

Troubleshooting

Was tun, wenn etwas schief geht?

Daniela Palleschi

2024-06-25

Inhaltsverzeichnis

Lernziele	1
1 Einrichten	1
2 Fehlermeldungen	2
3 Rendering-Fehler	7
4 Um Hilfe bitten	9
Session Info	13

Lernziele

In diesem Abschnitt werden wir lernen

- wie man mit allgemeinen Warnungen und Fehlermeldungen umgeht
- wie man Fehler beim Rendern von Quarto-Skripten behebt
- wo man Hilfe findet, wenn man mit fehlerhaftem Code nicht weiterkommt

1 Einrichten

```
p_load(tidyverse,  
       janitor,  
       here)
```

```
Error in p_load(tidyverse, janitor, here): could not find function "p_load"
```

1.1 Paket angeben

- wir müssen angeben, dass sie aus dem Paket `pacman` stammt
 - oder wir könnten dieser Zeile `library(pacman)` voranstellen

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

2 Fehlermeldungen

- Fehler- oder Warnmeldungen sind bei der Programmierung im Allgemeinen sehr häufig
 - auch erfahrene Programmierer machen oft die gleichen Fehler wie Sie
- Mit zunehmender Erfahrung werden Sie lernen, Codefehler effizienter zu erkennen und zu beheben

2.1 Syntaxfehler

- Syntaxfehler beziehen sich auf Fehler aufgrund von ungültigem Code
 - am häufigsten fehlende Komma oder Klammer
 - Tippfehler bei Funktions-, Objekt- oder Variablennamen

2.1.1 fehlende Zeichensetzung

- was ist das Problem mit diesem Code?

```
lexdec[c("Subject", "RT", "Correct"]
```

```
Error: <text>:1:36: unexpected '['  
1: lexdec[c("Subject", "RT", "Correct"]  
~
```

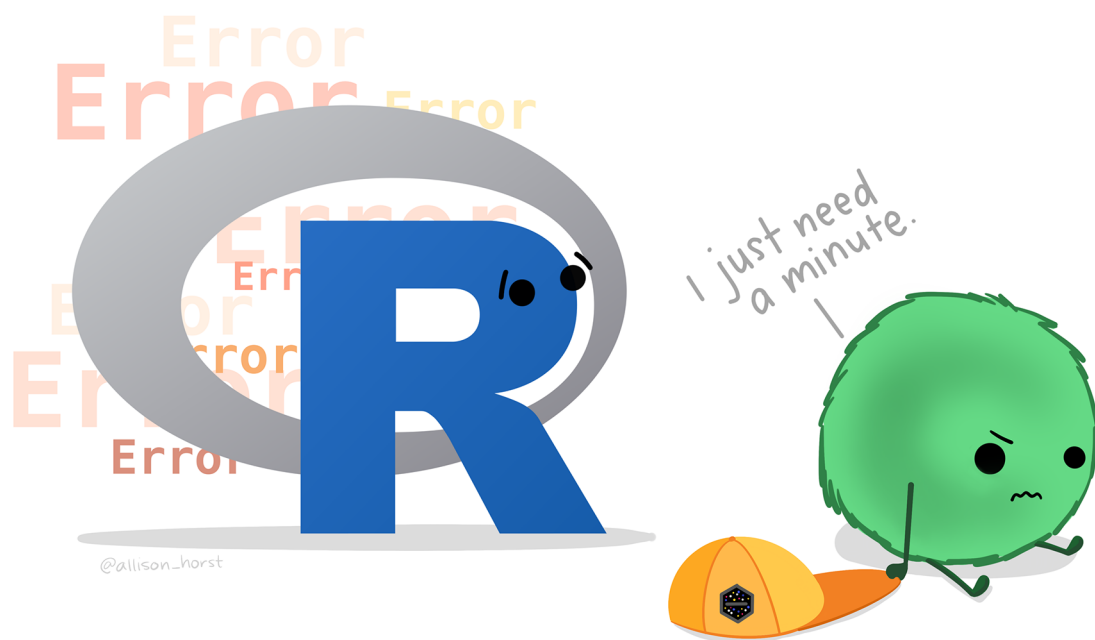


Abbildung 1: Kunstwerke von Alison Horst

2.1.2 Typfehler

- was ist das Problem mit diesem Code?

```
lexdec[c("Subject", "rt", "Correct")]
```

```
Error in `[.data.frame`(lexdec, c("Subject", "rt", "Correct")): undefined columns selected
```

2.1.3 Tidyverse-Meldungen

- das Tidyverse hat normalerweise sehr informative Meldungen

```
lexdec |>  
select("Subjct", "RT", "Correct")
```

```
Error in `select()`:  
! Can't select columns that don't exist.  
x Column `Subjct` doesn't exist.
```

- Hier wird uns genau gesagt, wo das Problem liegt: Wir haben keine Variable namens Subjct

```
lexdec |>  
select("subject", "RT", "Correct")
```

```
Error in `select()`:  
! Can't select columns that don't exist.  
x Column `subject` doesn't exist.
```

- Dieses Mal ist der Tippfehler auf die Groß- und Kleinschreibung zurückzuführen, denn unsere Variable Subjekt beginnt mit einem großen S

2.1.4 häufige Syntaxfehler

Einige häufige Syntaxfehler sind:

- falsche Großschreibung (z.B. Lexdec\$Subject statt lexdec\$Subject)
- Tippfehler (z. B. Länge(lexdec) statt Länge(lexdec))
- schließende Interpunktion, wie z. B. ein schließendes Anführungszeichen, eine Klammer oder eine eckige Klammer
- fortgesetzte Interpunktion, typischerweise fehlt ein Komma oder eine Pipe

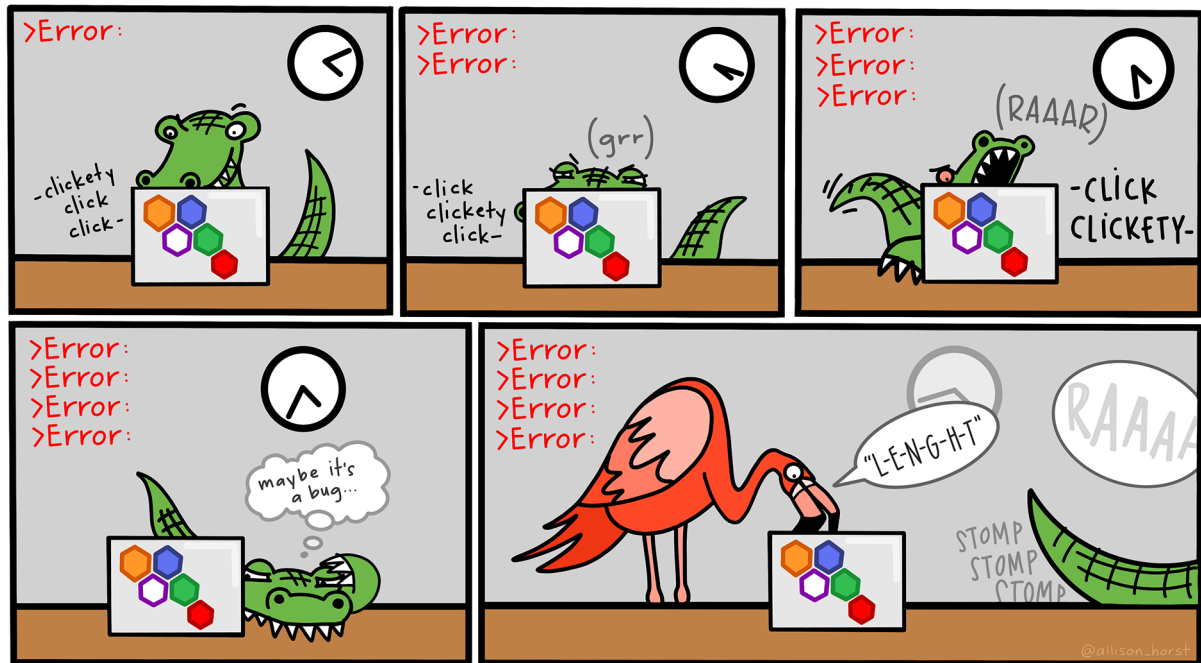


Abbildung 2: Kunstwerke von Alison Horst

2.2 RStudio-Syntax-Checker

- RStudio erkennt in der Regel Syntaxfehler und teilt Ihnen vor der Ausführung Ihres Codes mit, ob Sie etwas übersehen haben. Das folgende Code-Stück gibt zum Beispiel eine Fehlermeldung aus.

```
lexdec[c("Subject", "RT" "Correct")]
```

```
Error: <text>:1:26: unexpected string constant
1: lexdec[c("Subject", "RT" "Correct"
  ^
```

- Wir sehen ein rotes "X" neben der Zeilennummer (42) sowie eine rote Unterstreichung für den problematischen oder unerwarteten Code.

```
40  
41 {r}  
42 lexdec[c("Subject", "RT", "Correct")]  
43  
44
```

Abbildung 3: RStudio-Syntaxfehler-Warnung

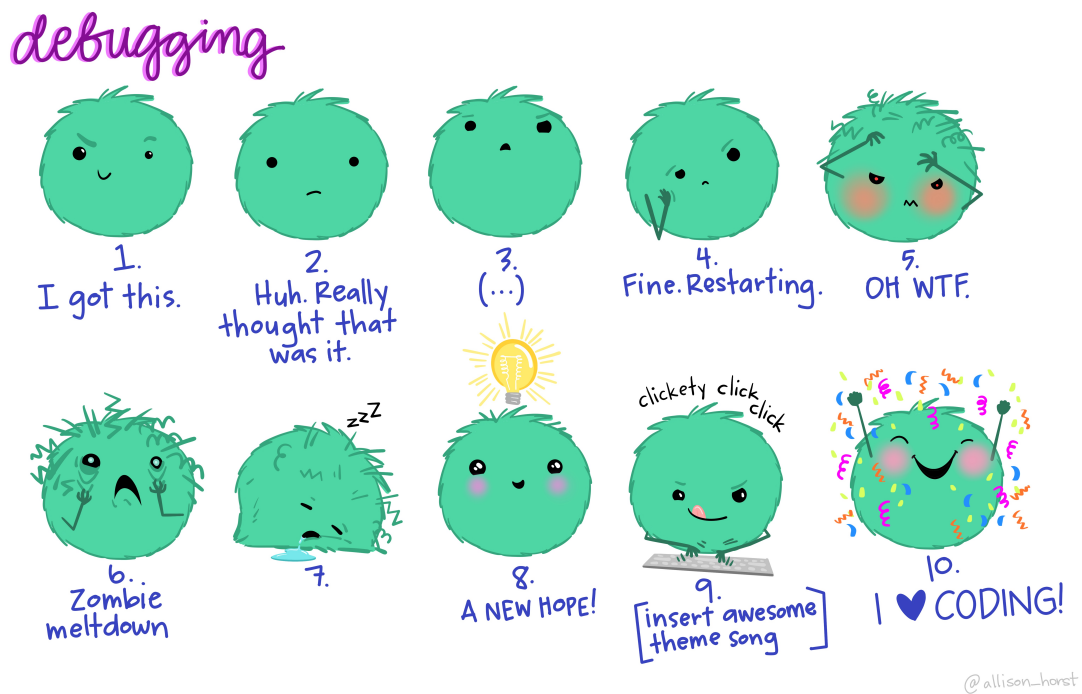


Abbildung 4: Kunstwerke von Alison Horst

3 Rendering-Fehler

- manchmal scheint unser Code gut zu laufen, aber wenn wir ein Quarto-Skript rendern, erhalten wir einen Fehler

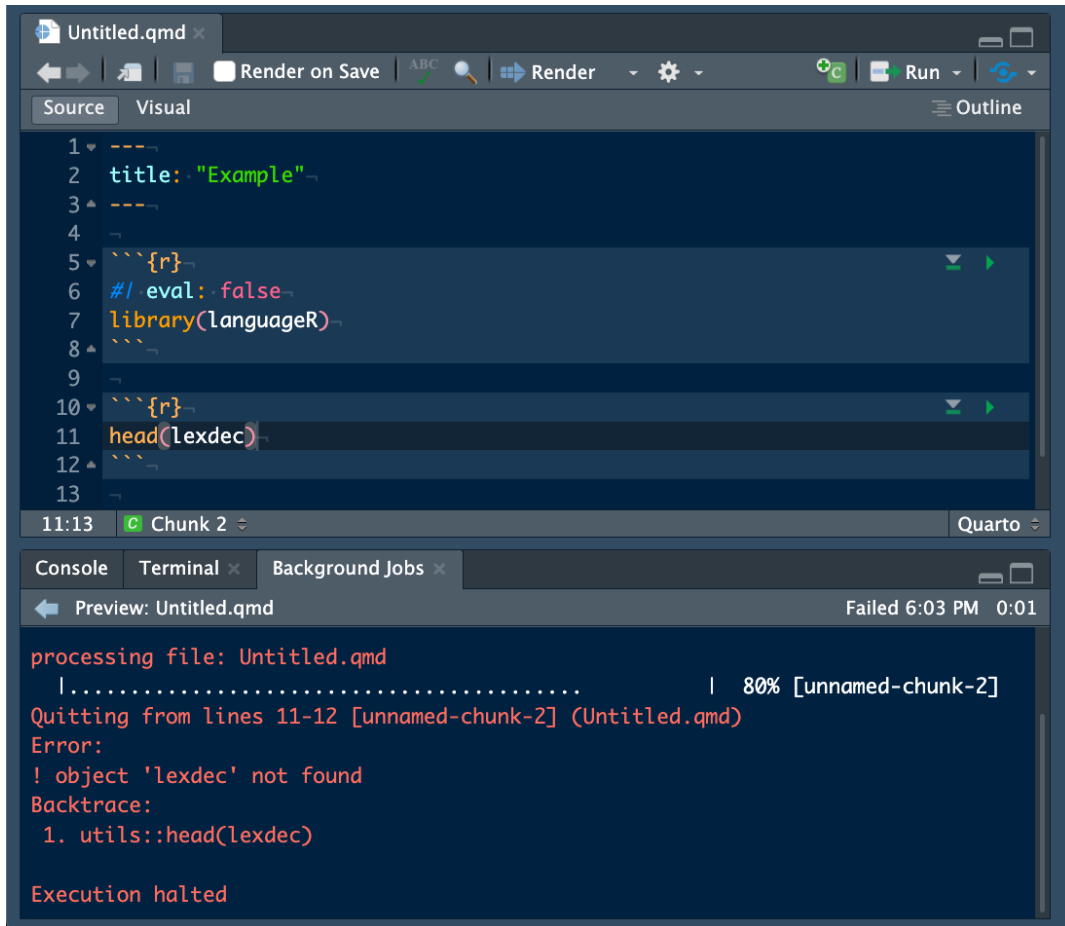


Abbildung 5: Kunstwerke von Alison Horst

3.1 eval: false

- Der erste Codeabschnitt in Abbildung 5 enthält die Option `eval: false`, und wird daher beim Rendern des Skripts nicht ausgeführt
- Der zweite Codeabschnitt versucht dann, die ersten 6 Zeilen des Datensatzes `lexdec` zu drucken, der Teil des Pakets `languageR` ist

- daher wird das Objekt 'lexdec' nicht gefunden.
- Dies ist insofern informativ: es sagt uns nicht explizit, dass es daran liegt, dass wir das Paket nicht geladen haben
 - Bei solchen Fehlern müssen wir manchmal nachforschen, indem wir unseren Code untersuchen.

3.2 Code-Zeilen und Chunk-Nummern

- vielleicht haben wir ein sehr langes Skript geschrieben und es ist nicht klar, wo die Quelle des Problems liegt, wenn wir nur den Fehlertext betrachten
 - wir erhalten auch Informationen darüber, *wo* Quarto auf ein Problem gestoßen ist: Quitting from lines 11-12 [unnamed-chunk-2] (Untitled.qmd)

3.3 RProjects

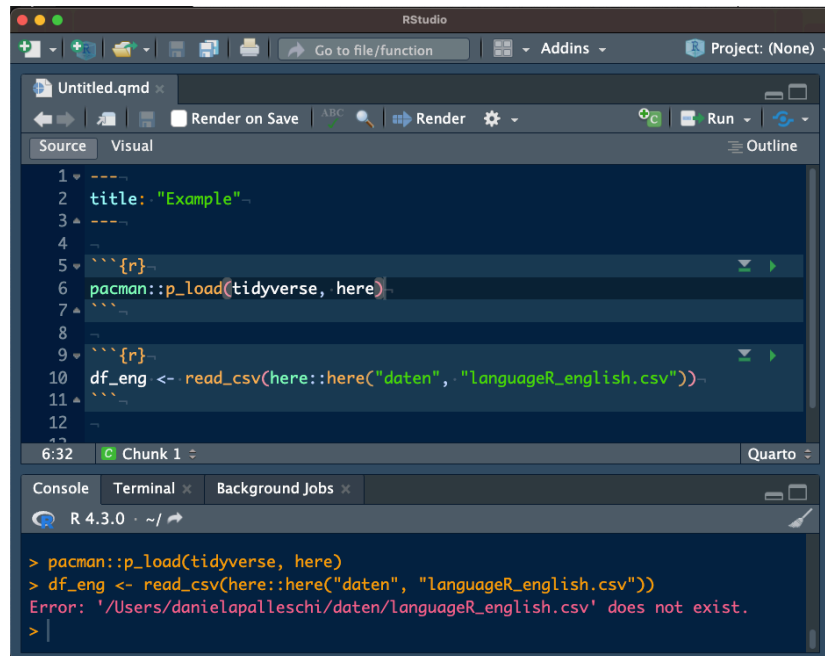


Abbildung 6: Kunstwerke von Alison Horst

RProjects

- Abbildung 6: "Users/danielapalleschi/daten/lanaugeR_english.csv" existiert nicht

- Diese Aussage ist richtig, da dieser Dateipfad auf meinem Rechner tatsächlich nicht existiert
- Beachten Sie, dass oben in der Mitte des Bildes “RStudio” steht und rechts daneben “Project: (None)”
 - wir arbeiten nicht innerhalb eines RProjekts

RProjects

- das gleiche Skript wird ohne Warnung ausgeführt, und Sie können sehen, dass wir innerhalb eines RProjekts mit dem Namen **r4ling** arbeiten
 - und die csv-Datei wird ohne Warnung geladen

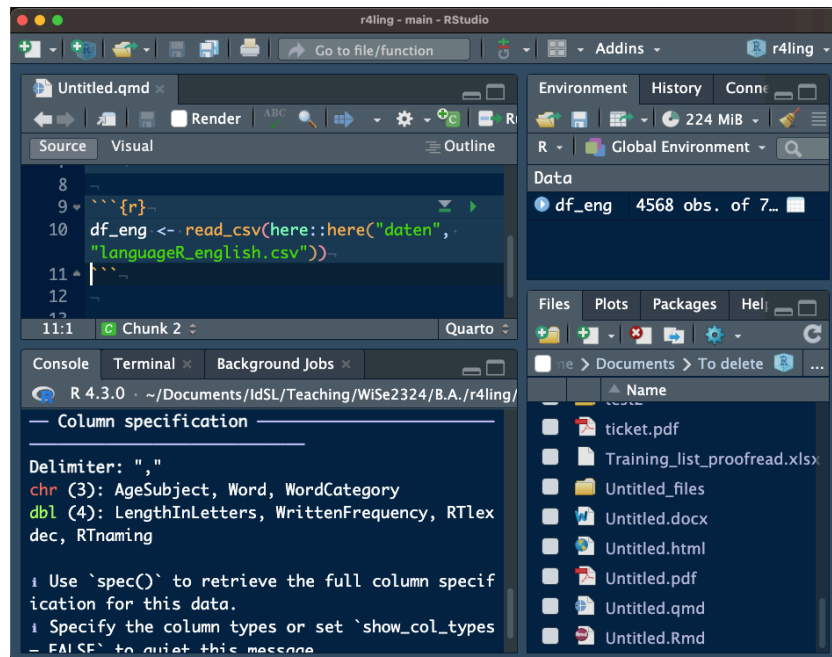


Abbildung 7: Kunstwerke von Alison Horst

4 Um Hilfe bitten

- für manche Probleme ist es schwierig, eine Lösung zu finden
- glücklicherweise gibt es eine sehr aktive R-Gemeinschaft im Internet
- wir werden uns hier die häufigsten und hilfreichsten ansehen

4.1 Google

- Im Zweifelsfall: Google!
 - Möglicherweise finden Sie einige Blogbeiträge oder Forenbeiträge, in denen das Problem und mögliche Lösungen diskutiert werden
- Dies ist ein guter erster Schritt, um das Problem zu lösen.

4.2 Stackoverflow

- eine Online-Community für Entwickler mit vielen Forenbeiträgen zu R-Programmierproblemen
- In der Regel finden Sie einen Thread, in dem jemand das gleiche Problem hatte wie Sie, und mehrere andere antworten mit möglichen Lösungen
- Wenn Sie wirklich nicht weiterkommen, können Sie sogar Ihre eigene Frage stellen! Dazu müssen Sie ein Konto erstellen

4.3 Dokumentation

- es gibt eine Fülle von Dokumentationen für Pakete (und Funktionen)
- in der Console: `?Paket` oder `?Funktion`

4.4 ChatGPT

Chatbots work best for small, discrete programming tasks, such as loading data, performing basic data manipulations and creating visualizations and websites.

— Perkel (2023)

- Wenn Sie ChatGPT Ihre Fehlermeldung oder sogar den problematischen Code geben, können Sie eine schnelle Erklärung des Problems und eine Lösung finden
 - Dies kann ein wertvolles Lernmittel sein

4.4.1 ChatGPT: Vorsicht

- ChatGPT wird Ihnen zum Beispiel nicht sagen können, dass Sie einen Variablennamen mit Kamelhülsen geschrieben haben (`cameCaseLooksLikeThis`), wenn Ihre Variablen mit Schlangenhülsen benannt sind (`snake_case_looks_like_this`)
 - oder wenn es einen anderen Tippfehler gibt, der nichts mit der Syntax zu tun hat
- Wenn Sie in der Lage sind, Fehler-/Warnmeldungen zu interpretieren und die Ursache des Problems ausfindig zu machen, ist das der schnellste Weg zur Lösung.

4.4.2 ChatGPT: Vertrauen, aber überprüfen

- es ist wichtig zu wissen, dass ChatGPT ein Tausendsassa ist, der nichts beherrscht
- “Vertrauen, aber überprüfen” (Perkel, 2023)
 - Vertrauen Sie nicht blind auf das, was die KI Ihnen sagt
- Seien Sie also skeptisch gegenüber den Lösungen, die ChatGPT Ihnen anbietet, vor allem, wenn Ihre Probleme komplexer werden
- Testen Sie immer, ob der vorgeschlagene Code nicht nur ohne eine weitere Warnung oder Fehlermeldung läuft, sondern ob er auch das erreicht, was Sie erwarten

4.4.3

Disclaimer

- ChatGPT und Datenschutz: Alles, was Sie in ChatGPT eingeben, wird an die Server von OpenAI gesendet und kann für Trainingszwecke verwendet werden. Geben Sie niemals sensible Informationen in ChatGPT ein.
- ChatGPT und akademische Integrität: Schauen Sie sich unbedingt die [Empfehlungen für das Umgehen mit Künstlicher Intelligenz in Prüfungen](#) der Humboldt-Universität an
- Aktualität: Darüber hinaus werden LLM (Large Language Models) auf Daten trainiert, die aktuell sein können oder auch nicht
 - zum Beispiel wird ChatGPT vielleicht nicht auf der neuesten Version eines Pakets trainiert und ist sich daher möglicherweise der Probleme mit einem veralteten Argument nicht bewusst
- Ethik der KI: weiteres, allgemeineres Problem bei der Verwendung von ChatGPT oder einem anderen LMM (Large Language Model)
 - is es ethisch vertretbar, sich zu sehr auf künstliche Intelligenz zu verlassen, bei der wiederholt festgestellt wurde, dass sie rassistische und geschlechtsspezifische Vorurteile hat?

Lernziele

In diesem Kapitel haben wir gelernt...



Abbildung 8: Ein Tweet über die Vorteile und Kosten der Verwendung von ChatGPT zur Korrektur von fehlerhaftem Code [Klicken Sie hier, um den Tweet zu sehen.](#)

- wie man mit allgemeinen Warnungen und Fehlermeldungen umgeht
- wie man Fehler beim Rendern von Quarto-Skripten behebt
- wo man Hilfe findet, wenn man mit fehlerhaftem Code nicht weiterkommt

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] languageR_1.5.0 here_1.0.1      janitor_2.2.0    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] gtable_0.3.5      jsonlite_1.8.8    compiler_4.4.0    renv_1.0.7
[5] tidyselect_1.2.1  snakecase_0.11.1  png_0.1-8         scales_1.3.0
[9] yaml_2.3.8        fastmap_1.1.1     R6_2.5.1          generics_0.1.3
[13] knitr_1.46        rprojroot_2.0.4   munsell_0.5.1     pillar_1.9.0
[17] tzdb_0.4.0        rlang_1.1.3       utf8_1.2.4        stringi_1.8.3
[21] xfun_0.43         timechange_0.3.0  cli_3.6.2         withr_3.0.0
[25] magrittr_2.0.3    digest_0.6.35     grid_4.4.0        rstudioapi_0.16.0
[29] hms_1.1.3         lifecycle_1.0.4   vctrs_0.6.5       evaluate_0.23
[33] glue_1.7.0        fansi_1.0.6       colorspace_2.1-0  pacman_0.5.1
[37] rmarkdown_2.26    tools_4.4.0       pkgconfig_2.0.3   htmltools_0.5.8.1
```

Literaturverzeichnis

Abschnitt 3.6 “Troubleshooting Error Messages” aus dem Webbuch [R for Graduate Students](#) von Wendy Huynh (2019)

Perkel, J. M. (2023). Six Tips for Better Coding with ChatGPT. *Nature*, 618(7964), 422–423.
<https://doi.org/10.1038/d41586-023-01833-0>