) \leq k-1$ we have at most $k-1$ eigenvalues ≥ 0 .

Further, using all CDF is equivalent to LDA. I.e. if the groups are separated based on CDF, they will also be by LDA as they are equivalent in this case. Based on the plot of the first 2 canonical discriminant functions we can conclude



The data has discriminative value, but a linear method will not give a perfect separation

Question 3.2.

The first canonical function clearly has the most information, while the second has very little discriminative power with regards to cookies and pizza. We find the 4 numerically largest variables from CAN1

In this question we have slight different results between R and SAS due to the different algorithm structure of Canonical discriminant analysis in these two programming languages.
Despite these small differences, you can answer this question correctly for 3 of the 4 largest variables using either SAS or R programming language.

In R:

```
library(MASS)
library(candisc)
```

```
data_P3_2 <- data_P3
# Define the classes (type)
data_P3_2$type <- as.factor(data_P3_2$type)
```

```
### Canonical discriminant analysis ###
### The results differ from SAS but the largest variables from CAN1 remain the same with slight different values
```

```
lm_mower <-
lm(cbind(Italianseasoning, Worcestershiresauce, allspice, almonds, anchovies, anise, apples, apricots, arugula, bac
```

```

on,bakingsoda,bananas,basil,bean,beef,bellpeppers,blackpepper,blueberries,bourbon,brandy,broccoli,butter,b
uttermilk,butterscotchchips,capers,cardamom,carrots,cayennepepper,celery,cheese,cherries,chickenstock,chil
i,chocolate,cinnamon,cloves,cocoapowder,coconut,coffeepowder,cookiedough,cookies,cornsyrup,cornmeal,c
ornstarch,cranberries,cream,creamcheese,cumin,curry,dough,eggplant,eggs,flour,foodcoloring,garlic,ginger,g
rahamcrackers,guava,hazelnuts,honey,hotsauce,icecream,icing,jalapeno,ketchup,leaves,lemons,lettuce,limes,
maplesyrup,margarine,marshmallows,mayonnaise,meringue,milk,molasses,mushrooms,mustard,nutmeg,oats,
oil,olives,onions,oranges,oregano,pancetta,paprika,parsley,peanutbutter,peas,pecans,peppermint,pepperoni,pi
ecrust,pinenuts,pineapples,pistachios,potatoes,prosciutto,puffpastry,pumpkinspice,raisins,raspberries,redpep
pers,ricotta,rosemary,rum,salami,salt,sandwichcookies,sausage,scallions,sesame,shallots,sherry,shrimps,sour
cream,spinach,strawberries,sugar,tartar,thyme,tomatoes,tortillas,vanilla,vinegar,wafercookies,walnuts,water,
whippingcream,wine,yeast,zucchini) ~ type, data = data_P3_2)
# perform canonical discriminant analysis
library(candisc)
#can_mower <- candisc(lm_mower,terms="type")
can_mower <- candisc(mod = lm_mower,term="type")
summary(can_mower)

# Std coefficients table
coef(can_mower)

# Extract coefficients table
coefficients_table <- as.data.frame(coef(can_mower))

# Order coefficients table based on Can1 column
coefficients_table_ordered <- coefficients_table[order(coefficients_table[, "Can1"]), ]
coefficients_table_ordered

# Display the top 10 rows
top_10 <- head(coefficients_table_ordered, 10)

# Display the bottom 10 rows
bottom_10 <- tail(coefficients_table_ordered, 10)

# Show the top and bottom 10 rows
print(top_10)
print(bottom_10)

```

Result in R:

```
> # Show the top and bottom 10 rows
> print(top_10)
```

	Can1	Can2
sugar	-0.3432247	0.09835478
leaves	-0.2623732	-0.17275713
cookies	-0.2304479	0.27953498
spinach	-0.2233612	-0.09906611
eggs	-0.1897330	-0.18748480
butter	-0.1837032	-0.14813950
ketchup	-0.1708996	-0.09840898
cookiedough	-0.1542593	0.06747824
ginger	-0.1456076	0.02341162
coconut	-0.1389637	0.18195160

```
> print(bottom_10)
```

	Can1	Can2
tomatoes	0.1111259	0.04625687
sausage	0.1212556	0.10037974
mayonnaise	0.1303914	0.07143720
salt	0.1305677	-0.02528758
basil	0.1506384	0.07887832
garlic	0.1564636	0.08793886
oil	0.1883039	-0.08511688
cheese	0.2327744	-0.03540310
dough	0.3053666	-0.10710428
yeast	0.4227127	0.11079719

According to the R results, the 4 largest numerically variables in Can1 are yeast, sugar, dough and leaves.

Question 3.3.

First we test whether there is a difference in mean value given these three variables. The usual test-statistic for this has – under the null-hypothesis – the following distribution

We use theorem

|||| Theorem 4.9

We use the same notation as given above. Now, let

$$T^2 = \frac{nm}{n+m} (\bar{X} - \bar{Y})^T S^{-1} (\bar{X} - \bar{Y}).$$

Then the critical region for a test of H_0 against H_1 at level α is equal to

$$C = \{x_1, \dots, x_n, y_1, \dots, y_m \mid \frac{n+m-p-1}{(n+m-2)p} t^2 > F(p, n+m-p-1)_{1-\alpha}\}$$

Here t^2 is the observed value of T^2 .

In R:

```
library(MASS)
# Prior probabilities for the classes
# SAS by default has equal prior probabilities, so we need equal prior in R to receive the same results
different_types <- c("Cookies", "Pastries")
num_types <- length(different_types)
prior <- rep(1/num_types, num_types)
```

```

#linear discriminant analysis
# Define the classes (region)
# Filter and keep only the rows where 'type' is 'Cookies' or 'Pastries'
filtered_data <- data_P3[data_P3$type %in% c("Cookies", "Pastries"), ]

# Print the filtered data
head(filtered_data)

filtered_data$class <- as.factor(filtered_data$type)
# Define the variables for the analysis
variables <- c("puffpastry", "water", "apples")
# Perform Linear Discriminant Analysis
z <- lda(class ~ ., data = filtered_data[, c("class", variables)],prior=prior)

Class_Level_Information = data.frame("Frequency" = z$counts,"Proportion"=z$counts/z$N,"Prior"=z$prior)
print("Class Level Information:")
Class_Level_Information

n <- nrow(filtered_data)
Classes <- nlevels(filtered_data$type)

paste0("DF Within Classes = ",n-Classes)
paste0("DF Between Classes = ",Classes-1)

zpred <- predict(z)

#Confusion Matrix:
print("Confusion Matrix Linear Discriminant Analysis:")
xtabs(~filtered_data$type+zpred$class)
confusion_matrix = xtabs(~filtered_data$type+zpred$class)

```

Result in R:

```

[1] "Class Level Information:"
> Class_Level_Information
      Frequency Proportion Prior
Cookies      803   0.5385647   0.5
Pastries     688   0.4614353   0.5

```

Thus $p=3$, $n=803$, $m=688$. We insert

$$F(p, n + m - p - 1) = F(3, 803 + 688 - 3 - 1) = F(3, 1487)$$

Question 3.4.

What is the misclassification rate?

In R:

```
##### misclassification rates #####
confusion_matrix = xtabs(~filtered_data$type+zpred$class)
error_cookies <- confusion_matrix["Cookies", "Pastries"]
frequency_cookies <- sum(Class_Level_Information["Cookies", "Frequency"])
misclassification_rate_cookies <- error_cookies/frequency_cookies

error_pastries <- confusion_matrix["Pastries", "Cookies"]
frequency_pastries <- sum(Class_Level_Information["Pastries", "Frequency"])
misclassification_rate_pastries <- error_pastries/frequency_pastries

misclassification_rate_total <- (misclassification_rate_cookies*Class_Level_Information["Cookies",
"Prior"]) + (misclassification_rate_pastries*Class_Level_Information["Pastries", "Prior"])

cat("The misclassification rate for cookies is:", misclassification_rate_cookies, "\n")
cat("The misclassification rate for pastries is:", misclassification_rate_pastries, "\n")
cat("The total misclassification rate:", misclassification_rate_total, "\n")
```

Result in R:

```
> cat("The misclassification rate for cookies is:", misclassification_rate_cookies, "\n")
The misclassification rate for cookies is: 0.08094645
> cat("The misclassification rate for pastries is:", misclassification_rate_pastries, "\n")
The misclassification rate for pastries is: 0.4055233
> cat("The total misclassification rate:", misclassification_rate_total, "\n")
The total misclassification rate: 0.2432349
```

Question 3.5.

A new recipe needs to be classified. In the ingredient list we find [*puffpastry water apples*] = [1 0 0], i.e. puffpastry but not apples or water.

In R:

```
# I have taken this code from DiscriMiner library because the library DiscriMiner doesn't run in my
# computer
# You can run the code below or install the DiscriMiner library if it works in your computer
# webpage: https://github.com/gastonstat/DiscriMiner/blob/master/R/linDA.R
#####
#####
my_verify <-
function(x, y, qualitative=FALSE, na.rm=na.rm)
{
  # x: matrix or data frame with explanatory variables
  # y: vector or factor with group memberships
  # qualitative: logical indicating verification for disqual
```

```

# na.rm: logical indicating missing values in x

# x matrix or data.frame
if (is.null(dim(x)))
  stop("\n'variables' is not a matrix or data.frame")
# no missing values allowed when na.rm=FALSE
if (!na.rm) {
  if (length(complete.cases(x)) != nrow(x))
    stop("\nSorry, no missing values allowed in 'variables'")
}
# check lengths of x and y
if (nrow(x) != length(y))
  stop("\n'variables' and 'group' have different lengths")
# y vector or factor
if (!is.vector(y) && !is.factor(y))
  stop("\n'group' must be a factor")
# make sure y is a factor
if (!is.factor(y)) y = as.factor(y)
# no missing values in y
if (any(!is.finite(y)))
  stop("\nNo missing values allowed in 'group'")
# quantitative or qualitative variables?
if (!qualitative)
{ # quantitative data
  # make sure is matrix
  if (!is.matrix(x)) x <- as.matrix(x)
  # only numeric values
  if (!is.numeric(x))
    stop("\n'variables' must contain only numeric values")
} else { # data frame with qualitative data
  # variables as data frame with factors
  fac_check = sapply(x, class)
  if (!is.data.frame(x) && any(fac_check != "factor"))
    stop("\nA data frame with factors was expected")
}

# verified inputs
if (is.null(colnames(x)))
  colnames(x) = paste(rep("X", ncol(x)), seq_len(ncol(x)), sep="")
if (is.null(rownames(x)))
  rownames(x) = 1L:nrow(x)
list(X=x, y=y)
}

my_linDA <-
function(X, y, learn, test, prior, prob)

```

```

{
# Perform a linear discriminant analysis
# X: matrix or data.frame with explanatory variables
# y: vector or factor with group membership
# learn: vector of learning observations
# test: vector of testing observations
# prior: vector of prior proportions
# prob: logical indicating results in probability terms

# how many observations
n = nrow(X[learn,])
ntest = length(test)
# how many groups
ng = nlevels(y[learn])
glevs = levels(y[learn])
# how many obs in each group
nobs_group = as.vector(table(y[learn]))
# group means
GM = my_groupMeans(X[learn,], y[learn])
# within-class covariance matrix
W = my_withinCov(X[learn,], y[learn])
# inverse of Within cov matrix
W_inv = solve(W)

# constant terms of classification functions
cons = rep(0, ng)
# coefficients of classification functions
Betas = matrix(0, nrow(W_inv), ng)
for (k in 1:ng)
{
  cons[k] = -(1/2) * GM[k,] %*% W_inv %*% GM[k,] + log(prior[k])
  Betas[,k] = t(GM[k,]) %*% W_inv
}
# Fisher's Discriminant Functions
FDF = rbind(cons, Betas)
rownames(FDF) = c("constant", colnames(X))
colnames(FDF) = glevs

# matrix of constant terms
A = matrix(rep(cons, ntest), ntest, ng, byrow=TRUE)
# apply discrim functions
Disc = X[test,] %*% Betas + A

# probability values
if (prob) {
  # exponential

```

```

Disc <- 1 - exp( -(Disc - apply(Disc, 1, min, na.rm=TRUE)))
# predicting classes
pred = Disc / drop(Disc %*% rep(1, ng))
# predicted class
pred_class = factor(max.col(pred), levels=seq_along(glevs), labels=glevs)
} else {
# predicted class
pred = apply(Disc, 1, function(u) which(u == max(u)))
names(pred) = NULL
# assign class values
pred_class = factor(pred, levels=seq_along(glevs), labels=glevs)
}
dimnames(Disc) = list(rownames(X[test,]), glevs)
# confusion matrix
conf = table(original=y[test], predicted=pred_class)
# results
res = list(FDF=FDF, conf=conf, Disc=Disc, pred_class=pred_class)
res
}

```

```

my_groupMeans <-
function(X, g)
{
# X: matrix of explanatory variables
# g: factor with group memberships

# how many groups
ng = nlevels(g)
# matrix with group means
Means = matrix(0, ng, ncol(X))
for (j in 1:ncol(X))
{
Means[,j] = tapply(X[,j], g, FUN=mean)
}
# add names
rownames(Means) = levels(g)
colnames(Means) = colnames(X)
# results
Means
}

```

```

my_withinCov <-
function(X, g, div_by_n=FALSE)
{
# X: matrix of explanatory variables
# g: factor with group memberships

```

```

# div_by_n: logical indicating division by num of observations

# how many observations
n = nrow(X)
# how many variables
ncx = ncol(X)
# group levels and number of levels
glevs = levels(g)
ng = nlevels(g)
# within cov matrix
Within = matrix(0, ncx, ncx)
for (k in 1:ng)
{
  tmp <- g == glevs[k]
  nk = sum(tmp)
  if (div_by_n) {
    Wk = ((nk-1)/n) * var(X[tmp,])
  } else {
    #Wk = ((nk-1)/(n-1)) * var(X[tmp,])
    # divide by degrees of freedom
    Wk = ((nk-1)/(n-ng)) * var(X[tmp,])
  }
  Within = Within + Wk
}
# result
Within
}

linDA <-
function(variables, group, prior = NULL, validation = NULL,
         learn = NULL, test = NULL, prob = FALSE)
{
  # Perform a linear discriminant analysis
  # variables: matrix or data.frame with explanatory variables
  # group: vector or factor with group membership
  # prior: vector of prior probabilities (NULL = proportions)
  # validation: NULL, "crossval", "learntest"
  # learn: vector of learn-set
  # test: vector of test-set
  # prob: logical indicating results in probability terms

  # check inputs
  verify_Xy = my_verify(variables, group, na.rm=FALSE)
  X = verify_Xy$X
  y = verify_Xy$y

```

```

# type of validation
if (is.null(validation)) {
  validation = "none"
} else {
  vali = validation %in% c("crossval", "learntest")
  if (!vali)
    stop("\nIncorrect type of validation")
}

# how many observations
n = nrow(X)
# how many variables
p = ncol(X)
# how many groups
ng = nlevels(y)
# how many obs in each group
nobs_group = as.vector(table(y))
# prior probabilities
if (!is.null(prior))
{
  if (!is.numeric(prior) || !is.vector(prior))
    stop("\n'prior' probabilities incorrectly defined")
  if (length(prior) != ng)
    stop("\n'prior' probabilities don't match number of groups")
  if (any(prior > 1) || any(prior < 0))
    stop("'prior' probabilities must range between [0,1]")
  if (round(sum(prior), 5) != 1)
    stop("'prior' probabilities don't add to 1")
} else {
  # prior as proportions
  prior = nobs_group / n
  props = prior
}
# group levels
glevs = levels(y)

## linDA with no validation
if (validation == "none") {
  get_linda = my_linDA(X, y, 1:n, 1:n, prior, prob)
  err = 1 - sum(diag(get_linda$conf)) / n
}

## linDA with learn-test sets validation
if (validation == "learntest")
{
  if (any(learn) <= 0 || any(learn) > n)

```

```

    stop("\nsubscript out of bounds in 'learn' set")
  if (any(test) <= 0 || any(test) > n)
    stop("\nsubscript out of bounds in 'test' set")
  # apply linDA
  get_linda = my_linDA(X, y, learn, test, prior, prob)
  # misclassification error rate
  err = 1 - sum(diag(get_linda$conf))/length(test)
}

## linDA with crossvalidation
if (validation == "crossval")
{
  # linDA for all observations
  get_linda = my_linDA(X, y, 1:n, 1:n, prior, prob)
  # elements in each group
  elems_group = vector("list", ng)
  for (k in 1:ng) {
    elems_group[[k]] = which(group == glevs[k])
  }
  # misclassification error rate
  mer = 0
  # 10 crossvalidation samples
  for (r in 1:10)
  {
    test = vector("list", ng)
    test_sizes = floor(n * props / 10)
    for (k in 1:ng) {
      test[[k]] = sample(elems_group[[k]], test_sizes[k])
    }
    test = unlist(test)
    learn = (1:n)[-test]
    # apply linDA
    linda_cv = my_linDA(X, y, learn, test, prior, prob)
    # misclassification error rate
    mer = mer + sum(diag(linda_cv$conf))/n
  }
  # total misclassification error rate
  err = 1 - mer
}

## specifications
specs = list(n=n, p=p, ng=ng, glevs=glevs,
             nob_group=nobs_group, validation=validation)
## results
structure(list(functions = get_linda$FDF,
               confusion = get_linda$conf,

```

```

    scores = get_linda$Disc,
    classification = get_linda$pred_class,
    error_rate = err,
    specs = specs),
    class = "linda")
}

#####
#####
lda.coef <-
linDA(filtered_data[,c("puffpastry","water","apples")],filtered_data[,ncol(filtered_data)],prior=prior)
# removing log(prior), see Definition 5.15
# if prior are equal, (+log(p)) is not needed.
lda.coef$functions[1,] = lda.coef$functions[1,] - log(prior)

print("Linear Discriminant Function for species:")
lda.coef$functions

```

Result in R:

```

[1] "Linear Discriminant Function for species:"
> lda.coef$functions
      Cookies  Pastries
constant -0.02489970 -1.078988
puffpastry 0.12380661  3.713769
water      0.61521403  2.670969
apples     -0.02423918  2.370819

```

For cookies we have $S_{cookies} : -0.0249 + 0.12381 = 0.0989$

For pastries we have $S_{pastries} : -1.07899 + 3.71377 = 2.6348$

Question 3.6.

We consider the Linear Discriminant Function for classifying between cookies and pastries using a subset of the variables: puffpastry water apples. We classify as cookies if the function is positive. Using equal priors but a loss of ten for classifying pastry wrong we get:

We use

|||| Theorem 5.4

Let $\pi_1 \sim N(\mu_1, \Sigma)$ and $\pi_2 \sim N(\mu_2, \Sigma)$. Then we have

$$\frac{f_1(x)}{f_2(x)} \geq c \Leftrightarrow x^T \Sigma^{-1} (\mu_1 - \mu_2) - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 \geq \log c$$

$$\Leftrightarrow \left[x^T \Sigma^{-1} \mu_1 - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 \right] - \left[x^T \Sigma^{-1} \mu_2 - \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 \right] \geq \log c.$$

In R:

```
lda_coef <- data.frame(lda.coef$functions)
# Add a new column "Cookies_Pastries" with the summation of "Cookies" and "Pastries"
lda_coef$Cookies_Pastries <- lda_coef$Cookies + lda_coef$Pastries

# Print the updated data frame
print(lda_coef)
```

Result in R:

```
> print(lda_coef)
      Cookies Pastries Cookies_Pastries
constant -0.02489970 -1.078988      1.054088
puffpastry 0.12380661  3.713769     -3.589963
water      0.61521403  2.670969     -2.055755
apples     -0.02423918  2.370819     -2.395058
```

$$[puffpastry\ water\ apples] \begin{bmatrix} -3.5900 \\ -2.0558 \\ -2.3951 \end{bmatrix} + 1.0541 > \log c$$

We then need to adjust the right hand side

||| Theorem 5.1

The *Bayes solution* to the classification problem is given by the region

$$R_1 = \left\{ x \mid \frac{f_1(x)}{f_2(x)} \geq \frac{L_{21}p_2}{L_{12}p_1} \right\}.$$

$$[puffpastry\ water\ apples] \begin{bmatrix} -3.5900 \\ -2.0558 \\ -2.3951 \end{bmatrix} + 1.0541 > \log \frac{10}{1}$$

$$[puffpastry\ water\ apples] \begin{bmatrix} -3.5900 \\ -2.0558 \\ -2.3951 \end{bmatrix} + 1.0541 > 2.3026$$

$$[puffpastry\ water\ apples] \begin{bmatrix} -3.5900 \\ -2.0558 \\ -2.3951 \end{bmatrix} - 1.2485 > 0$$

Problem 4.

Enclosure B with SAS program and SAS output belongs to this problem. We consider data giving the rates (pr. 1000 capita) of different types of crimes and the prevalence of different types of unemployment benefits and social security (pr. 1000 capita) for the 98 municipalities (kommuner) in Denmark (Source <http://www.statistikbanken.dk>)

We consider the following variables for crime

SAS-name	Meaning
C2	Sexual crimes
C19	Murder
C21	Simple violence
C22	Serious violence
C23	Especially serious violence
C30	Threats
C53	Robbery
C55	Vandalism
C64	Sale of narcotics
C74	Weapon possession

And for social security and benefits

SAS-name	Meaning
S1	Educational benefits (SU)
S3	Unemployment benefit
S4	Social security
S19	Flexjob (state supported jobs)
S33	Integration benefits
S36	Sickness benefit

We shall now investigate the relations between the crime rates and the social benefits by means of a Canonical Correlation Analysis.

Load the data in R

```
# Read the data from the csv file
data_P4=read.csv("socialcrime2018.csv")

# Select the specific variables
selected_variables <- data_P4[, c("C2", "C19", "C21", "C22", "C23", "C30", "C53", "C55", "C64", "C74",
"S1", "S3", "S4", "S19", "S33", "S36")]

# Create a new data frame with the selected variables
data_P4_4_1 <- data.frame(selected_variables)

# View the first 5 rows
head(data_P4_4_1)

# find the correlation of variables using in this problem
df_corr = cor(data_P4_4_1)
```

Question 4.1.

The first canonical correlation describes what fraction of the variation between V1 and W1

The squared correlation is the degree of variance explained, see page 29 and the *squared coefficient of correlation* represents the reduction in variance, i.e. the fraction of Y 's variance, which can be explained by X , since

$$\rho^2 = \frac{V(Y) - V(Y|X = x)}{V(Y)}.$$

In R:

```
## Computing Canonical correlations and Eigenvalues.
## var C2 C19 C21 C22 C23 C30 C53 C55 C64 C74 in SAS are the variables C2 C19 C21 C22 C23 C30
## C53 C55 C64 C74 for x in R in E..
## with S1 S3 S4 S19 S33 S36 in SAS are the variables S1 S3 S4 S19 S33 S36 for y in R in E..
Exx = as.matrix(df_corr[1:10,1:10])
Eyx = as.matrix(df_corr[11:16,1:10])
Exy = as.matrix(df_corr[1:10,11:16])
Eyy = as.matrix(df_corr[11:16,11:16])
invExx = solve(Exx)
invEyy = solve(Eyy)

# Add diagonal loading (regularization) to make the matrices positive definite
epsilon <- 1e-5 # A small positive value
Eyx <- Eyx + epsilon * diag(dim(Eyx)[1])
Eyy <- Eyy + epsilon * diag(dim(Eyy)[1])
#install.packages("eigen")
library("eigen")

#canonical correlation:
Cancorr = eigen(Eyx%*%invExx%*%Exy,Eyy,symmetric = TRUE)
values = sort(Cancorr$values,decreasing = TRUE)

#E is the residual variation after having predicted Y by means of X
H = Eyx%*%invExx%*%Exy
E = Eyy - Eyx%*%invExx%*%Exy
invE = solve(E)
Ev <- eigen(invE%*%H)
var = Ev$values
# Eigenvalues, Proportion and Cumulative proportion of Variance:
varPC <- var/sum(var)
cumu = c(1:6)
for (i in 1:6){
  cumu[i] = sum(varPC[1:i])
}
results <- data.frame("CanCor" = sqrt(values),"Squared CanCor" =
values,"eigenvalues"=var,"proportion"=varPC,
"cumulative" = cumu)
```

```
print(results)
```

Result in R:

```
> print(results)
  CanCor Squared.CanCor eigenvalues proportion cumulative
1 0.7319224 0.53571035 1.15382788 0.509770166 0.5097702
2 0.6092569 0.37119395 0.59031548 0.260805987 0.7705762
3 0.4415922 0.19500363 0.24224162 0.107024240 0.8776004
4 0.3932144 0.15461756 0.18289658 0.080805138 0.9584055
5 0.2686396 0.07216724 0.07778044 0.034364003 0.9927695
6 0.1268941 0.01610212 0.01636564 0.007230466 1.0000000
```

Question 4.2.

How many canonical correlations are significant at the 5% level?

In R:

```
library(CCP)
## Calculating p-values using F-approximations with Wilks test:
n = dim(data_P4_4_1)[1]
p = length(Exx[,1])
q = length(Eyy[,1])

HypTest <- p.asym(results$CanCor,n,p,q,tstat="Wilks")
print("Table with information about the Canonical Correlations similar to the output in SAS:")
results_all <- data.frame(results,HypTest)
results_all
```

Result in R:

```
> results_all
  CanCor Squared.CanCor eigenvalues proportion cumulative id stat approx df1 df2 p.value
1 0.7319224 0.53571035 1.15382788 0.509770166 0.5097702 wilks 0.1813730 2.7906371 60 434.6790 9.266424e-10
2 0.6092569 0.37119395 0.59031548 0.260805987 0.7705762 wilks 0.3906462 1.9454205 45 374.3820 4.770742e-04
3 0.4415922 0.19500363 0.24224162 0.107024240 0.8776004 wilks 0.6212507 1.3406529 32 311.3719 1.093048e-01
4 0.3932144 0.15461756 0.18289658 0.080805138 0.9584055 wilks 0.7717435 1.0999966 21 244.6242 3.483816e-01
5 0.2686396 0.07216724 0.07778044 0.034364003 0.9927695 wilks 0.9128927 0.6682582 12 172.0000 7.801681e-01
6 0.1268941 0.01610212 0.01636564 0.007230466 1.0000000 wilks 0.9838979 0.2847621 5 87.0000 9.202858e-01
```

We look at the p-value column and then we take the correct number of canonical correlations (significant at 5% level).

2 canonical correlations

Question 4.3.

In R:

```
#for Var variables:
Cancorr2 = geigen(Exy%*%invEyy%*%Eyx,Exx,symmetric = TRUE)

# Correlations between the Var variables and their Canonical variables
```

```
Cor_H = Exx%*%Cancorr2$vector
```

```
#Corresponding standardized canonical coefficients / correlations:
```

```
Coefficients = data.frame("Variables" = colnames(df_corr)[1:10],
  "CorrV1" = -round(Cor_H[,10],4),
  "CorrV2" = -round(Cor_H[,9],4),
  "CorrV3" = round(Cor_H[,8],4),
  "CorrV4" = round(Cor_H[,7],4),
  "CorrV5" = -round(Cor_H[,6],4),
  "CorrV6" = -round(Cor_H[,5],4))
```

```
print("Correlations between the Var variables and their Canonical variables:")
```

```
Coefficients
```

```
#Corresponding standardized canonical coefficients / correlations:
```

```
Coefficients = data.frame("Variables" = colnames(df_corr)[1:10],
  "CorrV1" = -round(Cancorr2$vector[,10],4),
  "CorrV2" = round(Cancorr2$vector[,9],4),
  "CorrV3" = round(Cancorr2$vector[,8],4),
  "CorrV4" = -round(Cancorr2$vector[,7],4),
  "CorrV5" = round(Cancorr2$vector[,6],4),
  "CorrV6" = round(Cancorr2$vector[,5],4))
```

```
print("Standardized Canonical Coefficients for the VAR Variables:")
```

```
Coefficients
```

Result in R:

```
[1] "Correlations between the var variables and their Canonical variables:"
> Coefficients
  variables  CorrV1  CorrV2  CorrV3  CorrV4  CorrV5  CorrV6
C2          C2  0.1215 -0.0076  0.1803 -0.2002 -0.0542  0.1088
C19         C19 -0.0395 -0.0823 -0.1312  0.1677  0.5034  0.0403
C21         C21  0.2936  0.6585 -0.5208 -0.1470  0.0405 -0.1223
C22         C22  0.4290  0.1232  0.1836  0.0735 -0.0600 -0.2128
C23         C23  0.1250 -0.0405 -0.0060 -0.1542 -0.6409  0.5784
C30         C30  0.4743  0.2980  0.4886 -0.3056  0.0505 -0.3116
C53         C53  0.9024 -0.2435 -0.1127  0.1281 -0.0777 -0.1329
C55         C55  0.4694  0.3228  0.1279 -0.3822  0.4902  0.3124
C64         C64  0.3488  0.2145  0.2763  0.6691 -0.0963  0.1675
C74         C74  0.3742  0.6711  0.5517 -0.0253  0.0148  0.1235
```

```
[1] "Standardized Canonical Coefficients for the VAR variables:"
> Coefficients
  variables  CorrV1  CorrV2  CorrV3  CorrV4  CorrV5  CorrV6
C2          C2 -0.1081  0.1916  0.0504  0.3483  0.1368 -0.1173
C19         C19 -0.0711  0.1312 -0.0734 -0.2682 -0.4731 -0.0976
C21         C21  0.0592 -0.6966 -0.8078  0.0510  0.1894  0.2184
C22         C22 -0.0312  0.0925  0.0418 -0.3190  0.2681  0.3505
C23         C23  0.0975  0.1113 -0.0535  0.2011  0.5940 -0.6180
C30         C30  0.1584  0.1656  0.5506  0.4947  0.0365  0.6129
C53         C53  0.8159  0.3342 -0.2691 -0.0100  0.0519  0.1169
C55         C55  0.3172  0.1522  0.0033  0.3232 -0.7688 -0.8075
C64         C64  0.1209 -0.0451  0.1583 -0.8227 -0.1181 -0.3494
C74         C74 -0.0220 -0.8098  0.3789 -0.1242  0.1544 -0.0589
```

The third canonical variate V3 can be interpreted as

A contrast between Threats and Weapon Possession against simple violence. Can be seen from the standardized coefficients, as well as the correlations

Question 4.4.

From the relation between V1 and W1, we see a clear link between robberies and the number of people on educational benefits. One may speculate on whether students that have run out of money, may be tempted to commit a robbery, or whether an underlying socioeconomic factor is the reason for this. We investigate this further. What is the correlation between C53 (robberies) and S1 (educational benefits) when we condition on S3 (Unemployment benefit)?

We use the formula from page 34

$$\rho_{y_1 y_2 | x} = \frac{\rho_{y_1 y_2} - \rho_{y_1 x} \rho_{y_2 x}}{\sqrt{(1 - \rho_{y_1 x}^2)(1 - \rho_{y_2 x}^2)}}.$$

We insert from the correlation matrix

$$\rho_{C53, S1 | S3} = \frac{\rho_{C53, S1} - \rho_{C53, S3} \rho_{S1, S3}}{\sqrt{(1 - \rho_{C53, S3}^2)(1 - \rho_{S1, S3}^2)}} = \frac{0.60196 - 0.52351 \cdot 0.59661}{\sqrt{(1 - 0.52351^2)(1 - 0.59661^2)}} = 0.4236$$

In R:

```
print("Correlation matrix:")
df_corr

# Find the value at the intersection of the "C53" row and "S1" column
p_c53_s1 <- df_corr["C53", "S1"]
p_c53_s3 <- df_corr["C53", "S3"]
p_s1_s3 <- df_corr["S1", "S3"]

correlation_value <- (p_c53_s1 - p_c53_s3 * p_s1_s3) / sqrt((1 - p_c53_s3^2) * (1 - p_s1_s3^2))
cat("The correlation value is:", correlation_value, "\n")
```

Result in R:

```
> df_corr
      C2      C19      C21      C22      C23      C30      C53      C55      C64
C2  1.00000000  0.046205505  0.15486954  0.180391673 -0.01896614  0.26100944  0.122030161  0.18970739  0.29333640
C19  0.04620551  1.000000000  0.13016409  0.035093577 -0.02962908  0.08871928 -0.002154332  0.07639123 -0.03092677
C21  0.15486954  0.130164095  1.000000000  0.277407365  0.01196876  0.35451988  0.090125318  0.39873319  0.13717148
C22  0.18039167  0.035093577  0.27740736  1.000000000  0.04345974  0.47501640  0.271726046  0.46647019  0.20448068
C23 -0.01896614 -0.029629078  0.01196876  0.043459742  1.000000000  0.03789379  0.023251692 -0.01481912  0.04642455
C30  0.26100944  0.088719284  0.35451988  0.475016398  0.03789379  1.000000000  0.209627095  0.47537716  0.25738543
C53  0.12203016 -0.002154332  0.09012532  0.271726046  0.02325169  0.20962709  1.000000000  0.12527069  0.25453098
C55  0.18970739  0.076391227  0.39873319  0.466470185 -0.01481912  0.47537716  0.125270694  1.000000000  0.03291812
C64  0.29333640 -0.030926770  0.13717148  0.204480677  0.04642455  0.25738543  0.254530981  0.03291812  1.000000000
C74  0.24254682 -0.007386386  0.22097786  0.367625425  0.08484474  0.56471893  0.149445092  0.49922446  0.34918537
S1  0.05637377  0.034334637  0.16692834  0.230192665  0.06336675  0.20517936  0.601957615  0.28869110  0.18611633
S3  0.05637512 -0.026576884  0.27552013  0.316873441  0.02893525  0.35626819  0.523509793  0.31332031  0.35004772
S4  0.06118521 -0.030586843  0.32470612  0.194514070 -0.05321769  0.36761524  0.121996303  0.35684915  0.15702316
S19 -0.06259093  0.036056784 -0.03836360 -0.155732298 -0.07580267 -0.16045551 -0.465992368 -0.12522492  0.01024155
S33 -0.11782574  0.061079662  0.20979740 -0.174392042 -0.05165517 -0.29637085 -0.266956118 -0.15255774 -0.12674632
S36  0.01452528 -0.106705253  0.17346969 -0.006904161  0.07369906  0.08807415 -0.272748562  0.04497395  0.05994042

      C74      S1      S3      S4      S19      S33      S36
C2  0.242546817  0.05637377  0.05637512  0.06118521 -0.06259093 -0.11782574  0.014525275
C19 -0.007386386  0.03433464 -0.02657688 -0.03058684  0.03605678  0.06107966 -0.106705253
C21  0.220977857  0.16692834  0.27552013  0.32470612 -0.03836360  0.20979740  0.173469686
C22  0.367625425  0.23019266  0.31687344  0.19451407 -0.15573230 -0.17439204 -0.006904161
C23  0.084844741  0.06336675  0.02893525 -0.05321769 -0.07580267 -0.05165517  0.073699063
C30  0.564718935  0.20517936  0.35626819  0.36761524 -0.16045551 -0.29637085  0.088074149
C53  0.149445092  0.60195761  0.52350979  0.12199630 -0.46599237 -0.26695612 -0.272748562
C55  0.499224463  0.28869110  0.31332031  0.35684915 -0.12522492 -0.15255774  0.044973955
C64  0.349185366  0.18611633  0.35004772  0.15702316  0.01024155 -0.12674632  0.059940419
C74  1.000000000  0.11838153  0.39404565  0.46744756  0.04333601 -0.20257299  0.290886006
S1  0.118381526  1.000000000  0.59661444  0.13060486 -0.39665140 -0.17463898 -0.350825758
S3  0.394045648  0.59661444  1.000000000  0.62056681 -0.39160192 -0.32616792 -0.066905284
S4  0.467447560  0.13060486  0.62056681  1.000000000 -0.06744594 -0.24439466  0.223828296
S19 0.043336006 -0.39665140 -0.39160192 -0.06744594  1.000000000  0.31397691  0.588472732
S33 -0.202572988 -0.17463898 -0.32616792 -0.24439466  0.31397691  1.000000000  0.100815409
S36 0.290886006 -0.35082576 -0.06690528  0.22382830  0.58847273  0.10081541  1.000000000

> cat("The correlation value is:", correlation_value, "\n")
The correlation value is: 0.4235697
```

Question 4.5.

The 95% confidence interval for the correlation between C53 (robberies) and S1 (educational benefits) is:

We use from page 40

|||| Theorem 1.40

Assume the situation is as in the previous theorem. We consider the hypothesis

$$H_0 : \rho_{ij|m+1,\dots,p} = \rho_0$$

versus

$$H_1 : \rho_{ij|m+1,\dots,p} \neq \rho_0.$$

We let

$$Z = \frac{1}{2} \log \frac{1 + R_{ij|m+1,\dots,p}}{1 - R_{ij|m+1,\dots,p}}$$

and

$$z_0 = \frac{1}{2} \log \frac{1 + \rho_0}{1 - \rho_0}.$$

Under H_0 we will have

$$(Z - z_0) \cdot \sqrt{n - (p - m) - 3} \text{ approx. } \sim N(0, 1).$$

Also shown in example 1.41.

We have $n=98$, we insert

$$P(-1.96 < (Z - z)\sqrt{98 - 0 - 3} < 1.96) \approx 95\%$$

$$P(-1.96 - 9.7468Z < -9.7468z < 1.96 - 9.7468Z) \approx 95\%$$

$$P(Z - 0.2011 < z < Z + 0.2011) \approx 95\%$$

We find

$$Z = \frac{1}{2} \log \frac{1 + 0.60196}{1 - 0.60196} = 0.6962$$

And we get the z-limits

$$[0.4951, 0.8973]$$

We then need to transform it

$$\left[\frac{e^{2 \cdot 0.4951} - 1}{e^{2 \cdot 0.4951} + 1}, \frac{e^{2 \cdot 0.8973} - 1}{e^{2 \cdot 0.8973} + 1} \right] = [0.4583, 0.7150]$$

In R:

```
Z= (1/2)*log((1+p_c53_s1)/(1-p_c53_s1))
cat("Z is equal to:", Z, "\n")

# Given values
range <- 0.2011

# Calculate the interval
lower_bound <- Z - range
upper_bound <- Z + range

# Create a vector containing the lower and upper bounds
interval <- c(lower_bound, upper_bound)

# Calculate the transform interval
lower_transform_bound <- (exp(2*interval[1])-1) / (exp(2*interval[1])+1)
upper_transform_bound <- (exp(2*interval[2])-1) / (exp(2*interval[2])+1)

# Create a vector containing the transform lower and upper bounds
transform_interval <- c(lower_transform_bound, upper_transform_bound)
cat("The transform interval is equal to:", transform_interval, "\n")
```

Result in R:

```
> cat("The transform interval is equal to:", transform_interval, "\n")
The transform interval is equal to: 0.458264 0.7149863
```

Problem 5.

Enclosure C with SAS program and SAS output belongs to this problem. We consider the relation between overall satisfaction with life, and the satisfaction with personal economy, family life, social relations and work, in 7 different income groups. The data is based on a questionnaire, where 0 means ‘not satisfied at all’, and 10 means ‘perfectly satisfied’. Data is from <http://www.statistikbanken.dk>

The variables are

SAS-name	Meaning
life	Overall satisfaction with life
econ	Satisfaction with economical situation
famiy	Satisfaction with family life
social	Satisfaction with social relations
work	Satisfaction with work

The observation are from the following income groups

Observation	Income group (DKK)
1	0-99.999
2	100.000-199.999
3	200.000-299.999
4	300.000-399.999
5	400.000-499.999
6	500.000-599.999
7	600.000 +

We consider two models: M1 and M2.

M1 with all variables

$$life = \mu + \beta_1 \cdot econ + \beta_2 \cdot famiy + \beta_3 \cdot social + \beta_4 \cdot work + \epsilon$$

where μ is the intercept and ϵ is the error term.

M2 is a model where we have performed stepwise model selection.

Load the data in R

```
# Create a dataframe with the data
data_satisfaction <- data.frame(
  life = c(7.3, 7.4, 7.6, 7.7, 7.8, 7.8, 8.2),
  econ = c(5.8, 6.4, 7.1, 7.7, 8.2, 8.4, 8.9),
  famiy = c(8.0, 8.1, 8.0, 8.1, 8.1, 8.1, 8.3),
  social = c(7.6, 7.6, 7.7, 7.7, 7.9, 7.8, 8.0),
  work = c(7.3, 7.5, 7.3, 7.4, 7.6, 7.9, 7.9)
)
```

```
# Print the dataframe
print(data_satisfaction)
```

Question 5.1.

What is the usual test statistic for M1 vs M2

In R:

```
##### Model M1 #####
# Fit a linear regression model
Model_M1 <- lm(life ~ econ+family+social+work,data=data_satisfaction)

# Summary of the Model_M1 model
summary(Model_M1)

##### Model M2 #####
# Initialize an empty model
best_model <- lm(life ~ 1, data = data_satisfaction)

# Create a list of predictor variables
predictors <- c("econ", "family", "social", "work")

# Initialize a list to keep track of selected predictors
selected_predictors <- character(0)

# Loop through predictors
for (predictor in predictors) {
  # Add the predictor to the model
  temp_model <- update(best_model, formula = as.formula(paste("life ~", paste(selected_predictors,
collapse = " + "), " + ", predictor)), data = data_satisfaction)

  # Fit the model
  fit <- summary(temp_model)

  # Get the p-value for the added variable
  p_value <- fit$coefficients[rownames(fit$coefficients) == predictor, "Pr(>|t|)"]

  # Check if p-value is less than 0.15
  if (p_value < 0.15) {
    best_model <- temp_model
    selected_predictors <- c(selected_predictors, predictor)
  }
}

# Final selected model
final_model_M2 <- best_model

# Display the selected predictors
cat("Selected Predictors:", selected_predictors, "\n")

# Summary of the final model
summary(final_model_M2)
```

```
anova(Model_M1)
anova_M1 = anova(Model_M1)
```

```
anova(final_model_M2)
anova_M2 = anova(final_model_M2)
```

```
cat("SS res M2:",anova_M2$`Sum Sq`[3], "\n")
cat("SS res M1:",anova_M1$`Sum Sq`[5], "\n")
cat("DF res M2:",anova_M2$Df[3], "\n")
cat("SS res M1:",anova_M1$Df[5], "\n")
```

Compare M1 vs M2

```
F_observed_Model_H1 = ((anova_M2$`Sum Sq`[3]-anova_M1$`Sum Sq`[5])/(anova_M2$Df[3]-
anova_M1$Df[5]))/(anova_M1$`Sum Sq`[5]/anova_M1$Df[5])
cat("The test statistic for model H1 is:", F_observed_Model_H1, "\n")
```

Result in R:

```
> anova(Model_M1)
Analysis of Variance Table

Response: life
      Df Sum Sq Mean Sq F value    Pr(>F)
econ    1  0.48255   0.48255  90.4236 0.01088 *
family  1  0.02431   0.02431   4.5560 0.16637
social  1  0.00710   0.00710   1.3303 0.36797
work    1  0.00393   0.00393   0.7369 0.48112
Residuals 2  0.01067   0.00534
```

```
> anova(final_model_M2)
Analysis of Variance Table

Response: life
      Df Sum Sq Mean Sq F value    Pr(>F)
econ    1  0.48255   0.48255  88.9297 0.000705 ***
family  1  0.02431   0.02431   4.4808 0.101710
Residuals 4  0.02170   0.00543
```

```
> cat("SS res M2:",anova_M2$`Sum Sq`[3], "\n")
SS res M2: 0.02170492
> cat("SS res M1:",anova_M1$`Sum Sq`[5], "\n")
SS res M1: 0.01067316
> cat("DF res M2:",anova_M2$Df[3], "\n")
DF res M2: 4
> cat("SS res M1:",anova_M1$Df[5], "\n")
SS res M1: 2
```

```
> cat("The test statistic for model H1 is:", F_observed_Model_H1, "\n")
The test statistic for model H1 is: 1.033598
```

Using the test statistic formula, we have:

Test statistic for $H_0: E(Y) \in H$ against $H_1: E(Y) \in M \setminus H$:

$$\frac{\|p_M(Y) - p_H(Y)\|^2 / (k - r)}{\|Y - p_M(Y)\|^2 / (n - k)} = \frac{(SS_{res}(Hyp) - SS_{res}(Mod)) / (DF_{res}(Hyp) - DF_{res}(Mod))}{SS_{res}(Mod) / DF_{res}(Mod)}$$

$$F = \frac{(0.02170 - 0.01067) / (4 - 2)}{0.01067 / 2} = 1.0337$$

Question 5.2.

The usual test-statistic for M1 vs M2 has – under the null-hypothesis – the following distribution

We use

||| Theorem 2.21

Let the situation be as above. Then the likelihood ratio test at level α of testing

$$H_0 : \mu \in H \quad \text{versus} \quad H_1 : \mu \in M \setminus H,$$

is equivalent to the test given by the critical region

$$C_\alpha = \{(y_1, \dots, y_n) \mid \frac{\|p_M(y) - p_H(y)\|^2 / (k-r)}{\|y - p_M(y)\|^2 / (n-k)} > F(k-r, n-k)_{1-\alpha}\}.$$

And readily gets the answer F(2,2)

Question 5.3.

What is the reduction in the fraction of variance described when moving from model M1 to M2

We find R^2 for the two models M1: 0.9798, M2: 0.9589

Answer = $0.9798 - 0.9589 = 0.0209$

In R:

```
# Calculate the R-squared value Model_M1
summary_model_M1 <- summary(Model_M1)
r_squared_M1 <- summary_model_M1$r.squared
```

```
# Print the R-squared value
cat("R-squared value:", r_squared_M1, "\n")
```

```
# Calculate the R-squared value Model_M2
summary_model_M2 <- summary(final_model_M2)
r_squared_M2 <- summary_model_M2$r.squared
```

```
# Print the R-squared value
cat("R-squared value:", r_squared_M2, "\n")

# Reduction
cat("The reduction is:", r_squared_M1-r_squared_M2, "\n")
```

Result in R:

```
> cat("R-squared value:", r_squared_M1, "\n") > cat("R-squared value:", r_squared_M2, "\n")
R-squared value: 0.9798075 R-squared value: 0.9589366

> # Reduction
> cat("The reduction is:", r_squared_M1-r_squared_M2, "\n")
The reduction is: 0.02087089
```

We now only consider model M2

Question 5.4.

What is the leverage of observation 1?

In R:

```
Obs <- 1:length(data_satisfaction$life)
DFFITS <- round(dffits(final_model_M2), 4)
R_student <- round(rstudent(final_model_M2),4)
HatDiagH <- round(hatvalues(final_model_M2),4)
Residual <- round(residuals(final_model_M2),4)
Covratio <- round(covratio(final_model_M2), 4)
Predicted_value <- round(predict(final_model_M2),4)

Stats <- data.frame(Obs,Predicted_value,Residual,R_student,HatDiagH,Covratio,DFFITS)
print(Stats)

# Find out the leverage - HatDiagH value for observation number 1
leverage_observation_1 <- Stats$HatDiagH[1]

# Print the leverage value for observation number 1
cat("Observation 1 leverage value is:", leverage_observation_1, "\n")
```

Result in R:

```
> print(Stats)
  Obs Predicted_value Residual R_student HatDiagH Covratio DFFITS
1   1          7.2657   0.0343    0.6251    0.5288    3.4840   0.6623
2   2          7.4740  -0.0740   -1.6888    0.4829    0.6176  -1.6320
3   3          7.5159   0.0841    1.7583    0.3583    0.4412   1.3139
4   4          7.7242  -0.0242   -0.3146    0.1541    2.5422  -0.1343
5   5          7.8205  -0.0205   -0.2876    0.2806    3.0366  -0.1796
6   6          7.8590  -0.0590   -1.0120    0.3705    1.5601  -0.7764
7   7          8.1408   0.0592    5.9493    0.8248    0.0065  12.9095
```

We look at HatDiagH column and we find that the observation 1 has leverage equal to 0.5288.

Question 5.5.

What is the 95% confidence interval for observation 7?

We use

|||| Theorem 2.15

Let the situation be as above. Then the $(1 - \alpha)$ -confidence interval for the expected value of a new observation Y will be

$$[u - t(n - k)_{1-\frac{\alpha}{2}} s \sqrt{c}, \quad u + t(n - k)_{1-\frac{\alpha}{2}} s \sqrt{c}].$$

We insert

$$[8.1408 - t(7 - 3)_{0.975} \sqrt{0.0054 \cdot 0.8248}, \quad 8.1408 + t(7 - 3)_{0.975} \sqrt{0.0054 \cdot 0.8248}]$$

$$[8.1408 - 2.776 \sqrt{0.0054 \cdot 0.8248}, \quad 8.1408 + 2.776 \sqrt{0.0054 \cdot 0.8248}]$$

$$[7.9555, \quad 8.3261]$$

$$[7.96, \quad 8.33]$$

In R:

```
# We consider the model M2
```

```
# Calculate confidence intervals for observation 7 (alpha = 0.05)
```

```
# Given values
```

```
mean_squared_error <- anova_M2$`Mean Sq`[3]
```

```
# Find out the leverage - HatDiagH value for observation number 7
```

```
leverage_obs_7 <- Stats$HatDiagH[7]
```

```
Predicted_value_obs_7 <- Stats$Predicted_value[7]
```

```
df <- 7 - 3
```

```
t_value <- qt(0.975, df = df) # 0.975 (1-α/2) corresponds to a 95% confidence interval
```

```
# Calculate the lower and upper bounds of the confidence interval
```

```
lower_bound <- Predicted_value_obs_7 - t_value * sqrt(mean_squared_error*leverage_obs_7)
```

```
upper_bound <- Predicted_value_obs_7 + t_value * sqrt(mean_squared_error*leverage_obs_7)
```

```
# Print the confidence interval
cat("Confidence Interval (95%) for observation 7: [", lower_bound, " , " , upper_bound, "]\n")
```

Result in R:

```
> cat("Confidence Interval (95%) for observation 7: [", lower_bound, " , " , upper_bound, "]\n")
Confidence Interval (95%) for observation 7: [ 7.955057 , 8.326543 ]
```

Question 5.6.

What is the 95% prediction interval for observation 7

We use

|||| Theorem 2.17

Let us assume that a new observation taken at (z_1, \dots, z_k) has a variance $c_1\sigma^2$. Furthermore, it is independent of the earlier observations. In that case a $(1 - \alpha)$ -prediction interval for the new observation equals the interval

$$[u - t(n - k)_{1-\frac{\alpha}{2}} s \sqrt{c + c_1}, u + t(n - k)_{1-\frac{\alpha}{2}} s \sqrt{c + c_1}].$$

In R:

```
# Calculate confidence intervals for observation 7 (alpha = 0.05)
# Given values

mean_squared_error <- anova_M2$`Mean Sq`[3]

# Find out the leverage - HatDiagH value for observation number 7
leverage_obs_7 <- Stats$HatDiagH[7]
Predicted_value_obs_7 <- Stats$Predicted_value[7]
df <- 7 - 3
t_value <- qt(0.975, df = df) # 0.975 (1-a/2) corresponds to a 95% prediction interval

# Calculate the lower and upper bounds of the confidence interval
lower_bound <- Predicted_value_obs_7 - t_value *
sqrt((mean_squared_error*leverage_obs_7)+(mean_squared_error*1))
upper_bound <- Predicted_value_obs_7 + t_value *
sqrt((mean_squared_error*leverage_obs_7)+(mean_squared_error*1))

# Print the confidence interval
cat("Prediction Interval (95%) for observation 7: [", lower_bound, " , " , upper_bound, "]\n")
```

Result in R:

```
> # Print the confidence interval
> cat("Prediction Interval (95%) for observation 7: [", lower_bound, " , " , upper_bound, "]\n")
Prediction Interval (95%) for observation 7: [ 7.864522 , 8.417078 ]
```

$$\begin{aligned} & [8.1408 - t(7-3)_{0.975} \sqrt{0.0054 \cdot 0.8248 + 0.0054 \cdot 1}, \quad 8.1408 + t(7-3)_{0.975} \sqrt{0.0054 \cdot 0.8248 + 0.0054 \cdot 1}] \\ & [8.1408 - 2.776 \sqrt{0.0054 \cdot 0.8248 + 0.0054 \cdot 1}, \quad 8.1408 + 2.776 \sqrt{0.0054 \cdot 0.8248 + 0.0054 \cdot 1}] \\ & [7.87, \quad 8.42] \end{aligned}$$

Question 5.7.

What is the 95% confidence interval for the econ parameter

We use from page 108

$$\hat{V}(\hat{\theta}_{i_0}) = \hat{\sigma}^2 \left\{ \left(\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} \right)^{-1} \right\}_{ii}$$

||| Theorem 2.23

Let the situation be as above. Then the critical region for testing H_0 against H_1 at significance level α is

$$C_\alpha = \left\{ (y_1, \dots, y_n) \mid \hat{\theta}_{i_0} < c - t(f)_{1-\frac{\alpha}{2}} \sqrt{\hat{V}(\hat{\theta}_{i_0})} \text{ or } \hat{\theta}_{i_0} > c + t(f)_{1-\frac{\alpha}{2}} \sqrt{\hat{V}(\hat{\theta}_{i_0})} \right\}$$

Where $f = n - \text{rk}(x)$

In R:

```
# Create the matrix 'x' with an intercept, econ, and family
x <- as.matrix(cbind(intercept = 1, econ = data_satisfaction$econ, family = data_satisfaction$family))

# Define matrix invxx
invxx <- solve(t(x) %*% x)

# Set the row and column names
rownames(invxx) <- colnames(x)
colnames(invxx) <- colnames(x)

# Print the 'invxx' matrix with named rows and columns
print("X'X Inverse:")
print(invxx)

print("Parameter estimates:")
final_model_M2$coefficients

invxx_econ <- invxx["econ", "econ"]
mean_squared_error <- anova_M2$`Mean Sq`[3]
```

```

V_theta <- mean_squared_error*invxx_econ

# Calculate confidence intervals for econ parameter (alpha = 0.05)
# Find out the parameter estimate value for econ
Econ_parameter_estimate <- final_model_M2$coefficients[[2]]
df <- 7 - 3
t_value <- qt(0.975, df = df) # 0.975 (1-a/2) corresponds to a 95% confidence interval

# Calculate the lower and upper bounds of the confidence interval
lower_bound <- Econ_parameter_estimate - t_value * sqrt(V_theta)
upper_bound <- Econ_parameter_estimate + t_value * sqrt(V_theta)

# Print the confidence interval
cat("Confidence Interval (95%) econ parameter: [", lower_bound, " , ", upper_bound, "]\n")

```

Result in R:

```

> cat("Confidence Interval (95%) econ parameter: [", lower_bound, " , ", upper_bound, "]\n")
Confidence Interval (95%) econ parameter: [ 0.08408449 , 0.3009272 ]

```

Analytically, we start by finding

$$V(\theta_i) = 0.0054 \cdot 0.281030445 = 0.0015$$

We then have

$$\left[0.19251 - t(7 - 3)_{0.975} \sqrt{0.0015}, \quad 0.19251 + t(7 - 3)_{0.975} \sqrt{0.0015} \right]$$

$$\left[0.19251 - 2.776 \sqrt{0.0015}, \quad 0.19251 + 2.776 \sqrt{0.0015} \right]$$

$$[0.0844, \quad 0.3007]$$

$$[0.084, \quad 0.301]$$

Problem 6.

Enclosure D with SAS program and SAS output belongs to this problem. We again consider the data from problem 4, but now only the crime variables.

We consider the following variables for crime

SAS-name	Meaning
C2	Sexual crimes
C19	Murder
C21	Simple violence
C22	Serious violence
C23	Especially serious violence
C30	Threats
C53	Robbery
C55	Vandalism
C64	Sale of narcotics
C74	Weapon possession

We seek to investigate the underlying patterns in crime by running a principal component analysis on all crime variables.

Load the data in R

```
# Read the data from the csv file
```

```
data=read.csv("socialcrime2018.csv")
```

```
# Select the specific variables
```

```
selected_P6_variables <- data[, c("C2", "C19", "C21", "C22", "C23", "C30", "C53", "C55", "C64", "C74")]
```

```
# Create a new data frame with the selected variables
```

```
data_P6 <- data.frame(selected_P6_variables)
```

Question 6.1.

How many factors do we need to account for 90 % of the variance?

In R:

```
### When we have cor=FALSE, PCA is performed on the covariance matrix,
```

```
### which takes into account the variances and covariances between variables
```

```
# Principal Component Analysis on the covariance matrix.
```

```
pca <- princomp(data_P6,cor=FALSE,scores=TRUE)
```

```
# variance for each Principal Component:
```

```
var <- pca$sdev^2
```

```
# proportion of variance:
```

```
varPC <- var/sum(var)
```

```
# cumulative variance:
```

```

cumu = c(1:10)
for (i in 1:10){
  cumu[i] = sum(varPC[1:i])
}
results_PCA <- data.frame("eigenvalues"=var,"proportion"=varPC,
                          "cumulative" = cumu)

# Format the eigenvalues, proportion, and cumulative columns with the specified format
results_PCA$eigenvalues <- sprintf("%.8f", as.numeric(results_PCA$eigenvalues))
results_PCA$proportion <- sprintf("%.4f", as.numeric(results_PCA$proportion))
results_PCA$cumulative <- sprintf("%.4f", as.numeric(results_PCA$cumulative))

print("Eigenvalues of the Covariance matrix:")
results_PCA

```

Result in R:

```

[1] "Eigenvalues of the Covariance matrix:"
> results_PCA

```

	eigenvalues	proportion	cumulative
Comp.1	0.16396660	0.6165	0.6165
Comp.2	0.04003897	0.1505	0.7670
Comp.3	0.03645228	0.1371	0.9041
Comp.4	0.01128039	0.0424	0.9465
Comp.5	0.00938293	0.0353	0.9818
Comp.6	0.00227074	0.0085	0.9903
Comp.7	0.00138109	0.0052	0.9955
Comp.8	0.00115364	0.0043	0.9998
Comp.9	0.00004634	0.0002	1.0000
Comp.10	0.00000203	0.0000	1.0000

Question 6.2.

The usual test statistic for the last 4 eigenvalues being equal is

We use

||| Theorem 6.8

If we are using the estimated *variance-covariance matrix* $\hat{\Sigma}$, the test statistic for testing the hypothesis above becomes

$$Z_1 = -n^* \log \frac{\det \hat{\Sigma}}{\hat{\lambda}_1 \cdot \dots \cdot \hat{\lambda}_m \cdot \hat{\lambda}_*^{k-m}} = -n^* \log \frac{\hat{\lambda}_{m+1} \cdot \dots \cdot \hat{\lambda}_k}{\hat{\lambda}_*^{k-m}},$$

where

$$n^* = n - m - \frac{1}{6}(2(k - m) + 1 + \frac{2}{k - m}),$$

and

$$\hat{\lambda}_* = (\text{tr } \hat{\Sigma} - \hat{\lambda}_1 - \dots - \hat{\lambda}_m) / (k - m) = (\hat{\lambda}_{m+1} + \dots + \hat{\lambda}_k) / (k - m).$$

The critical region using a test at level α is approximately

$$\{(x_1, \dots, x_n) | z_1 > \chi^2(\frac{1}{2}(k - m + 2)(k - m - 1))_{1-\alpha}\}.$$

In SAS:

We find the eigenvalues and number of observations

7	0.00139533	Observations 98 Variables 10
8	0.00116553	
9	0.00004682	
10	0.00000205	

$$n^* = 98 - 6 - \frac{1}{6}\left(2(10 - 6) + 1 + \frac{2}{10 - 6}\right) = 90.4167$$

$$\begin{aligned} Z_2 &= -90.4167 \log \frac{0.00139533 \cdot 0.00116553 \cdot 0.00004682 \cdot 0.00000205}{\left[\frac{(0.00139533 + 0.00116553 + 0.00004682 + 0.00000205)}{4}\right]^4} \\ &= -90.4167 \log \frac{1.5609 \cdot 10^{-16}}{(6.5243 \cdot 10^{-4})^4} = -90.4167 \log(8.6148 \cdot 10^{-4}) = 638.0580 \end{aligned}$$

In R:

```
# We perform the usual test statistic for the last 4 eigenvalues
# We have the last to eigenvalues from the output with k=10, m=6, and n=98
k = 10
m = 6
```

```

n = nrow(data_P6)
n_star = n-m-((1/6)*((2*(k-m)+1+(2/(k-m))))))

eig7 = 0.00138109
eig8 = 0.00115364
eig9 = 0.00004634
eig10 = 0.00000203
lambda_star = (eig7+eig8+eig9+eig10)/(k-m)
z2 = -n_star*log((eig7*eig8*eig9*eig10)/(lambda_star^(k-m)))
cat("The test statistic for the last 4 eigenvalues is equal to:", z2, "\n")

```

Result in R:

```

> cat("The test statistic for the last 4 eigenvalues is equal to:", z2, "\n")
The test statistic for the last 4 eigenvalues is equal to: 638.0214

```

Question 6.3.

From the score plots we see at least four clear outliers: observation 36 (Guldborgsund), 39 (Lolland), 59 (Fanø), and 75 (Samsø). Out of these, find the two where problems with vandalism (C55) are *lowest*.

In R:

```
# First, Second and Third principal components
pca$loadings[,1]
pca$loadings[,2]
pca$loadings[,3]

# Format the eigenvalues, proportion, and cumulative columns with the specified format
First_PC <- sprintf("%.6f", as.numeric(pca$loadings[,1]))
Second_PC <- sprintf("%.6f", as.numeric(pca$loadings[,2]))
Third_PC <- sprintf("%.6f", as.numeric(pca$loadings[,3]))

# Create a data frame
loadings_PCA <- data.frame(
  Prin1 = First_PC,
  Prin2 = Second_PC,
  Prin3 = Third_PC
)

# Print the data frame
print("First, Second and Third principal components:")
print(loadings_PCA)

#####
# Create a vector of characters from 1 to 98
observations <- as.character(1:98)

# Load the ggrepel package
library(ggrepel)
# scores plot for PC1 and PC2
scores = data.frame(pca$scores)
ggplot(scores, aes(x = scores[,1], y = scores[,2])) +
  geom_point(col = 'red', size = 2) +
  geom_text_repel(aes(label = observations), box.padding = 0.5, size = 3, max.overlaps = Inf) +
  labs(x = 'principal component 1', y = 'principal component 2') +
  ggtitle('Component Scores')

# scores plot for PC1 and PC3
scores = data.frame(pca$scores)
ggplot(scores, aes(x = scores[,1], y = scores[,3])) +
  geom_point(col = 'red', size = 2) +
  geom_text_repel(aes(label = observations), box.padding = 0.5, size = 3, max.overlaps = Inf) +
  labs(x = 'principal component 1', y = 'principal component 3') +
  ggtitle('Component Scores')
```

```
# scores plot for PC2 and PC3
scores = data.frame(pca$scores)
ggplot(scores,aes(x = scores[,2],y = scores[,3]))+
  geom_point(col = 'red',size = 2)+
  geom_text_repel(aes(label = observations),box.padding = 0.5,size=3,max.overlaps = Inf)+
  labs(x = 'principal component 2',y = 'principal component 3')+
  ggtitle('Component Scores')
```

Result in R:

We look at the principal components table:

```
> print("First, Second and Third principal components:")
[1] "First, Second and Third principal components:"
> print(loadings_PCA)
      Prin1      Prin2      Prin3
1  0.067496  0.047888  0.101836
2  0.001408 -0.003900  0.001322
3  0.288262 -0.685112  0.650626
4  0.061321  0.008694  0.018889
5  0.000046  0.000485  0.000602
6  0.208282  0.126746  0.239590
7  0.018183  0.012023  0.020934
8  0.842662 -0.130116 -0.518551
9  0.015356  0.044196  0.076453
10 0.393140  0.702243  0.483090
```

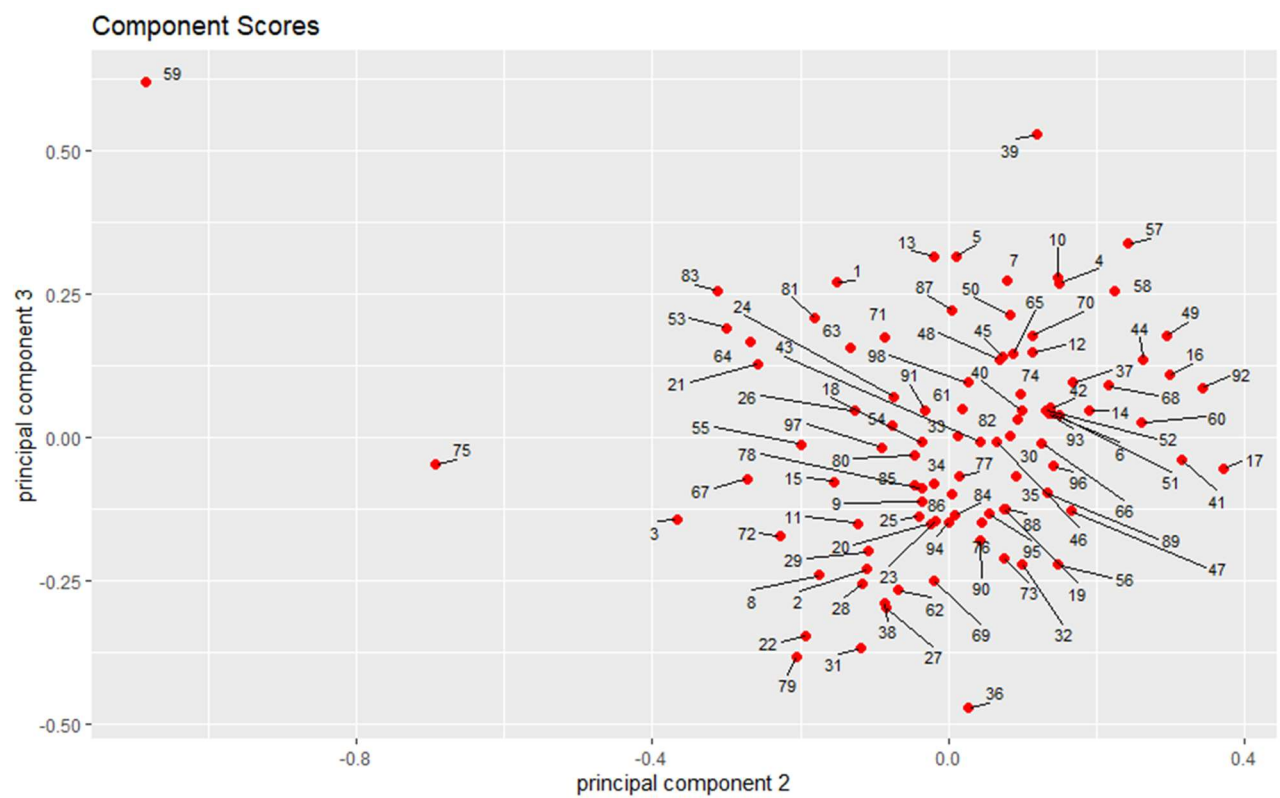
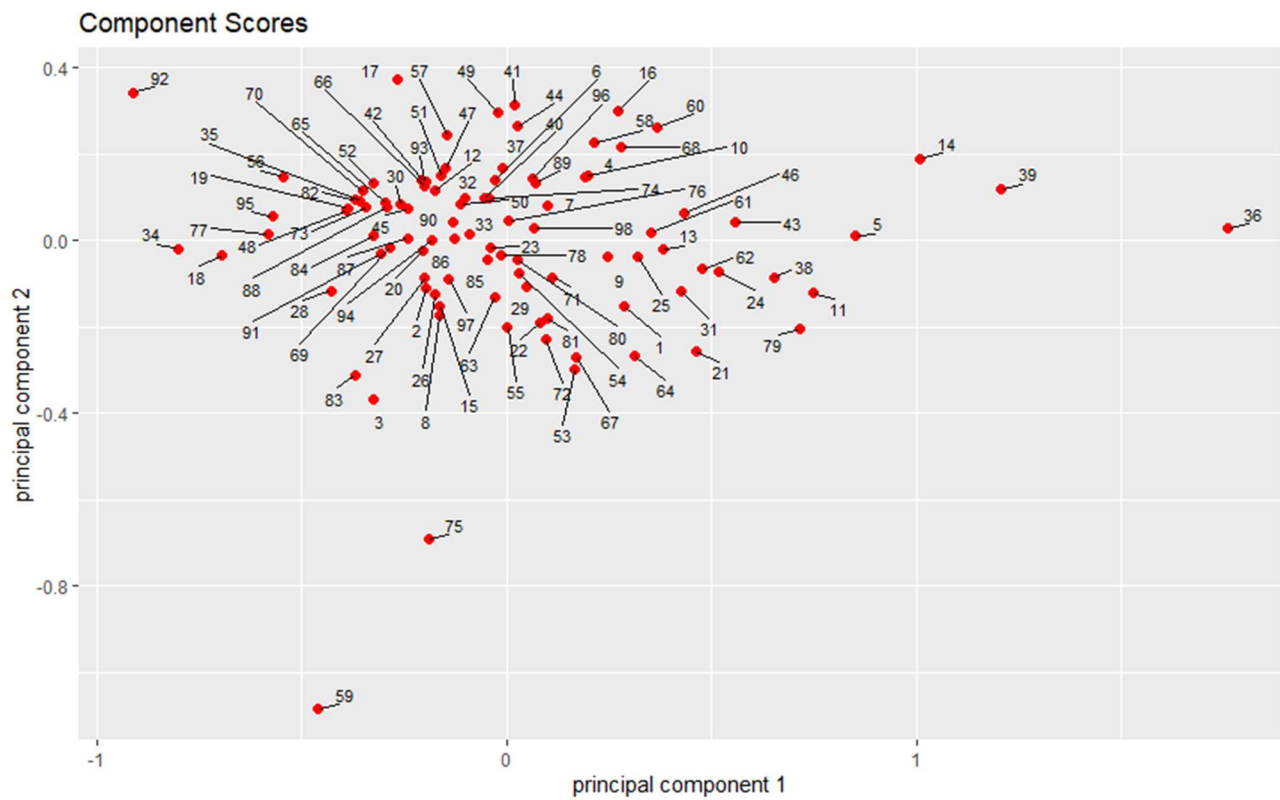
So for the problem to be *lowest*, we need a low score on component 1.

We read from the score plots

Observation	Component 1	Component 2	Component 3
Guldborgsund – 36	1.8	0	-0.5
Lolland – 39	1.2	0.1	0.5
Fanø – 59	-0.5	-1.1	0.6
Samsø - 75	-0.1	-0.6	0

This seems to indicate that Fanø and Samsø has the lowest problem with vandalism

Result in R:



LAST PAGE:

END OF THE EXAM SET