

# Troubleshooting

Was tun, wenn etwas schief geht?

Daniela Palleschi

Humboldt-Universität zu Berlin

2024-02-14

# 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

# Einrichten

```
1 p_load(tidyverse,  
2        janitor,  
3        here)
```

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

# Paket angeben

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

```
1 pacman::p_load(tidyverse,  
2                   janitor,  
3                   here,  
4                   languageR)
```

# 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

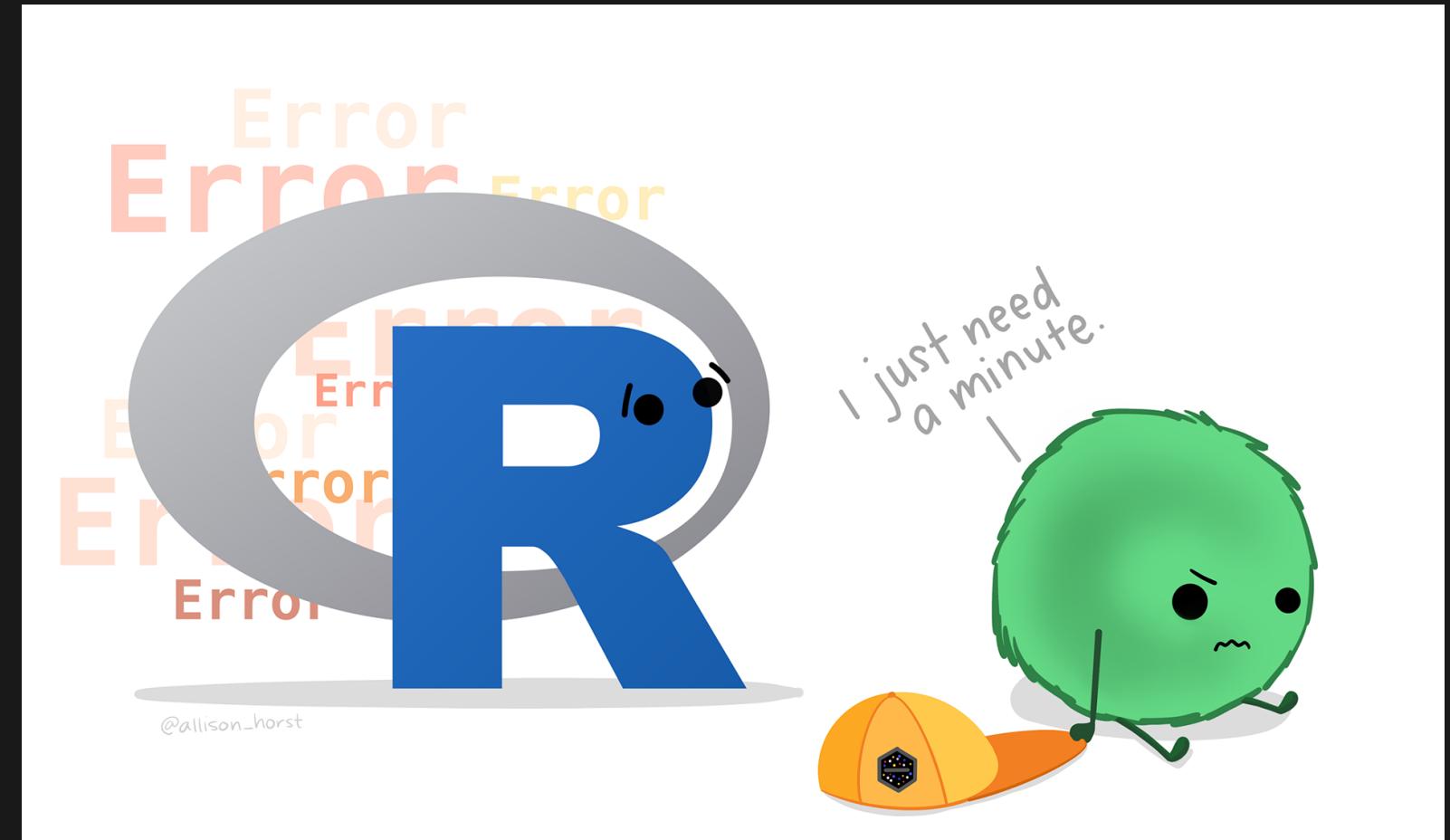


Abbildung 1: Kunstwerke von Alison Horst

# Syntaxfehler

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

# fehlende Zeichensetzung

- was ist das Problem mit diesem Code?

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

# Typfehler

- was ist das Problem mit diesem Code?

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

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

# Tidyverse-Meldungen

- das Tidyverse hat normaleweise sehr informative Meldungen

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

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

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

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

```
Error in `select()`:
! Can't subset columns that don't exist.
✖ 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

# häufige Syntaxfehler

Einige häufige Syntaxfehler sind:

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

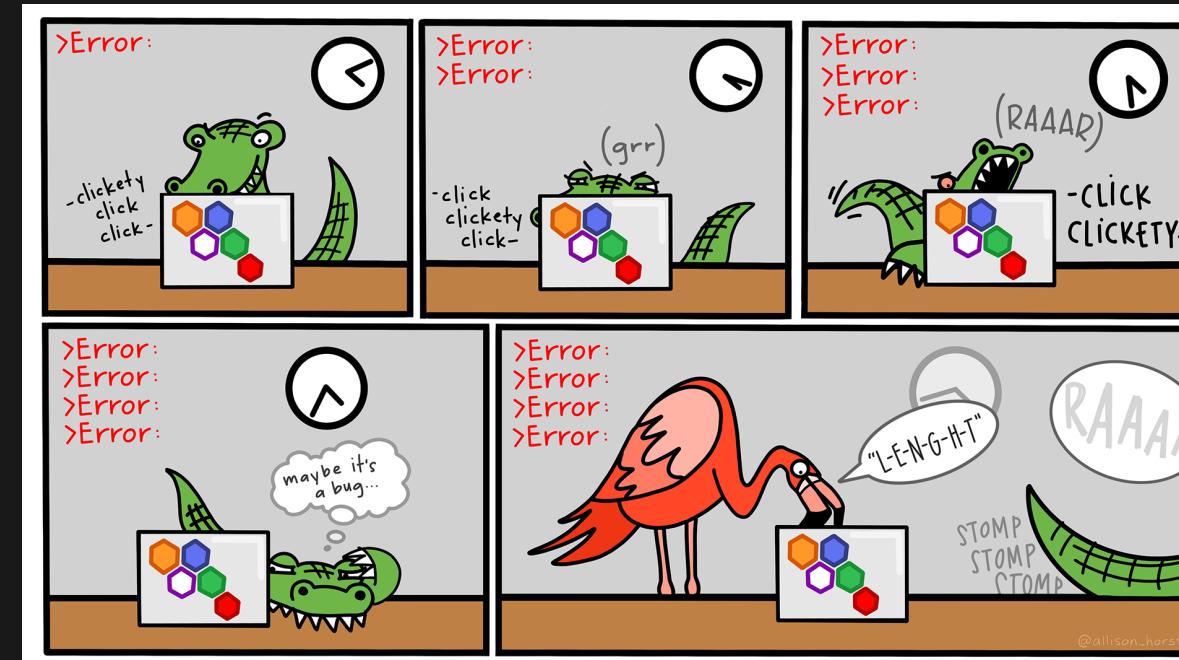


Abbildung 2: Kunstwerke von Alison Horst

# 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.

```
1 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.

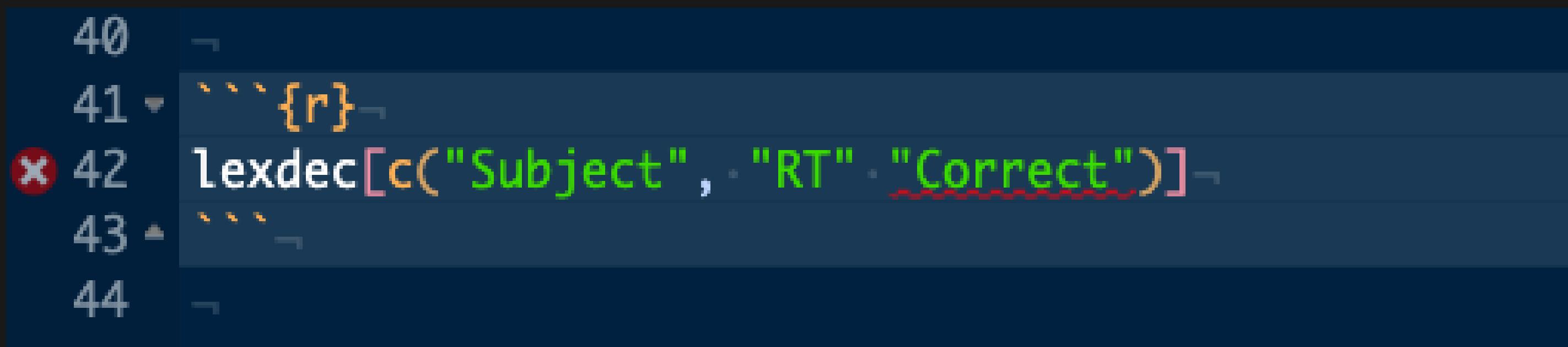
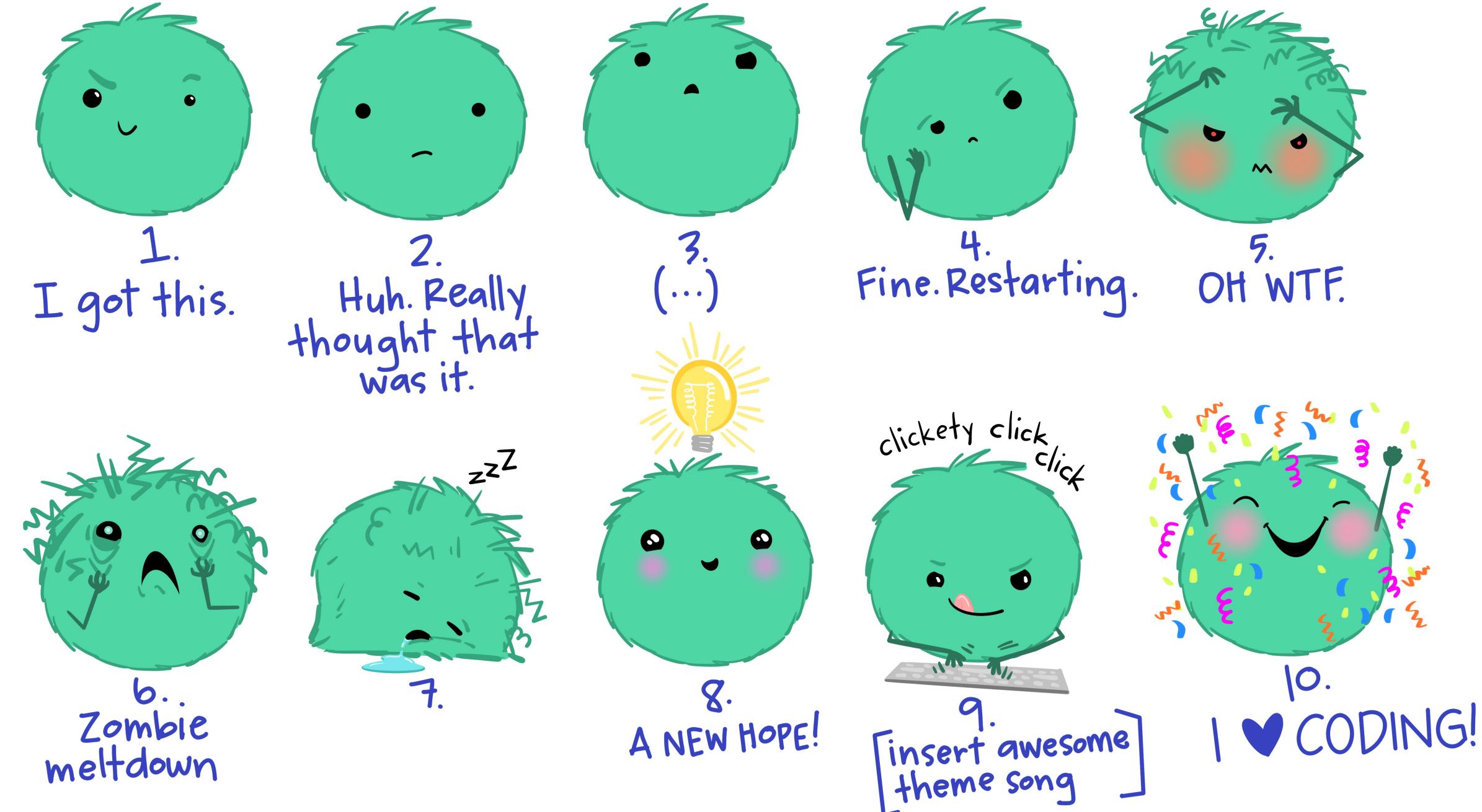


Abbildung 3: RStudio-Syntaxfehler-Warnung

# debugging



@allison\_horst

Abbildung 4: Kunstwerke von Alison Horst

# Rendering-Fehler

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

The screenshot shows a Quarto development interface with the following details:

- Title Bar:** Untitled.qmd
- Toolbar:** Render on Save, Render, Run
- Source View:** Shows R code with syntax highlighting. Lines 11 and 12 are highlighted in red, indicating an error.
- Console View:** Shows the output of the rendering process.

```
processing file: Untitled.qmd
|.....| 80% [unnamed-chunk-2]
Quitting from lines 11-12 [unnamed-chunk-2] (Untitled.qmd)
Error:
! object 'lexdec' not found
Backtrace:
1. utils::head(lexdec)

Execution halted
```
- Status Bar:** Failed 6:03 PM 0:01

Abbildung 5: Kunstwerke von Alison Horst

# **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.

# 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)`

# RProjects

The screenshot shows the RStudio interface with a Quarto document titled "Untitled.qmd" open in the Source tab. The code in the document is:

```
1 ---  
2 title: "Example"  
3 ---  
4  
5 ```{r}  
6 pacman::p_load(tidyverse, here)  
7 ---  
8  
9 ```{r}  
10 df_eng <- read_csv(here::here("daten", "languageR_english.csv"))  
11 ---  
12  
13
```

The R console at the bottom shows the following output:

```
> pacman::p_load(tidyverse, here)  
> df_eng <- read_csv(here::here("daten", "languageR_english.csv"))  
Error: '/Users/danielapalleschi/daten/languageR_english.csv' does not exist.  
>
```

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

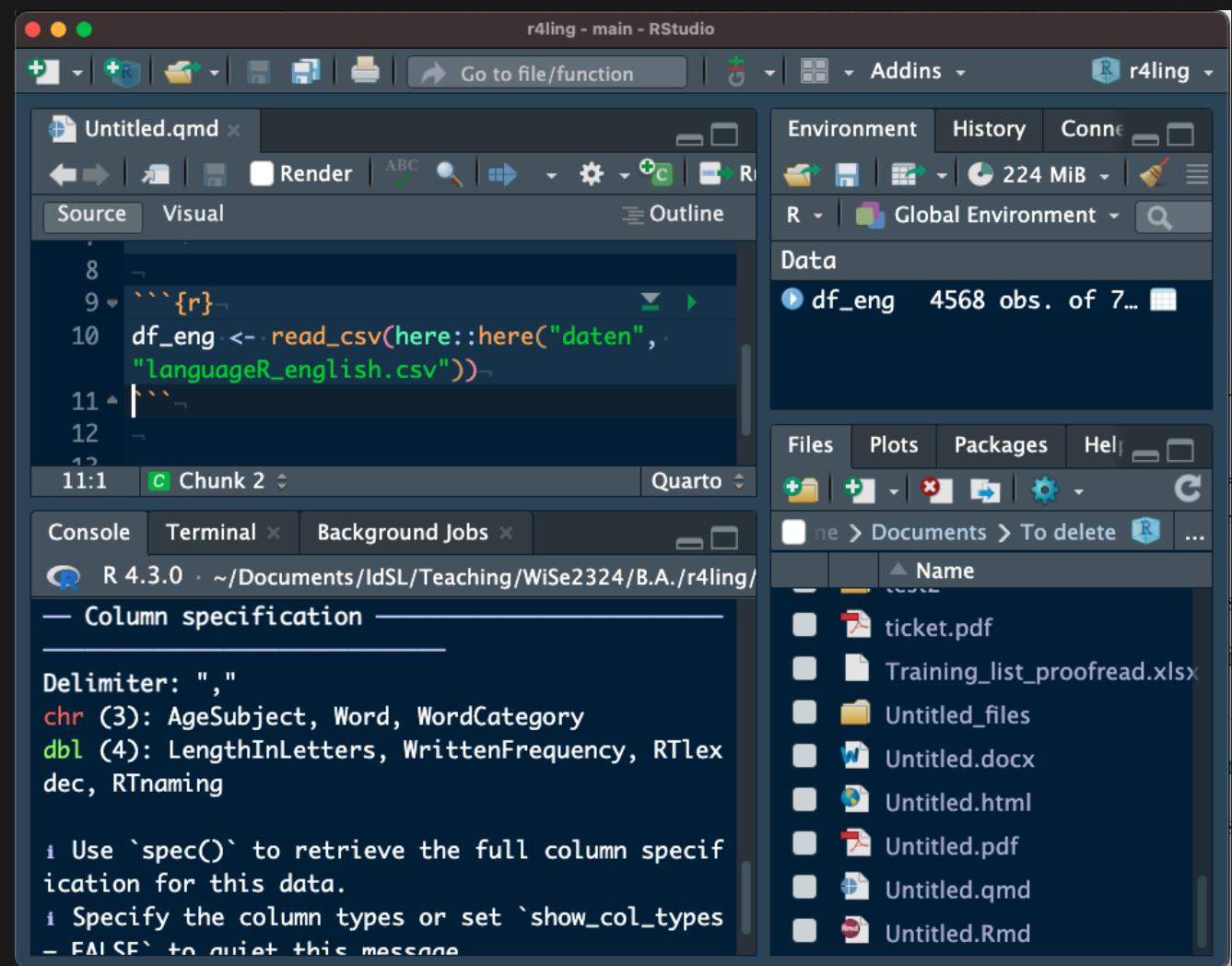


Abbildung 7: Kunstwerke von Alison Horst

# 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

# 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.

# 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

# Dokumentation

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

# 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

## ChatGPT: Vorsicht

- ChatGPT wird Ihnen zum Beispiel nicht sagen können, dass Sie einen Variablenamen 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.



# 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

## 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 rassische 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).

# 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>

