Prof. Dr. M. Kateri,
Dr. W. Herff,
L. Kaufmann, M.Sc.

RWTH Aachen, SS 2022
Release date: April 18th, 2022
Solution: Week starting from April 25th, 2022

Applied Data Analysis

R-Laboratory 3

Central Limit Theorem – Simple Linear Models

Useful packages and functions:

- `table()`
- `barplot()`
- `sprintf()`
- `rbinom()`
- `title()`
- `axis()`
- `dplyr`
- `dplyr::mutate()`
- `pairs()`
- `abline()`
- `lm()`
- `MASS`
- `MASS::ginv()`
- `predict()`
- `poly()`
- `I()`
- `qf()`

Task 9

(a) Draw a random sample of size $m = 30$ from a $\mathcal{B}(n, p)$-distribution, the Binomial distribution with parameter $n = 12$ and $p = 0.7$, applying the R-function `rbinom`.

(b) Construct the bar plot and add the probability mass function (pmf) of the generating distribution.

(c) Calculate the mean ($\bar{x}$) and the variance (s^2) of your sample and write them in the title of the figure. Furthermore, calculate

$$T_{m,n,p} := \sqrt{m}\left(\frac{\bar{x} - np}{\sqrt{np(1-p)}}\right)$$

directing the output to the console.

(d) Write a function with arguments m, n and p which draws a new random sample of size m from a $\mathcal{B}(n, p)$-distribution and returns the value of $T_{m,n,p}$.

(e) Apply the function from (d) 10,000 times for $m \in \{5, 30, 500\}$, with $n = 12$ and $p = 0.7$. For each m, create a histogram with 16 breaks for the returned values. What do you observe?

Task 10

(a) Download the CSV-file *Solar.csv* from the RWTHmoodle space of the course Applied Data Analysis (Tutorial/Praktikum, no 11.04012). Import the data as a `data.frame` object into the R workspace and transform the attribute `batch` to type `factor`.

(b) Create a scatterplot matrix of the attributes `Pmax`, `Imax`, `Umax`, `Isc` and `Uoc`. Differentiate the points by `batch` using colors.

(c) Create Box-plots for `Uoc` for each batch in one figure.

(d) For the data of *Solar.csv*, create an (`Pmax`, `Isc`) scatterplot. Differentiate the points by `batch` using colors and add a linear regression line. Compute the parameter vector

   (i) via Example I.4.6 and Theorem I.4.9 of the lecture,

   (ii) via the function `lm`.

   Hint: `lm` needs an argument `formula`. An object of class `formula` takes the form "*response*∼*terms*", where *terms* describes the predictors for *response*. The intercept of a linear model is given as default. If there is no intercept in the model you need to add "-1" to *terms*. The `formula Isc`∼`Pmax` describes the simple linear regression model above.

(e) Add corresponding colored regression lines based on the observations from batch 1 and batch 4.

(f) Predict the missing values of `Isc` based on the regression in (d).

(g) Save the `data.frame` into an `.RData` file.

Task 11

(a) Download the CSV-file *rent.csv* from RWTHmoodle. Import the data as a `data.frame` object into the R workspace.

(b) Create a scatterplot of the attributes `rent.sqm` (y-axis) and `space` (x-axis). Add a linear regression line (you may use the function `lm`) to the scatterplot. Does the linear regression describes the data well? Is there a transformation of one of the two variables which possibly allows the creation of a better fitting linear model?
Hint: Create a scatterplot of `rent.sqm` (y-axis) and `1/space` (x-axis).

(c) Create a regression model with the approach

$$\texttt{rent.sqm} = a + \frac{b}{\texttt{space}}$$

for real valued parameters $a, b \in \mathbb{R}$ (it is a linear model in the parameters). Add the regression curve to the first scatterplot in (b). Does this model provide a better description of the relation between `rent.sqm` and `space` than the simple linear regression of (b) based on your visual impression?
Hint: You can add the term $\frac{b}{\texttt{space}}$ to the formula of linear model by adding `I(1/space)` to the argument `formula` of `lm`.

Task 12

(a) Download the white-space-separated file *cars2.dat* from RWTHmoodle. Import the data as a `data.frame` object into the R workspace.

(b) Create a scatterplot of the attributes `dist` (y-axis) and `speed` (x-axis) of the `cars2` data set.

(c) Add a quadratic regression curve to the scatterplot by using a linear model with the approach

$$\texttt{dist} = a + b \cdot \texttt{speed} + c \cdot \texttt{speed}^2 \qquad (+)$$

for real valued parameters $a, b, c \in \mathbb{R}$ (it is linear in the parameters).
Hint: You can add the term $c \cdot \texttt{speed}^2$ to the formula of linear model by adding `I(speed^2)` to the argument `formula` of `lm`. Alternatively, you can use the function `poly` to create a polynomial predictor for a linear model. In the latter case, it is recommended to compute the points for the regression curve using the function `predict`.

(d) Test the hypotheses

$$\mathrm{H}_0 \colon c = 0 \qquad \text{versus} \qquad \mathrm{H}_1 \colon c \neq 0$$

on the significance level $\alpha = 0.05$ for the parameter c of the linear model with the approach $(+)$ via the F-test of Testing procedure I.4.40 of the lecture. Consider the conditions of the F-test to be satisfied. Does the test reject the null hypothesis?

```
######################
#########TASK9########
######################

#a) get the random sample
rbinom_task9 = rbinom(30, 12, 0.7)

#b) construct the bar plot
barplot(table(rbinom_task9))

#c) calculate mean and variance of the sample
# and write them in the figure

T.mnp = sqrt(30)*((mean(rbinom_task9) - 12*0.7)/sqrt(12*0.7*0.3))

#d)
Tmnp <- function(m, n, p){

  return(sqrt(m)* (mean(rbinom(m, n, p)) - n*p )/sqrt(n*p*(1-p)))

}

# (e)
k=10000
m.vec=c(5,30,500)
size=12
p = 0.7
z = seq(-4,4,0.01) #for the plot of dnorm
for(m in m.vec){
  T.vec=c()
  for(i in 1:k){
    T.vec=c(T.vec,calc.T(m,n,p))
  }
  hist(T.vec,nclass=16,freq=FALSE) #histogram with 16 breaks
  lines(z,dnorm(z),col="red",lty=3) #add density of standard normal distribution on the interval from -4 to
}
```

```
# Task10 from b)
#b) scatter plot for multiple columns
pairs(~Pmax+Imax+Umax+Isc+Uoc,
      data=solar_task10,
      col = solar_task10$batch,
      main="Solar scatter plot")
#c)
ggplot(solar_task10, aes(batch, Uoc, color = batch)) + geom_boxplot(outlier.colour="red",
outlier.shape=8,outlier.size=4)
#d)
pairs(~Pmax+Isc,
      data=solar_task10,
      col = solar_task10$batch,
      main="Solar scatter plot Pmax and Isc")
#i) compute the parameter by using I.4.6 and I.4.9
# Isc~Pmax
# insert the vector of ones
new_pmax = cbind(rep(1, nrow(solar_task10)), solar_task10$Pmax)

# achieve the parameter manually
parameter_manual = ginv(new_pmax) %*% solar_task10$Isc[!is.na(solar_task10$Isc)]

# to ensure, there is no NA records
X = solar_task10$Pmax[!is.na(solar_task10$Isc)]
Y = solar_task10$Isc[!is.na(solar_task10$Isc)]

# ii)
# fit the linear model
fit = lm(Y ~ X)
plot(solar_task10$Pmax,solar_task10$Isc,col=c("red","blue","green","orange")[solar_task10$batch])
reg.par.lm = fit$coefficients
#
# > fit$coefficients
# (Intercept)        Pmax
# 4.49334242  0.03713745

abline(fit, col = "orange")

Pmax.df = data.frame(solar_task10$Pmax)
predicted_Isc = predict(fit, newdata = solar_task10)
batch4 = which(solar_task10$batch == 4)
batch1 = which(solar_task10$batch == 1)

fit.batch1 = lm(solar_task10$Isc[batch1] ~ solar_task10$Pmax[batch1], data = solar_task10)

abline(fit.batch1$coefficients, col = "blue")

abline(lm(solar_task10$Isc[batch4] ~ solar_task10$Pmax[batch4], data = solar_task10), col = "red")

# e) predict the regression of missing values in Isc
Isc_NA = which(is.na(solar_task10$Isc) == TRUE)
solar_task10$Isc[Isc_NA] = predicted_Isc[predicted_Isc]
```

```
#######################
#########TASK11########
#######################


#a)
rent_task11 = read.csv2("R-Lab-Datasets/rent.csv", header = TRUE, sep = ";")

#b)
plot(rent_task11$space, rent_task11$rent.sqm)
lm.rent = lm(rent.sqm ~ space , data = rent_task11)
abline(lm.rent, col = "red")

#c)
plot(1/rent_task11$space,rent_task11$rent.sqm)
lm.rent.2 = lm(rent.sqm ~ 1/space , data = rent_task11)
abline(lm.rent.2, col = "blue")
```

```
#######################
#########TASK12########
#######################

#a)
cars_task12 = read.table("R-Lab-Datasets/cars2.dat", header = TRUE, sep = " ")

#b)
plot(cars_task12$speed, cars_task12$dist)

#c)
speed = cars_task12$speed
dist = cars_task12$dist

lm.cars2.qd = lm(dist ~ poly(speed,2))

plot(speed, dist)

speed.qd = (cars_task12$speed)^2

cars_task12$speed2 = speed.qd

speed_secon = seq(min(speed), max(speed), length.out = 100)

speed_secon_grid = data.frame(speed = speed_secon)

predicted.dist = predict(lm.cars2.qd, speed_secon_grid)

lines(speed_secon, predicted.dist,col = "blue")

#d)

B0 = cbind(1, speed)
B = cbind(B0,  speed^2)

Q0 = B0 %*% solve(t(B0) %*% B0) %*% t(B0)
Q = B %*% solve(t(B) %*% B) %*% t(B)

r0 = 2
r = 3

numerator = t(dist) %*%(Q - Q0) %*% dist
denominator = t(dist) %*%(diag(nrow(Q)) - Q) %*% dist/(nrow(B) - r)

F_statistic = numerator / denominator

lm.cars2.normal = lm(dist ~ speed)
```