Thema 1: Table 1

JSA

3/24/23

# extract R code from qmd-File  
# knitr::purl("t1\_table\_1/t1\_ablauf.qmd", output = "t1\_table\_1/t1\_ablauf.R")  
  
suppressPackageStartupMessages(library(tidyverse))  
suppressPackageStartupMessages(library(ggplot2))  
suppressPackageStartupMessages(library(flextable))  
suppressPackageStartupMessages(library(officer))

## Beispieldaten für Teil 1 erstellen

Wir erstellen Dummydaten, die aus 5 Variablen bestehen.

* Identifier der Studienteilnehmer
* Behandlungsarm
* Geschlecht
* Alter
* Blutdruck

# Startwert für Zufallszahlengenerator  
set.seed(123)  
# Zahl der Studienteilnehmer festlegen  
n <- 500  
# Folge 1 bis 500 speichern  
id <- seq(1, n)  
# Zufallswerte aus der Menge {"Test", "Reference"} mit den Wahrscheinlichkeiten (0.5, 0.5)  
treatment <- sample(c("Test", "Reference"), n, replace = TRUE, prob = c(0.5, 0.5))  
table(treatment)

treatment  
Reference Test   
 265 235

# Zufallswerte aus der Menge {"Female", "Male"} mit den Wahrscheinlichkeiten (0.5, 0.5)  
gender <- sample(c("Female", "Male"), n, replace = TRUE, prob = c(0.5, 0.5))  
table(gender)

gender  
Female Male   
 258 242

table(treatment, gender)

gender  
treatment Female Male  
 Reference 135 130  
 Test 123 112

# Gleichförmig verteilte Zufallszahlen aus dem Bereich 18 bis 65  
age <- round(runif(n, 18, 65), 0)  
summary(age)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 18.00 29.00 41.00 41.22 53.00 65.00

# Normalverteilte Zufallszahlen mit Mittelwert 140 und SD 5 und Offset für Referenz-Behandlungsarm  
bp <- round(rnorm(n, 140, 5) + ifelse(treatment == "Reference", rnorm(1, 5, 1), 0), 0)  
# Speichern als data.frame, rechteckige Tabellenstruktur wie in einem SAS data set.  
df <- data.frame(id, treatment, gender, age, bp)  
# head() gibt die ersten Zeilen einer Tabelle aus  
head(df)

id treatment gender age bp  
1 1 Reference Male 31 154  
2 2 Test Male 46 139  
3 3 Reference Male 26 149  
4 4 Test Male 58 141  
5 5 Test Male 58 139  
6 6 Reference Male 40 145

## Kategorielle Variablen auszählen

Das Geschlecht wird als kategorielle Variable erfasst und stratifiziert nach Behandlungsarm ausgezählt.

Die Anzahl der Studienteilnehmer wird bestimmt und als Basis für die Berechnung des Prozentanteils benutzt.

Die absolute und die relative Häufigkeit werden als String zusammengefasst und in der Variablen value gespeichert.

n\_trt <- df %>% group\_by(treatment) %>% summarize(cnt\_trt = n())  
n\_ref <- n\_trt %>% filter(treatment == "Reference") %>% select(cnt\_trt) %>% as.numeric()  
n\_test <- n\_trt %>% filter(treatment == "Test") %>% select(cnt\_trt) %>% as.numeric()  
n\_tot <- nrow(df)  
  
n\_trt

# A tibble: 2 × 2  
 treatment cnt\_trt  
 <chr> <int>  
1 Reference 265  
2 Test 235

n\_trt\_gender <- df %>%   
 group\_by(treatment, gender) %>%   
 summarize(cnt\_gender = n()) %>%  
 ungroup()

`summarise()` has grouped output by 'treatment'. You can override using the  
`.groups` argument.

n\_trt\_gender

# A tibble: 4 × 3  
 treatment gender cnt\_gender  
 <chr> <chr> <int>  
1 Reference Female 135  
2 Reference Male 130  
3 Test Female 123  
4 Test Male 112

df1 <- left\_join(n\_trt\_gender, n\_trt, by = c("treatment" = "treatment"))  
df1$pct <- round(df1$cnt\_gender / df1$cnt\_trt \* 100, 1)  
df1$value <- paste0(df1$cnt\_gender, " (", df1$pct, "%)")  
df1

# A tibble: 4 × 6  
 treatment gender cnt\_gender cnt\_trt pct value   
 <chr> <chr> <int> <int> <dbl> <chr>   
1 Reference Female 135 265 50.9 135 (50.9%)  
2 Reference Male 130 265 49.1 130 (49.1%)  
3 Test Female 123 235 52.3 123 (52.3%)  
4 Test Male 112 235 47.7 112 (47.7%)

Die Tabelle wird nach Selektion auf die relevanten Variablen transponiert, um die Behandlungsarme als Spalten darzustellen.

df1\_t <- pivot\_wider(df1 %>% select(treatment, gender, value), names\_from = "treatment", values\_from = "value")  
df1\_t

# A tibble: 2 × 3  
 gender Reference Test   
 <chr> <chr> <chr>   
1 Female 135 (50.9%) 123 (52.3%)  
2 Male 130 (49.1%) 112 (47.7%)

Es werden für das Aneinanderhängen der Teilergebnisse Spalten ergänzt und umbenannt.

df1\_t$Order <- 1  
df1\_t$stat\_order <- 1  
df1\_t$Label <- "Gender"  
df1\_t <- df1\_t %>% rename("Category" = "gender") %>%   
 select(Order, stat\_order, Label, Category, Test, Reference)  
df1\_t

# A tibble: 2 × 6  
 Order stat\_order Label Category Test Reference   
 <dbl> <dbl> <chr> <chr> <chr> <chr>   
1 1 1 Gender Female 123 (52.3%) 135 (50.9%)  
2 1 1 Gender Male 112 (47.7%) 130 (49.1%)

## Kontinuierliche Variable *Alter* beschreiben

Das Alter liegt als kontinuierliche Variable vor.

Die deskriptiven Statistiken werden nach Behandlungsarm stratifiziert berechnet.

Hier werden beispielhaft Mittelwert, Standardabweichung, Median und Anzahl der Werte berechnet.

df2 <- df %>% group\_by(treatment) %>%  
 summarize(m = mean(age),  
 s = sd(age),  
 md = median(age),  
 n = n()) %>%  
 ungroup()  
df2$m <- as.character(round(df2$m, 2))  
df2$s <- as.character(round(df2$s, 3))  
df2$ms <- paste0(df2$m, " \U00B1 ", df2$s)  
df2$md <- as.character(df2$md)  
df2$n <- as.character(df2$n)  
df2

# A tibble: 2 × 6  
 treatment m s md n ms   
 <chr> <chr> <chr> <chr> <chr> <chr>   
1 Reference 41.44 13.191 42 265 41.44 ± 13.191  
2 Test 40.98 13.527 41 235 40.98 ± 13.527

Die Tabelle wird nach Selektion zunächst in eine Langform gebracht, um die statistischen Größe zu vereinheitlichen und die Sortierreihenfolge zu setzen.

df2\_t <- pivot\_longer(df2 %>% select(treatment, n, ms, md), cols = -c("treatment"), names\_to = "statistic", values\_to = "value")  
df2\_t

# A tibble: 6 × 3  
 treatment statistic value   
 <chr> <chr> <chr>   
1 Reference n 265   
2 Reference ms 41.44 ± 13.191  
3 Reference md 42   
4 Test n 235   
5 Test ms 40.98 ± 13.527  
6 Test md 41

df2\_t <- df2\_t %>%   
 mutate(stat\_order = case\_when(  
 statistic == "n" ~ 1,  
 statistic == "ms" ~ 2,  
 statistic == "md" ~ 3  
 )) %>%   
 mutate(stat\_long = case\_when(  
 statistic == "n" ~ "n",  
 statistic == "ms" ~ "Mean \U00B1 SD",  
 statistic == "md" ~ "Median"  
 ))  
df2\_t

# A tibble: 6 × 5  
 treatment statistic value stat\_order stat\_long  
 <chr> <chr> <chr> <dbl> <chr>   
1 Reference n 265 1 n   
2 Reference ms 41.44 ± 13.191 2 Mean ± SD  
3 Reference md 42 3 Median   
4 Test n 235 1 n   
5 Test ms 40.98 ± 13.527 2 Mean ± SD  
6 Test md 41 3 Median

Die Tabelle wird nach Selektion auf die relevanten Variablen transponiert, um die Behandlungsarme als Spalten darzustellen.

df2\_tt <- pivot\_wider(df2\_t %>%   
 select(treatment, stat\_long, value, stat\_order),   
 id\_cols = c("stat\_long", "stat\_order"),   
 names\_from = "treatment",   
 values\_from = c("value"))  
df2\_tt

# A tibble: 3 × 4  
 stat\_long stat\_order Reference Test   
 <chr> <dbl> <chr> <chr>   
1 n 1 265 235   
2 Mean ± SD 2 41.44 ± 13.191 40.98 ± 13.527  
3 Median 3 42 41

Es werden für das Aneinanderhängen der Teilergebnisse Spalten ergänzt und umbenannt.

df2\_tt$Order <- 2  
df2\_tt$Label <- "Age"  
df2\_tt <- df2\_tt %>% rename("Category" = "stat\_long") %>%   
 select(Order, stat\_order, Label, Category, Test, Reference)  
df2\_tt

# A tibble: 3 × 6  
 Order stat\_order Label Category Test Reference   
 <dbl> <dbl> <chr> <chr> <chr> <chr>   
1 2 1 Age n 235 265   
2 2 2 Age Mean ± SD 40.98 ± 13.527 41.44 ± 13.191  
3 2 3 Age Median 41 42

## Kontinuierliche Variable *Blutdruck* beschreiben

Diese Variable wird sinngemäß berechnet.

Der Code ist identisch und wird hier nicht wiederholt. Er lädt direkt dazu ein, ihn als Funktion zu implementieren.

# A tibble: 3 × 6  
 Order stat\_order Label Category Test Reference   
 <dbl> <dbl> <chr> <chr> <chr> <chr>   
1 2 1 Blood pressure [mmHg] n 235 265   
2 2 2 Blood pressure [mmHg] Mean ± SD 140.16 ± 5.385 146.09 ± 4.813  
3 2 3 Blood pressure [mmHg] Median 140 146

## Kombinieren der Variablen

Die drei Teiltabellen werden jetzt zusammengesetzt, noch einmal nach Variable und statistischer Größe sortiert und dann auf die auszugebenden Variablen reduziert.

So könnte eine Datei aussehen, die man an PROC REPORT übergibt.

dfc <- rbind(df1\_t, df2\_tt, df3\_tt) %>% select(Label, Category, Test, Reference)  
  
head(dfc, 9)

# A tibble: 8 × 4  
 Label Category Test Reference   
 <chr> <chr> <chr> <chr>   
1 Gender Female 123 (52.3%) 135 (50.9%)   
2 Gender Male 112 (47.7%) 130 (49.1%)   
3 Age n 235 265   
4 Age Mean ± SD 40.98 ± 13.527 41.44 ± 13.191  
5 Age Median 41 42   
6 Blood pressure [mmHg] n 235 265   
7 Blood pressure [mmHg] Mean ± SD 140.16 ± 5.385 146.09 ± 4.813  
8 Blood pressure [mmHg] Median 140 146

Bei R kann man hier die Funktion flextable() nutzen, die aus einem data.frame ein flextable-Objekt generiert.

Eine Funktionsreferenz von flextable() findet sich unter <https://davidgohel.github.io/flextable/reference/index.html>.

ft <- dfc %>% flextable() %>% autofit()  
ft

| Label | Category | Test | Reference |
| --- | --- | --- | --- |
| Gender | Female | 123 (52.3%) | 135 (50.9%) |
| Gender | Male | 112 (47.7%) | 130 (49.1%) |
| Age | n | 235 | 265 |
| Age | Mean ± SD | 40.98 ± 13.527 | 41.44 ± 13.191 |
| Age | Median | 41 | 42 |
| Blood pressure [mmHg] | n | 235 | 265 |
| Blood pressure [mmHg] | Mean ± SD | 140.16 ± 5.385 | 146.09 ± 4.813 |
| Blood pressure [mmHg] | Median | 140 | 146 |

## Kopfzeilen und Fußzeilen hinzufügen

In den folgenden Zeilen werden Kopf- und Fußzeilen hinzugefügt, um die Tabelle näher zu bezeichnen.

Die Linien werden zunächst entfernt und am Schluss wieder kontrolliert hinzugefügt.

ft <- ft %>%   
 add\_header\_lines(c("Study title", "1.2.3 Table title", "FAS")) %>%  
 add\_footer\_lines(c("Abbreviations and comments", "Program name / Run date / Data cut date")) %>%  
 border\_remove()  
ft

| Study title | | | |
| --- | --- | --- | --- |
| 1.2.3 Table title | | | |
| FAS | | | |
| Label | Category | Test | Reference |
| Gender | Female | 123 (52.3%) | 135 (50.9%) |
| Gender | Male | 112 (47.7%) | 130 (49.1%) |
| Age | n | 235 | 265 |
| Age | Mean ± SD | 40.98 ± 13.527 | 41.44 ± 13.191 |
| Age | Median | 41 | 42 |
| Blood pressure [mmHg] | n | 235 | 265 |
| Blood pressure [mmHg] | Mean ± SD | 140.16 ± 5.385 | 146.09 ± 4.813 |
| Blood pressure [mmHg] | Median | 140 | 146 |
| Abbreviations and comments | | | |
| Program name / Run date / Data cut date | | | |

## Spaltenköpfe ändern

In den ersten beiden Spalten werden die Beschriftungen entfernt.

In den beiden Behandlungsarm-Spalten wird die Anzahl an Studienteilnehmern pro Behandlungsarm hinzugefügt.

ft <- ft %>% set\_header\_labels(  
 Label = "",  
 Category = "",   
 Test = paste0("Test\n(n = ", n\_test, ")"),  
 Reference = paste0("Reference\n(n = ", n\_ref, ")"),  
 Total = paste0("Total\n(n = ", n\_tot, ")")  
)  
ft

| Study title | | | |
| --- | --- | --- | --- |
| 1.2.3 Table title | | | |
| FAS | | | |
|  |  | Test (n = 235) | Reference (n = 265) |
| Gender | Female | 123 (52.3%) | 135 (50.9%) |
| Gender | Male | 112 (47.7%) | 130 (49.1%) |
| Age | n | 235 | 265 |
| Age | Mean ± SD | 40.98 ± 13.527 | 41.44 ± 13.191 |
| Age | Median | 41 | 42 |
| Blood pressure [mmHg] | n | 235 | 265 |
| Blood pressure [mmHg] | Mean ± SD | 140.16 ± 5.385 | 146.09 ± 4.813 |
| Blood pressure [mmHg] | Median | 140 | 146 |
| Abbreviations and comments | | | |
| Program name / Run date / Data cut date | | | |

## Wiederholungen ausblenden

ft\_merged <- ft %>%  
 merge\_v(j = "Label") %>%   
 valign(j = c(1), valign = "top", part = "body")  
ft\_merged

| Study title | | | |
| --- | --- | --- | --- |
| 1.2.3 Table title | | | |
| FAS | | | |
|  |  | Test (n = 235) | Reference (n = 265) |
| Gender | Female | 123 (52.3%) | 135 (50.9%) |
| Male | 112 (47.7%) | 130 (49.1%) |
| Age | n | 235 | 265 |
| Mean ± SD | 40.98 ± 13.527 | 41.44 ± 13.191 |
| Median | 41 | 42 |
| Blood pressure [mmHg] | n | 235 | 265 |
| Mean ± SD | 140.16 ± 5.385 | 146.09 ± 4.813 |
| Median | 140 | 146 |
| Abbreviations and comments | | | |
| Program name / Run date / Data cut date | | | |

## Linien hinzufügen

Die Rahmenlinien werden hinzugefügt. Hierzu wird das Paket officer benötigt.

my\_border = fp\_border(color="black", width = 2)  
  
ft\_merged <- ft\_merged %>%  
 hline\_top(border = my\_border, part = "footer") %>%  
 hline\_bottom(border = my\_border, part = "header") %>%  
 hline\_bottom(border = my\_border, part = "body") %>%  
 hline(i = c(3), border = my\_border, part = "header") %>%  
 hline(i = c(2, 5), border = my\_border, part = "body") %>%  
 hline(i = c(6), j = c(1), border = my\_border, part = "body")  
ft\_merged

| Study title | | | |
| --- | --- | --- | --- |
| 1.2.3 Table title | | | |
| FAS | | | |
|  |  | Test (n = 235) | Reference (n = 265) |
| Gender | Female | 123 (52.3%) | 135 (50.9%) |
| Male | 112 (47.7%) | 130 (49.1%) |
| Age | n | 235 | 265 |
| Mean ± SD | 40.98 ± 13.527 | 41.44 ± 13.191 |
| Median | 41 | 42 |
| Blood pressure [mmHg] | n | 235 | 265 |
| Mean ± SD | 140.16 ± 5.385 | 146.09 ± 4.813 |
| Median | 140 | 146 |
| Abbreviations and comments | | | |
| Program name / Run date / Data cut date | | | |

In der letzten Zeile gibt es evtl. einen Bug, der zur Entfernung eines Teils der Abschlusslinie oberhalb der Fußzeilen führt, wenn man die Zellen merged. Der Workaround ist der letzte hline-Aufruf.

Die Linie ist vollständig, wenn man die Tabelle als Worddokument speichert.

## Speichern als Word-Dokument

Mit dem Paket officer kann man ein flextable-Objekt als Worddokument speichern.

my\_doc <- read\_docx() %>% body\_add\_flextable(value = ft\_merged)  
print(my\_doc, target = "t1\_table\_one.docx")