Datenvisualisierung 3

Darstellung der zusammenfassenden Statistik

Daniela Palleschi

2024-01-24

Inhaltsverzeichnis

1 Readings

- Kapital 27 (A field guide to base R) in Wickham et al. (2023)
- course website: Ch. 12: base R

Learning objectives

Today we will...

- learn what base R is
- compare base R and tidyverse
- learn base R equivalents of tidyverse verbs

2 base R

- basic software containing the R programming languag
  - contains the `base` package which is required to run R
- includes several packages such as `utils` and `stats` (among others)
  - installed when you install R

2.1 tidyverse

- the Tidyverse (Wickham et al., 2019a) is a family of R-packages designed to facilitate cleaning and wrangling data
  - tidyverse packages "haben eine gemeinsame Designphilosophie und eine gemeinsame Grammatik und Datenstruktur, so dass das Erlernen eines Pakets das Erlernen des nächsten erleichtert." (Wickham et al., 2019b). - tidyverse was written in the R programming language

2.2 base R vs. tidyverse

- main goal of base R is stability
  - not many or often changes to functionality of functions
- the tidyverse is constantly adding, updating, and changing functions
- this means that tidyverse code is prone to "breaking": tidyverse code that runs today might not run in a few years if some functions or arguments have been "deprecated"

2.3 Controversy

- some people strongly prefer to use base R or the tidyverse
  - arguments for tidyverse: more human-readable, tidier, simpler for non-programmers
  - arguments for base R: "truer" R-programming, more stable
- generally, having a strong background in one and at least a basic literacy of the other is wise

2.3.1 Twitter debates

- in this Tweet we see the original post from Prof. Zorn stating that knowing the tidyverse does not equate knowing R
  - but there were many replies highlighting the benefits of the tidyverse
  - from instructors, professors (like Bodo Winter, who wrote a statistics books for linguists using R), and data scientists working in industry

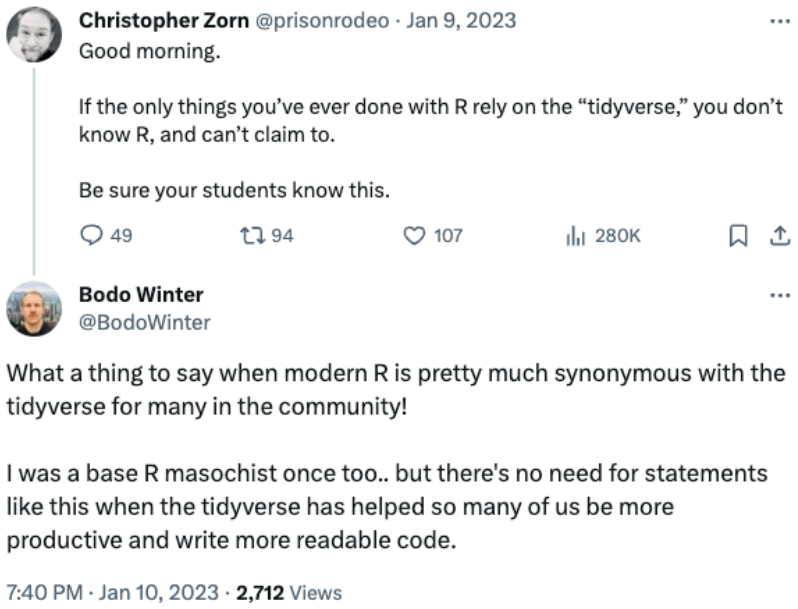


Abbildung 1: A tweet exchange about base R versus the tidyverse (original tweet above, with reply below). Click here to view tweet.

2.4 My preference

- I obviously think learning the tidyverse is important
  - the tidyverse is human centred, and we are not programmers or computer scientists
- not everybody agrees with me, but there are also a lot of people who do

3 Set-up

```
pacman::p_load(
  tidyverse,
  here
)
```

3.1 Read in data

- we'll now see our first comparison of tidyverse code versus base R code

3.2 CSV: tidyverse

Listing 1 `tidyverse`

```
df_tidy <-
  read_csv(
    here("daten", "languageR_english.csv")
  )
```

3.3 CSV: base R

Listing 2 `base R`

```
df_base <-
  read.csv(
    here("daten", "languageR_english.csv")
  )
```

3.4 Comparing outcome

- how many columns?

```
length(df_tidy)
```

```
[1] 7
```

```
length(df_base)
```

```
[1] 7
```

- what are the column names?

```
names(df_base)
```

```
[1] "AgeSubject"       "Word"             "LengthInLetters"  "WrittenFrequency"
[5] "WordCategory"     "RTlexdec"         "RTnaming"
```

```
names(df_tidy)
```

```
[1] "AgeSubject"       "Word"             "LengthInLetters"  "WrittenFrequency"
[5] "WordCategory"     "RTlexdec"         "RTnaming"
```

- how many rows?

```
nrow(df_tidy)
```

```
[1] 4568
```

```
nrow(df_base)
```

```
[1] 4568
```

- the data structure is identical

4 Wrangling columns and rows

- we'll see base R alternatives to our most common `dplyr` verbs

Listing 3 `tidyverse`

```
  df_tidy |>
    select(AgeSubject)
```

4.1 Extract variables: tidyverse

```
# A tibble: 10 x 1
   AgeSubject
   <chr>
 1 young
 2 young
 3 young
 4 young
 5 young
 6 young
 7 young
 8 young
 9 young
10 young
```

4.2 Extract variables: base R

- the dollar sign (`$`) can be used to extract a column from a dataframe (or tibble)
- this will give us a vector, whereas `dplyr::select()` preserves the dataframe/tibble attributes of the column

Listing 4 `base R`

```
  df_base$AgeSubject
```

```
 [1] "young" "young" "young" "young" "young" "young" "young" "young" "young"
[10] "young" "young" "young" "young" "young" "young" "young" "young" "young"
```

4.3 Extract variables: base R

- or we can use `dataframe[row,column]`

Listing 5 `base R`

```
# using variable name
```

```
df_base[,"AgeSubject"]
```

- we can use the name of a column in quotation marks

```
 [1] "young" "young" "young" "young" "young" "young" "young" "young" "young"
[10] "young" "young" "young" "young" "young" "young" "young" "young" "young"
```

- or we can give the index of the column, where 1 means the first column, 2 means the second column, and so on

Listing 6 `base R`

```
# using variable index
df_base[,1]
```

```
 [1] "young" "young" "young" "young" "young" "young" "young" "young" "young"
[10] "young" "young" "young" "young" "young" "young" "young" "young" "young"
```

4.4 Multiple variables: tidyverse

Listing 7 `tidyverse`

```
df_tidy |>
  select(AgeSubject, RTlexdec)
```

```
# A tibble: 10 x 2
   AgeSubject RTlexdec
   <chr>         <dbl>
 1 young          695.
 2 young          600.
 3 young          547.
 4 young          617.
```

```
 5 young          633.
 6 young          687.
 7 young          584.
 8 young          527.
 9 young          741.
10 young          536.
```

4.5 Multiple variables: baseR

- for this we need `c()`

Listing 8 `base R`

```
# using variable name
df_base[,c("AgeSubject", "RTlexdec")]
```

```
   AgeSubject RTlexdec
1       young   694.89
2       young   600.40
3       young   547.27
4       young   616.60
5       young   633.08
6       young   686.75
7       young   584.40
8       young   526.82
9       young   741.48
10      young   536.38
```

Listing 9 `base R`

```
# using variable index
df_base[,c(1, 6)]
```

```
   AgeSubject RTlexdec
1       young   694.89
2       young   600.40
3       young   547.27
```

```
4       young   616.60
5       young   633.08
6       young   686.75
7       young   584.40
8       young   526.82
9       young   741.48
10      young   536.38
```

4.6 Extract/Filter observations: tidyverse

- with the `filter()` function from `dplyr`

Listing 10 `tidyverse`

```
df_tidy |>
  filter(RTlexdec > 600 & RTnaming < 480)
```

```
# A tibble: 856 x 7
   AgeSubject Word  LengthInLetters WrittenFrequency WordCategory RTlexdec
   <chr>      <chr>           <dbl>            <dbl> <chr>           <dbl>
 1 young      doe                 3             3.91 N                695.
 2 young      pork                4             5.02 N                617.
 3 young      prop                4             4.77 N                687.
 4 young      arc                 3             4.89 N                741.
 5 young      tile                4             4.08 N                647.
 6 young      slope               5             5.80 N                633.
 7 young      pith                4             2.48 N                696.
 8 young      blitz               5             4.19 N                672.
 9 young      port                4             6.08 N                683.
10 young      plan                4             7.46 N                636.
# i 846 more rows
# i 1 more variable: RTnaming <dbl>
```

4.7 Extract/Filter observations: base R

- add these conditional statements into `[,]`
  - we need to include the dataframe name with the dollar sign preceding the column name

Listing 11 `base R`

```
df_base[df_base$RTlexdec > 600 & df_base$RTnaming < 480,]
```

```
   AgeSubject  Word LengthInLetters WrittenFrequency WordCategory RTlexdec
1       young   doe               3         3.912023            N   694.89
4       young  pork               4         5.017280            N   616.60
6       young  prop               4         4.770685            N   686.75
9       young   arc               3         4.890349            N   741.48
17      young  tile               4         4.077537            N   647.07
18      young slope               5         5.802118            N   632.54
22      young  pith               4         2.484907            N   695.86
26      young blitz               5         4.189655            N   671.59
29      young  port               4         6.084499            N   683.36
34      young  plan               4         7.462789            N   636.10
   RTnaming
1     466.4
4     460.3
6     477.1
9     453.8
17    459.3
18    476.2
22    473.3
26    469.5
29    459.3
34    470.4
```

4.8 Select single data points: tidyverse

- use `filter()` and `select()` (which we've already done before)

Listing 12 `tidyverse`

```
df_tidy |>
  filter(RTlexdec > 600, RTnaming < 480) |>
  select(AgeSubject, RTlexdec)
```

```
# A tibble: 10 x 2
```

```
   AgeSubject RTlexdec
   <chr>         <dbl>
 1 young          695.
 2 young          617.
 3 young          687.
 4 young          741.
 5 young          647.
 6 young          633.
 7 young          696.
 8 young          672.
 9 young          683.
10 young          636.
```

4.9 Select single data points: base R

- combine row and column values in `[,]`

Listing 13 `base R`

```
df_base[df_base$RTlexdec > 600 & df_base$RTnaming < 480,c("AgeSubject", "RTlexdec")]
```

```
   AgeSubject RTlexdec
1       young   694.89
4       young   616.60
6       young   686.75
9       young   741.48
17      young   647.07
18      young   632.54
22      young   695.86
26      young   671.59
29      young   683.36
34      young   636.10
```

4.10 Select single data points: base R

- again, you can replace the column names with the index value

```
   AgeSubject RTlexdec
1       young   694.89
```

Listing 14 `base R`

```
df_base[df_base$RTlexdec > 600 & df_base$RTnaming < 480,c(1, 6)]
```

```
4       young   616.60
6       young   686.75
9       young   741.48
17      young   647.07
18      young   632.54
22      young   695.86
26      young   671.59
29      young   683.36
34      young   636.10
```

4.11 Create new variables: tidyverse

- with the `mutate()` function from `dplyr`

Listing 15 `tidyverse`

```
df_tidy |>
  mutate(rt_lexdec_s = RTlexdec/1000)
```

```
# A tibble: 4,568 x 8
   AgeSubject Word   LengthInLetters WrittenFrequency WordCategory RTlexdec
   <chr>      <chr>            <dbl>            <dbl> <chr>           <dbl>
 1 young      doe                  3             3.91 N                695.
 2 young      whore                5             4.52 N                600.
 3 young      stress               6             6.51 N                547.
 4 young      pork                 4             5.02 N                617.
 5 young      plug                 4             4.89 N                633.
 6 young      prop                 4             4.77 N                687.
 7 young      dawn                 4             6.38 N                584.
 8 young      dog                  3             7.16 N                527.
 9 young      arc                  3             4.89 N                741.
10 young      skirt                5             5.93 N                536.
# i 4,558 more rows
# i 2 more variables: RTnaming <dbl>, rt_lexdec_s <dbl>
```

4.12 Create new variables: tidyverse

- define the new variable name (with `dataframe$variable`) and assign the value with the assignment operator `<-`

Listing 16 `base R`

```
df_base$rt_lexdec_s <- df_base$RTlexdec/1000
```

4.13 Summarise: tidyverse

- `summarise()` from `dplyr`

Listing 17 `tidyverse`

```
  df_tidy |>
  summarise(
    mean_lexdec = mean(RTlexdec),
    sd_lexdec = sd(RTlexdec),
    mean_naming = mean(RTnaming, na.rm = T),
    sd_naming = sd(RTnaming, na.rm = T)
  )
```

```
# A tibble: 1 x 4
  mean_lexdec sd_lexdec mean_naming sd_naming
        <dbl>     <dbl>       <dbl>     <dbl>
1        708.      115.        566.      101.
```

4.14 Summarise: tidyverse

- we have to create new objects containing the value of each operation + combine them into a data frame using the `data.frame()` function
- there are many alternative ways to do this, but this is the simplest if we only want to produce a few summary statistics

```
  mean_lexdec sd_lexdec mean_naming sd_naming
1    708.1336  114.8599    565.9233  100.8153
```

Listing 18 `base R`

```
data.frame(mean_lexdec = mean(df_base$RTlexdec),
           sd_lexdec = sd(df_base$RTlexdec),
           mean_naming = mean(df_base$RTnaming, na.rm = T),

           sd_naming = sd(df_base$RTnaming, na.rm = T))
```

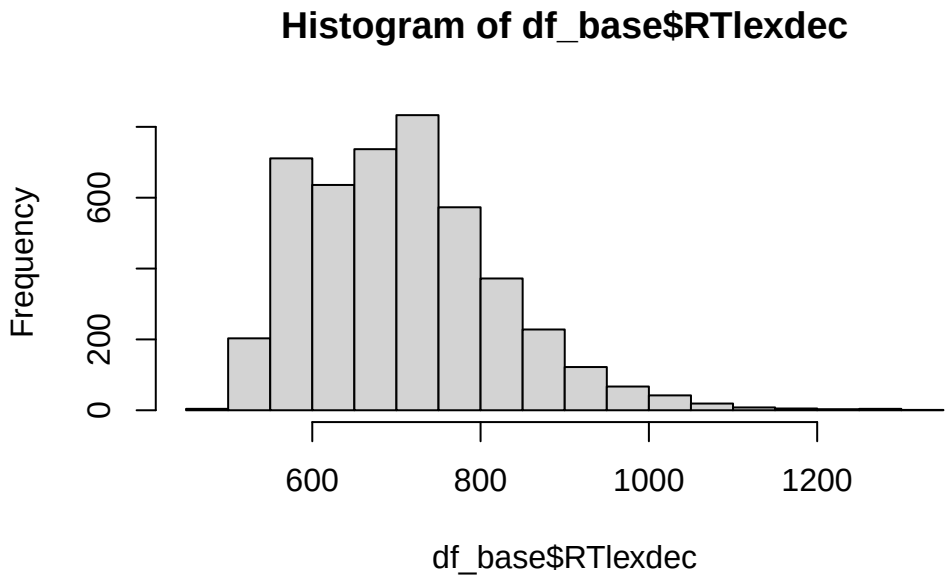
5 Plots

- `ggplot2` is popular even among people who don't use the tidyverse + this is because it has some useful features and a clean look

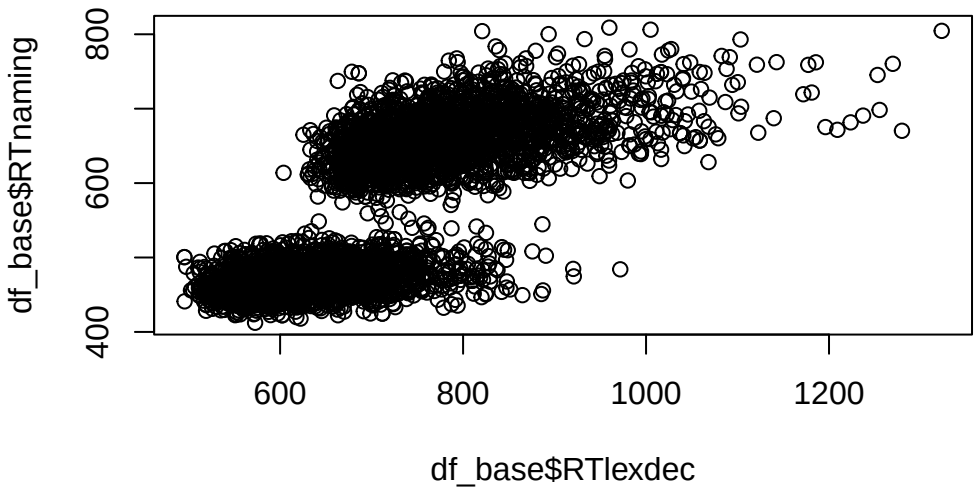
5.1 Plots: base R

- can be useful when you want to produce simple plots to get a first look at your data
  - the most useful functions are `hist()` and `plot()`
  - note that these functions work with vectors, which is why we have to use `$` to extract the columns from the data frame

```
hist(df_base$RTlexdec)
```



```r
plot(df_base$RTlexdec, df_base$RTnaming)
```



5.2 Plots: tidyverse

- as we've seen before:

```r
library(patchwork)

# histogram
fig_hist <-
  df_base |>
  ggplot() +
  aes(x = RTlexdec) +
  geom_histogram()

# scatter plot
fig_scatter <-
  df_base |>
  ggplot() +
  aes(x = RTlexdec, y = RTnaming) +
  geom_point()

fig_hist + fig_scatter
```

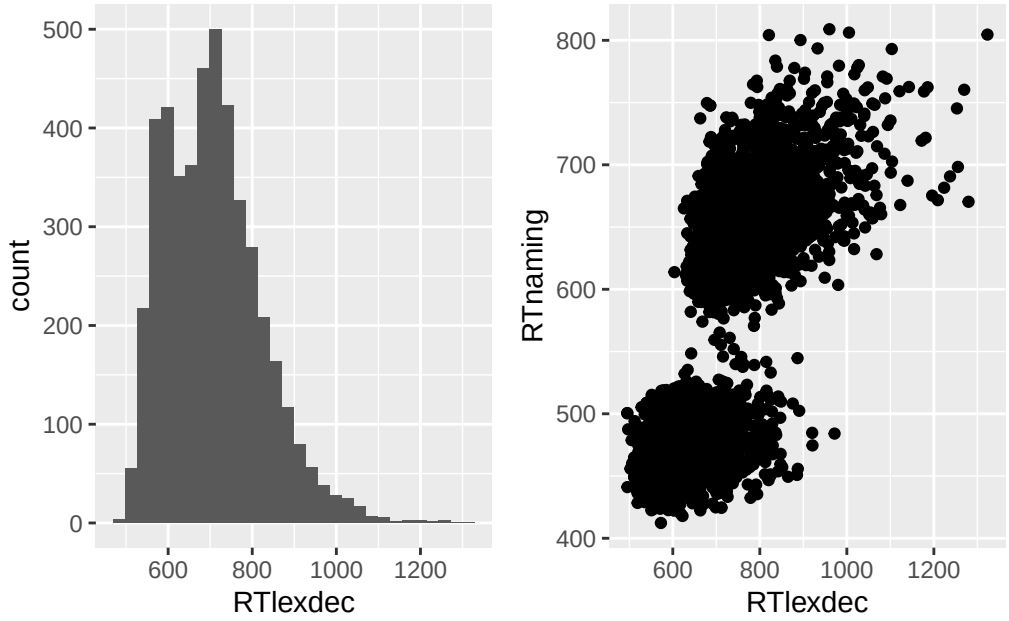


Abbildung 2: Histogram and scatterplot with ggplot2

Learning objectoves

In this chapter we...

- learned what Base R is
- compared Base R and Tidyverse
- learn Base R equivalents of common Tidyverse verbs

6 Tasks

Convert the following tidyverse code to base R. We will again use the `languageR_english.csv` dataset.

6.1 Read-in

```
df_eng <-
  read_csv(here("daten", "languageR_english.csv"))
```

6.2 Selecting columns

```
df_eng |>
  select(Word, WrittenFrequency)
```

```
# A tibble: 10 x 2
   Word   WrittenFrequency
   <chr>             <dbl>
 1 doe                3.91
 2 whore              4.52
 3 stress             6.51
 4 pork               5.02
 5 plug               4.89
 6 prop               4.77
 7 dawn               6.38
 8 dog                7.16
 9 arc                4.89
10 skirt              5.93
```

6.3 Filtering rows

```
df_eng |>
  filter(WrittenFrequency > 5.6)
```

```
# A tibble: 10 x 7
   AgeSubject Word   LengthInLetters WrittenFrequency WordCategory RTlexdec
   <chr>      <chr>            <dbl>            <dbl> <chr>           <dbl>
 1 young      stress               6             6.51 N                547.
 2 young      dawn                 4             6.38 N                584.
 3 young      dog                  3             7.16 N                527.
 4 young      skirt                5             5.93 N                536.
 5 young      are                  3            11.3  N                611.
 6 young      pipe                 4             6.00 N                563.
 7 young      guard                5             6.59 N                559.
 8 young      slope                5             5.80 N                633.
 9 young      pile                 4             6.16 N                595.
10 young      tide                 4             6.08 N                598.
# i 1 more variable: RTnaming <dbl>
```

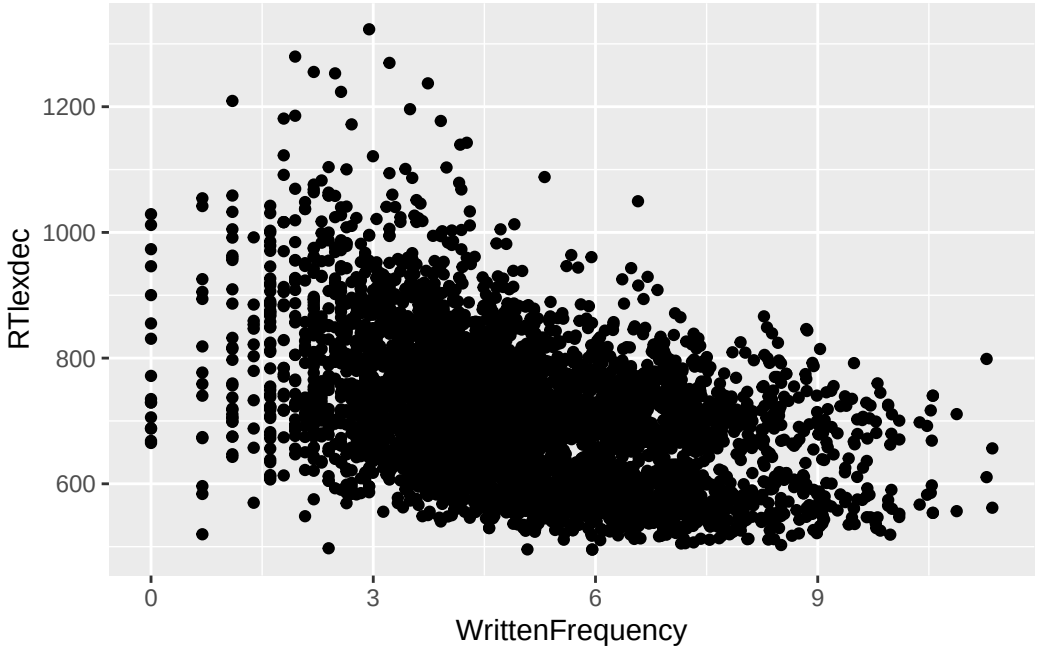
6.4 Selecting rows and columns

```
df_eng |>
  filter(WrittenFrequency > 5.6 & AgeSubject == "old") |>
  select(AgeSubject, Word, WrittenFrequency)
```

```
# A tibble: 10 x 3
   AgeSubject Word   WrittenFrequency
   <chr>      <chr>              <dbl>
 1 old        stress              6.51
 2 old        dawn                6.38
 3 old        dog                 7.16
 4 old        skirt               5.93
 5 old        are                11.3
 6 old        pipe                6.00
 7 old        guard               6.59
 8 old        slope               5.80
 9 old        pile                6.16
10 old        tide                6.08
```

6.5 Scatterplot

```
df_eng |>
  ggplot() +
  aes(x = WrittenFrequency, y = RTlexdec) +
  geom_point()
```



6.6 Tidyverse versus base R

What is your impression of base R versus the tidyverse? Based on what you've seen, would you prefer one over the other, or would you prefer one in certain cases only? There's no correct answer here.

Session Info

Hergestellt mit R version 4.3.0 (2023-04-21) (Already Tomorrow) und RStudioversion 2023.9.0.463 (Desert Sunflower).

```
print(sessionInfo(),locale = F)
```

```
R version 4.3.0 (2023-04-21)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.2.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;
```

```
attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] patchwork_1.1.3 janitor_2.2.0   here_1.0.1      lubridate_1.9.2
 [5] forcats_1.0.0   stringr_1.5.0   dplyr_1.1.3     purrr_1.0.2
 [9] readr_2.1.4     tidyr_1.3.0     tibble_3.2.1    ggplot2_3.4.3
[13] tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] utf8_1.2.3       generics_0.1.3   stringi_1.7.12   hms_1.1.3
 [5] digest_0.6.33    magrittr_2.0.3   evaluate_0.21    grid_4.3.0
 [9] timechange_0.2.0 fastmap_1.1.1    rprojroot_2.0.3  jsonlite_1.8.7
[13] fansi_1.0.4      scales_1.2.1     cli_3.6.1        rlang_1.1.1
[17] crayon_1.5.2     bit64_4.0.5      munsell_0.5.0    withr_2.5.0
[21] yaml_2.3.7       tools_4.3.0      parallel_4.3.0   tzdb_0.4.0
[25] colorspace_2.1-0 pacman_0.5.1     vctrs_0.6.3      R6_2.5.1
[29] lifecycle_1.0.3  snakecase_0.11.0 bit_4.0.5        vroom_1.6.3
[33] pkgconfig_2.0.3  pillar_1.9.0     gtable_0.3.4     glue_1.6.2
[37] xfun_0.39        tidyselect_1.2.0 rstudioapi_0.14  knitr_1.44
[41] farver_2.1.1     htmltools_0.5.5  rmarkdown_2.22   labeling_0.4.3
[45] compiler_4.3.0
```

Literaturverzeichnis

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019a). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., … Yutani, H. (2019b). Welcome to the Tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). *R for Data Science* (2. Aufl.).