```
#########################################################################################################################
# In dieser Datei sind alle für eigene Datenanalysen zu individualisierenden Variablen- und Programmeinträge farblich hinterlegt!
#########################################################################################################################
```

```
#############################################################################
# Quellenangaben:
# Diese Datei entstand in Form einer Kombination aus Literatur-, Internet- und Vortragsunterlagenrecherche:
# Björn, W. (2022). Statistik mit R. Schnelleinstieg. R einfach lernen in 14 Tagen. mitp
# Luhmann, M. (2020). R für Einsteiger. Einführung in die Statistik-Software für die Sozialwissenschaften (5. Auflage). Beltz
# sowie über Google auffindbare Tutorials bzw. in Form persönlicher Kommunikation mit ChatGPT zwischen 1. und 30. Juli 2024.
# Weitere Beispiele entstammen den Übungsunterlagen der die VO STADA am Institut für Publizistik- und Kommunikationswissen-
# schaft der Uni Wien im Wintersemester 2024 begleitenden Übungen (UE STADA).
# Besonderer Dank gilt dabei dem Team rund um Ariadne Neureiter.

# Eine Seite grundlegender Einstiegstipps in R findet sich bei Döring, 2023, S. 1043 (Döring, N. (2023). Forschungsmethoden
# und Evaluation in den Sozial- und Humanwissenschaften (6. Auflage). Springer).

# Analysewege in R sind vielfältig und besitzen meist mehrere Alternativen. Die in diesem Skript angeführten Befehle stellen
# den Versuch dar, jeweils möglichst einfachen Code zu definieren (mit wenig und kurzem Befehlscode auszukommen).
# Alle angeführten Beispiele sind adaptierbar und gelten bloß als Referenz für verschiedene statistische Analyseroutinen.
# SIE STELLEN KEINE VOLLWERTIGE AUSWERTUNG EINES DATENFILES DAR!
```

```
################################################################################
# Start mit R: Installieren von (1.) R und (2.) RStudio unter
# https://www.rstudio.com/products/rstudio/download/#download

# In RStudio unter "Tools > Global Options..." bei "Appearance" ein "Editor theme" wählen.

# Den Dateipfad einstellen: HIER sollten alle Daten liegen, mit denen gearbeitet wird.
# "Session > Set Working Directory > Choose Directory..." und entsprechenden Ordner am eigenen PC auswählen.

# Diese Datei in die Working Directory kopieren und einlesen (= anklicken und mit RStudio öffnen).
# Die Datei enthält die für das gebräuchliche Vorgehen beim Analysieren von Daten wichtigsten Schritte.
# Verändert werden müssen für die Analyse eigener Daten bloß die Datei- und Variablennamen sowie
# unter Anführungszeichen stehende Angaben, die sich auf
# https://howtodo.at/r/BUCHdaten.csv (Format Text) oder
# https://howtodo.at/r/BUCHdaten.xlsx (Format Excel) oder
# https://howtodo.at/r/BUCHdaten.sav (Format SPSS) beziehen.

# In der ebenfalls frei verfügbaren Datei
# https://howtodo.at/r/HowToDo.R.pdf sind die für eigene Datenanalysen zu verändernden Datei- und Variablennamen
# sowie andere Angaben farblich hinterlegt.

# Der zu den Daten gehörende Fragebogen findet sich unter
# https://howtodo.at/r/Fragebogen_BUCHdaten.pdf

# Die Seitenangaben [in eckigen Klammern] in den Kommentaren beziehen sich auf
# Braunecker, C. (2023). How to do Statistik und SPSS. Eine Gebrauchsanleitung. facultas/utb

# Jeder Text neben einem Rautezeichen gilt als Kommentar und nicht als Befehlstext.

# Befehl ausführen mit Cursor irgendwo innerhalb des Befehls (der sich auch übermehrere Zeilen erstrecken kann),
# und "STRG + ENTER".
# Nur jene Befehle anpassen und ausführen (bzw. in ein eigenes Skript unter "File > New File > R Script" kopieren)
# und weiterentwickeln, die zur eigenen Auswertung benötigt werden.

# R-Befehle werden sequenziell (der Reihe nach) abgearbeitet. Treten Fehler auf, ist es oft hilfreich, mit vorherigen
# Programmteilen neu zu beginnen (diese zu markieren und neu ablaufen zu lassen) bzw. im schlimmsten Fall die
# gesamten Daten neu zu laden und am Skriptanfang neu zu beginnen.

# Wenn vor Befehlen etwas von "if (!require(...)) {install.packages("...")} steht, benötigen diese Befehle
# ein eigenes Programmpaket (eines von mehr als 20.000 frei erhältlichen). Dieses jeweilige Paket muss einmalig
# installiert werden (das macht der "if (!require ...)"-Befehl automatisch), ab dann nur noch aufgerufen werden
# (jeweils in der daruntersteheende Zeile, "library(...)"). Jedes Programmpaket umfasst mehrere (andere) Befehle.
```

```
# Dieselben Befehle können auch in unterschiedlichen Programmpaketen vorkommen, führen dann aber oft zu anderen
# Ergebnis-Outputs.

# Gutes Gelingen! Claus Braunecker, Sommer 2024
```

```
################################################################################
# Daten einlesen
################################################################################
# CSV-Daten einlesen
data <- read.csv2("D:/_projekt/_spss/daten & fragebogen/BUCHdaten.csv", na = "NA", dec = ",", header=TRUE)
View(data)

# oder
# Excel-Daten einlesen
if (!require(readxl)) {install.packages("readxl")}
library(readxl)
BUCHdaten <- read_excel("D:/_projekt/_spss/daten & fragebogen/BUCHdaten.xlsx")
data <- BUCHdaten
View(data)

# oder
# SPSS-Daten einlesen
if (!require(haven)) {install.packages("haven")}
library(haven)
BUCHdaten <- read_sav("D:/_projekt/_spss/daten & fragebogen/BUCHdaten.sav")
data <- BUCHdaten
View(data)
```

```
################################################################################
# Datenbeschriftungen [Seite 127 - 129]
################################################################################
# Variablenlabels definieren (eine exemplarische Auswahl)
# Bringt wenig Mehrwert, weil's R nicht in  Routinen übernimmt.
if (!require(Hmisc)) {install.packages("Hmisc")}
library(Hmisc)
label(data$f_01) <- "Lesen Sie gerne?"
label(data$f_02) <- "Haben Sie in den letzten 12 Monaten einen Roman gelesen?"
label(data$f_03) <- "Haben Sie in den letzten 12 Monaten ein Fachbuch gelesen?"
label(data$f_04) <- "Wo kaufen Sie Bücher lieber?"
label(data$f_12) <- "Geschlecht"

# Wertebeschriftungen definieren (eine exemplarische Auswahl
# Die Beschriftungen überschreiben die Codierungen, werden aber in die Routinen übernommen.
# ACHTUNG: Keine Codierungen, mit denen Berechnungen durchgeführt werden sollen, mit Textbezeichnungen überschreiben!
# ACHTUNG!!! Nach dem Einlesen der Daten nur EIN Mal laufen lassen, sonst sind auf diesen Variablen keine Daten mehr vorhanden!
class(data$f_01)
class(data$f_04)
class(data$f_05_txt)
data$f_01 <- factor(data$f_01, levels=c(0,1), labels=c("nein","ja"))
data$f_02 <- factor(data$f_02, levels=c(0,1), labels=c("nein","ja"))
data$f_03 <- factor(data$f_03, levels=c(0,1), labels=c("nein","ja"))
data$f_04 <- factor(data$f_04, levels=c(1,2), labels=c("in einer Buchhandlung","im Versandhandel"))
data$f_12 <- factor(data$f_12, levels=c(1,2,3), labels=c("weiblich","männlich", "divers"))

# Check der vergebenen Wertelabels in einer einfachen Häufigkeitszählung
table(data$f_01)
table(data$f_02)
table(data$f_03)
table(data$f_04)
table(data$f_08)
table(data$f_12)
```

```
################################################################################
# Datenhandling [Seite 124; 133 - 140]
################################################################################
################################################################################
# Fälle auswählen
# Fälle auswählen, wenn numerisch codiert
# table(data$f_04)
# data_filter <- subset(data, f_04 == 1)
# table(data_filter$f_04)

# Fälle auswählen wenn als Text codiert
table(data$f_04)
data_gefiltert <- subset(data, f_04 == "in einer Buchhandlung")
table(data_gefiltert$f_04)



# Umcodieren
# Code als fehlenden Wert [Seite 130 - 132] definieren
if (!require(car)) {install.packages("car")}
library(car)
# Bundesland "Wien" (Code 1) wird zu fehlendem Wert (nur beispielhaft für eine Umcodierung, hier sinnfrei)
data$f_13 <- recode(data$f_13, "1=NA")
table(data$f_13)

# Bundesland "Wien" wird rückcodiert zu Code 1 (nur beispielhaft für eine Umcodierung, hier sinnfrei)
data$f_13 <- recode(data$f_13, "NA=1")
table(data$f_13)
```

```
# Gruppenbildung (umcodieren)
if (!require(car)) {install.packages("car")}
library(car)
table(data$f_14)
# Gruppe 1: 15 bis 30 Jahre alt, Gruppe 2: 31 bis 50 Jahre, Gruppe 3: 51 Jahre plus
data$f_14_gruppiert <- recode(data$f_14, "1:30=1; 31:50=2; 51:100=3")
# oder mit "lo" - niedrigster Wert und "hi" - höchster Wert
data$f_14_gruppiert <- recode(data$f_14, "lo:30=1; 31:50=2; 51:hi=3")
if (!require(gmodels)) {install.packages("gmodels")}
library(gmodels)
# Kontrolle, ob alles funktioniert hat
CrossTable(data$f_14, data$f_14_gruppiert,
           prop.c = FALSE, prop.r = FALSE, prop.t = FALSE, expected = FALSE,
           prop.chisq = FALSE)



# Richtungsbereinigen von invers gepolten Items in Itembatterien
if (!require(car)) {install.packages("car")}
library(car)
data$f_09_2_bereinigt <- recode(data$f_09_2, "1=6; 2=5; 3=4; 4=3; 5=2; 6=1")
data$f_09_4_bereinigt <- recode(data$f_09_4, "1=6; 2=5; 3=4; 4=3; 5=2; 6=1")
data$f_09_6_bereinigt <- recode(data$f_09_6, "1=6; 2=5; 3=4; 4=3; 5=2; 6=1")
data$f_09_9_bereinigt <- recode(data$f_09_9, "1=6; 2=5; 3=4; 4=3; 5=2; 6=1")
# Kontrolle, ob alles funktioniert hat (ob sich die Codierungen der Datensätze wie beabsichtigt verändert haben)
data$f_09_2
data$f_09_2_bereinigt
data$f_09_4
data$f_09_4_bereinigt
data$f_09_6
data$f_09_6_bereinigt
data$f_09_9
data$f_09_9_bereinigt
```

```
# Mittelwertsindex (über richtungsbereinigte Frage 9) bilden
data$f_09_index <- rowMeans(subset(data,
    select = c(
      f_09_1,
      f_09_2_bereinigt,
      f_09_3,
      f_09_4_bereinigt,
      f_09_5,
      f_09_6_bereinigt,
      f_09_7,
      f_09_8,
      f_09_9_bereinigt
      )), na.rm = TRUE)
data$f_09_index
describe(data$f_09_index)
```

```
################################################################################
# Deskriptive Statistiken
################################################################################
# Übersicht über eine Variable und deren Codierungen
if (!require(epiDisplay)) {install.packages("epiDisplay")}
library(epiDisplay)
tab1(data$f_04)

# Erstellen einer Häufigkeitstabelle [Seite 143 - 145] für eine Variable (ähnlich SPSS)
if (!require(summarytools)) {install.packages("summarytools")}
library(summarytools)
freq_table <- freq(data$f_08, report.nas = TRUE)
print(freq_table)


# Einfaches Säulendiagramm (kategorial)
barplot(table(data$f_01),
        xlab="Lesen Sie gerne?",
        names.arg=c("nein", "ja"),
        ylab="Anzahl",
        main="Überschrift",
        cex.axis=0.8, cex.names=0.8, axis.lty=1, las=2)

# Einfaches Säulendiagramm (metrisch, in Rot)
barplot(table(data$f_08),
        xlab="Ausgaben Fachliteratur",
        ylab="Anzahl",
        main="Überschrift",
        col = "red",
        cex.axis=0.8, cex.names=0.8, axis.lty=1, las=2)

# Einfaches Balkendiagramm (metrisch)
barplot(table(data$f_08),
        xlab="Ausgaben Fachliteratur",
        ylab="Anzahl",
        main="Überschrift",
        horiz=TRUE,
        cex.axis=0.8, cex.names=0.8, axis.lty=1, las=2)
```

```
# Boxplot [Seite 63]
boxplot(data$f_08,
        xlab = "Ausgaben für Fachliteratur",
        ylab = "Anzahl",
        main = "Überschrift")

# Histogramm
hist(data$f_08,
     xlab = "Ausgaben für Fachliteratur",
     ylab = "Anzahl",
     main = "Überschrift")


# Erstellen einer deskriptiven (metrischen) Übersicht über ALLE Variablen [Seite 149 - 152]
if (!require(Hmisc)) {install.packages("Hmisc")}
library(Hmisc)
describe(data)

# Erstellen einer deskriptiven (metrischen) Übersicht über einzelne Variablen
summary(data$f_08)

# ... oder eine andere Variante:
# Eventuell vorhandene andere und aktivierte Bibliothek psych wird abgewählt
if ("package:psych" %in% search()) {detach("package:psych", unload = TRUE)}
describe(data$f_08)
describe(data$f_16)
describe(data$f_17)

# ... oder eine andere Variante:
# Derselbe Befehl mit anderer Programmbibliothek
if (!require(psych)) {install.packages("psych")}
library(psych)
describe(data$f_08)
describe(data$f_16)
describe(data$f_17)
```

```
################################################################################
# Kreuztabelle [Seite 152 - 159]
################################################################################
if (!require(gmodels)) {install.packages("gmodels")}
library(gmodels)

# P-Wert soll NICHT wissenschaftlich formatiert werden.
# Diese Anweisung gilt bis zum Ende der aktuellen R-Sitzung.
options(scipen = 999)

CrossTable(data$f_01, data$f_03, prop.c = TRUE, prop.r = FALSE, prop.t = FALSE, expected = TRUE)
CrossTable(data$f_12, data$f_03, prop.c = TRUE, prop.r = FALSE, prop.t = FALSE, expected = TRUE)


# Bei kleinem Sample Fishers exakter Test [Seite 159]
# CrossTable(data$f_01, data$f_03, prop.c = TRUE, prop.r = FALSE, prop.t = FALSE, expected = TRUE, fisher = TRUE)


# Gruppiertes Balkendiagramm
auswertung <- table(data$f_01, data$f_03)
barplot (auswertung, legend = TRUE, beside = TRUE, xlab="Lesen Sie gerne?", ylab="Haben Sie ein Fachbuch gelesen?")
```

```
##############################################################################
# Mittelwertsvergleich [Seite 162 - 168; 175 - 177; 185 - 186]
##############################################################################
# Erstellen eines Fehlerbalkendiagramms
if (!require(psych)) {install.packages("psych")}
library(psych)

# Einfaches Fehlerbalkendiagramm
error.bars(data$f_06_1, eyes = FALSE, ylab = ("muss mir sympathisch sein"), xlab = "Konfidenzintervall")

# Gruppiertes Fehlerbalkendiagramm
error.bars.by(data$f_06_1, data$f_04, eyes = FALSE, ylab = ("muss mir sympathisch sein"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_2, data$f_04, eyes = FALSE, ylab = ("muss optisch ansprechend sein"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_3, data$f_04, eyes = FALSE, ylab = ("muss leicht verständlich sein"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_4, data$f_04, eyes = FALSE, ylab = ("muss interessante Inhalte haben"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_5, data$f_04, eyes = FALSE, ylab = ("muss einen leicht lesbaren Text haben"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_6, data$f_04, eyes = FALSE, ylab = ("muss einen hohen persönlichen Nutzen haben"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_7, data$f_04, eyes = FALSE, ylab = ("muss rasch Informationen liefern"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_8, data$f_04, eyes = FALSE, ylab = ("muss übersichtlich gestaltet sein"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
error.bars.by(data$f_06_9, data$f_04, eyes = FALSE, ylab = ("muss immer wieder Neues zu entdecken haben"),
              xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))
```

```
################################################################################
# T-Test für UNabhängige Stichproben(teile) [Seite 175 - 177] und U-Test [Seite 185 - 186]
if (!require(psych)) {install.packages("psych")}
library(psych)
describeBy(data$f_06_2, data$f_04)
boxplot(data$f_06_2~data$f_04)

# Normalverteilungsdiagramm je Subgruppe
if (!require(ggpubr)) {install.packages("ggpubr")}
library(ggpubr)
ggqqplot(data, "f_06_2", facet.by = "f_04")


# Normalverteilungstest je Subgruppe
# Fälle auswählen, wenn numerisch codiert
data_filter <- subset(data, f_04 == 1)
shapiro.test(data_filter$f_06_2)
data_filter <- subset(data, f_04 == 2)
shapiro.test(data_filter$f_06_2)

# Fälle auswählen, wenn als Text codiert
# data_gefiltert <- subset(data, f_04 == "in einer Buchhandlung")
# shapiro.test(data_gefiltert$f_06_2)
# data_gefiltert <- subset(data, f_04 == "im Versandhandel")
# shapiro.test(data_gefiltert$f_06_2)

# Test auf Varianzhomogenität
f04_kategorien <- factor(data$f_04, levels = c(1, 2), labels = c("in Buchhandlung", "im Versandhandel"))
#print(f04_kategorien)
if (!require(car)) {install.packages("car")}
library(car)
leveneTest(data = data, f_06_2~f04_kategorien)


# T-Test für UNabhängige Stichproben(teile)
t.test(data = data, f_06_2~f_04, var.equal = TRUE)
# Hier eigentlich nicht rechenbar wegen fehlender Normalverteilung und fehlender Varianzhomogenität,
# deshalb besser: Welch-T-Test
t.test(data = data, f_06_2~f_04)

# U-Test
# U-Test ist hier die parameterfreie Wahl.
wilcox.test(data$f_06_2~data$f_04, exct = FALSE, conf.int = TRUE)
```

```
# Fehlerbalkendiagramm zur Visualisierung des signifikanten Mittelwertsunterschieds
if (!require(psych)) {install.packages("psych")}
library(psych)
error.bars.by(data$f_06_2, data$f_04, eyes = FALSE, ylab = ("muss optisch ansprechend sein"), xlab = "Konfidenzintervall",
              v.labels=cbind("kaufe lieber in Buchhandlung", "lieber im Versandhandel"))


################################################################################
# T-Test für ABhängige Stichproben(teile) [Seite 168 - 170]
t.test(data$f_06_1, data$f_06_3, paired = TRUE)
```

```
################################################################################
# Korrelation [Seite 188 - 192]
################################################################################
# Normalverteilungsdagramme und -tests
if (!require(ggpubr)) {install.packages("ggpubr")}
library(ggpubr)
ggqqplot(data, "f_14")
ggqqplot(data, "f_17")
gghistogram(data, "f_14")
gghistogram(data, "f_17")
shapiro.test(data$f_14)
shapiro.test(data$f_17)
ks.test(data$f_14, "pnorm", mean=mean(data$f_14), sd=sd(data$f_14))
ks.test(data$f_17, "pnorm", mean=mean(data$f_17), sd=sd(data$f_17))


# Pearson-Korrelation
cor.test(data$f_14, data$f_17)

# Im vorliegenden Fall wegen fehlender Normalverteilung ratsam eher die
# Spearman-Korrelation
cor.test(data$f_14, data$f_17, method="spearman")


# Streudiagramm
plot(data$f_17, data$f_14, xlab = "Bücher nicht ganz freiwillig pro Jahr", ylab = "Alter",  main = "Streudiagramm")
```