

# Troubleshooting

Was tun, wenn etwas schief geht?

Daniela Palleschi

2024-02-14

## Inhaltsverzeichnis

<b>Lernziele</b>	<b>1</b>
<b>1 Einrichten</b>	<b>2</b>
<b>2 Fehlermeldungen</b>	<b>2</b>
<b>3 Rendering-Fehler</b>	<b>6</b>
<b>4 Um Hilfe bitten</b>	<b>10</b>
<b>Session Info</b>	<b>13</b>

## 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?

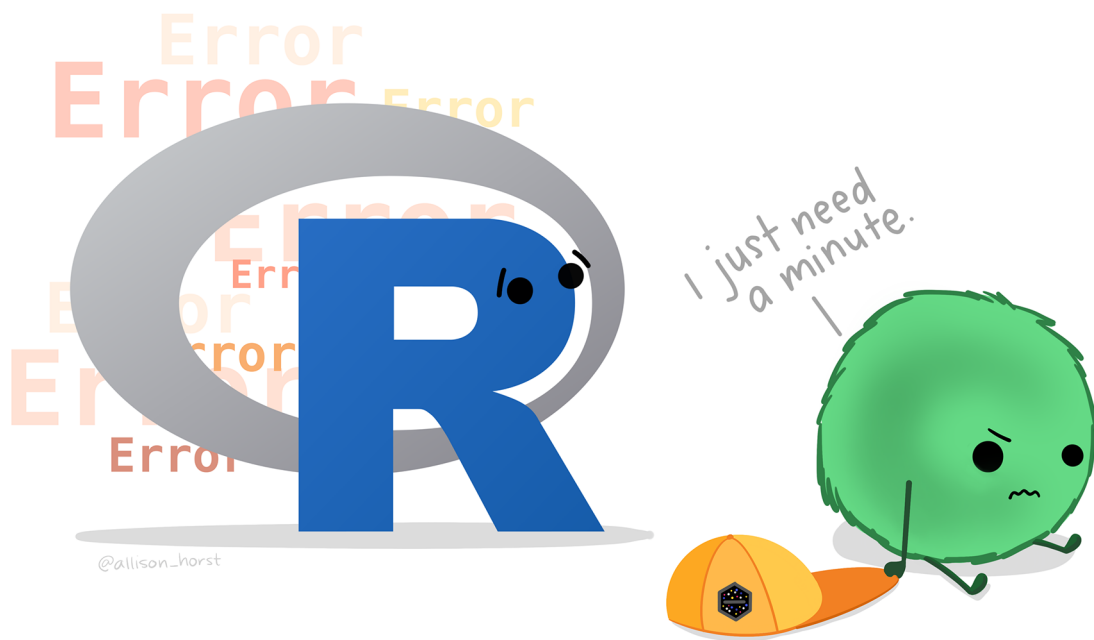


Abbildung 1: Kunstwerke von Alison Horst

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

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

### 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 subset 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 subset 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

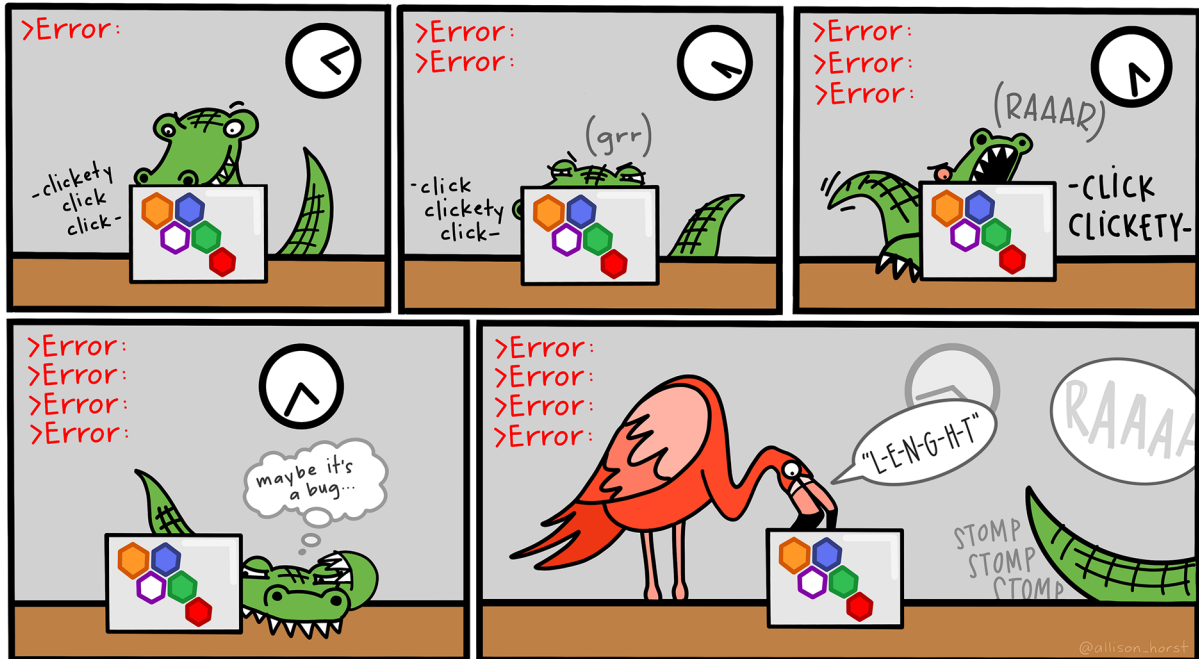


Abbildung 2: Kunstwerke von Alison Horst

## 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.

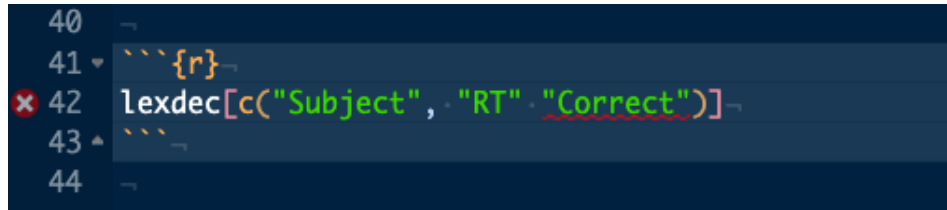


Abbildung 3: RStudio-Syntaxfehler-Warnung

### 3 Rendering-Fehler

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

#### 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)`

# debugging



1.  
I got this.



2.  
Huh. Really  
thought that  
was it.



3.  
(...)



6..  
Zombie  
meltdown



7.



8.  
A NEW HOPE

Abbildung 4: Kunstwerke von Alison Horst

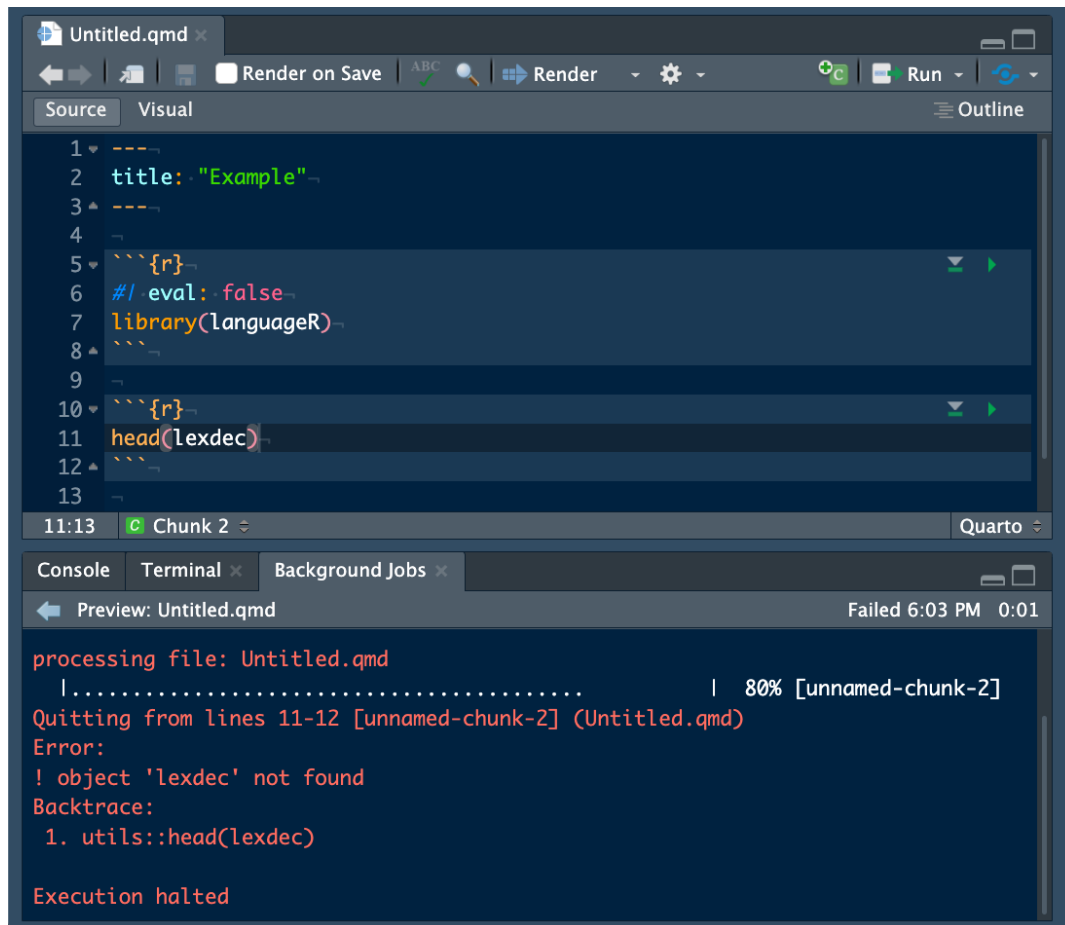


Abbildung 5: Kunstwerke von Alison Horst



### 3.3 RProjects

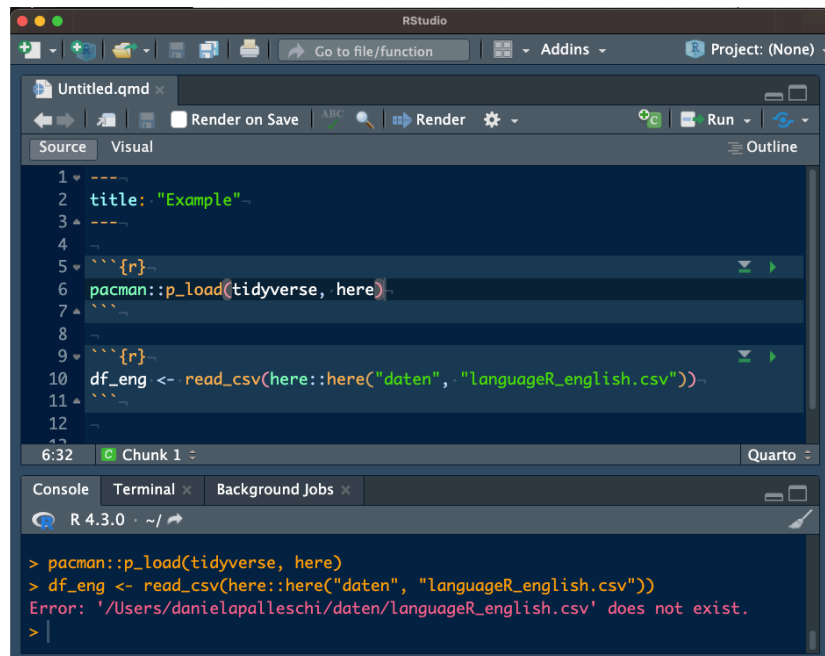


Abbildung 6: Kunstwerke von Alison Horst

### RProjects

- Abbildung 6: "Users/danielapalleschi/daten/langaugeR\_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

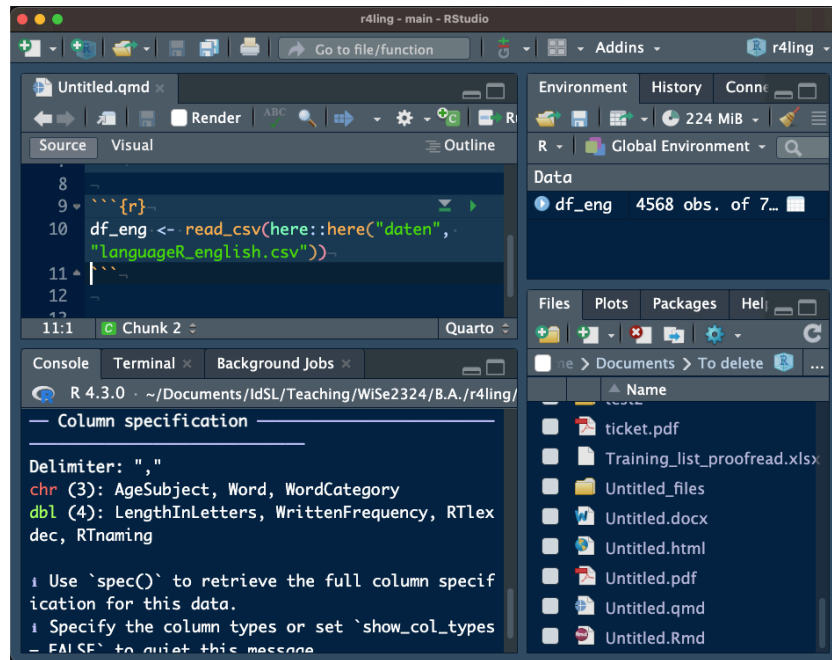


Abbildung 7: Kunstwerke von Alison Horst

## 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



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.](#)

### 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...

- 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 4.3.0 (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] languageR\_1.5.0 here\_1.0.1 janitor\_2.2.0 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] gtable\_0.3.4 jsonlite\_1.8.7 compiler\_4.3.0 tidyselect\_1.2.0  
[5] snakecase\_0.11.0 png\_0.1-8 scales\_1.2.1 yaml\_2.3.7  
[9] fastmap\_1.1.1 R6\_2.5.1 generics\_0.1.3 knitr\_1.44  
[13] rprojroot\_2.0.3 munsell\_0.5.0 pillar\_1.9.0 tzdb\_0.4.0  
[17] rlang\_1.1.3 utf8\_1.2.3 stringi\_1.7.12 xfun\_0.39  
[21] timechange\_0.2.0 cli\_3.6.1 withr\_2.5.0 magrittr\_2.0.3  
[25] digest\_0.6.33 grid\_4.3.0 rstudioapi\_0.14 hms\_1.1.3  
[29] lifecycle\_1.0.3 vctrs\_0.6.3 evaluate\_0.21 glue\_1.6.2  
[33] fansi\_1.0.4 colorspace\_2.1-0 pacman\_0.5.1 rmarkdown\_2.22  
[37] jpeg\_0.1-10 tools\_4.3.0 pkgconfig\_2.0.3 htmltools\_0.5.5

## 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>