Stat 717 Final Exam

2023-12-14

```
#Download the dataset MorderGE.RData ... Run the following code to perform the k-means clustering of the 30 samples using 3 clusters

load("C:/Users/Jonathan/Downloads/MorderGE.RData")
#header(Morder)
#type
morder.kmeans <- kmeans(Morder, centers=3)
```

```
#1i: Use table function to report the number of samples in each cluster

ncluster <- table(morder.kmeans$cluster)
ncluster
```

```
##
##  1  2  3
##  9 10 11
```

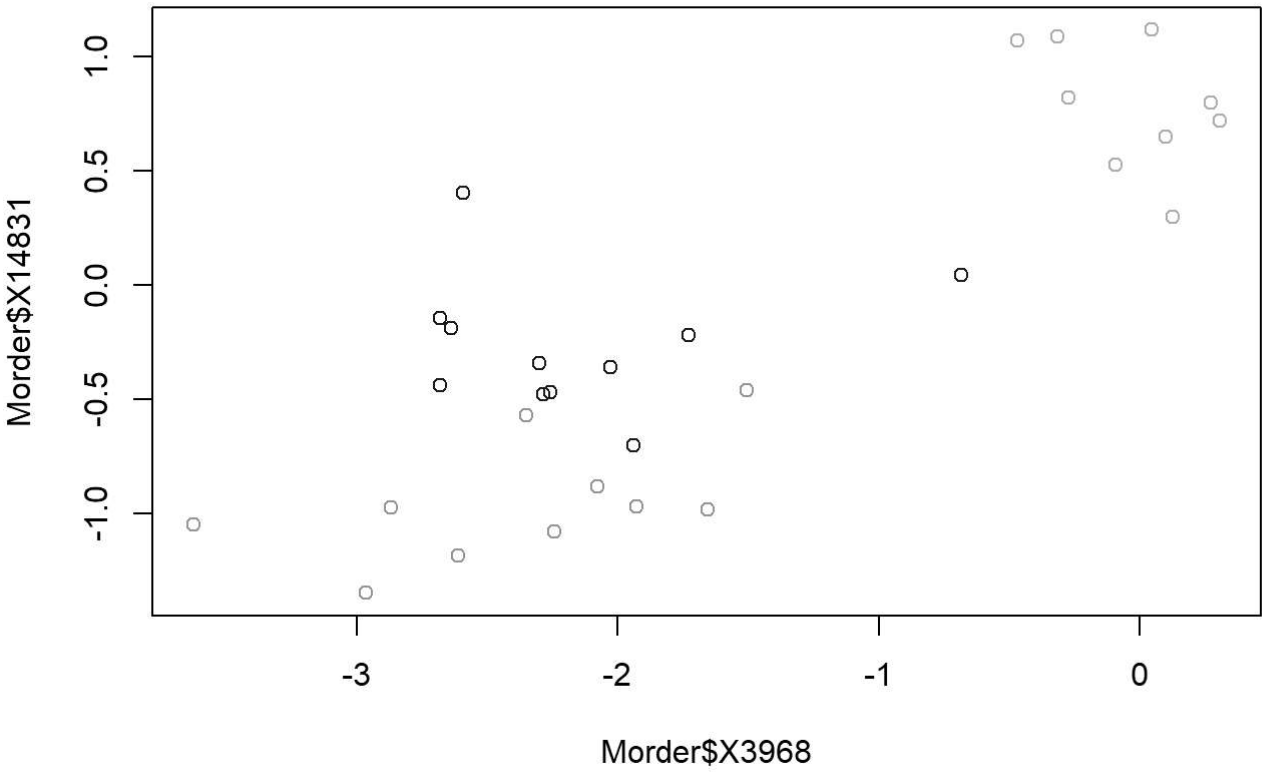
```
#1ii: The type vector (which is also in MorderGE.RData) contains the T cell type of each sample. We expect that the three clusters obtained from k-means should match the three cell types. Compare type with your k-means cluster and report how many samples are wrongly clustered.

tcluster <- table(morder.kmeans$cluster, type)
tcluster
```

```
##    type
##     effector memory naive
##   1        0      0     9
##   2       10      0     0
##   3        0      9     2
```

```
#1iii: Morder$X3968 and Morder$X14831 contain the first two columns of Morder (i.e. the gene expression measurements of the first two genes.) Use either plot or ggplot to plot Morder$X14831 against Morder$X3968 and color the points according to the cluster ID from the k-means output.

plot(Morder$X3968,Morder$X14831, col=c("orange","green", "blue")[morder.kmeans$cluster])
```



#2: Download the dataset DNASeqs.RData and then load it into your R workspace. The dataframe dna.seq has 11 rows and 1620 columns, where each row vector is the DNA sequence of some protein. The length of the sequenc e is 1620, and thus each column represents one nucleotide. Run the following code to compute the distance m atrix, dna.dist.

```
hamming <- function(x, y){
sum(x != y)}
load("C:/Users/Jonathan/Downloads/DNASeqs.RData")
n <- nrow(dna.seq)
D <- matrix(0, nrow=n, ncol=n)
row.names(D) = row.names(dna.seq)
for (i in 2:n){
for (j in 1:(i-1)){
D[i, j] <- hamming(dna.seq[i,], dna.seq[j,])}}
dna.dist = as.dist(D)
```

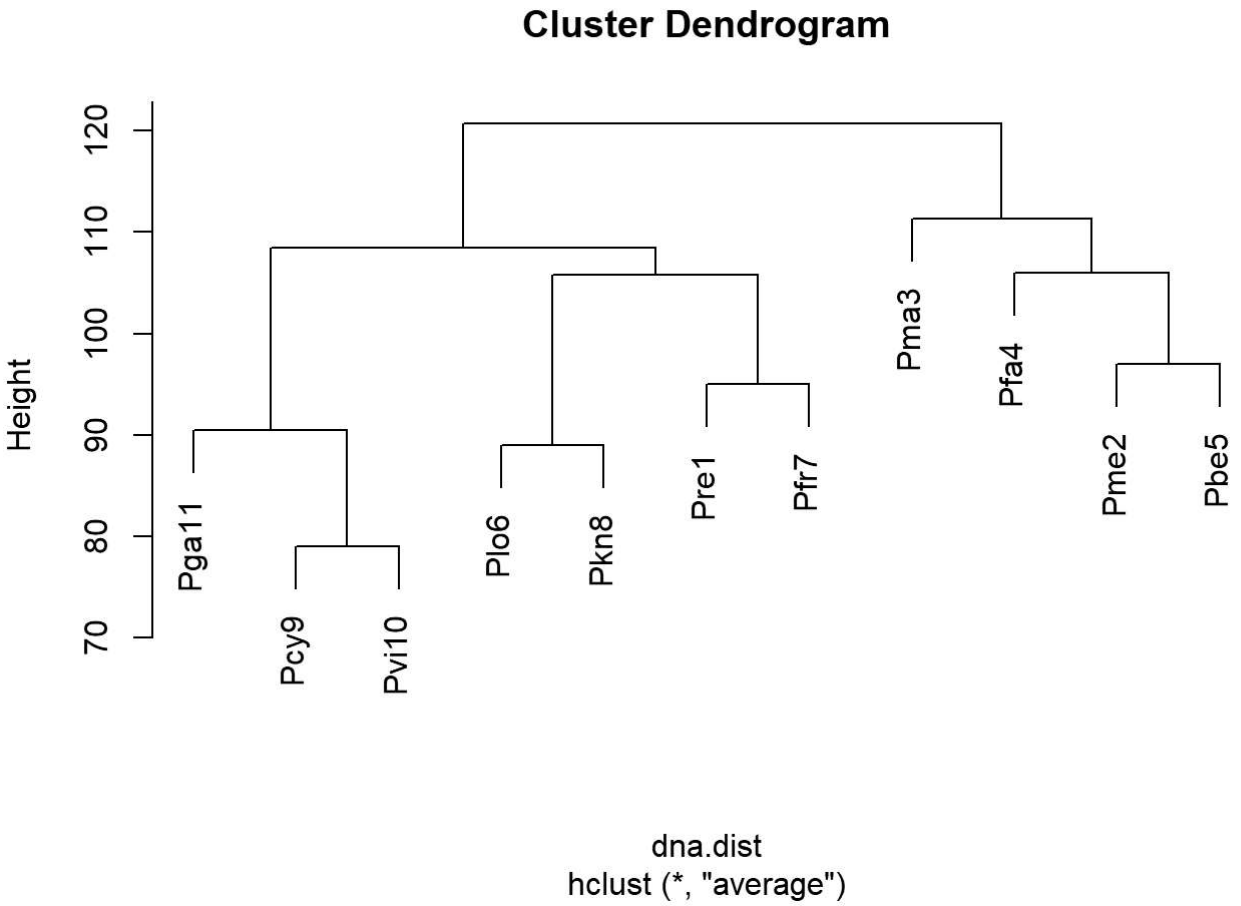
#2i: Note that we defined the Hamming distance ourselves. For two DNA sequences, x and y, the Hamming dista nce is the number of positions where x and y are different. Why can't we use the Euclidean distance for our data?

```
"We cannot apply to euclidean norm to this data because the data is not a group of points in an n-dimension
al space; it is categorical"
```

```
## [1] "We cannot apply to euclidean norm to this data because the data is not a group of points in an n-di
mensional space; it is categorical"
```

```
#2ii: Plot the UPGMA phylogenetic using dna.dist as the input distance matrix and average linkage.

upgma.tree <- hclust(dna.dist, method = "average")
plot(upgma.tree)
```




```
#2iii: Which two proteins are clustered in the first step of building the UPGMA tree?

row.names(dna.seq)[upgma.tree$order[1:2]]
```

```
## [1] "Pga11" "Pcy9"
```

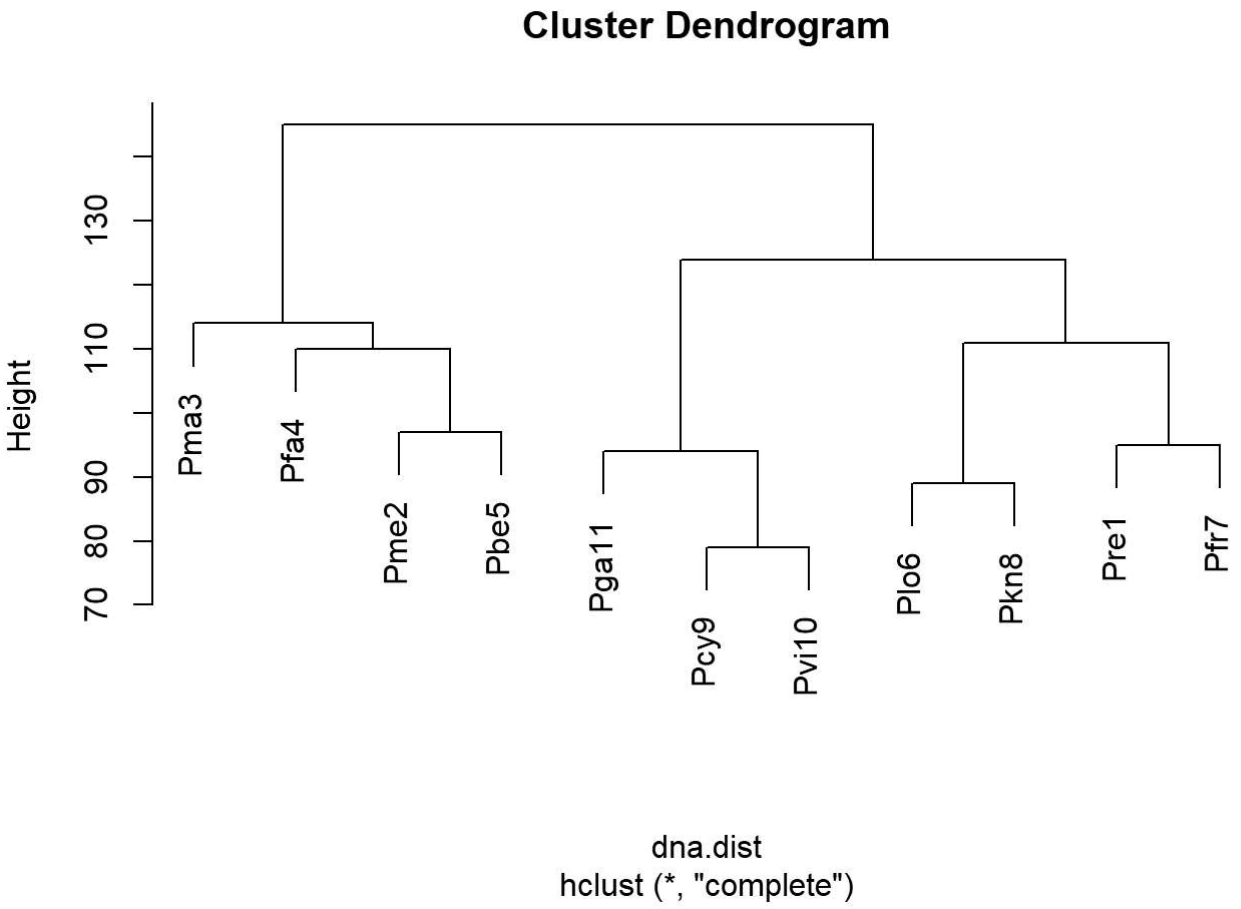
```
#2iv: What is the maximum node height in the UPGMA tree?

max(upgma.tree$height)
```

```
## [1] 120.6786
```

```
#2v: Plot the phylogenetic tree using dna.dist and complete linkage

upgma.tree1 <- hclust(dna.dist, method = "complete")
plot(upgma.tree1)
```



#2vi: For the tree built using complete linkage, observe that the two proteins connected at the first step are the same as those for the UPGMA tree. Why?

"The complete and average linkage are the same in the first step because each cluster has one sample and th
e max/average of each cluster are the same."

```
## [1] "The complete and average linkage are the same in the first step because each cluster has one sample
and the max/average of each cluster are the same."
```

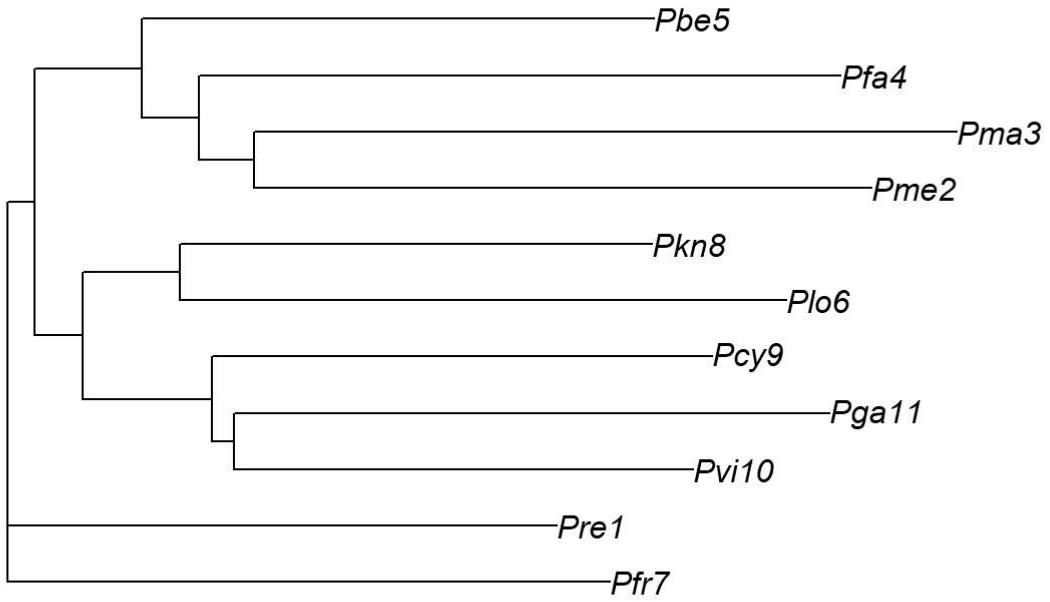
#2vii: The maximum node height for the tree built using complete linkage is obviously greater than that for the UPGMA tree. Why?

"Maximum node height for complete linkage is greater than the average linkage because the maximum distance
between clusters will almost always be greater than the average distance."

```
## [1] "Maximum node height for complete linkage is greater than the average linkage because the maximum di
stance between clusters will almost always be greater than the average distance."
```

#2viii: Draw the phylogenetic tree using neighbor joining. (Note: To use function nj, you need to first ins tall and load the package ape.)

```
library("ape")
plot(nj(dna.dist))
```




```
#Part 2

library("HSAUR2")
```

```
## Loading required package: tools
```

```
data <- USArrests
```

```
#Exercise 2.1: In our class we mentioned the use of correlation-based distance and Euclidean distance as di
ssimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalen
t. Assume each observation has been centered to have mean zero and standard deviation one, and let rij deno
te the correlation between the ith and jth observations. Then the quantity 1 – rij is proportional to the s
quared Euclidean distance between the ith and jth observations. Using the data, show that this proportional
ity holds.


summary(as.dist(1 -cor(t(USArrests))) / dist(USArrests)^2)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 6.420e-09 1.497e-06 3.941e-06 1.128e-05 1.025e-05 2.867e-04
```

```
#Exercise 2.2: Section 3.3 on page 65 gives a formula for calculating the proportion of the total variation (PTV) explained by the principal components. We also saw that the PTV can be obtained using the sdev output of the prcomp function. Calculate the PTV using these two approaches – they should deliver the same results.

pca <- prcomp(USArrests)
summary(pca)
```

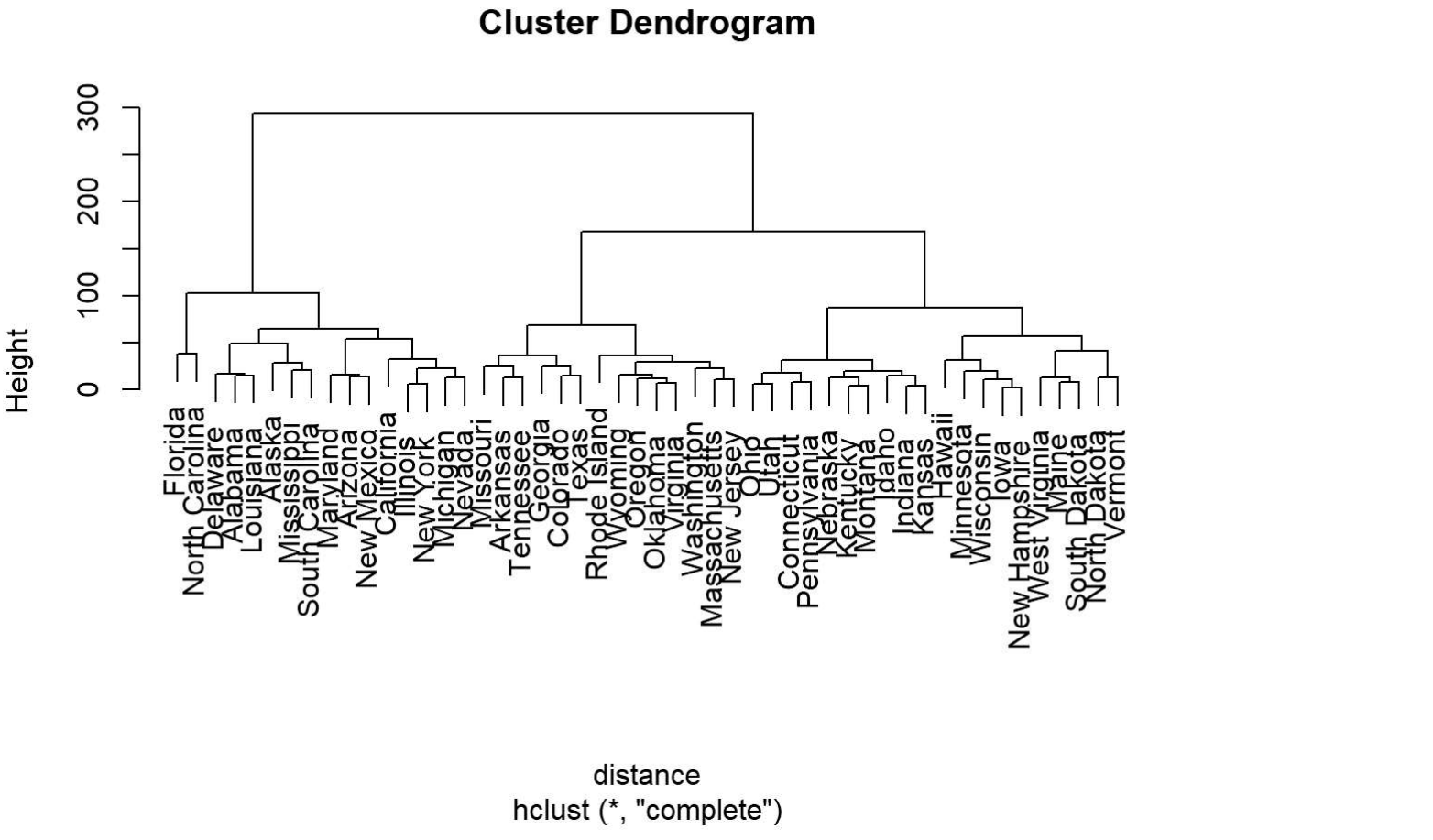
```
## Importance of components:
##                           PC1      PC2    PC3     PC4
## Standard deviation     83.7324 14.21240 6.4894 2.48279
## Proportion of Variance  0.9655  0.02782 0.0058 0.00085
## Cumulative Proportion   0.9655  0.99335 0.9991 1.00000
```

```
round(pca$sdev^2/sum(pca$sdev^2), digits=4)
```

```
## [1] 0.9655 0.0278 0.0058 0.0008
```

```
#Exercise 2.3 (i): Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

distance <- dist(USArrests, method = "euclidean")
state_hclust <- hclust(distance, method = "complete")
plot(state_hclust)
```



```
#Exercise 2.3 (ii):Cut the dendrogram at a height that results in three distinct clusters. Which states bel
ong to which clusters?

clusters <- cutree(state_hclust, k = 3)
state_names <- rownames(USArrests)

cat("Cluster 1:", state_names[clusters == 1], "\n", "\n")
```

```
## Cluster 1: Alabama Alaska Arizona California Delaware Florida Illinois Louisiana Maryland Michigan Missi
ssippi Nevada New Mexico New York North Carolina South Carolina
##
```

```
cat("Cluster 2:", state_names[clusters == 2], "\n", "\n")
```

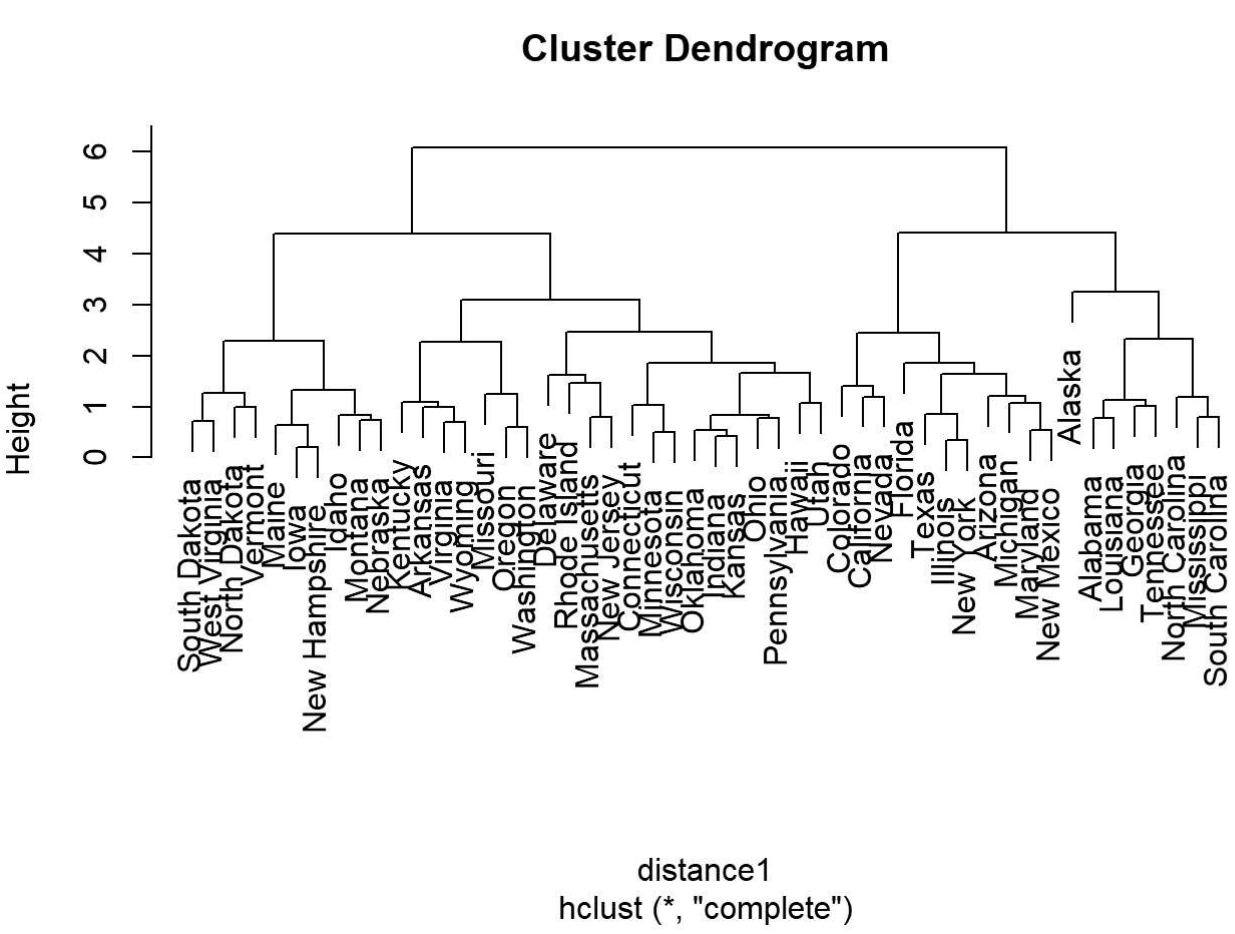
```
## Cluster 2: Arkansas Colorado Georgia Massachusetts Missouri New Jersey Oklahoma Oregon Rhode Island Tenn
essee Texas Virginia Washington Wyoming
##
```

```
cat("Cluster 3:", state_names[clusters == 3], "\n", "\n")
```

```
## Cluster 3: Connecticut Hawaii Idaho Indiana Iowa Kansas Kentucky Maine Minnesota Montana Nebraska New Ha
mpshire North Dakota Ohio Pennsylvania South Dakota Utah Vermont West Virginia Wisconsin
##
```

```
#Exercise 2.3 (iii): Hierarchically cluster the states using complete linkage and Euclidean distance, after
scaling the variables to have standard deviation one.

scaled_arrests <- scale(USArrests)
distance1 <- dist(scaled_arrests, method = "euclidean")
state_hclust_scaled <- hclust(distance1, method = "complete")
plot(state_hclust_scaled)
```



#Exercise 2.4 (iv): What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Pro vide a justification for your answer.

"Scaling changed the results of the hierarchical clustering and seemed to increase dissimilarity in the tre e. Scaling should be done before inter-observation dissimilarities are computed; especially when variables are in different units"

```
## [1] "Scaling changed the results of the hierarchical clustering and seemed to increase dissimilarity in
the tree. Scaling should be done before inter-observation dissimilarities are computed; especially when var
iables are in different units"
```

#Exercise 3.1:Do a correspondence analysis for the car-ratings. Explain how this table can be considered as a contingency table. The data are averaged ratings for 24 car types from a sample of 40 persons. The marks range from 1 (very good) to 6 (very bad)

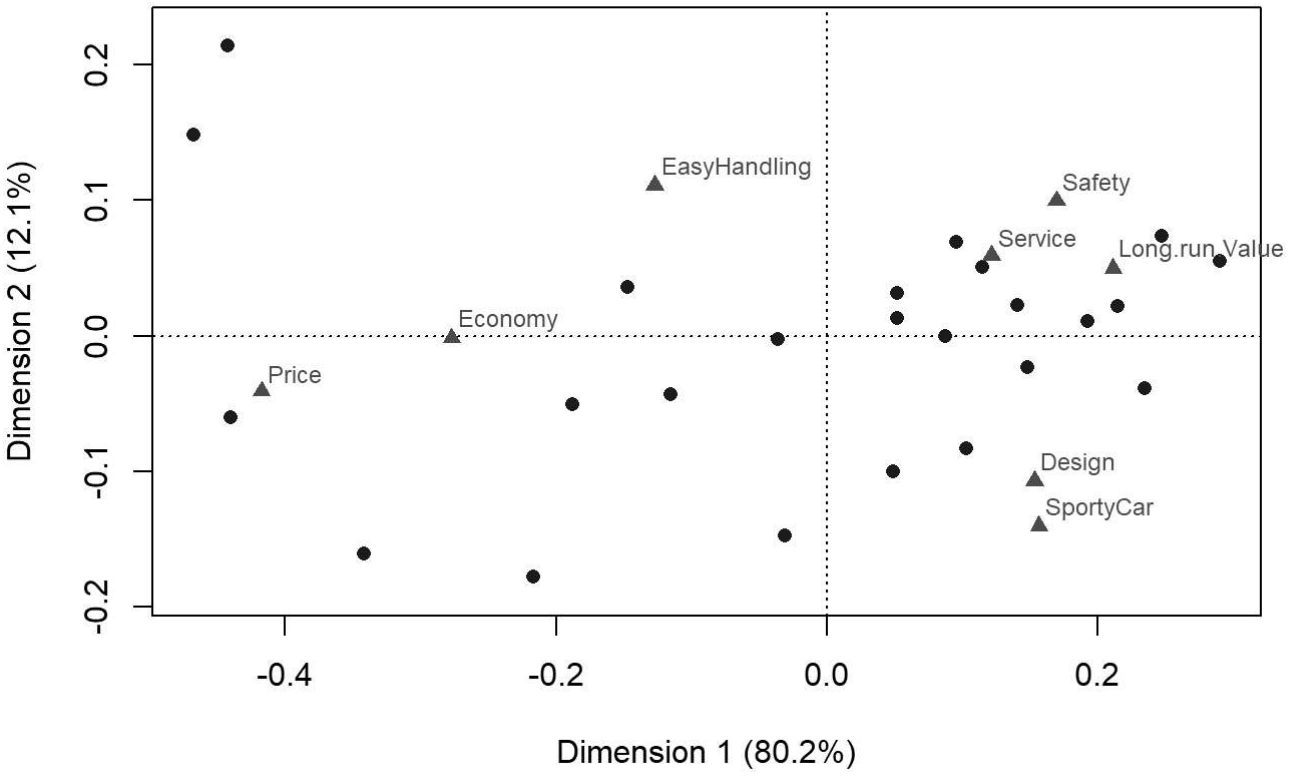
```
library(ca)

cars <- read.table("C:/Users/Jonathan/Downloads/cars.txt",header =T)
cars_ca <- ca(cars[ , -c(1:2)])

cars
```

```
##          Type   Model Economy Service Long.run.Value Price Design SportyCar
## 1        Audi     100     3.9     2.8            2.2   4.2    3.0       3.1
## 2         BMW 5series     4.8     1.6            1.9   5.0    2.0       2.5
## 3     Citroen      AX     3.0     3.8            3.8   2.7    4.0       4.4
## 4     Ferrari     N/A     5.3     2.9            2.2   5.9    1.7       1.1
## 5        Fiat     Uno     2.1     3.9            4.0   2.6    4.5       4.4
## 6        Ford  Fiesta     2.3     3.1            3.4   2.6    3.2       3.3
## 7     Hyundai     N/A     2.5     3.4            3.2   2.2    3.3       3.3
## 8      Jaguar     N/A     4.6     2.4            1.6   5.5    1.3       1.6
## 9        Lada  Samara     3.2     3.9            4.3   2.0    4.3       4.5
## 10      Mazda     323     2.6     3.3            3.7   2.8    3.7       3.0
## 11   Mercedes     200     4.1     1.7            1.8   4.6    2.4       3.2
## 12 Mitsubishi  Galant     3.2     2.9            3.2   3.5    3.1       3.1
## 13     Nissan   Sunny     2.6     3.3            3.9   2.1    3.5       3.9
## 14       Opel   Corsa     2.2     2.4            3.0   2.6    3.2       4.0
## 15       Opel  Vectra     3.1     2.6            2.3   3.6    2.8       2.9
## 16    Peugeot     306     2.9     3.5            3.6   2.8    3.2       3.8
## 17    Renault      19     2.7     3.3            3.4   3.0    3.1       3.4
## 18      Rover     N/A     3.9     2.8            2.6   4.0    2.6       3.0
## 19     Toyota Corolla     2.5     2.9            3.4   3.0    3.2       3.1
## 20      Volvo     N/A     3.8     2.3            1.9   4.2    3.1       3.6
## 21    Trabant     601     3.6     4.7            5.5   1.5    4.1       5.8
## 22         VW    Golf     2.4     2.1            2.0   2.6    3.2       3.1
## 23         VW  Passat     3.1     2.2            2.1   3.2    3.5       3.5
## 24   Wartburg     1.3     3.7     4.7            5.5   1.7    4.8       5.2
##    Safety EasyHandling
## 1     2.4          2.8
## 2     1.6          2.8
## 3     4.0          2.6
## 4     3.3          4.3
## 5     4.4          2.2
## 6     3.6          2.8
## 7     3.3          2.4
## 8     2.8          3.6
## 9     4.7          2.9
## 10    3.7          3.1
## 11    1.4          2.4
## 12    2.9          2.6
## 13    3.8          2.4
## 14    2.9          2.4
## 15    2.4          2.4
## 16    3.2          2.6
## 17    3.0          2.7
## 18    3.2          3.0
## 19    3.2          2.8
## 20    1.6          2.4
## 21    5.9          3.1
## 22    3.1          1.6
## 23    2.8          1.8
## 24    5.5          4.0
```

```
plot(cars_ca)
```




```
"The data can be considered a contigency table because car-ratings are displayed, each with a frequency of one (one score per category, per car)"
```

```
## [1] "The data can be considered a contigency table because car-ratings are displayed, each with a frequency of one (one score per category, per car)"
```

```
#Exercise 3.2: Write an R function to compute the chi-square statistic of independence. Test the null using for the bachelor data (file bachelors.txt). The data consists of observations of 202,100 bachelors from France and give the frequencies for different sets of modalities classified into regions. The variables (modalities) are: A=Philosophy-Letters, B=Economics and Social Sciences, C=Mathematics and Physics, D=Mathematics and Natural Sciences, E=Mathematics and Techniques, F=Industrial Techniques, G=Economic Techniques, H=Computer Techniques.

bachelors <- read.table("C:/Users/Jonathan/Downloads/bachelors.txt", header = T)

my_table <- bachelors[ , -c(1,2,11)]
chisq.test(my_table)
```

```
## Warning in chisq.test(my_table): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  my_table
## X-squared = 4354.5, df = 147, p-value < 2.2e-16
```

```
#Exercise 3.3: Do correspondence analysis of the U.S. crime data (file UScrime.txt), and determine the abso
lute contributions for the first three axes. How can you interpret the third axis? Try to identify the stat
es with one of the four regions to which it belongs. Do you think the four regions have a different behavio
r with respect to crime? This is a data set consisting of 50 measurements of 7 variables. It contains the n
umber of crimes in the 50 states of the U.S. classified according to 7 categories. Region is 1 for Northeas
t, 2 for Midwest, 3 for South and 4 for West. Division is 1 for New England, 2 for Mid Atlantic, 3 for E N
central, 4 for W N Central, 5 for S Atlantic, 6 for E S Central, 7 for W S Central, 8 for Mountain and 9 fo
r Pacific.

crime <- read.table("C:/Users/Jonathan/Downloads/uscrime.txt", header = T)
crime_ca <- ca(crime[ , -c(1)])

print(crime_ca)
```

```
##
##  Principal inertias (eigenvalues):
##            1        2        3        4        5        6        7
## Value      0.118156 0.044516 0.002318 0.001265 0.000834 0.000535 5.4e-05
## Percentage 70.46%   26.54%   1.38%    0.75%    0.5%     0.32%    0.03%
##            8       9     10
## Value      1.8e-05 7e-06 1e-06
## Percentage 0.01%   0%    0%
##
##
##  Rows:
##              [,1]     [,2]     [,3]      [,4]     [,5]     [,6]      [,7]
## Mass     0.010890 0.003603 0.003617  0.005360 0.001937 0.003507  0.021521
## ChiDist  0.157920 0.438801 0.528755  1.327417 2.939495 1.421319  0.708959
## Inertia  0.000272 0.000694 0.001011  0.009444 0.016733 0.007085  0.010817
## Dim. 1  -0.311850 1.030244 0.863911  3.659057 7.094324 4.093162  1.506395
## Dim. 2   0.529878 1.077179 1.763254 -1.397328 7.127634 0.536438 -2.282220
##              [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## Mass     0.005679  0.017507  0.016707  0.013507  0.021513  0.021952  0.019103
## ChiDist  1.470129  0.534824  0.483314  0.229103  0.356140  0.249830  0.068344
## Inertia  0.012273  0.005008  0.003903  0.000709  0.002729  0.001370  0.000089
## Dim. 1   3.930131  0.686241  1.054120  0.613604  0.692665  0.639808 -0.075194
## Dim. 2  -2.729852 -2.257559 -1.499865 -0.411002 -1.235422 -0.458069 -0.150815
##             [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
## Mass     0.027558  0.018514  0.023636  0.021944  0.023980  0.024471  0.026529
## ChiDist  0.131414  0.122182  0.054593  0.283916  0.278992  0.227717  0.181815
## Inertia  0.000476  0.000276  0.000070  0.001769  0.001867  0.001269  0.000877
## Dim. 1  -0.367369 -0.308152 -0.062174 -0.767172 -0.747342 -0.600943 -0.444885
## Dim. 2   0.129200  0.208718  0.099401  0.484425  0.514383  0.436562  0.460045
##            [,22]    [,23]     [,24]     [,25]     [,26]    [,27]     [,28]
## Mass    0.002082 0.005854  0.014956  0.008271  0.018484 0.011270  0.020226
## ChiDist 2.560835 1.107493  0.188149  0.044540  0.144133 0.176939  0.135319
## Inertia 0.013656 0.007180  0.000529  0.000016  0.000384 0.000353  0.000370
## Dim. 1  5.971004 3.096941  0.476572 -0.034062  0.127922 0.437591 -0.026655
## Dim. 2  7.178567 0.190489 -0.385515 -0.013086 -0.548071 0.130683 -0.420218
##             [,29]     [,30]     [,31]     [,32]     [,33]     [,34]     [,35]
## Mass     0.022621  0.013997  0.014803  0.017453  0.015494  0.017406  0.016678
## ChiDist  0.329689  0.051959  0.116283  0.075261  0.156021  0.142430  0.076965
## Inertia  0.002459  0.000038  0.000200  0.000099  0.000377  0.000353  0.000099
## Dim. 1   0.784888  0.087139  0.162391 -0.126020 -0.435331 -0.341888  0.154321
## Dim. 2  -0.856367 -0.109878 -0.414070 -0.126713 -0.077290  0.269901 -0.053730
##             [,36]     [,37]     [,38]     [,39]     [,40]     [,41]     [,42]
## Mass     0.022890  0.086236  0.045258  0.026040  0.030410  0.033816  0.038211
## ChiDist  0.127101  0.182644  0.304861  0.294389  0.285840  0.168740  0.253283
## Inertia  0.000370  0.002877  0.004206  0.002257  0.002485  0.000963  0.002451
## Dim. 1  -0.343813 -0.495038 -0.836237 -0.706506 -0.738701 -0.368806 -0.645335
## Dim. 2   0.214330 -0.305570  0.472870  0.782323  0.619248  0.519398  0.571402
##            [,43]     [,44]     [,45]     [,46]     [,47]     [,48]    [,49]
## Mass    0.003078  0.027244  0.035204  0.023245  0.031358  0.057462 0.003077
## ChiDist 2.619117  0.216294  0.248091  0.085228  0.184248  0.267794 1.471253
## Inertia 0.021114  0.001275  0.002167  0.000169  0.001065  0.004121 0.006660
## Dim. 1  7.141381 -0.465860 -0.502031 -0.000719 -0.421520  0.309312 3.162943
## Dim. 2  3.754501  0.672494  0.817350  0.350210  0.532497 -1.154487 4.570860
##            [,50]
## Mass    0.003843
## ChiDist 1.666510
```

```
## Inertia 0.010674
## Dim. 1  3.785313
## Dim. 2  4.749773
##
##
##  Columns:
##         land.area population   murder     rape   robbery  assault burglary
## Mass     0.876768   0.071181 0.000103 0.000234  0.001464 0.002033 0.013703
## ChiDist  0.127165   0.967131 1.172515 1.128963  1.451783 1.195651 1.237205
## Inertia  0.014178   0.066579 0.000142 0.000299  0.003086 0.002906 0.020975
## Dim. 1  -0.368080   2.224263 2.390253 2.536810  3.593965 2.759613 3.183746
## Dim. 2   0.059881  -2.806099 1.575093 1.960130 -0.113661 1.859560 2.508917
##          larcery auto.theft   region division
## Mass    0.028875   0.005521 0.000040 0.000077
## ChiDist 1.219946   1.725024 1.052018 1.095236
## Inertia 0.042974   0.016430 0.000044 0.000092
## Dim. 1  2.942435   4.325038 1.949949 1.749075
## Dim. 2  3.160287   3.082566 2.942707 3.020914
```

```
cat("\n\n","The contributions are the percentages relating to each eigenvalue. .7046, .2654 and .0138 for t
he first three axes, totaling .9838","\n\n")
```

```
##
##
##  The contributions are the percentages relating to each eigenvalue. .7046, .2654 and .0138 for the first
three axes, totaling .9838
```

```
cat("The third axis explains a very small amount. It may be worthwhile to only use the first two axes to ha
ve easily visualizable data","\n\n")
```

```
## The third axis explains a very small amount. It may be worthwhile to only use the first two axes to have
easily visualizable data
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:ape':
##
##     where
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
crime_byregion <- crime %>%
  group_by(region) %>%
  summarise(across(c(4:10), sum))

print(crime_byregion)
```

```
## # A tibble: 4 × 8
##   region murder  rape robbery assault burglary larcery auto.theft
##    <int>  <dbl> <dbl>   <dbl>   <int>    <int>   <int>      <int>
## 1      1   33.9  87.7    999.     837     8145   12146       4242
## 2      2   59.1 172.    1161.    1205     9229   22804       3915
## 3      3  168.  261.    1558.    2973    13056   25807       4731
## 4      4   81.6 259.    1160.    1756    15219   35435       5505
```

```
chisq.test(crime_byregion)
```

```
## Warning in chisq.test(crime_byregion): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  crime_byregion
## X-squared = 3146.2, df = 21, p-value < 2.2e-16
```

```
"I've grouped the crime counts by region and ran the chi-squared test. The p-value is very close to zero. W
e reject the null hypthesis; assume the rows and columns are dependent. Therefore, the regions are likely t
o have different behavior with respect to crime."
```

```
## [1] "I've grouped the crime counts by region and ran the chi-squared test. The p-value is very close to
zero. We reject the null hypthesis; assume the rows and columns are dependent. Therefore, the regions are l
ikely to have different behavior with respect to crime."
```

```
#Exercise 3.4: Consider the food data (file food.txt). Given that all of the variables are measured in the
same units (dollars), explain how this table can be considered as a contingency table. Perform a correspond
ence analysis and compare the results to those obtained with the PCA analysis of the correlation matrix. Th
e data set consists of the average expenditures on food for several different types of families (manual wor
kers = MA, employees = EM, managers = CA) with different numbers of children (2,3,4 or 5 children).

food <- read.table("C:/Users/Jonathan/Downloads/food.txt", header = T)
food
```

```
##    ID Workertype bread vegetables fruits meat poultry milk wine
## 1   1        MA2   332        428    354 1437     526  247  427
## 2   2        EM2   293        559    388 1527     567  239  258
## 3   3        CA2   372        767    562 1948     927  235  433
## 4   4        MA3   406        563    341 1507     544  324  407
## 5   5        EM3   386        608    396 1501     558  319  363
## 6   6        CA3   438        843    689 2345    1148  243  341
## 7   7        MA4   534        660    367 1620     638  414  407
## 8   8        EM4   460        699    484 1856     762  400  416
## 9   9        CA4   385        789    621 2366    1149  304  282
## 10 10        MA5   655        776    423 1848     759  495  486
## 11 11        EM5   584        995    548 2056     893  518  319
## 12 12        CA5   515       1097    887 2630    1167  561  284
```

```
cat("This is a contingency table because it ranks dollars sold in each food type against ID and all of the 'counts' are in the same units","\n\n")
```

```
## This is a contingency table because it ranks dollars sold in each food type against ID and all of the 'c
ounts' are in the same units
```

```
food_ca <- ca(food[ , -c(1,2)])
food_pca <- prcomp(cor(food[ ,-c(1,2)]))

summary(food_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4    PC5      PC6       PC7
## Standard deviation     1.1675 0.4922 0.09732 0.02498 0.0125 0.003256 6.521e-17
## Proportion of Variance 0.8437 0.1500 0.00586 0.00039 0.0001 0.000010 0.000e+00
## Cumulative Proportion  0.8437 0.9937 0.99951 0.99990 1.0000 1.000000 1.000e+00
```

```
print(food_ca)
```

```
##
##  Principal inertias (eigenvalues):
##            1        2        3        4        5        6
## Value      0.013928 0.005225 0.000997 0.000521 0.000298 0.000115
## Percentage 66.06%   24.78%   4.73%    2.47%    1.41%    0.55%
##
##
##  Rows:
##              [,1]     [,2]     [,3]      [,4]      [,5]     [,6]      [,7]
## Mass     0.061286 0.062593 0.085679  0.066857  0.067494 0.098799  0.075811
## ChiDist  0.181111 0.074172 0.130121  0.139561  0.089177 0.169863  0.157466
## Inertia  0.002010 0.000344 0.001451  0.001302  0.000537 0.002851  0.001880
## Dim. 1  -0.828075 0.350821 0.640596 -1.099452 -0.675910 1.339296 -1.295295
## Dim. 2   1.996719 0.218593 1.257648  0.637440  0.159762 0.642529 -0.332688
##              [,8]     [,9]     [,10]     [,11]     [,12]
## Mass     0.082951 0.096332  0.088914  0.096610  0.116673
## ChiDist  0.056667 0.177484  0.177569  0.128990  0.160258
## Inertia  0.000266 0.003035  0.002804  0.001607  0.002996
## Dim. 1  -0.431313 1.423997 -1.435921 -0.234348  0.924140
## Dim. 2   0.109725 0.241978 -0.412079 -1.681475 -1.446716
##
##
##  Columns:
##             bread vegetables    fruits     meat  poultry      milk      wine
## Mass     0.087575   0.143518  0.099012 0.369921 0.157471  0.070239  0.072265
## ChiDist  0.200073   0.083077  0.142744 0.050997 0.133191  0.246872  0.300675
## Inertia  0.003506   0.000991  0.002017 0.000962 0.002793  0.004281  0.006533
## Dim. 1  -1.577401   0.063275  1.072840 0.309974 1.037322 -1.588950 -1.986725
## Dim. 2  -0.604676  -0.933605 -0.173342 0.350648 0.230157 -2.099050  2.568132
```

```
#Exercise 7.1:

library(lavaan)
```

```
## This is lavaan 0.6-16
## lavaan is FREE software! Please report any bugs.
```

```
entries <- c(1,-.04,.61,.45,.03,-.29,-.30,.45,.30,
             0,1,-.07,-.12,.49,.43,.30,-.31,-.17,
             0,0,1,.59,.03,-.13,-.24,.59,.32,
             0,0,0,1,-.08,-.21,-.19,.63,.37,
             0,0,0,0,1,.47,.41,-.14,-.24,
             0,0,0,0,0,1,.63,-.13,-.15,
             0,0,0,0,0,0,1,-.26,-.29,
             0,0,0,0,0,0,0,1,.40,
             0,0,0,0,0,0,0,0,1
             )

pain <- matrix(entries, nrow=9, ncol=9)

model_txt = '
  f1 =~ Q1 + Q3 + Q4 + Q8
  f2 =~ Q2 + Q5 + Q6 + Q7
  f1 ~~ f2
'

colnames(pain) <- c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6","Q7","Q8","Q9")
rownames(pain) <- c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6","Q7","Q8","Q9")
pain
```

```
##       Q1    Q2    Q3    Q4    Q5    Q6    Q7  Q8 Q9
## Q1  1.00  0.00  0.00  0.00  0.00  0.00  0.00 0.0  0
## Q2 -0.04  1.00  0.00  0.00  0.00  0.00  0.00 0.0  0
## Q3  0.61 -0.07  1.00  0.00  0.00  0.00  0.00 0.0  0
## Q4  0.45 -0.12  0.59  1.00  0.00  0.00  0.00 0.0  0
## Q5  0.03  0.49  0.03 -0.08  1.00  0.00  0.00 0.0  0
## Q6 -0.29  0.43 -0.13 -0.21  0.47  1.00  0.00 0.0  0
## Q7 -0.30  0.30 -0.24 -0.19  0.41  0.63  1.00 0.0  0
## Q8  0.45 -0.31  0.59  0.63 -0.14 -0.13 -0.26 1.0  0
## Q9  0.30 -0.17  0.32  0.37 -0.24 -0.15 -0.29 0.4  1
```

```
cfa_mod = sem(model_txt, sample.cov=pain, sample.nobs=123)

summary(cfa_mod)
```

```
## lavaan 0.6.16 ended normally after 28 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        17
##
##   Number of observations                           123
##
## Model Test User Model:
##
##   Test statistic                                63.749
##   Degrees of freedom                                19
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   f1 =~
##     Q1                1.000
##     Q3                1.209    0.169    7.158    0.000
##     Q4                1.131    0.165    6.873    0.000
##     Q8                1.134    0.165    6.884    0.000
##   f2 =~
##     Q2                1.000
##     Q5                1.116    0.240    4.649    0.000
##     Q6                1.572    0.296    5.304    0.000
##     Q7                1.381    0.266    5.198    0.000
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   f1 ~~
##     f2               -0.107    0.044   -2.409    0.016
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.551    0.083    6.640    0.000
##    .Q3                0.347    0.070    4.990    0.000
##    .Q4                0.427    0.073    5.812    0.000
##    .Q8                0.425    0.073    5.790    0.000
##    .Q2                0.714    0.100    7.114    0.000
##    .Q5                0.645    0.095    6.816    0.000
##    .Q6                0.305    0.087    3.521    0.000
##    .Q7                0.461    0.086    5.394    0.000
##     f1                0.441    0.114    3.868    0.000
##     f2                0.278    0.097    2.854    0.004
```

```r
#Exercise 7.2:

library(lavaan)

entries3 <- c(0,0,0,0,0,100,
           0,0,0,0,100,54,
           0,0,0,100,-29,-35,
           0,0,100,67,-28,-37,
           0,100,44,56,-30,-36,
           100,66,52,47,-29,-41)

alien <- matrix(entries3, nrow=6, ncol=6) / 100
alien <- alien[,c(6:1)]

colnames(alien) <- c("Educ", "SEI", "Anomia71", "Powles71", "Anomia67", "Powles67")
rownames(alien) <- c("Powles67", "Anomia67", "Powles71", "Anomia71", "SEI", "Educ")

alien
```

```
##           Educ   SEI Anomia71 Powles71 Anomia67 Powles67
## Powles67  1.00  0.00     0.00     0.00     0.00        0
## Anomia67  0.66  1.00     0.00     0.00     0.00        0
## Powles71  0.52  0.44     1.00     0.00     0.00        0
## Anomia71  0.47  0.56     0.67     1.00     0.00        0
## SEI      -0.29 -0.30    -0.28    -0.29     1.00        0
## Educ     -0.41 -0.36    -0.37    -0.35     0.54        1
```

```r
model2 <- '
  SES =~ Educ + SEI
  Alienation67 =~ Anomia67 + Powles67
  Alienation71 =~ Anomia71 + Powles71

  Alienation67 ~ SES
  Alienation71 ~ SES + Alienation67

  Anomia67 ~~ Anomia71
'

cfa_mod2 = sem(model2, sample.cov=alien, sample.nobs=932)

summary(cfa_mod2)
```

```
## lavaan 0.6.16 ended normally after 28 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        16
##
##   Number of observations                           932
##
## Model Test User Model:
##
##   Test statistic                                 6.390
##   Degrees of freedom                                 5
##   P-value (Chi-square)                           0.270
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SES =~
##     Educ              1.000
##     SEI               0.754    0.062   12.261    0.000
##   Alienation67 =~
##     Anomia67          1.000
##     Powles67          1.154    0.060   19.332    0.000
##   Alienation71 =~
##     Anomia71          1.000
##     Powles71          1.088    0.055   19.660    0.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Alienation67 ~
##     SES              -0.495    0.048  -10.299    0.000
##   Alienation71 ~
##     SES              -0.185    0.043   -4.297    0.000
##     Alienation67      0.600    0.048   12.428    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .Anomia67 ~~
##    .Anomia71          0.155    0.020    7.870    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Educ              0.284    0.054    5.295    0.000
##    .SEI               0.592    0.040   14.676    0.000
##    .Anomia67          0.428    0.031   13.658    0.000
##    .Powles67          0.236    0.034    6.972    0.000
##    .Anomia71          0.383    0.031   12.180    0.000
##    .Powles71          0.268    0.033    8.142    0.000
##     SES               0.715    0.068   10.463    0.000
```

```
##    .Alienation67      0.397    0.037   10.870    0.000
##    .Alienation71      0.308    0.027   11.263    0.000
```

```
#Exercise 7.3:

library(lavaan)

entries1 <- c(1,.37,.42,.53,.38,.81,.35,.42,.40,.24,
              0,1,.33,.14,.10,.34,.65,.32,.14,.15,
              0,0,1,.38,.20,.49,.20,.75,.39,.17,
              0,0,0,1,.24,.58,-.04,.46,.73,.15,
              0,0,0,0,1,.32,.11,.26,.19,.43,
              0,0,0,0,0,1,.34,.46,.55,.24,
              0,0,0,0,0,0,1,.18,.06,.15,
              0,0,0,0,0,0,0,1,.54,.2,
              0,0,0,0,0,0,0,0,1,.16,
              0,0,0,0,0,0,0,0,0,1
             )

mental <- matrix(entries1, nrow=10, ncol=10)

colnames(mental) <- c("V1", "S1", "R1", "N1", "W1", "V2", "S2", "R2", "N2", "W2")
rownames(mental) <- c("V1", "S1", "R1", "N1", "W1", "V2", "S2", "R2", "N2", "W2")

model_txt1 = '
  f1 =~ V1 + S1 + R1 + N1 + W1
  f2 =~ V2 + S2 + R2 + N2 + W2
  f1 ~~ f2
'

mental
```

```
##      V1   S1   R1    N1   W1   V2   S2   R2   N2 W2
## V1 1.00 0.00 0.00  0.00 0.00 0.00 0.00 0.00 0.00  0
## S1 0.37 1.00 0.00  0.00 0.00 0.00 0.00 0.00 0.00  0
## R1 0.42 0.33 1.00  0.00 0.00 0.00 0.00 0.00 0.00  0
## N1 0.53 0.14 0.38  1.00 0.00 0.00 0.00 0.00 0.00  0
## W1 0.38 0.10 0.20  0.24 1.00 0.00 0.00 0.00 0.00  0
## V2 0.81 0.34 0.49  0.58 0.32 1.00 0.00 0.00 0.00  0
## S2 0.35 0.65 0.20 -0.04 0.11 0.34 1.00 0.00 0.00  0
## R2 0.42 0.32 0.75  0.46 0.26 0.46 0.18 1.00 0.00  0
## N2 0.40 0.14 0.39  0.73 0.19 0.55 0.06 0.54 1.00  0
## W2 0.24 0.15 0.17  0.15 0.43 0.24 0.15 0.20 0.16  1
```

```
cfa_mod1 = sem(model_txt1, sample.cov=mental, sample.nobs=123)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                 is not positive definite;
##                 use lavInspect(fit, "cov.lv") to investigate.
```

```
summary(cfa_mod1)
```

```
## lavaan 0.6.16 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        21
##
##   Number of observations                           123
##
## Model Test User Model:
##
##   Test statistic                               236.862
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   f1 =~
##     V1                1.000
##     S1                0.523    0.116    4.516    0.000
##     R1                0.827    0.112    7.392    0.000
##     N1                0.871    0.111    7.840    0.000
##     W1                0.466    0.116    4.011    0.000
##   f2 =~
##     V2                1.000
##     S2                0.390    0.107    3.649    0.000
##     R2                0.790    0.096    8.204    0.000
##     N2                0.735    0.098    7.485    0.000
##     W2                0.351    0.108    3.263    0.001
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   f1 ~~
##     f2                0.706    0.109    6.500    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .V1                0.419    0.060    6.984    0.000
##    .S1                0.835    0.107    7.830    0.000
##    .R1                0.600    0.079    7.629    0.000
##    .N1                0.557    0.074    7.538    0.000
##    .W1                0.867    0.111    7.836    0.000
##    .V2                0.305    0.051    5.952    0.000
##    .S2                0.887    0.114    7.806    0.000
##    .R2                0.564    0.076    7.459    0.000
##    .N2                0.621    0.082    7.570    0.000
##    .W2                0.907    0.116    7.814    0.000
##     f1                0.573    0.118    4.860    0.000
##     f2                0.686    0.125    5.497    0.000
```