

A:\17 Cloud\OneDrive\Documents\GitHub\r_projekt_chirurgie\scripts\05_korrelation_v5.R

```
1 # -----
2 # 05_korrelation.R
3 # Berechnung, Visualisierung & Korrelationsanalyse
4 cat("----- \n")
5 cat("05_korrelation.R \n")
6 cat("----- \n")
7 # -----
8
9 # 1) Pakete installieren und laden, wenn nötig
10 cat("🔍 Überprüfe, ob die Pakete ggplot2, stargazer und gridExtra installiert sind...\n")
11
12 # Warnungen vorübergehend deaktivieren
13 options(warn = -1)
14
15 # 1.1) ggplot2 Paket laden (automatisch installieren, wenn nötig)
16 if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2", type = "binary")
17 library(ggplot2)
18
19 # 1.2) stargazer Paket laden (automatisch installieren, wenn nötig)
20 if (!requireNamespace("stargazer", quietly = TRUE)) install.packages("stargazer", type = "binary")
21 library(stargazer)
22
23 # 1.3) gridExtra Paket laden (automatisch installieren, wenn nötig)
24 if (!requireNamespace("gridExtra", quietly = TRUE)) install.packages("gridExtra", type = "binary")
25 library(gridExtra)
26
27 cat("✅ Alle Pakete wurden erfolgreich geladen.\n")
28
29 # 2) Ordnerstruktur erstellen, wenn noch nicht vorhanden
30 cat("🔍 Überprüfe, ob der Ordner 'export/05_korrelation' existiert...\n")
31 output_dir <- "export/05_korrelation"
32 if (!dir.exists(output_dir)) {
33   dir.create(output_dir, recursive = TRUE)
34   cat("✅ Ordner wurde erstellt.\n")
35 } else {
36   cat("✅ Ordner existiert bereits.\n")
37 }
38
39 # 3) Einheitliches Theme für Plots
40 theme_white_report <- function(base_size = 14) {
41   theme_bw(base_size = base_size) +
42     theme(
43       plot.background = element_rect(fill = "white", color = NA),
44       panel.background = element_rect(fill = "white", color = "black"),
45       text = element_text(color = "black"),
46       axis.text = element_text(color = "black"),
47       axis.title = element_text(color = "black"),
48       plot.title = element_text(face = "bold")
49     )
50 }
51
52 # 4) Funktionen berechnen, je nach Aufgabenstellung
53 cat("🔍 Berechnung der Korrelationsmatrix...\n")
54
55 # Subset für Korrelationsanalyse erstellen
56 subset_cor <- subset(daten, select = c(Operationsdauer, Blutverlust, Komplikationsrisiko))
57
58 # Korrelationsmatrix berechnen
59 cor_matrix <- cor(subset_cor)
60
61 # Korrelationswerte ausgeben
62 cat("----- \n")
63 cat("Korrelationskoeffizienten: \n")
64 cat("----- \n")
65 cat("Operationsdauer & Blutverlust:", round(cor(subset_cor$Operationsdauer, subset_cor$Blutverlust), 3), "\n")
66 cat("Operationsdauer & Komplikationsrisiko:", round(cor(subset_cor$Operationsdauer, subset_cor$Komplikationsrisiko), 3), "\n")
67 cat("Blutverlust & Komplikationsrisiko:", round(cor(subset_cor$Blutverlust, subset_cor$Komplikationsrisiko), 3), "\n")
68 cat("----- \n")
69 cat("----- \n")
70
71 # 5) Plots in R-Studio unter Reiter Plots ausgeben
72 cat("🖼 Alle Plots werden in R-Studio unter dem Reiter 'Plots' angezeigt.\n")
73
74 # Hilfsfunktion für Korrelationsplot mit Trendlinie
75 plot_correlation <- function(xvar, yvar, xlabel, ylabel, daten, title) {
76   ggplot(daten, aes(x = .data[[xvar]], y = .data[[yvar]])) +
```

```

76     geom_point(alpha = 0.5, color = "gray40") +
77     geom_smooth(method = "lm", color = "blue", linewidth = 1, se = FALSE) +
78     labs(title = title, x = xlabel, y = ylabel) +
79     theme_white_report()
80 }
81
82 # Korrelationsplots erstellen und anzeigen
83 plot_ops_blut <- plot_correlation("Operationsdauer", "Blutverlust", "Operationsdauer (Min)", "Blutverlust (ml)", daten,
84 "Korrelation: OP-Dauer & Blutverlust")
85 print(plot_ops_blut)
86
87 plot_ops_komp <- plot_correlation("Operationsdauer", "Komplikationsrisiko", "Operationsdauer (Min)", "Komplikationsrisiko", daten,
88 "Korrelation: OP-Dauer & Komplikationsrisiko")
89 print(plot_ops_komp)
90
91 plot_blut_komp <- plot_correlation("Blutverlust", "Komplikationsrisiko", "Blutverlust (ml)", "Komplikationsrisiko", daten,
92 "Korrelation: Blutverlust & Komplikationsrisiko")
93 print(plot_blut_komp)
94
95 # 6) Plots exportieren
96 cat("📄 Exportiere die Korrelationsplots als PNG...\n")
97
98 ggsave(file.path(output_dir, "plot_ops_blut.png"), plot = plot_ops_blut, width = 7, height = 5)
99 ggsave(file.path(output_dir, "plot_ops_komp.png"), plot = plot_ops_komp, width = 7, height = 5)
100 ggsave(file.path(output_dir, "plot_blut_komp.png"), plot = plot_blut_komp, width = 7, height = 5)
101
102 cat("✅ Alle Plots wurden erfolgreich exportiert.\n")
103
104 # 7) Stargazer-Tabelle für die Korrelationsmatrix speichern und im RStudio Viewer anzeigen
105 cat("📄 Erstelle stargazer-Tabelle für die Korrelationsmatrix als .html...\n")
106
107 # HTML-Datei für die Korrelationsmatrix erzeugen
108 html_output <- file.path(output_dir, "korrelationsmatrix.html")
109 capture.output(stargazer(cor_matrix, type = "html", out = html_output), file = NULL)
110
111 # HTML-Datei im RStudio Viewer anzeigen
112 if (interactive()) {
113   viewer <- getOption("viewer")
114   viewer(html_output)
115 }
116
117 cat("✅ stargazer-Tabelle wurde als .html-Datei erstellt und im RStudio Viewer angezeigt.\n")
118
119 # Warnungen wieder aktivieren
120 options(warn = 0)

```