R course exercises Kiefer Schmidt WS 2018/19 part 2

Exercises using the chi-square test

Exercise 1:

A chi-square test can be used to test based on contingency tables if categorial variables are significantly correlated. The dataset Cars93 implemented in the R-package „MASS“ will serve as an example. If not already installed please install the package „MASS“. Afterwards open the dataset Cars93 and safe it under the varable „autos“.

First check the structure of the dataset using str(). The dataset contains a number of variables at different factor levels, which can serve as categorial variables.

Use a chi-square test to test, if the „Type“ of the cars is significantly correlated with „Airbags“. To do this, first generate a dataframe containing „Type“ and „AirBags“ before generating a table using the function table().

```
> library(MASS)
> autos <- Cars93
> str(autos)
'data.frame':   93 obs. of  27 variables:
 $ Manufacturer      : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4 4 4 5 ...
 $ Model             : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1 6 24 54 74 73 35 ...
 $ Type              : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3 2 ...
 $ Min.Price         : num  12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33 ...
 $ Price             : num  15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7 ...
 $ Max.Price         : num  18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36.3 ...
 $ MPG.city          : int  25 18 20 19 22 22 19 16 19 16 ...
 $ MPG.highway       : int  31 25 26 26 30 31 28 25 27 25 ...
 $ AirBags           : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1 2 2 2 2 2 2 ...
 $ DriveTrain        : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3 2 2 ...
 $ Cylinders         : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4 4 4 5 ...
 $ EngineSize        : num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
 $ Horsepower        : int  140 200 172 172 208 110 170 180 170 200 ...
 $ RPM               : int  6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 ...
 $ Rev.per.mile      : int  2890 2335 2280 2535 2545 2565 1570 1320 1690 1510 ...
```

```
 $ Man.trans.avail   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...
 $ Fuel.tank.capacity: num  13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...
 $ Passengers        : int  5 5 5 6 4 6 6 6 5 6 ...
 $ Length            : int  177 195 180 193 186 189 200 216 198 206 ...
 $ Wheelbase         : int  102 115 102 106 109 105 111 116 108 114 ...
 $ Width             : int  68 71 67 70 69 69 74 78 73 73 ...
 $ Turn.circle       : int  37 38 37 37 39 41 42 45 41 43 ...
 $ Rear.seat.room    : num  26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...
 $ Luggage.room      : int  11 15 14 17 13 16 17 21 14 18 ...
 $ Weight            : int  2705 3560 3375 3405 3640 2880 3470 4105 3495 3620 ...
 $ Origin            : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ...
 $ Make              : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3 5 6 7 9 8 10 ...
```

```
> autoframe <- data.frame(autos$AirBags, autos$Type)
> autotable <- table(autos$AirBags, autos$Type)
> autotable

                  Compact Large Midsize Small Sporty Van
 Driver & Passenger       2     4       7     0      3   0
 Driver only              9     7      11     5      8   3
 None                     5     0       4    16      3   6
>
```

> chisq.test(autotable)

Pearson's Chi-squared test

data: autotable

X-squared = 33.001, df = 10, p-value = 0.0002723

Warnmeldung:

In chisq.test(autotable) : Chi-Quadrat-Approximation kann inkorrekt sein

Exercise for the principal component analysis

Exercise 1:

Load the „wine“ dataset using

```
wine <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data", sep=",")
```

, Quelle https://archive.ics.uci.edu/ml/datasets/Wine.

First name the columns using the function below. Cvs in the first column is used to declare the classes. Extract the columns for Cvs, alcohol, and proanthocyanins. On this reduced dataset conduct a PCA. Show the results in form of a PCA plot.

Naming the columns

> colnames(wine) <- c("Cvs","Alcohol","Malic acid","Ash","Alcalinity of ash", "Magnesium", "Total phenols", "Flavanoids", "Nonflavanoid_phenols", "Proanthocyanins", "Color_intensity", "Hue", "OD280/OD315_of_diluted_wines", "Proline")

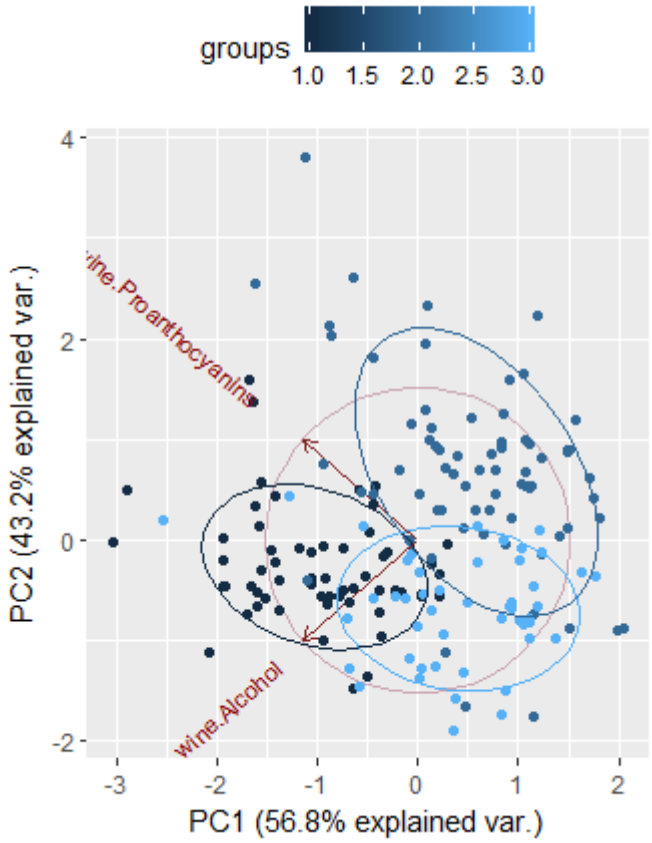
>

The first columns indicates the classes

> wineClasses <- factor(wine$Cvs)

> wine_subset <- data.frame(wine$Cvs, wine$Alcohol, wine$Proanthocyanins)

```
# Extract the subset of the data for the PCA
> wine_subset
   wine.Cvs wine.Alcohol wine.Proanthocyanins
1         1       14.23                 2.29
2         1       13.20                 1.28
3         1       13.16                 2.81
4         1       14.37                 2.18

# PCA...
> winepca <− prcomp( wine_subset[, c(2, 3)] , center=T, scale=T)
> summary(winepca)
Importance of components:
                          PC1    PC2
Standard deviation     1.0662 0.9291
Proportion of Variance 0.5684 0.4316
Cumulative Proportion  0.5684 1.0000

# and finally do the plot
> cvs <- wine[,1]
> winevis <− ggbiplot ( winepca , obs.scale = 1 , var.scale = 1 , groups = cvs , ellipse  = T, circle = T)
> winevis <− winevis + theme(legend.direction = "horizontal" , legend.position = "top")
> print(winevis)
```



Exercises to diverse packages

Exercise 1:

In the following sequences count the frequence of „tata“. Also indicate the length of the given sequences.

> sequenzen <- c("tcctctctctatcttatactctttatag", "tttgcataaaaaaaatatagaaa", "aaaagaaactggatagtgaatttgctatg", "taagtgtaaaagttataca")

> sequenzen

[1] "tcctctctctatcttatactctttatag" "tttgcataaaaaaaatatagaaa" "aaaagaaactggatagtgaatttgctatg"

[4] "taagtgtaaaagttataca"

> str_count(sequenzen, "tata")

[1] 2 1 0 1

> str_length(sequenzen)

[1] 28 23 29 19

Exercise 2:

Plot a heatmap of the following gene expression data using the package plotly. Use the function ly().

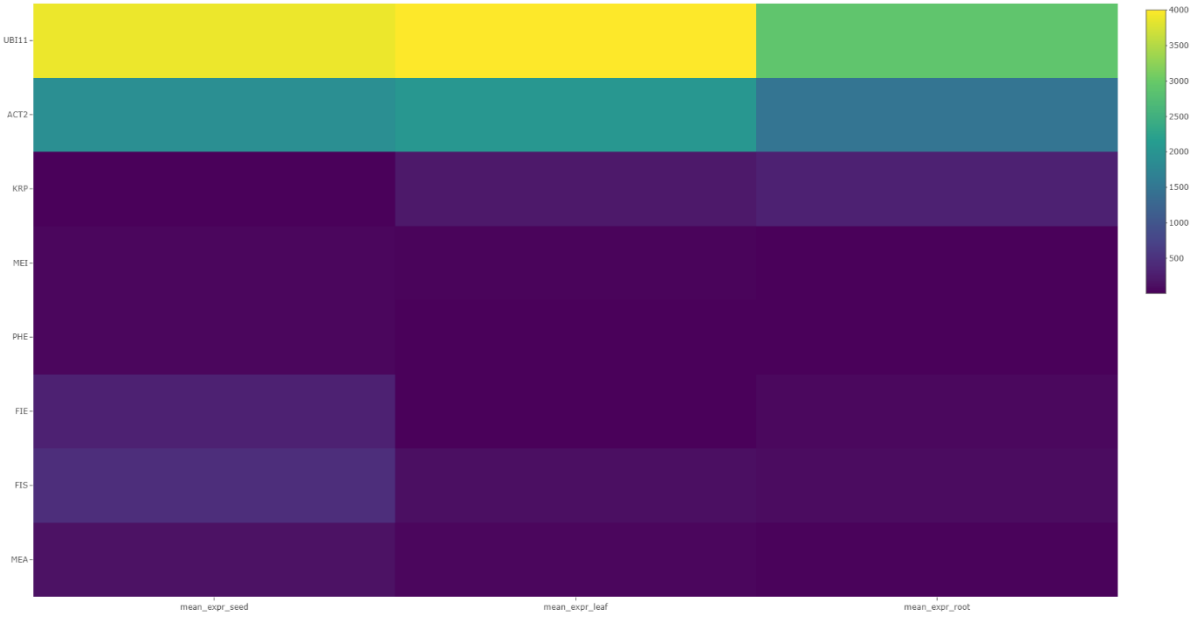
	mean_expr_seed	mean_expr_leaf	mean_expr_root
MEA	130	45	13
FIS	456	103	89
FIE	307	3	50
PHE	40	5	3
MEI	39	20	5
KRP	2	198	298
ACT2	1898	2044	1478
UBI11	3890	4002	2900

Generation of the data as matrix

```
> rownames <- c("MEA", "FIS", "FIE", "PHE", "MEI", "KRP", "ACT2", "UBI11")

> colnames <- c("mean_expr_seed", "mean_expr_leaf", "mean_expr_root")

> data <- matrix(c(130, 456, 307, 40, 39, 2, 1898, 3890, 45, 103, 3, 5, 20, 198, 2044, 4002, 13, 89, 50, 3, 5, 298, 1478, 2900), ncol = 3, dimnames = list(rownames, colnames))
```

doing the plot

```
> p <- plot_ly(x = colnames, y = rownames, z = data, type = "heatmap")

> p
```


UBI11
ACT2
KRP
MEI
PHE
FIE
FIS
MEA
mean_expr_seed
mean_expr_leaf
mean_expr_root
4000
3500
3000
2500
2000
1500
1000
500