

# Reproduzierbare Forschung mit R und Quarto – Einleitung

Workshop für die GSE Wuppertal

Björn S. Siepe

June 27, 2025

# Einstieg

# Plan für Heute

## 1. Einleitung

- Was ist Reproduzierbarkeit?
- Einfache Strategien (“Easy Wins”)
- Einführung in dynamische Berichte

## 2. Praktischer Teil

- Erstellung dynamischer Berichte mit Quarto
- Typische Herausforderungen und Lösungen

## 3. Fortgeschrittene Themen

- Nächste Schritte für mehr Reproduzierbarkeit
- Tools und Best Practices

## 4. Zusammenfassung & Abschlussdiskussion

- Was haben wir gelernt?
- Offene Fragen & Ausblick

# Sehr grober Zeitplan

Uhrzeit	Programmpunkt
10:00 – 10:15	Begrüßung & Einführung
10:15 – 11:15	<b>Einleitung</b>
11:15 – 11:20	 Kurze Pause
11:20 – 12:30	<b>Praktischer Teil 1</b>
12:30 – 13:15	 Mittagspause
13:15 – 14:20	<b>Praktischer Teil 2</b>
14:20 – 14:30	 Kurze Pause
14:30 – 15:30	<b>Fortgeschrittene Themen</b>
15:30 – 16:00	<b>Zusammenfassung &amp; Abschlussdiskussion</b>

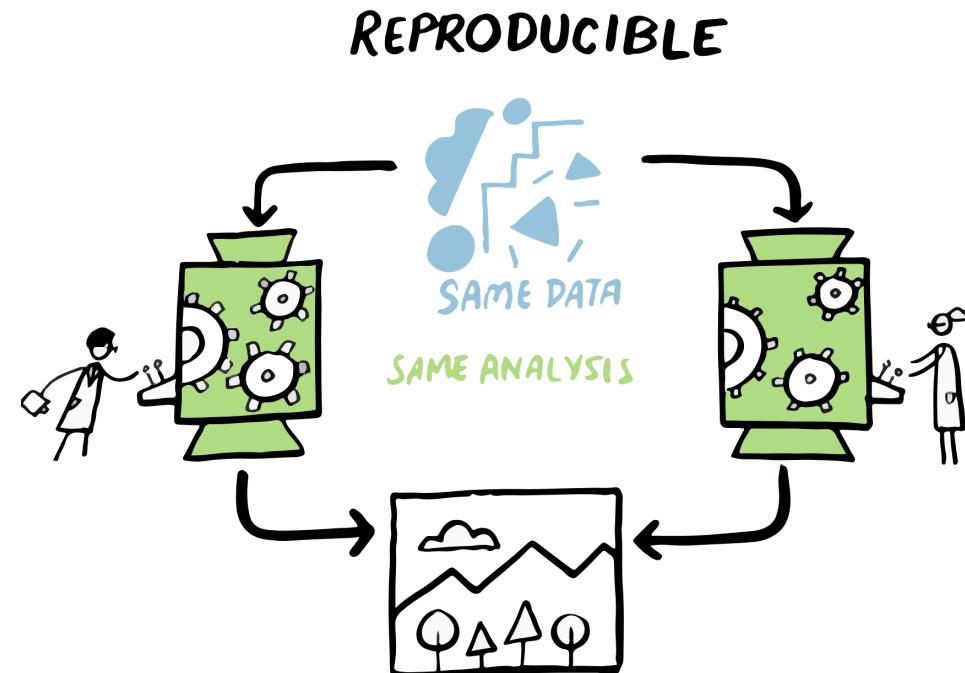
# Vorabinformation

Teile dieses Workshops beruhen teils verbatim oder sehr nah auf folgenden Ressourcen, welche ich mit einer CC-BY Lizenz verwende: - [CRS Primer on dynamic reporting](#) - Molo, Fraga González, and Pawel (2025) - Peikert and Diemerling (2025) - Teils recht opinionated  
TODO Link zu Materialien nochmals.

# Computational reproducibility

Eine sehr einfache Definition:

Die gleiche Analyse der gleichen Daten führt zu den gleichen Ergebnissen



# Example

## Computational reproducibility of Jupyter notebooks from biomedical publications ⓘ

Sheeba Samuel ✉, Daniel Mietchen ✉ Author Notes

GigaScience, Volume 13, 2024, giad113,

<https://doi.org/10.1093/gigascience/giad113> 

Published: 11 January 2024 Article history ▾

- Es wurde versucht, 22.578 Jupyter-Notebooks, die mit 3.467 Publikationen aus PubMed verknüpft sind, erneut auszuführen.
- Für 10.388 dieser Notebooks konnten alle angegebenen Abhängigkeiten erfolgreich installiert werden
- 1.203 Notebooks liefen ohne Fehler durch
- 879 Notebooks (4 %) lieferten Ergebnisse, die mit denen des Originals identisch waren

→ Computational reproducibility kann schwierig sein!

# Terminologie

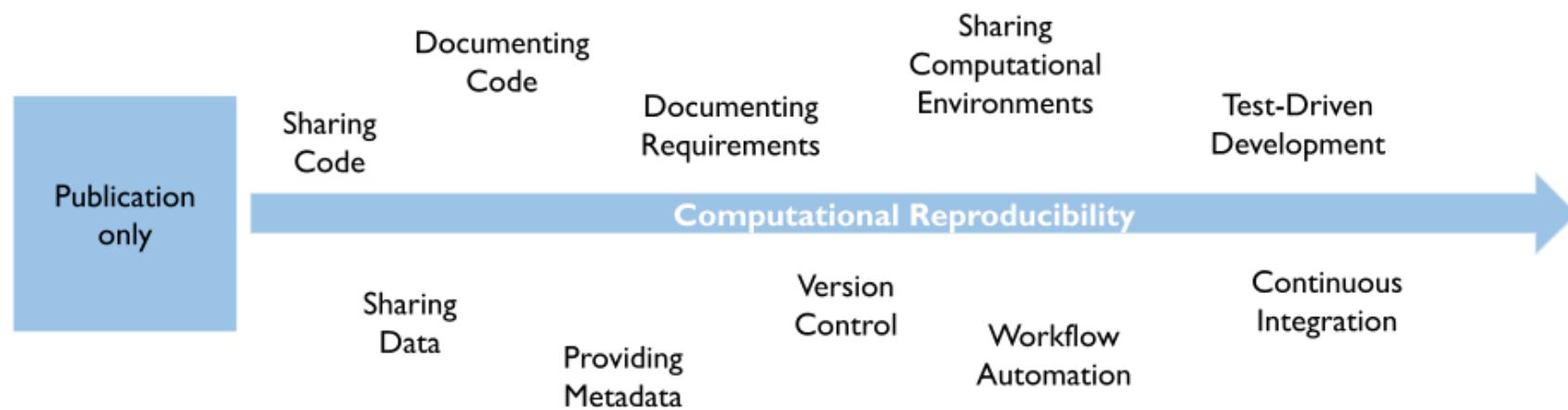
		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

The Turing Way Community ([2022](#))

# Computational Reproducibility

Molo, Fraga González, and Pawel (2025)

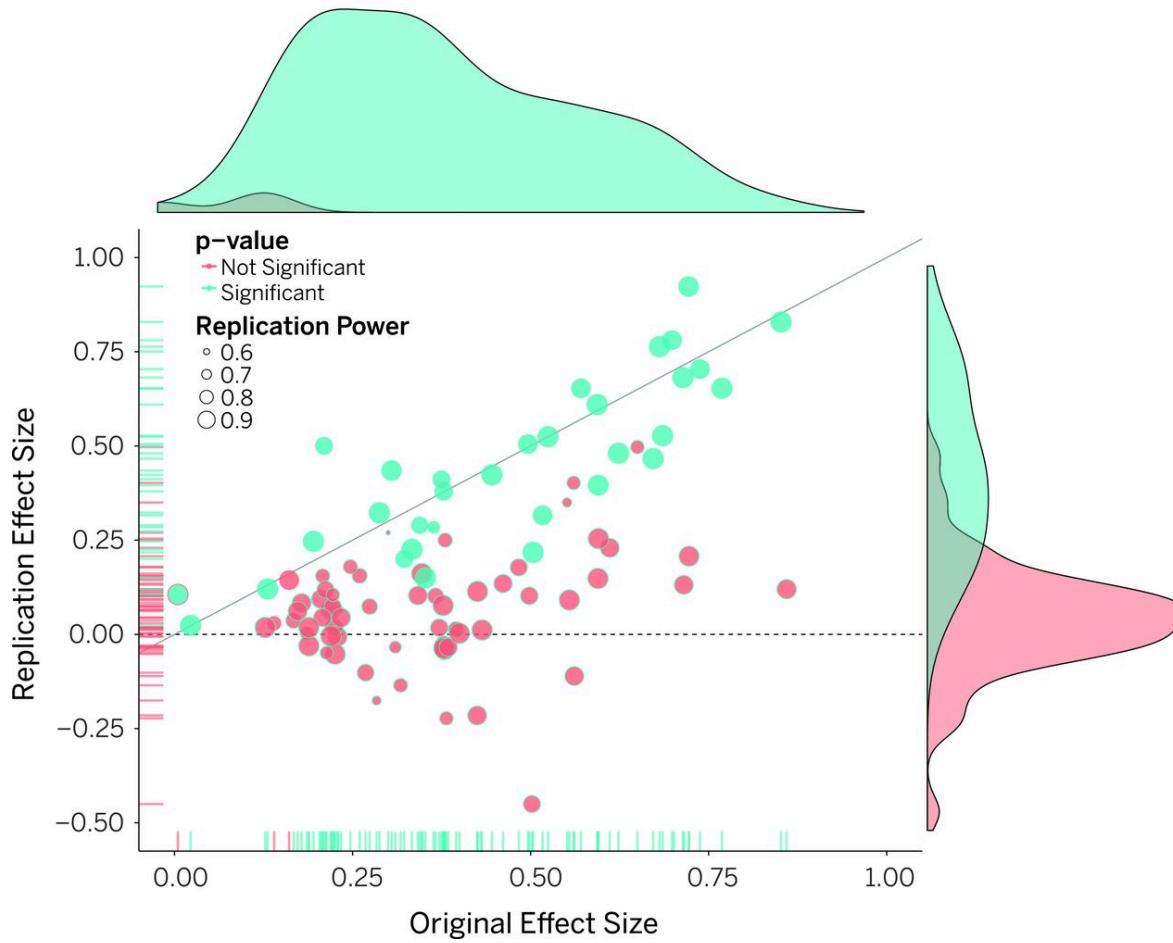
Reproducible	Replicable
Robust	Generalisable



From Molo, Fraga González, and Pawel (2025), inspired by Peng (2011) and Peikert and Brandmaier (2021)

# Replicability: Beispiel

Reproducible	Replicable
Robust	Generalisable



Open Science Collaboration (2015)

# Robustness: Beispiel

Reproducible	Replicable
Robust	Generalisable

## REPRODUCIBILITY GOES ON TRIAL IN ECOLOGY

With same data sets, 246 biologists get different results – showing how analytical choices drive conclusions.

By Anil Oza

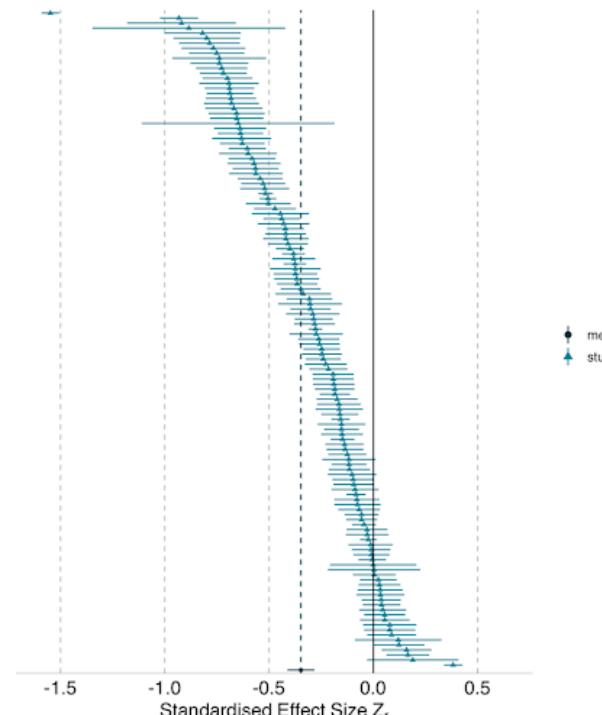
In a massive exercise to examine reproducibility, more than 200 biologists analysed the same sets of ecological data – and got widely divergent results. The first sweeping study of its kind in ecology demonstrates how much results in the field can vary, not because of differences in the environment, but because of scientists' analytical choices (E. Gould *et al.* Preprint at EcoEvoRxiv <https://doi.org/gswhwr>; 2023).

"There can be a tendency to treat individual papers' findings as definitive," says Hannah Fraser, an ecology meta researcher at the University of Melbourne in Australia and a co-author of the study. But the work shows that "we really can't be relying on any individual result or any individual study to tell us the whole story".

Variation in results might not be surprising, but quantifying that variation in a formal study could catalyse a larger movement to improve reproducibility, says Brian Nosek,

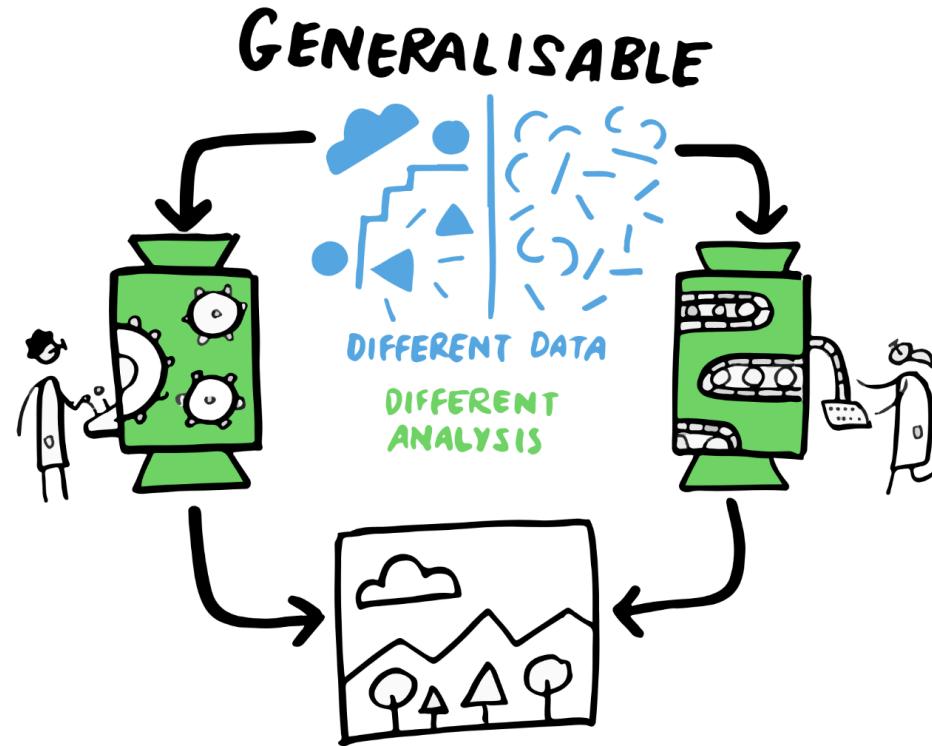
Nature | Vol 622 | 26 October 2023 | 677

Gould et al. (2023)



# Generalizability

Reproducible	Replicable
Robust	Generalisable



The Turing Way Community and Scriberia ([2024](#))

→ Reproducibility + Robustness + Replicability necessary for Generalizability!

# Jeder Schritt zählt

Aber reproduzierbare Forschung ist schwierig und kann sich überwältigend anfühlen.

Kein Grund zur Verzweiflung! Mit jedem Schritt::

- seid Ihr einen Schritt näher an reproduzierbarer Forschung
- verbessert Ihr die Qualität eurer Forschung und erleichtert euer Leben
- lernt Ihr breit anwendbare technische Skills



# Easy Wins

# Projektstruktur

## Vorher: Chaos

- Paper\_final\_wirklich\_final.docx
- daten.xlsx code\_alt.R
- code\_neu\_funktioniert.R
- plot1.png

## Nachher: Ordnung

```
projekt/
└── data/
    ├── raw/
    └── processed/
    └── scripts/
    └── output/
    └── README.md
```



### Praxis-Tipp

Erstellt eine Standard-Ordnerstruktur und kopiert diese für neue Projekte!

# Dateinamen

## ✗ Problematisch

- Daten.xlsx
- neue analyse.R
- plot final FINAL.png
- Fragebogen - Kopie (2).docx



## ✓ Besser

- 2025-06-28\_fragebogen-daten-roh.xlsx
- 01\_datenbereinigung.R
- 02\_deskriptive-statistik.R
- abb01\_altersverteilung.png

# Code-Dokumentation

## ✗ Unzureichend:

Unklarer Name, keine Dokumentation:

```
1 logmlis <- function(x){  
2   lapply(x, function(x){  
3     y <- unlist(x)  
4     y <- y[!is.na(y)]  
5     y <- y[y != ""]  
6     return(mean(log(y)))  
7   })  
8 }
```



## Besser:

Klarer Name, konzise Dokumentation:

```
1 log_mean_list <- function(x) {  
2   # Calculates the logarithmic mean of a list of numerical values  
3   # Removes NA and empty values before the calculation  
4   y <- unlist(x)  
5   y <- y[!is.na(y)]  
6   y <- y[y != ""]  
7   return(mean(log(y)))  
8 }
```

# Code-Dokumentation

- Viele unterschiedliche Meinungen
- In der Wissenschaft: Lieber etwas zu viel als zu wenig



## Praxis-Tipp

Nach Peikert and Diemerling (2025):

1. Was standardisiert ist, muss nicht dokumentiert werden.
2. Was einfach ist, braucht nur wenig Dokumentation.
3. Was konsistent ist, muss nur einmal dokumentiert werden.

# Dateipfade



## Absoluter Pfad (schlecht):

```
1 daten <- read.csv("C:/Users/Maria/Desktop/Masterarbeit/daten.csv")
```

Problem: Funktioniert nur auf Ihrem Computer!



## Relativer Pfad (gut):

```
1 daten <- read.csv("data/raw/fragebogen-daten.csv")
```

Noch besser: Verwendung des `here` packages (funktioniert mit verschiedenen Systemen):

```
1 library(here)
2 daten <- read.csv(here("data", "raw", "fragebogen-daten.csv"))
```



# Aufteilung in verschiedene Skripte

## Ein großes Skript:

```
1 # complete_analysis.R (500+ lines)
2 # Data import, cleaning, analysis, plots, export...
```

## Modulare Aufteilung:

- `01_data_import.R` - Rohdaten laden
- `02_data_cleaning.R` - Präprozessierung
- `03_descriptive_analysis.R` - Deskriptive Statistiken
- `04_inferential_statistics.R` - Inferenz
- `05_visualization.R` - Abbildungen
- `06_export.R` - Export

**Vorteile:** Übersichtlicher, leichter zu debuggen, bessere Zusammenarbeit

# Funktionsbasiertes Programmieren

## ✗ Code-Wiederholung:

```
1 # Standardization for variable 1
2 data$var1_std <- (data$var1 - mean(data$var1, na.rm = TRUE)) /
3                 sd(data$var1, na.rm = TRUE)
4
5 # Standardization for variable 2
6 data$var2_std <- (data$var2 - mean(data$var2, na.rm = TRUE)) /
7                 sd(data$var2, na.rm = TRUE)
8
9 # Standardization for variable 3...
```



## Funktion erstellen:

```
1 standardize_variable <- function(x) {
2   # Z-standardization of a numeric variable
3   (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
4 }
5
6 # Application
7 data$var1_std <- standardize_variable(data$var1)
8 data$var2_std <- standardize_variable(data$var2)
9 data$var3_std <- standardize_variable(data$var3)
10
11 # Or even better
12 data <- data |>
13   mutate(across(starts_with("var"), standardize_variable))
```

# Vermeiden von “magic numbers”

## ✗ Unklare Zahlen im Code:

```
1 # What do these numbers mean?  
2 clean_data <- data[data$age >= 18 & data$age <= 65, ]  
3 final_data <- clean_data[clean_data$score > 2.5, ]  
4 significant <- results[results$p_value < 0.05, ]
```



## Benannte Konstanten:

```
1 # Clear definitions at the beginning of the script  
2 MIN_AGE <- 18  
3 MAX_AGE <- 65  
4 MIN_SCORE <- 2.5  
5 ALPHA_LEVEL <- 0.05  
6  
7 # Usage in code  
8 clean_data <- data[data$age >= MIN_AGE &  
9                 data$age <= MAX_AGE, ]  
10 final_data <- clean_data[clean_data$score > MIN_SCORE, ]  
11 significant <- results[results$p_value < ALPHA_LEVEL, ]
```

**Vorteil:** Code wird selbsterklärend und leichter anpassbar

# Session Info

## Warum wichtig?

- **Reproduzierbarkeit:** Welche R-Version und Pakete wurden verwendet?
- **Fehlerdiagnose:** Unterschiedliche Paketversionen können verschiedene Ergebnisse liefern
- **Dokumentation:** Vollständige Nachvollziehbarkeit der Analyse



### Am Ende jedes Skripts:

```
1 # Session information for reproducibility  
2 sessionInfo()
```



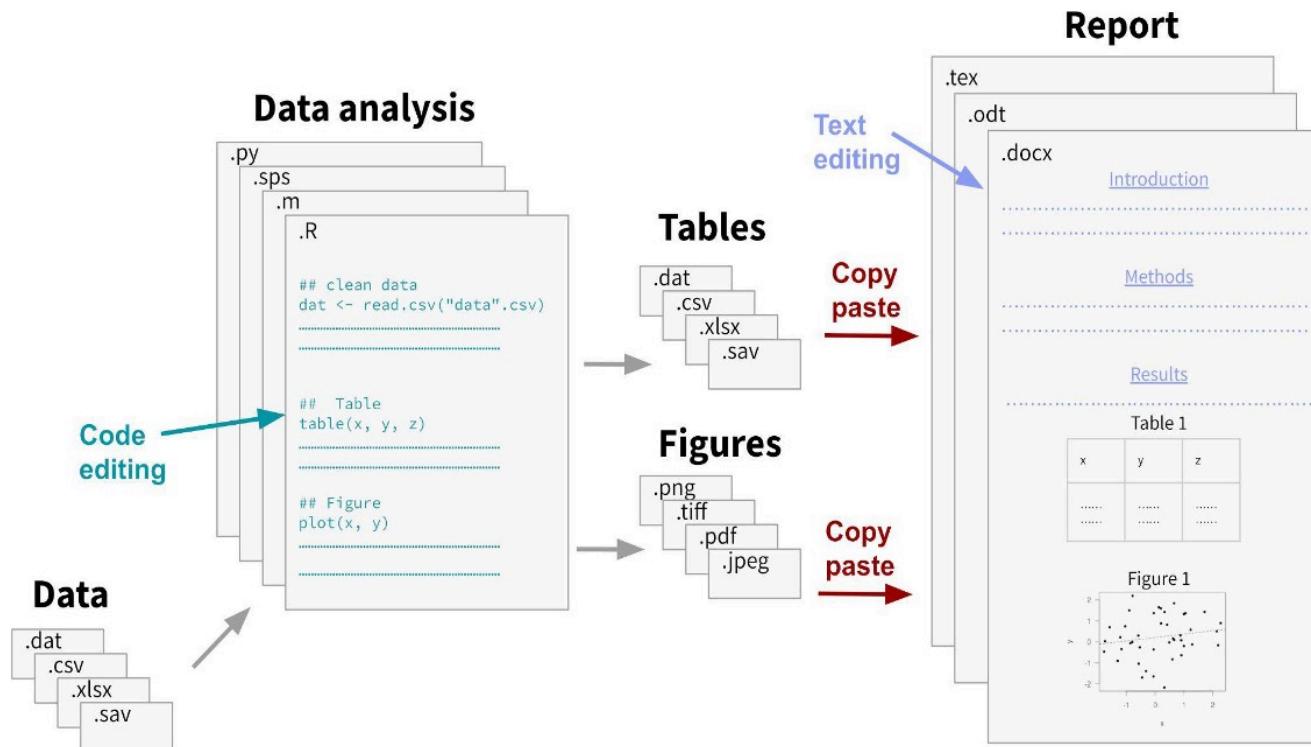
#### Praxis-Tipp

Entweder in Quarto direkt rendern oder in separatem File speichern:

```
1 writeLines(capture.output(sessionInfo()), "session_info.txt")
```

# Dynamische Reports

# Typischer, nicht-dynamischer Workflow



Molo, Fraga González, and Pawel (2025)

# Typische Probleme



cleandata.R

- unklarer Ausführungsreihenfolge
- Zeitaufwendiges Copy-Pasten
- Fehleranfälliges Copy-Pasten



data.csv



plots.R



stats.R



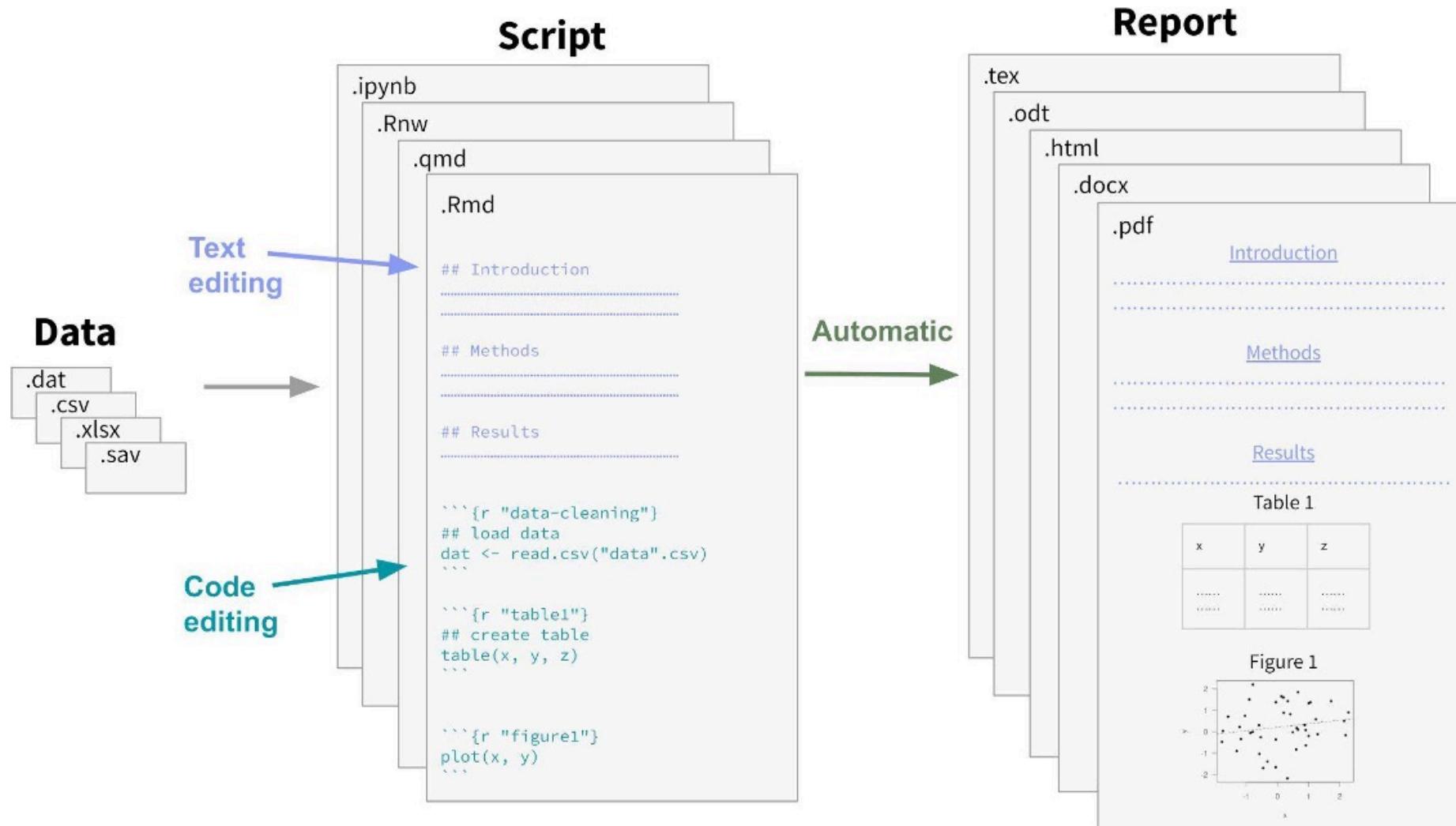
tables.R

Molo, Fraga González, and Pawel (2025)

# Dynamische Reports: Das Ziel

Daten → dynamischer Report → Manuskript

# Ein Workflow für dynamische Reports



# Komponenten eines dynamischen Reports

YAML Header

M↓ Text

</> Code chunk

→ Code output

a</> Inline code

```
---  
title: "An example"  
format: html  
---  
<# Section 1  
The code below reads a table and  
plots the data  
```{r}  
df <- read_xlsx('table.csv')  
plot(df)  
```
```



This table has `r nrow(df)` rows

# Text: Typische Markup-Sprachen

## HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Document</title>
</head>
<body>
  <h1>Heading Level 1</h1>
  <h2>Heading Level 2</h2>
  <p><strong>Bold text</strong>
and <em>italic text</em> in a
paragraph.</p>

<ol> <li>First item in an ordered
list</li> <li>Second item in an ordered
list</li> </ol>

<ul> <li>Unordered list item</li>
<li>Another unordered list item</li>
</ul> <p><code>Inline code</code>
and</p>

<pre>Block code</pre>

</body>
</html>
```

## LaTeX

```
\documentclass{article}
\begin{document}
\section{Heading Level 1}
\subsection{Heading Level 2}

\textbf{Bold text} and \textit{italic
text} in a paragraph.

\begin{enumerate}
\item First item in an ordered list
\item Second item in an ordered list
\end{enumerate}

\begin{itemize}
\item Unordered list item
\item Another unordered list item
\end{itemize}

\texttt{Inline code} and
\begin{verbatim} Block code
```\end{verbatim}
```

## Markdown

```
# Heading Level 1
## Heading Level 2

**Bold text** and *italic text*
in a paragraph.

1. First item in an ordered list
2. Second item in an ordered list

- Unordered list item
- Another unordered list item

`Inline code` and

``````

Block code
``````
```

# Manche Implementierungen

|                  | Datei  | Code             | Markup   | Output Formate                  |
|------------------|--------|------------------|----------|---------------------------------|
| Sweave           | .Rnw   | R                | LaTeX    | pdf, tex                        |
| R Markdown       | .Rmd   | R                | Markdown | pdf, html, tex, docx, pptx      |
| Jupyter Notebook | .ipynb | Julia, Python, R | Markdown | pdf, html, tex, py, ...         |
| Quarto           | .qmd   | Julia, Python, R | Markdown | pdf, html, tex, docx, pptx, ... |

# Warum Quarto?

- **open source**
- **modern**, wird kontinuierlich weiterentwickelt
- **einfache Handhabung**
- **Mehrere Programmiersprachen** (z.B. R, Python, Julia)
- **vielfältige Editoren/IDEs** (z. B. RStudio, VS Code)
- **allgemeines wissenschaftliches Publikationssystem** (z. B. Artikel, Folien, Websites)



<https://quarto.org/quarto.png>

# Quarto in RStudio

The screenshot shows the RStudio interface with a Quarto file open in the Source editor and its rendered output in the Preview tab.

**Source Editor (example-quarto.qmd):**

```
1 ---  
2 title: Example Quarto file  
3 subtitle: Reproducible and Dynamic Reporting with R and Quarto  
4 author:  
5   - name: 'Fabio Molo'  
6     orcid: 0000-0001-9320-5854  
7 format:  
8   html:  
9     toc-depth: 4  
10  pdf: default  
11 execute:  
12   eval: true  
13 bibliography: references.bib  
14 ---  
15 ## Load libraries  
16  
17 Let us first load all necessary libraries.  
18  
19 ```{r}  
20 #| warning: false  
21 library("dplyr") # data wrangling  
22 library("ggplot2") # plotting  
23 library("kableExtra") # tables  
24 ````  
25  
26 ## Read the data  
27  
28 For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" (https://data.stadt-zuerich.ch/dataset/bev\_vornamen\_baby\_od3700). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.  
29  
30 ```{r}  
31 url <- "https://data.stadt-zuerich.ch/dataset/bev_vornamen_baby_od3700/download"  
10:13 Example Quarto file
```

**Handwritten annotations:**

- A yellow curly brace groups lines 1-14 under the heading "instructions: YAML".
- A purple curly brace groups lines 15-24 under the heading "text: markdown".
- A blue curly brace groups lines 25-31 under the heading "code: R".

**Preview Tab (workshop-quarto-2025-uzh):**

**Example Quarto file**  
Reproducible and Dynamic Reporting with R and Quarto

AUTHOR  
Fabio Molo

**Load libraries**

Let us first load all necessary libraries.

```
library("dplyr") # data wrangling  
library("ggplot2") # plotting  
library("kableExtra") # tables
```

**Read the data**

For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" ([https://data.stadt-zuerich.ch/dataset/bev\\_vornamen\\_baby\\_od3700](https://data.stadt-zuerich.ch/dataset/bev_vornamen_baby_od3700)). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.

```
url <- "https://data.stadt-zuerich.ch/dataset/bev_vornamen_baby_od3700"  
dat <- read.csv(url)  
datclean <- dat |>  
  rename(year = StichtagDatJahr,  
         name = Vorname,
```

# Quarto in RStudio: PDF Output

The screenshot shows the RStudio interface with a Quarto file open in the Source tab and its rendered output in the Preview tab.

**Source Tab:**

```
1 ---  
2 title: Example Quarto file  
3 subtitle: Reproducible and Dynamic Reporting with R and Quarto  
4 author:  
5   - name: 'Fabio Molo'  
6   orcid: 0000-0001-9320-5854  
7 format:  
8   html:  
9     toc-depth: 4  
10  pdf: default  
11 execute:  
12   eval: true  
13 bibliography: references.bib  
14 ---  
15 ## Load libraries  
16  
17 Let us first load all necessary libraries.  
18  
19 ````{r}  
20 #| warning: false  
21 library("dplyr") # data wrangling  
22 library("ggplot2") # plotting  
23 library("kableExtra") # tables  
24 ````  
25  
26 ## Read the data  
27  
28 For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" (https://data.stadt-zuerich.ch/dataset/bev\_voramen\_baby\_od3700). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.  
29  
30 ````{r}  
31 url <- "https://data.stadt-zuerich.ch/dataset/bev_voramen_baby_od3700/download/BEV3700D3700.csv"  
10:15 Example Quarto file
```

**Preview Tab:**

**Example Quarto file**  
Reproducible and Dynamic Reporting with R and Quarto  
Fabio Molo

**Load libraries**

Let us first load all necessary libraries.

```
library("dplyr") # data wrangling  
library("ggplot2") # plotting  
library("kableExtra") # tables
```

**Read the data**

For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" ([https://data.stadt-zuerich.ch/dataset/bev\\_voramen\\_baby\\_od3700](https://data.stadt-zuerich.ch/dataset/bev_voramen_baby_od3700)). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.

```
url <- "https://data.stadt-zuerich.ch/dataset/bev_voramen_baby_od3700/download/BEV3700D3700.csv"  
dat <- read.csv(url)  
datclean <- dat %>  
  rename(year = StichtagDatJahr,  
         name = Vorname,  
         sex = SexLang,  
         births = AnzGebuWir) %>  
  mutate(sex = ifelse(sex == "weiblich", "female", "male"))
```

**Summaries the data**

We will now compute the most common name by year and sex.

# Quarto in RStudio: MS Word output

The screenshot illustrates the process of generating Microsoft Word output from Quarto source code within the RStudio environment.

**Left Panel (Quarto Source Code):**

```
workshop-quarto-2025-untitled-1.Rmd
example-quarto.qmd x
Source Visual
example Next Prev All Replace
In selection Match case Whole word Regex Wrap
1 ---  
2 title: Example Quarto file  
3 subtitle: Reproducible and Dynamic Reporting with R and Quarto  
4 author:  
5 - name: 'Fabio Molo'  
6 orcid: 0000-0001-9320-5854  
7 format:  
8   html:  
9     toc-depth: 4  
10    pdf: default  
11    docx: default  
12 execute:  
13   eval: true  
14 bibliography: references.bib  
15 ---  
16 ## Load libraries  
17  
18 Let us first load all necessary libraries.  
19  
20 ``{r}  
21 #| warning: false  
22 library("dplyr") # data wrangling  
23 library("ggplot2") # plotting  
24 library("kableExtra") # tables  
25 ``  
26  
27 ## Read the data  
28  
29 For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" (https://data.stadt-zuerich.ch/dataset/bev\_vornamen\_baby\_od3700). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.  
7:8 Example Quarto file
```

**Right Panel (MS Word Output Preview):**

The preview shows the generated Microsoft Word document with the following structure:

- Title:** Example Quarto file
- Section:** Reproducible and Dynamic Reporting with R and Quarto
- Author:** Fabio Molo
- Text:** Load libraries
- Text:** Let us first load all necessary libraries.
- Text:** Read the data
- Text:** For this example we will use the data set "Vornamen neugeborener Mädchen und Knaben mit Wohnsitz in der Stadt Zürich" ([https://data.stadt-zuerich.ch/dataset/bev\\_vornamen\\_baby\\_od3700](https://data.stadt-zuerich.ch/dataset/bev_vornamen_baby_od3700)). It contains the names of newborn children in the city of Zurich since the year 1993. We will now import the data and rename the variables into English.
- Text:** summaries
- Text:** We will now compute the most common name by year and sex.
- Text:** summaries > kable(caption = "Most common name of newborns in the city of Zurich by year and sex")

# Weitere Literatur

- CRS Primer zu dynamischen Reports: <https://doi.org/10.5281/zenodo.7565735>
- Quarto Intro Tutorial: <https://quarto.org/docs/get-started/hello/rstudio.html>
- Quarto Authoring Tutorial: <https://quarto.org/docs/get-started/authoring/>
- Quarto Layout: <https://quarto.org/docs/authoring/article-layout.html>
- R4DS Kapitel zu Quarto: <https://r4ds.hadley.nz/quarto>
- Yihui Xie's Blog: [With Quarto coming, is R Markdown going away? No.](#)
- Reproducible manuscripts with Quarto: [Folien von Mine Cetinkaya-Rundel](#)
- Quarto/RMarkdown – What's different?: [Folien von Ted Laderas](#)

# Literatur

- Gould, Elliot, Hannah S. Fraser, Timothy H. Parker, Shinichi Nakagawa, Simon C. Griffith, Peter A. Vesk, Fiona Fidler, et al. 2023. “Same Data, Different Analysts: Variation in Effect Sizes Due to Analytical Decisions in Ecology and Evolutionary Biology,” October. <https://ecoevorxiv.org/repository/view/6000/>.
- Molo, Fabio, Gorka Fraga González, and Samuel Pawel. 2025. “Workshop ‘Reproducible and Dynamic Reporting with r and Quarto – Getting Started’ at UZH.” <https://doi.org/doi.org/10.5281/zenodo.14169002>.
- Open Science Collaboration. 2015. “Estimating the Reproducibility of Psychological Science.” *Science* 349 (6251): aac4716. <https://doi.org/10.1126/science.aac4716>.
- Peikert, Aaron, and Andreas M. Brandmaier. 2021. “A Reproducible Data Analysis Workflow with r Markdown, Git, Make, and Docker.” *Quantitative and Computational Methods in Behavioral Sciences* 1. <https://doi.org/10.5964/qcmb.3763>.
- Peikert, Aaron, and Hannes Diemerling. 2025. “Reproducible Research in r: A Workshop on How to Do the Same Thing More Than Once.” Zenodo. <https://doi.org/10.5281/zenodo.15038481>.
- Peng, Roger D. 2011. “Reproducible Research in Computational Science.” *Science* 334