

(12) Freie Arbeit

Im Folgenden finden Sie die Lösungen der Programmieraufgaben. Die Zwischenüberschriften geben an, auf die Inhalte welcher Einheit des Vorkurses sich die Lösungen beziehen.

(2) R und VS Code

1. Über die Kurswebseite wird die Datei `Salaries.csv` bereitgestellt. Laden Sie diese Datei herunter und speichern Sie sie auf Ihrem PC.
2. Platzieren Sie die Datei in einem Ordner auf Ihrem PC und legen Sie das R-Skript `Programmieraufgabe.r` an. Öffnen Sie es in VS Code.

```
# Salaries.csv  
# Programmieraufgabe.r
```

(6) Datenmanagement

3. Laden Sie den Inhalt der CSV-Datei mit `read.csv` in einem Dataframe `D1`. Speichern Sie die Anzahl der Zeilen von `D1` in die Variable `n`.

```
# Daten einlesen  
filename = 'Salaries.csv'           # Dateiname  
D1        = read.csv(filename)      # Dataframe  
n          = nrow(D1)               # Anzahl Datenpunkte
```

(5) Listen und Dataframes

4. Erzeugen Sie einen neuen Dataframe `D2`, indem Sie aus `D1` die Spalten `salary`, `yrs.since.phd` und `yrs.service` extrahieren. Geben Sie die ersten 10 Zeilen von `D2` aus.

```
# Daten reduzieren  
D2      = D1[,c("salary", "yrs.since.phd",      # Variablen extrahieren  
               "yrs.service")]  
# D2     = data.frame(D1$salary, D1$yrs.since.phd, # alternative Lösung  
#               D1$yrs.service)  
# D2     = D1[,c(6,3,4)]                        # alternative Lösung  
  
# Daten ausgeben  
print(D2[1:10,])                             # erste 10 Zeilen
```

	salary	yrs.since.phd	yrs.service
1	139750	19	18
2	173200	20	16
3	79750	4	3
4	115000	45	39
5	141500	40	41
6	97000	6	6
7	175000	30	23
8	147765	45	45
9	119250	21	20
10	129000	18	18

```
# head(D2, n=10)
```

```
# alternative Lösung
```

(8) Berechnung von Deskriptivstatistiken

5. Ermitteln Sie Stichprobenmittel, Stichprobenvarianz und Stichprobenstandardabweichung aller Variablen in D2 und speichern Sie diese in einen neuen Dataframe D2_var.

```
# Statistiken berechnen
library(dplyr)
D2_var = rbind(
  summarise(D2, across(everything(), mean)),
  summarise(D2, across(everything(), var)),
  summarise(D2, across(everything(), sd))) %>%
  as.data.frame()

# Statistiken ausgeben
rownames(D2_var) = c("Mean", "Var", "SD")
print(D2_var)
```

```
# R-Paket laden
```

```
# Kombination von Zeilen
```

```
# Stichprobenmittel
```

```
# Stichprobenvarianz
```

```
# Stichprobenstandardabweichung
```

```
# Konvertierung in Dataframe
```

```
# Zeilen benennen
```

```
# Ergebnisausgabe
```

	salary	yrs.since.phd	yrs.service
Mean	113706.46	22.31486	17.61461
Var	917425865.05	166.07485	169.15665
SD	30289.04	12.88700	13.00602

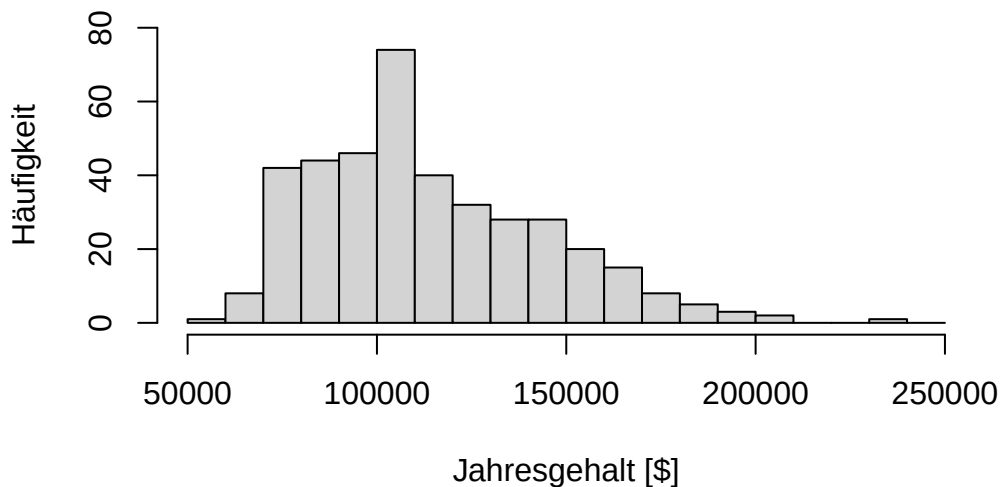
(9) Häufigkeitsverteilungen und Verteilungsfunktionen

6. Erstellen Sie ein Histogramm der Variable `salary` mit Klassen zwischen 50.000 \$ und 250.000 \$ und einer Klassenbreite von 10.000 \$. Speichern Sie die Abbildung als `Histogramm_1.pdf`.

```
# Variable "salary"
y      = D1$salary                # Dataframe-Variable
breaks = seq(50000, 250000, 10000) # Klassengrenzen

# Histogramm erzeugen
hist(y,                                # Histogramm
     breaks = breaks,                  # Klassengrenzen
     xlim   = c(min(breaks), max(breaks)), # x-Achsendaten
     ylim   = c(0, 80),                # y-Achsendaten
     xlab   = "Jahresgehalt [$]",      # x-Achsenbezeichnung
     ylab   = "Häufigkeit",            # y-Achsenbezeichnung
     main   = sprintf("Jahresgehalt, k = %d", # Abbildungstitel
                      length(breaks)))
```

Jahresgehalt, k = 21



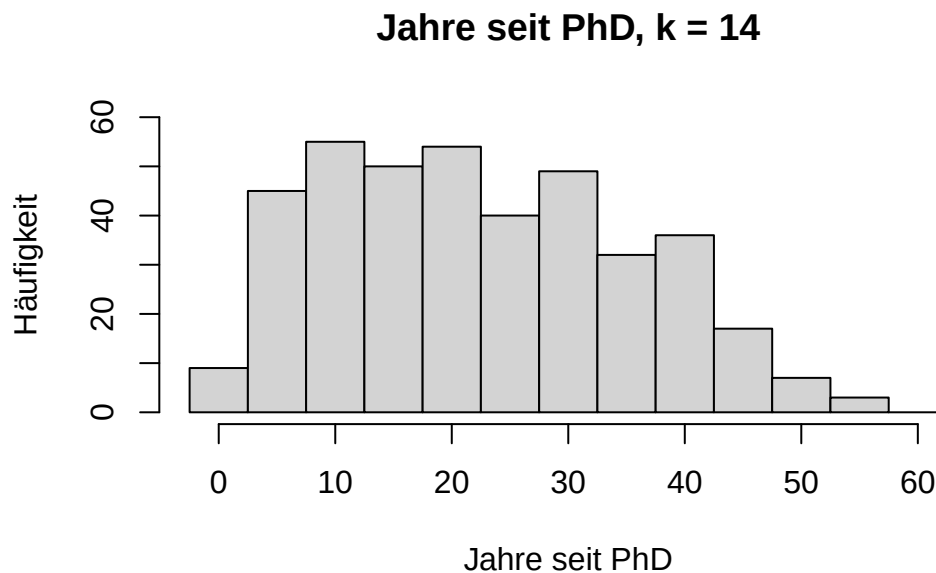
```
# Histogramm abspeichern
dev.copy2pdf(                          # PDF-Speicherung
  file = "Histogramm_1.pdf",           # Dateiname
  width = 7,                           # Breite
  height = 4)                          # Höhe
```

pdf
2

7. Erstellen Sie ein Histogramm der Variable `yrs.since.phd` mit Klassenzentren von 0 bis 60 Jahren in Schritten von 5 Jahren. Speichern Sie die Abbildung als `Histogramm_2.pdf`.

```
# Variable "Visits"
y      = D1$yrs.since.phd           # Dataframe-Variable
breaks = seq(0, 65, 5) - 2.5        # Klassengrenzen -2.5,+2.5,...,57.5,62.5
                                           # = Klassenzentren 0,5,...,55,60

# Histogramm erzeugen
hist(y,                               # Histogramm
     breaks = breaks,                # Klassengrenzen
     xlim   = c(min(breaks), max(breaks)), # x-Achsendaten
     ylim   = c(0, 60),              # y-Achsendaten
     xlab   = "Jahre seit PhD",       # x-Achsenbezeichnung
     ylab   = "Häufigkeit",          # y-Achsenbezeichnung
     main   = sprintf("Jahre seit PhD, k = %d", # Abbildungstitel
                      length(breaks)))
```



```
# Histogramm abspeichern
dev.copy2pdf(                          # PDF-Speicherung
  file = "Histogramm_2.pdf",           # Dateiname
  width = 7,                           # Breite
  height = 4)                          # Höhe
```

pdf
2

(10) Maße der zentralen Tendenz und der Datenvariabilität

8. Ermitteln Sie Stichprobenmittel, Stichprobenmedian und Stichprobenmodalwert aller Variablen in D2 und speichern Sie diese in einen neuen Dataframe D2_mean.

```
# Statistiken berechnen
library(dplyr)
D2_mean = rbind(
  summarise(D2, across(everything(), mean)),
  summarise(D2, across(everything(), median)),
  summarise(D2, across(everything(), ~ NA))) %>%
  as.data.frame()
for (i in 1:ncol(D2)) {
  H = as.data.frame(table(D2[,i]))
  names(H) = c("a", "h")
  D2_mean[3,i] = as.numeric(as.character(
    H$a[H$h == max(H$h)]))
}

# Statistiken ausgeben
rownames(D2_mean) = c("Mean", "Median", "Mode")
print(D2_mean)
```

	salary	yrs.since.phd	yrs.service
Mean	113706.5	22.31486	17.61461
Median	107300.0	21.00000	16.00000
Mode	92000.0	4.00000	3.00000

(7) Strukturiertes Programmieren, (3) Arithmetik und Variablen

9. Erzeugen Sie einen n-dimensionalen Vektor z, bei dem alle Einträge zunächst mit NA befüllt werden.
10. Benutzen Sie eine for-Schleife und if-else-Klauseln, um das i-te Element von z auf 1 zu setzen, wenn der i-te Eintrag von D1\$sex gleich "Female" ist, und auf 0, wenn dieser gleich "Male" ist.

```
# Vektor erzeugen
z = matrix(rep(NA,n), nrow=n)
# Vektor z

# Vektor befüllen
for (i in 1:n) {
  if (D1$sex[i] == "Female") { z[i] = 1 }
  else { z[i] = 0 }
}
```

(4) Vektoren und Matrizen, (3) Arithmetik und Variablen

11. Extrahieren Sie die Variablen `yrs.since.phd` und `salary` in die n -dimensionalen Vektoren `x` und `y`. Erzeugen Sie eine $n \times 3$ Matrix `X`, deren erste Spalte ein Einsvektor ist, deren zweite Spalte mit `z` befüllt wird und deren dritte Spalte der Variable `x` entspricht.
12. Berechnen Sie für ein lineares Regressionsmodell mit dem Datenvektor `y` und der Designmatrix `X` den Betaparameterschätzer mithilfe der Formel $\hat{\beta} = (X^T X)^{-1} X^T y$. Sie sollten folgendes Ergebnis erhalten:

```
# Variablen extrahieren
x      = as.matrix(D1$yrs.since.phd)      # Variable x
y      = as.matrix(D1$salary)             # Variable y

# Designmatrix erstellen
X      = matrix(c(rep(1,n), z, x), nrow = n) # Matrix X

# Parameterschätzer berechnen
beta_hat = solve(t(X) %*% X) %*% t(X) %*% y # Betaparameterschätzer
cat("Betaparameterschätzer : ", round(beta_hat, digits = 3))
```

Betaparameterschätzer : 93105.44 -7923.624 958.079