

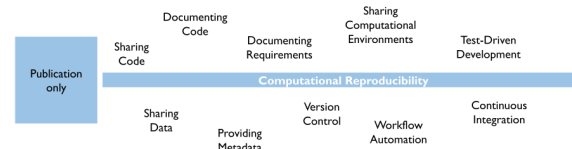
Reproduzierbare Forschung mit R und Quarto – Fortgeschrittenes

Workshop für die GSE Wuppertal

Björn S. Siepe

June 27, 2025

Reproducibility ist ein Spektrum



Molo, Fraga González, and Pawel (2025)

- Ziel heute: Verstehen, **was möglich ist** auf der rechten Seite
- ~~Ziel heute: Tiefes technisches Verständnis für fortgeschrittene Arbeitsabläufe ~~

Dynamische Reports lösen copy-paste Probleme, aber...

- Wie behält man den Überblick über verschiedene **Dateiversionen**?
 - `Paper-final.qmd` → `Paper-final2.qmd` → `Paper-final2b.qmd` → 🤖
 - Lösung: **Git**
- Woher weiß man, welche Dateien in welcher **Reihenfolge** ausgeführt werden sollen?
 - `functions.qmd`, `analysis.qmd`, `plots.qmd`, `paper.qmd`, 🤖
 - Lösung: **Make**
- Wie stellt man sicher, dass der Code noch läuft, wenn sich **Softwareversionen** ändern?
 - Heute: R Version 4.5, ggplot2 3.5.2 → **Nächstes Jahr**: R Version **?**, ggplot2 **?**
 - Lösung: **Docker**
- Sehr schönes Paper über einen solchen Workflow: Peikert and Brandmaier (2021)

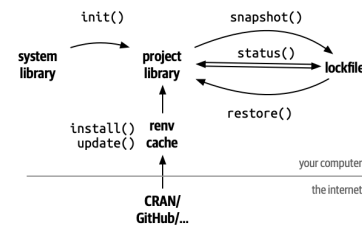
Molo, Fraga González, and Pawel (2025)

Nützliche Tools 1: **renv**

- R package für Reproduzierbarkeit von R-Projekten

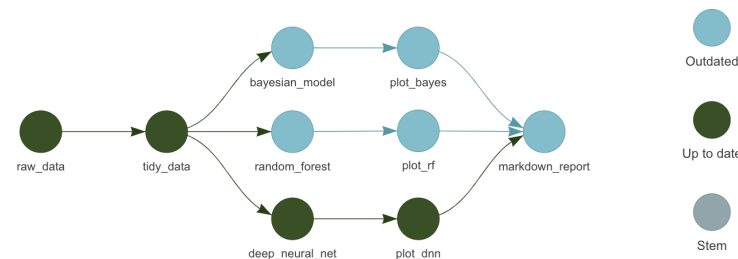
::: .column width="50%"

```
1 ```{r}
2 renv::init() # renv initialisieren
3 renv::snapshot() # Abhängigkeiten erfassen
4 renv::restore() # Abhängigkeiten wiederherstellen
5 renv::dependencies() # Abhängigkeiten auflisten
6 ```
```



::: .column width="50%"

Nützliche Tools 2: **targets** Pipeline für Reproduzierbarkeit von R-Projekten

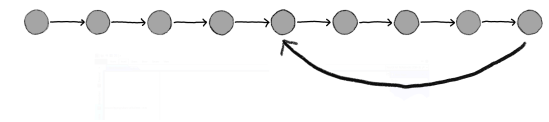


Versionskontrolle mit Git

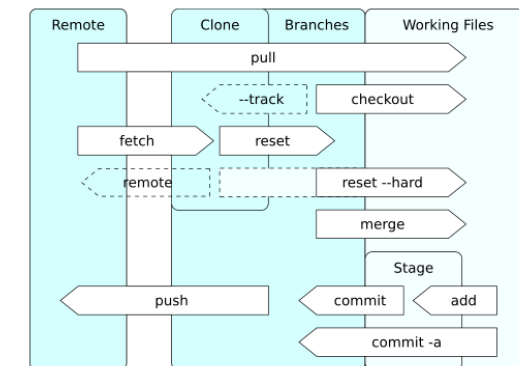
- **De-facto-Standard** Versionskontrollsystem
- Kann **in der Zeit zurückgehen** zu früheren Versionen
- Kann **Änderungen** zwischen Versionen verfolgen
- Kann **verzweigen und zusammenführen**
- Nützlich für **Kollaboration** und **Dateitransfer**
- **Command-Line-Schnittstelle** (Lernkurve)
- **Grafische Schnittstelle** in vielen Editoren/IDEs (z. B. RStudio, VS Code)
- **Git (Software) ≠ GitHub/GitLab** (Git-Repository-Hosting-Dienste)
- Snapshots ermöglichen die **dauerhafte Archivierung** (mit DOI) auf Daten-Repositories (z. B. Zenodo)



<https://git-scm.com/downloads/logos>



The Turing Way Community and Scriberia (2024)



https://commons.wikimedia.org/wiki/File:Git_operations.svg

Versionskontrolle mit Git - Wo starten?

- Git installieren
- GitHub Account erstellen
- Git in RStudio aktivieren
- RStudio-Projekt erstellen
- RStudio-Projekt mit Git initialisieren
- RStudio-Projekt mit GitHub verbinden

Gute Ressource: - Happy Git with R: <https://happygitwithr.com/>

Automatisierung mit Make

- Tool für **komputationale Rezepte**
- Automatisierung von mehreren Schritten in einem

Arrabbiata sauce, or sugo all'arrabbiata in Italian, is a spicy sauce for pasta made from garlic, tomatoes, and dried red chili peppers cooked in olive oil.

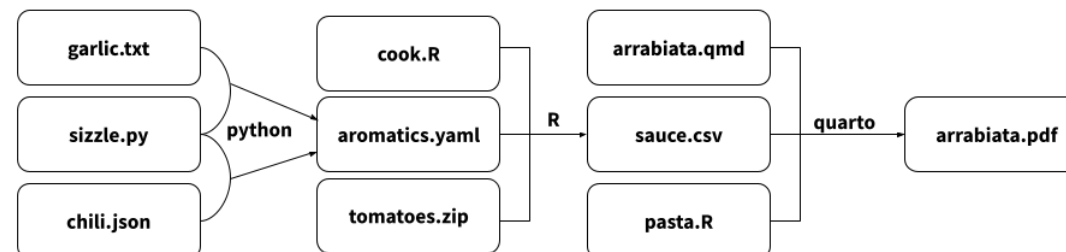
https://en.wikipedia.org/wiki/Arrabbiata_sauce

Makefile

```
1 arrabiata.pdf: arrabiata.qmd sauce.csv pasta.R
2   quarto render arrabiata.qmd
3
4 sauce.csv: cook.R tomatoes.zip aromatics.yaml
5   R -e "source(cook.R)"
6
7 aromatics.yaml: sizzle.py garlic.txt chili.json
8   python sizzle.py garlic.txt
9   python sizzle.py chili.json
```

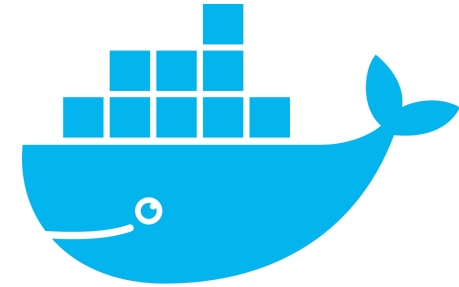
Terminal

```
1 make arrabiata.pdf # execute recipe
```



Containerisierung mit Docker

- **Software Container** beinhalten das komputationale Umfeld für die Ausführung von Code
- Komponenten eines Containers:
 - Operating System (OS)
 - R und R-Pakete (oder andere Programmiersprachen)
 - Numerische Bibliotheken (z.B., LAPACK, BLAS)
 - Andere Software (z.B., LaTeX, Java)
- **Docker** ist ein beliebtes Containerisierungstool
- **Rocker** ist ein Docker Image für R (<https://rocker-project.org/>)

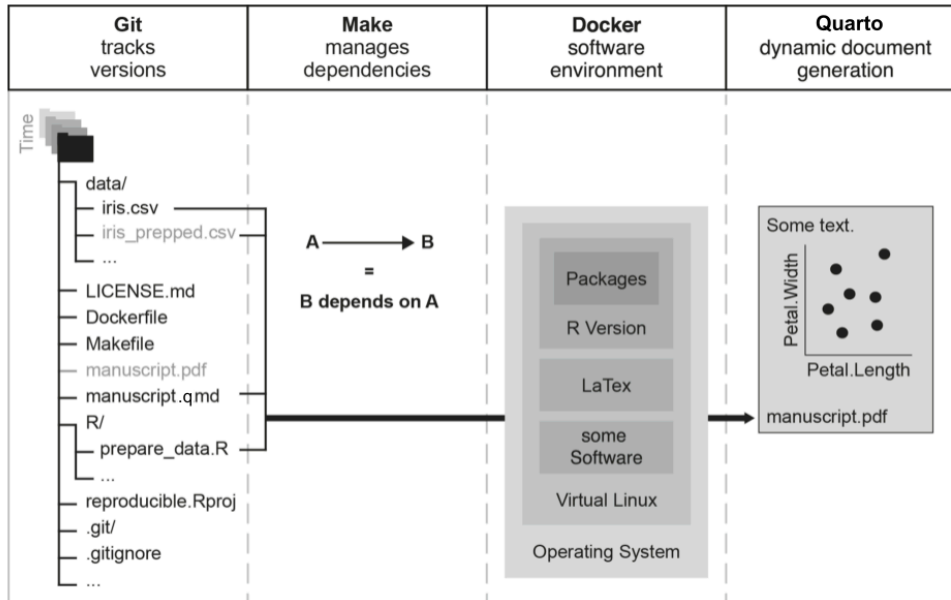


<https://commons.wikimedia.org/wiki/File:Docker-svgrepo-com.svg>

Dockerfile

```
1 # Rocker container image mit R 4.4
2 FROM rocker/verse:4.4
3
4 # R Packages zum damaligen Zeitpunkt installieren
5 RUN install2.r ggplot2 dplyr kableExtra
```


Alles zusammen



Terminal

```
1 # alles in einem Schritt ausführen
2 make docker-report
```

→ [report.html](#)

Nach Peikert and Brandmaier (2021) und Molo, Fraga González, and Pawel (2025)

Zusätzliches Thema: Code Reviews

- **Code Reviews** sind eine Methode, um die Qualität von Code zu verbessern
- TODO

Abschlussdiskussion

- Welche Tools könnten am nützlichsten sein?
- Welche Tools sind am schwierigsten zu lernen?
- Wo sind die größten verbleibenden Hürden?

Zusammenfassung

- Dynamische Reports + Versionskontrolle + Automatisierung + Containerisierung
 - verbessert **Reproduzierbarkeit, Wiederverwendbarkeit, Zusammenarbeit**
 - verleiht der Forschung mehr **Transparenz, Glaubwürdigkeit**
- Es gibt eine **Lernkurve** , aber ...
 - beginnt einfach und lernen Sie **Schritt für Schritt** 💡 .
 - die Tools werden **stetig weiterentwickelt** 🚀 .
 - investierte Lernzeit kann Ihnen in Zukunft **Zeit sparen** ⌚
 - Fähigkeiten sind **innerhalb/außerhalb von Academia** nützlich 💰 👨🔬

Quarto für andere Dinge

Kontakt

Kontakt

Vielen Dank für die Aufmerksamkeit! 🎉

Literatur

- Landau, William Michael. 2021. “The Targets r Package: A Dynamic Make-Like Function-Oriented Pipeline Toolkit for Reproducibility and High-Performance Computing.” *Journal of Open Source Software* 6 (57): 2959. <https://doi.org/10.21105/joss.02959>.
- 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>.
- 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>.
- The Turing Way Community, and Scriberia. 2024. “Illustrations from the Turing Way: Shared Under CC-BY 4.0 for Reuse.” Zenodo. <https://doi.org/10.5281/ZENODO.3332807>.
- Ushey, Kevin, and Hadley Wickham. 2025. *Renv: Project Environments*. <https://doi.org/10.32614/CRAN.package.renv>.