

## Aufgabe8Lösung.rmd

```
1 ---
2 title: "Lösungen der Übungsaufgaben"
3 author: "Doga Ceren Bozkurt-Sezin Özer"
4 date: "2024-11-24"
5 output: html_document
6 ---.c- .wemö
7
8
9 ```{r}
10 # ITSDaten einlesen aus der CSV-Datei
11 ITSDaten_Gruppe3_MTZ <- read.csv2(file = "ITSDaten_Gruppe3_MTZ.csv", fileEncoding = "utf-8",
12 stringsAsFactors = TRUE)
13
14 # Verteilung der Leberversagen-Werte in der Tabelle berechnen
15 leberversagen_verteilung <- table(ITSDaten_Gruppe3_MTZ$Leberversagen) /
16 nrow(ITSDaten_Gruppe3_MTZ)
17
18 # Ausgabe der Verteilung
19 print(leberversagen_verteilung)
20 ```
21
22 ```{r}
23 # CSV-Datei einlesen
24 ITSDaten_Gruppe3_MTZ <- read.csv2(file = "ITSDaten_Gruppe3_MTZ.csv", fileEncoding = "utf-8",
25 stringsAsFactors = TRUE)
26
27 # Filtern der Daten für Patienten mit einer Aufenthaltsdauer von genau einem Tag (LOS = 1)
28 ITSDaten_LOS1 <- ITSDaten_Gruppe3_MTZ[ITSDaten_Gruppe3_MTZ$LOS == 1, ]
29
30 # Entfernen des Ausreißers mit einer Körpertemperatur von 9,1 Celsius
31 ITSDaten_LOS1 <- ITSDaten_LOS1[ITSDaten_LOS1$Temperatur > 10, ]
32
33 print(ITSDaten_LOS1)
34 ```
35
36 ```{r}
37 ITSDaten_ML_Schätzer <- fitdistr(ITSDaten_LOS1$Temperatur, densfun = "normal")
38 print(ITSDaten_ML_Schätzer)
39 ```
40
41 Wir erhalten 37.7711538 °C als Schätzwert für den Mittelwert und 1.1785481 °C für die SD.
42
43 ```{r}
44 temperatur <- ITSDaten_ML_Schätzer$Temperatur
45
46 ```
47
48 ```{r}
49 median(ITSDaten_LOS1$Temperatur)
50 mad(ITSDaten_LOS1$Temperatur)
51 ```
```

```

50 | Der Median liegt bei 37.45 °C und der MAD bei 1.11195 °C. Diese Werte sind ähnlich zu den
    | Werten des ML-Schätzers.
51 | Da Mittelwert und Median ähnlich sind, spricht dies für eine recht symmetrische
    | Datenverteilung.
52 | Da der ML-Schätzer für die SD 0.1 °C größer als der MAD ist,
53 | könnte die Datenverteilung evtl. etwas breiter sein als bei einer Normalverteilung.
54 |
55 |
56 | ```{r}
57 | # RMX-Schätzer
58 | distrModOptions(show.details = "minimal")
59 | roblox(ITSDaten_LOS1$Temperatur, eps.lower = 0.00, eps.upper = 0.05, k = 3)
60 | ```
61 |
62 |
63 | Wir erhalten 37.69 °C als Schätzerwert für den Mittelwert und 1.169 °C für die SD.
64 | Die Werte sind sehr ähnlich zu den bereits berechneten Schätzwerten.
65 |
66 |
67 | ```{r}
68 | rmx <- rmx(ITSDaten_LOS1$Temperatur, model = "norm",
69 |           eps.lower = 0.00, eps.upper = 0.05)
70 | res.rmx
71 | ```
72 |
73 |
74 | Wir haben diesen Schritt übersprungen, da es kein Paket gab.
75 |
76 |
77 | ```{r}
78 | normCI(ITSDaten_LOS1$Temperatur, boot = TRUE, bootci.type = "stud")
79 | ```
80 |
81 |
82 | Wir erhalten als 95% Konfidenzintervall für den Mittelwert [37.46, 38.12] °C.
83 | Für die SD ergibt sich [1.012, 1.494] °C.
84 |
85 |
86 | ```{r}
87 | ## x: Vektor der Beobachtungen
88 | ## i: Vektor der Indizes der Bootstrap-Stichprobe
89 | RMXEst <- function(x, i){
90 |   res <- roblox(x[i], eps.lower = 0.00, eps.upper = 0.05, k = 3)
91 |   c(estimate(res), diag(asvar(res)))
92 | }
93 | ```
94 |
95 | ```{r}
96 | nr.cpus <- detectCores()-1
97 | ```
98 |
99 | ```{r, cache = TRUE}
100 | ## Bootstrap-Schätzungen
101 | boot.out <- boot(ITSDaten_LOS1$Temperatur, statistic = RMXEst, R = 999)

```

```

102 ## Bootstrap-Konfidenzintervall für den Mittelwert
103 boot.ci(boot.out, index = c(1,3), type = "stud")
104 ## Bootstrap-Konfidenzintervall für die Standardabweichung
105 boot.ci(boot.out, index = c(2,4), type = "stud")
106 ```
107
108
109 Als 95% Konfidenzintervall für den Mittelwert erhalten wir [37.37, 38.02] °C und
110 für die SD ergibt sich [0.963, 1.471] °C. Die Ergebnisse sind demnach sehr ähnlich
111 zu den Ergebnissen der Funktion normCI.
112
113 ```{r}
114 ## Bootstrap-Konfidenzintervall für den Median
115 medianCI(ITSDaten_LOS1$Temperatur, method = "boot", bootci.type = "perc")
116 ## Bootstrap-Konfidenzintervall für den MAD
117 madCI(ITSDaten_LOS1$Temperatur, method = "boot", bootci.type = "perc")
118 ```
119
120 Wir erhalten [37.30, 38.05] °C als 95% Konfidenzintervall für den Median und
121 [0.741, 1.631] °C für den MAD.
122 median
123 37.45
124 MAD
125 1.11195
126
127 ```{r}
128 ## x: Vektor der Beobachtungen
129 ## i: Vektor der Indizes der Bootstrap-Stichprobe
130 RMXEst <- function(x, i){
131   res <- roblox(x[i], eps.lower = 0.01, eps.upper = 0.05, k = 3)
132   c(estimate(res), diag(asvar(res)))
133 }
134 ```
135
136 ```{r, cache = TRUE}
137 ## Bootstrap-Schätzungen
138 boot.out <- boot(ITSDaten_LOS1$Temperatur, statistic = RMXEst, R = 999)
139 ## Bootstrap-Konfidenzintervall für den Mittelwert
140 boot.ci(boot.out, index = c(1,3), type = "stud")
141 ## Bootstrap-Konfidenzintervall für die Standardabweichung
142 boot.ci(boot.out, index = c(2,4), type = "stud")
143 ```
144
145 ```{r}
146 confint(res.rmx, method = "boot", type = "stud")
147 ```
148
149 rmx fehlt.
150
151 Es ergibt sich [37.34, 38.06] °C für das Konfidenzintervall des Mittelwertes und
152 [ 0.968, 1.490] °C für die Standardabweichung. Die Werte befinden sie sich innerhalb des
153 Konfidenzintervalls.
154

```

```

155
156 Histogramm
157 ```{r}
158 ggplot(ITSDaten_LOS1, aes(x=Temperatur)) +
159   geom_histogram(aes(y=after_stat(density)),
160     breaks = seq(from = 33, to = 42, by = 0.5),
161     fill = "darkgrey") +
162   geom_density(color = "orange") + ylab("Dichte") +
163   stat_function(fun = dnorm, args = list(mean = 37.9, sd = 1.17),
164     color = "darkred", linewidth = 2) +
165   stat_function(fun = dnorm, args = list(mean = 37.8, sd = 1.04),
166     color = "darkblue", linewidth = 2) +
167   stat_function(fun = dnorm, args = list(mean = 37.8, sd = 1.14),
168     color = "darkgreen", linewidth = 2) +
169   annotate("text", x = c(40, 40, 40), y = c(0.31, 0.29, 0.27),
170     col = c("darkred", "darkblue", "darkgreen"),
171     label = c("ML-Schätzer", "Median und MAD", "RMX-Schätzer"))+
172   ggtitle("Maximale Körpertemperatur")
173 ```
174
175
176 ggsave("plot.png", width = 8, height = 6)
177 wir haben das Histogramm als png gespeichert.
178
179
180 ```{r}
181 gg1 <- ggplot(ITSDaten_LOS1, aes(sample = Temperatur)) +
182   qqplotr::stat_pp_band(dparams = list(mean = 37.9, sd = 1.17)) +
183   qqplotr::stat_pp_point(dparams = list(mean = 37.9, sd = 1.17)) +
184   qqplotr::stat_pp_line() +
185   xlab("Beobachtete Maximale Körpertemperatur") +
186   ylab("Modell für Maximale Körpertemperatur") +
187   ggtitle("pp-Plot für ML-Schätzer")
188 gg2 <- ggplot(ITSDaten_LOS1, aes(sample = Temperatur)) +
189   qqplotr::stat_qq_band(dparams = list(mean = 37.9, sd = 1.17), identity = TRUE) +
190   qqplotr::stat_qq_point(dparams = list(mean = 37.9, sd = 1.17)) +
191   qqplotr::stat_qq_line(dparams = list(mean = 37.9, sd = 1.17), identity = TRUE) +
192   xlab("Modell für Maximale Körpertemperatur") +
193   ylab("Beobachtete Maximale Körpertemperatur") +
194   ggtitle("qq-Plot für ML-Schätzer")
195 grid.arrange(gg1, gg2, nrow = 1)
196 ```
197
198 ggsave("plot2.png", width = 8, height = 6)
199 wir haben das qqplot für ML-Schätzer als png gespeichert.
200
201 ```{r}
202 gg1 <- ggplot(ITSDaten_LOS1, aes(sample = Temperatur)) +
203   qqplotr::stat_pp_band(dparams = list(mean = 37.8, sd = 1.14)) +
204   qqplotr::stat_pp_point(dparams = list(mean = 37.8, sd = 1.14)) +
205   qqplotr::stat_pp_line() +
206   xlab("Beobachtete Maximale Körpertemperatur") +
207   ylab("Modell für Maximale Körpertemperatur") +
208   ggtitle("pp-Plot für RMX-Schätzer")

```

```
209 gg2 <- ggplot(ITSDaten_LOS1, aes(sample = Temperatur)) +
210   qqplotr::stat_qq_band(dparams = list(mean = 37.8, sd = 1.14), identity = TRUE) +
211   qqplotr::stat_qq_point(dparams = list(mean = 37.8, sd = 1.14)) +
212   qqplotr::stat_qq_line(dparams = list(mean = 37.8, sd = 1.14), identity = TRUE) +
213   xlab("Modell für Maximale Körpertemperatur") +
214   ylab("Beobachtete Maximale Körpertemperatur") +
215   ggtitle("qq-Plot für RMX-Schätzer")
216 grid.arrange(gg1, gg2, nrow = 1)
217 ```
218
219 ggsave("plot3.png", width = 8, height = 6)
220 Wir haben das qqplot für RMX-Schätzer als png gespeichert.
221 Die Anpassung ist ähnlich gut wie im Fall des ML-Schätzers.
222
```