

# base R

Ein Vergleich mit dem Tidyverse

Daniela Palleschi

2024-07-09

## Inhaltsverzeichnis

<b>1</b>	<b>Lesungen</b>	<b>1</b>
	<b>Lernziele</b>	<b>1</b>
<b>2</b>	<b>base R</b>	<b>2</b>
<b>3</b>	<b>Set-up</b>	<b>4</b>
<b>4</b>	<b>Mit Spalten und Zeilen hantieren</b>	<b>6</b>
<b>5</b>	<b>Plots</b>	<b>14</b>
<b>6</b>	<b>Aufgaben</b>	<b>17</b>
	<b>Session Info</b>	<b>19</b>

## 1 Lesungen

- [Kapital 27 \(A field guide to base R\)](#) in Wickham et al. (2023)
- Kurs-Website: [Kap. 12: base R](#)

## Lernziele

Heute werden wir...

- lernen, was base R ist
- Base R und Tidyverse vergleichen
- die Base-R-Äquivalente der Tidyverse-Verben kennenlernen

## 2 base R

- Basissoftware, die die Programmiersprache R enthält
  - enthält das Paket `base`, das zum Ausführen von R erforderlich ist
- enthält mehrere Pakete wie `utils` und `stats` (neben anderen)
  - wird installiert, wenn Sie R installieren

### 2.1 tidyverse

- das [Tidyverse](#) (Wickham et al., 2019a) ist eine Familie von R-Paketen, die das Bereinigen und Verwirren von Daten erleichtern sollen
  - tidyverse-Pakete “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 wurde in der Programmiersprache R geschrieben

### 2.2 base R vs. tidyverse

- Hauptziel von base R ist Stabilität
  - nicht viele oder häufige Änderungen an der Funktionalität der Funktionen
- tidyverse fügt ständig Funktionen hinzu, aktualisiert sie und ändert sie
- das bedeutet, dass der Code von tidyverse anfällig für “Brüche” ist: tidyverse-Code, der heute läuft, läuft vielleicht in ein paar Jahren nicht mehr, wenn einige Funktionen oder Argumente “veraltet” sind

## 2.3 Kontroverse

- einige Leute bevorzugen die Verwendung von Base R oder Tidyverse
  - Argumente für tidyverse: besser lesbar, aufgeräumter, einfacher für Nicht-Programmierer
  - Argumente für base R: “wahrere” R-Programmierung, stabiler
- Im Allgemeinen ist es ratsam, eine gute Kenntnis der einen und zumindest Grundkenntnisse der anderen Methode zu haben.

### 2.3.1 Twitter-Debatten

- In diesem Tweet sehen wir den ursprünglichen Beitrag von Prof. Zorn, der besagt, dass die Kenntnis des Tidyversums nicht gleichbedeutend mit der Kenntnis von R
  - aber es gab viele Antworten, die die Vorteile von Tidyverse hervorhoben
  - von Dozenten, Professoren (wie Bodo Winter, der ein Statistikbuch für Linguisten mit R geschrieben hat (Winter, 2019)) und Datenwissenschaftlern, die in der Industrie arbeiten

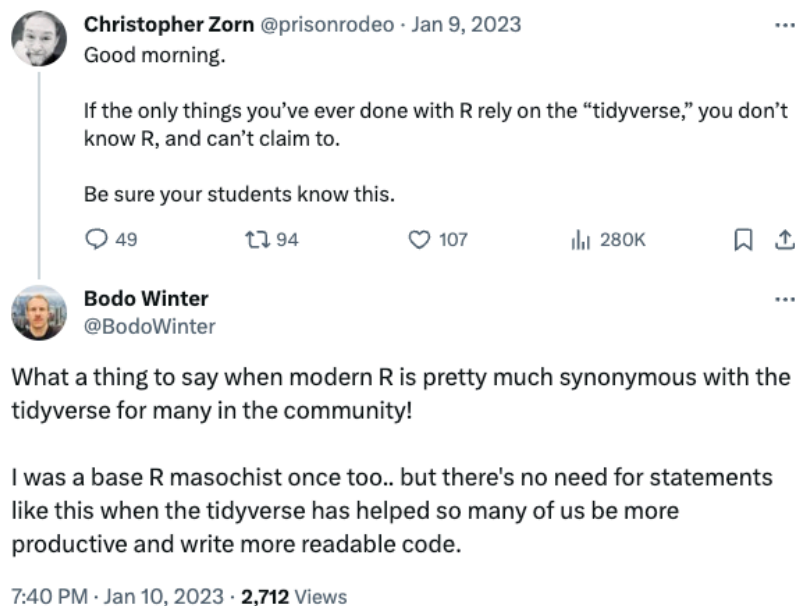


Abbildung 1: A tweet exchange about base R versus the tidyverse (original tweet above, with reply below). [Click here to view tweet.](#)

## 2.4 Meine Vorliebe

- Ich denke natürlich, dass das Erlernen des Tidyverse wichtig ist
  - das Tidyverse ist menschenzentriert, und wir sind keine Programmierer oder Informatiker
- nicht jeder stimmt mir zu, aber es gibt auch viele Leute, die mir zustimmen

## 3 Set-up

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

### 3.1 Daten einlesen

- Jetzt sehen wir unseren ersten Vergleich zwischen dem tidyverse-Code und dem Basis-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 Vergleich der Ergebnisse

- wie viele Spalten?

```
length(df_tidy)
```

```
[1] 7
```

```
length(df_base)
```

```
[1] 7
```

- Wie lauten die Spaltennamen?

```
names(df_base)
```

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

```
names(df_tidy)
```

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

- wie viele Zeilen?

```
nrow(df_tidy)
```

```
[1] 4568
```

```
nrow(df_base)
```

```
[1] 4568
```

- die Datenstruktur ist identisch

## 4 Mit Spalten und Zeilen hantieren

- sehen wir uns die Basis-R-Alternativen zu den gebräuchlichsten `dplyr`-Verben an

### 4.1 Variablen extrahieren: tidyverse

---

#### Listing 3 tidyverse

---

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

---

```
# 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 Variablen extrahieren: base R

- das Dollarzeichen (\$) kann verwendet werden, um eine Spalte aus einem Datenrahmen (oder Tibble) zu extrahieren
- dies ergibt einen Vektor, während `dplyr::select()` die Datenrahmen-/Tibble-Attribute der Spalte beibehält

---

**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 Variablen extrahieren: base R

- oder wir können Datenrahmen[Zeile,Spalte] verwenden
- wir können den Namen einer Spalte in Anführungszeichen setzen

---

**Listing 5** base R

---

```
# using variable name
df_base[,"AgeSubject"]
```

---

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

- oder wir können den Index der Spalte angeben, wobei 1 für die erste Spalte steht, 2 für die zweite Spalte und so weiter

---

**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 Mehrere Variablen: tidyverse

```
# A tibble: 10 x 2
  AgeSubject RTlexdec
  <chr>         <dbl>
```

---

**Listing 7 tidyverse**

---

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

---

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 Mehrere Variablen: baseR

- dafür brauchen wir `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

---

AgeSubject RTlexdec



---

**Listing 9** base R

---

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

---

```
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 Extrahieren/Filtern von Beobachtungen: tidyverse

- mit der Funktion `filter()` von `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 Extrahieren/Filtern von Beobachtungen: base R

- fügen Sie diese bedingten Anweisungen in [,] ein
  - wir müssen den Datenrahmennamen mit dem Dollarzeichen vor dem Spaltennamen einschließen

---

### 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 Einzelne Datenpunkte auswählen: tidyverse

- Filter() und Select() verwenden (was wir schon vorher gemacht haben)

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

---

**Listing 12 tidyverse**

---

```
df_tidy |>
  filter(RTlexdec > 600, RTnaming < 480) |>
  select(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 Einzelne Datenpunkte auswählen: base R

- Zeilen- und Spaltenwerte in [,] kombinieren

---

**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 Einzelne Datenpunkte auswählen: base R

- Auch hier können Sie die Spaltennamen durch den Indexwert ersetzen

---

### Listing 14 base R

---

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

---

	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

## 4.11 Neue Variablen erstellen: tidyverse

- mit der Funktion `mutate()` von `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 Neue Variablen erstellen: tidyverse

- Definieren Sie den Namen der neuen Variable (mit `dataframe$variable`) und weisen Sie den Wert mit dem Zuweisungsoperator `<-` zu

---

### Listing 16 base R

---

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

---

## 4.13 Zusammenfassen: tidyverse

- `Zusammenfassen()` von `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 Zusammenfassen: tidyverse

- wir müssen neue Objekte erstellen, die den Wert jeder Operation enthalten, und sie mit der Funktion `"data.frame()"` zu einem Datenrahmen zusammenfassen

- Es gibt viele alternative Möglichkeiten, dies zu tun, aber dies ist die einfachste, wenn wir nur ein paar zusammenfassende Statistiken erstellen wollen

---

**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))
```

---

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

## 5 Plots

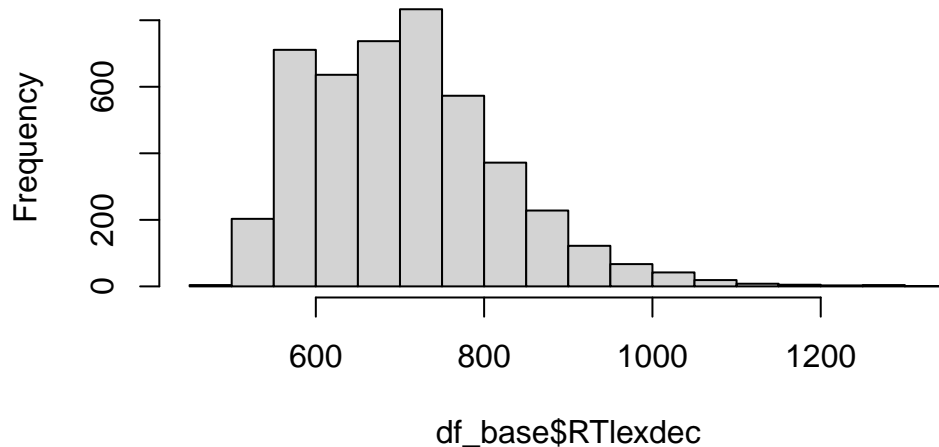
- ggplot2 ist auch bei Leuten beliebt, die tidyverse nicht benutzen
  - das liegt daran, dass es einige nützliche Funktionen und ein sauberes Aussehen hat

### 5.1 Plots: base R

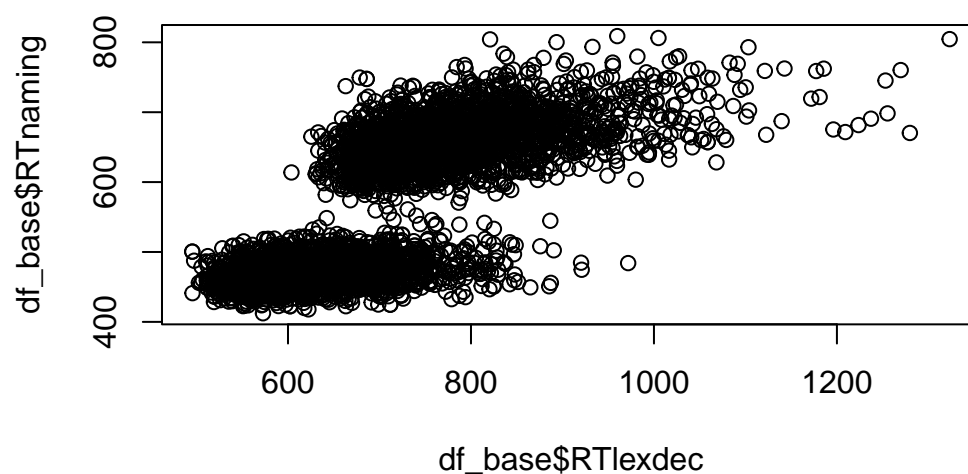
- kann nützlich sein, wenn Sie einfache Diagramme erstellen wollen, um einen ersten Blick auf Ihre Daten zu erhalten
  - Die nützlichsten Funktionen sind “hist()” und “plot()”.
  - Beachten Sie, dass diese Funktionen mit Vektoren arbeiten, weshalb wir \$ verwenden müssen, um die Spalten aus dem Datenrahmen zu extrahieren.

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

**Histogram of df\_base\$RTlexdec**



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



## 5.2 Plots: tidyverse

- wie wir es schon gesehen haben:

```
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

```

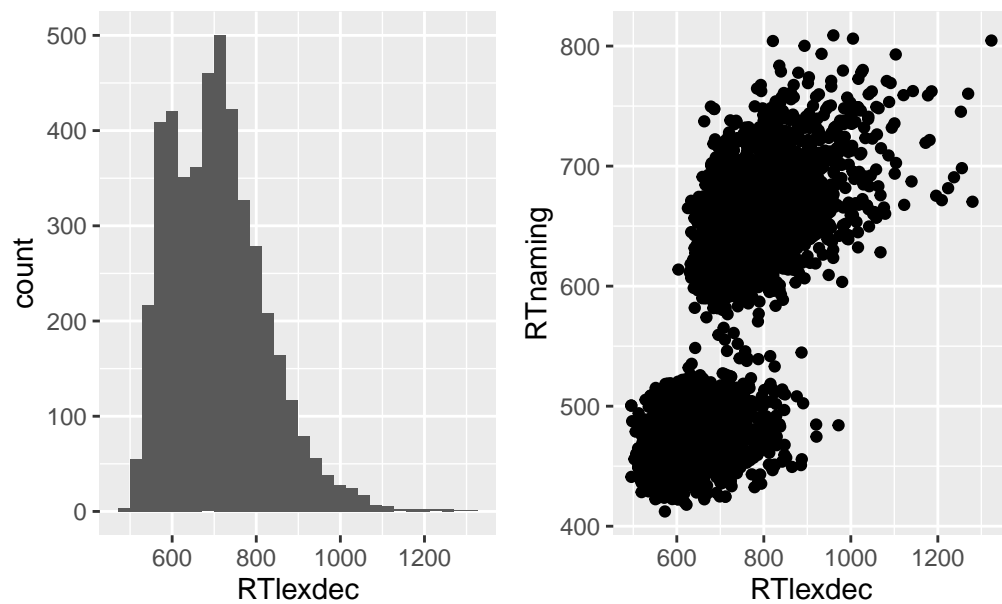


Abbildung 2: Histogram and scatterplot with ggplot2

## Lernziele

Heute haben wir...

- gelernt, was Base R ist
- Base R und Tidyverse verglichen
- lernen die Base R-Entsprechungen gängiger Tidyverse-Verben



## 6 Aufgaben

Konvertieren Sie den folgenden tidyverse-Code in Base R. Wir werden wieder den Datensatz “languageR\_english.csv” verwenden.

### 6.1 Daten einlesen

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

### 6.2 Extrahieren von Spalten

```
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 Zeilen filtern

```
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>
```

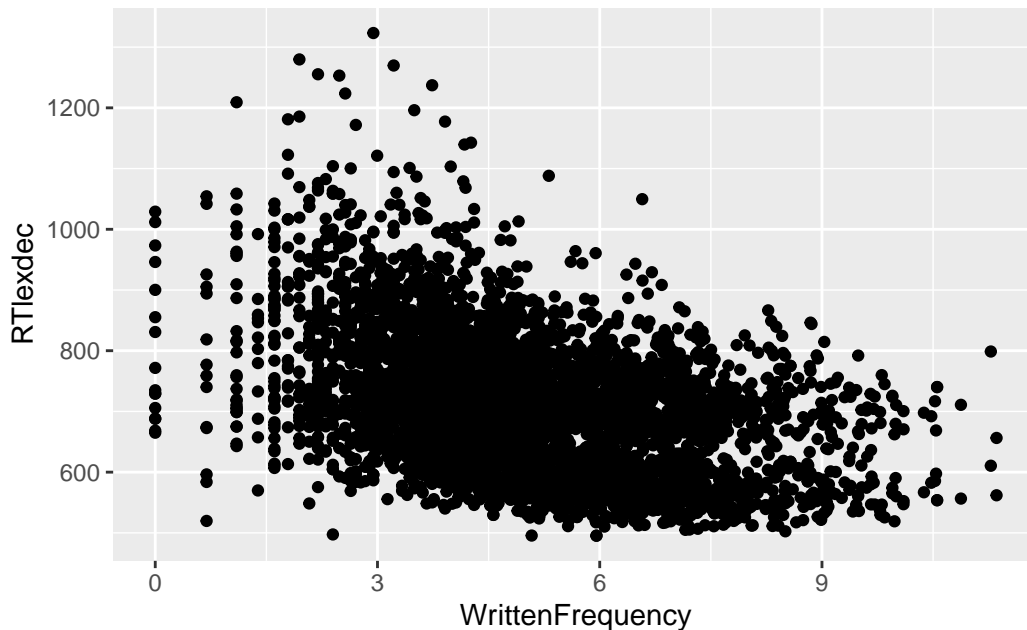
## 6.4 Filterung von Zeilen und Extraktion von Spalten

```
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 Streudiagramm

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



## 6.6 Tidyverse versus Basis-R

Wie ist Ihr Eindruck von Base R im Vergleich zu Tidyverse? Würden Sie, basierend auf dem, was Sie gesehen haben, das eine dem anderen vorziehen, oder würden Sie das eine nur in bestimmten Fällen vorziehen? Hier gibt es keine richtige Antwort.

## Session Info

Hergestellt mit R version 4.4.0 (2024-04-24) (Puppy Cup) und RStudioversion 2023.9.0.463 (Desert Sunflower).

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

```
R version 4.4.0 (2024-04-24)
Platform: aarch64-apple-darwin20
Running under: macOS Ventura 13.2.1
```

```

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;

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

other attached packages:
[1] patchwork_1.2.0  janitor_2.2.0    here_1.0.1      lubridate_1.9.3
[5] forcats_1.0.0    stringr_1.5.1    dplyr_1.1.4     purrr_1.0.2
[9] readr_2.1.5      tidyr_1.3.1      tibble_3.2.1    ggplot2_3.5.1
[13] tidyverse_2.0.0

loaded via a namespace (and not attached):
[1] utf8_1.2.4      generics_0.1.3   renv_1.0.7       stringi_1.8.3
[5] hms_1.1.3       digest_0.6.35    magrittr_2.0.3    evaluate_0.23
[9] grid_4.4.0      timechange_0.3.0 fastmap_1.1.1     rprojroot_2.0.4
[13] jsonlite_1.8.8  tinytex_0.50     fansi_1.0.6       scales_1.3.0
[17] cli_3.6.2       rlang_1.1.3      crayon_1.5.2      bit64_4.0.5
[21] munsell_0.5.1   withr_3.0.0      yaml_2.3.8        tools_4.4.0
[25] parallel_4.4.0  tzdb_0.4.0       colorspace_2.1-0  pacman_0.5.1
[29] vctrs_0.6.5     R6_2.5.1         lifecycle_1.0.4   snakecase_0.11.1
[33] bit_4.0.5       vroom_1.6.5      pkgconfig_2.0.3   pillar_1.9.0
[37] gtable_0.3.5    glue_1.7.0       xfun_0.43         tidyselect_1.2.1
[41] rstudioapi_0.16.0 knitr_1.46       farver_2.1.1      htmltools_0.5.8.1
[45] labeling_0.4.3  rmarkdown_2.26   compiler_4.4.0

```

## Literaturverzeichnis

- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Golemund, 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., Golemund, 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., & Golemund, G. (2023). *R for Data Science* (2. Aufl.).

Winter, B. (2019). Statistics for Linguists: An Introduction Using R. In *Statistics for Linguists: An Introduction Using R*. Routledge. <https://doi.org/10.4324/9781315165547>