

A:\17 Cloud\OneDrive\Documents\GitHub\r\_projekt\_chirurgie\scripts\03\_lagemasse\_v5.R

```
1 # -----
2 # 03_lagemasse.R
3 # Berechnung und grafische Darstellung der Lagemaße
4 # inkl. Histogramme und stargazer-Tabelle (nur Mean, Median, Mode)
5 cat("----- \n")
6 cat("03_lagemasse.R \n")
7 cat("----- \n")
8 # -----
9
10 # 1) Packages laden und installieren, wenn nötig
11 cat("🔍 Überprüfe, ob die Pakete ggplot2, stargazer und dplyr installiert sind...\n")
12
13 # Warnungen vorübergehend deaktivieren
14 options(warn = -1)
15
16 # 1.1) ggplot2 Paket laden (automatisch installieren, wenn nötig)
17 if (!requireNamespace("ggplot2", quietly = TRUE)) {
18   install.packages("ggplot2")
19 }
20 library(ggplot2)
21
22 # 1.2) stargazer Paket laden (automatisch installieren, wenn nötig)
23 if (!requireNamespace("stargazer", quietly = TRUE)) {
24   install.packages("stargazer")
25 }
26 library(stargazer)
27
28 # 1.3) dplyr Paket laden (automatisch installieren, wenn nötig)
29 if (!requireNamespace("dplyr", quietly = TRUE)) {
30   install.packages("dplyr")
31 }
32 library(dplyr)
33
34 cat("✅ Alle Pakete wurden erfolgreich geladen.\n")
35
36 # 2) Ordnerstruktur erstellen, wenn noch nicht vorhanden
37 cat("🔍 Überprüfe, ob der Ordner 'export/03_lagemasse' existiert...\n")
38 export_dir <- "export/03_lagemasse"
39
40 if (!dir.exists(export_dir)) {
41   dir.create(export_dir, recursive = TRUE)
42   cat("✅ Ordner wurde erstellt.\n")
43 } else {
44   cat("✅ Ordner existiert bereits.\n")
45 }
46
47 # 3) Einheitliches Theme für Plots
48 theme_white_report <- function(base_size = 14) {
49   theme_bw(base_size = base_size) +
50     theme(
51       plot.background = element_rect(fill = "white", color = NA),
52       panel.background = element_rect(fill = "white", color = "black"),
53       text = element_text(color = "black"),
54       axis.text = element_text(color = "black"),
55       axis.title = element_text(color = "black"),
56       plot.title = element_text(face = "bold")
57     )
58 }
59
60 # 4) Funktionen berechnen, je nach Aufgabenstellung
61 cat("📂 Lade den Datensatz 'chirurgische_komplikationen.RData'...\n")
62 load("data/chirurgische_komplikationen.RData")
63
64 # -----
65 # Lagemaße berechnen
66
67 # Modus (häufigster Wert) berechnen
68 modus <- function(x) {
69   ux <- unique(x)
70   ux[which.max(tabulate(match(x, ux)))]
71 }
72
73 # Funktion zur Berechnung der Lagemaße
74 berechne_lagemasse <- function(x, name) {
75   data.frame(
```

```

76     Variable = name,
77     Mean = round(mean(x), 2),
78     Median = round(median(x), 2),
79     Mode = modus(x)
80   )
81 }
82
83 # Berechnungen der Lagemaße für verschiedene Variablen
84 lagemaße_df <- rbind(
85   berechne_lagemasse(daten$Operationsdauer, "Operationsdauer"),
86   berechne_lagemasse(daten$Blutverlust, "Blutverlust"),
87   berechne_lagemasse(daten$Komplikationsrisiko, "Komplikationsrisiko")
88 )
89
90 cat("✅ Lagemaße wurden berechnet.\n")
91
92 # -----
93 # Stargazer-Tabelle als txt speichern
94
95 cat("📄 Erstelle stargazer-Tabelle als .txt...\n")
96
97 # Stargazer für die Lagemaße in einer strukturierten Textdatei ausgeben (nur Mean, Median, Mode)
98 stargazer(lagemaße_df[, c("Variable", "Mean", "Median", "Mode")], type = "text", summary = FALSE, rownames = FALSE,
99   out = file.path(export_dir, "lagemasse_tabelle.txt"))
100
101 cat("✅ stargazer-Tabelle wurde als .txt-Datei erstellt und gespeichert.\n")
102
103 # -----
104 # Plots für Lagemaße erstellen
105
106 plot_lagemasse <- function(var_name, var_label, y_label, daten, sprache = "en") {
107   x <- daten[[var_name]]
108   m <- mean(x)
109   md <- median(x)
110   mo <- modus(x)
111
112   if (sprache == "de") {
113     titel <- paste("Lagemaße:", var_label)
114     undertitel <- "Blau = Mittelwert, Rot = Median, Grün = Modus"
115   } else {
116     titel <- paste("Measures of Central Tendency:", var_label)
117     undertitel <- "Blue = Mean, Red = Median, Green = Mode"
118   }
119
120   ggplot(daten, aes(x = x)) +
121     geom_histogram(aes(y = ..density..), binwidth = 20, fill = "gray80", color = "black") +
122     geom_density(color = "darkgray", linewidth = 1) +
123     geom_vline(xintercept = m, color = "blue", linetype = "dashed") +
124     geom_vline(xintercept = md, color = "red", linetype = "solid") +
125     geom_vline(xintercept = mo, color = "green", linetype = "dotted") +
126     labs(
127       title = titel,
128       subtitle = undertitel,
129       x = var_label,
130       y = y_label
131     ) +
132     theme_white_report()
133 }
134
135 # Plots erstellen und speichern
136 cat("📄 Erstelle Plots für alle Lagemaße...\n")
137
138 # Duration of Surgery
139 p_duration <- plot_lagemasse("Operationsdauer", "Operationsdauer (Minuten)", "Dichte", daten, "de")
140 ggsave(file.path(export_dir, "lagemasse_duration_de.png"), plot = p_duration, width = 8, height = 6)
141 print(p_duration) # Plot im Reiter "Plots" anzeigen
142
143 # Blood Loss
144 p_bloodloss <- plot_lagemasse("Blutverlust", "Blutverlust (ml)", "Dichte", daten, "de")
145 ggsave(file.path(export_dir, "lagemasse_bloodloss_de.png"), plot = p_bloodloss, width = 8, height = 6)
146 print(p_bloodloss) # Plot im Reiter "Plots" anzeigen
147
148 # Complication Risk
149 p_risk <- plot_lagemasse("Komplikationsrisiko", "Komplikationsrisiko (Skalenwert)", "Dichte", daten, "de")
150 ggsave(file.path(export_dir, "lagemasse_risk_de.png"), plot = p_risk, width = 8, height = 6)
151 print(p_risk) # Plot im Reiter "Plots" anzeigen
152

```

```
153 | cat("✅ Alle Plots wurden erstellt und gespeichert.\n")
154 |
155 | # 5) Plots in R-Studio unter Reiter Plots ausgeben
156 | cat("📊 Alle Plots werden in R-Studio unter dem Reiter 'Plots' angezeigt.\n")
157 |
158 | # Nach dem Plot die Warnungen wieder aktivieren
159 | options(warn = 0)
160 |
```