



# Reproducible Reporting







in der Bildungsforschung

GEBF Open Science Summer 2021 - Samuel Merk, Jürgen Schneider



# Inhaltlicher Überblick

1. Reproduzierbarkeit von Datenanalysen/Datenmanagement

2. R-spezifische Werkzeuge

- Code 
- Code  + Daten 
- Ausführbarer Code  + verlinkte Daten 
- Wissenschaftskommunikation 

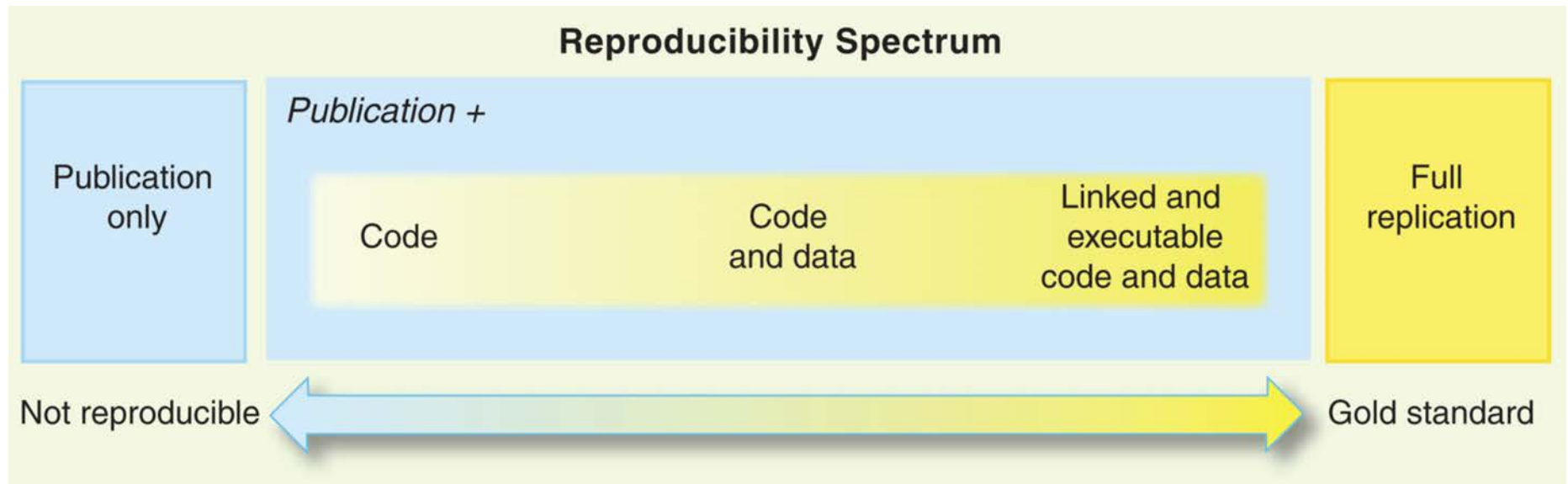
3. Gut in R integrierbare Werkzeuge

- Versionierung mit Git 
- Make-like files 

4. Wie einsteigen?

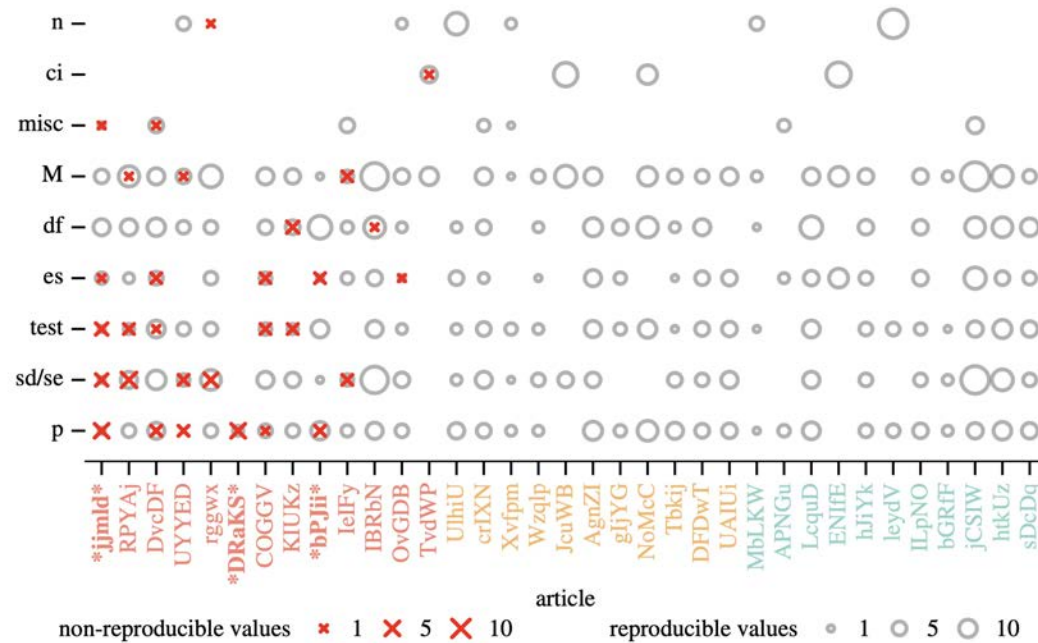
An article [...] is not the scholarship itself, it is merely advertising of the scholarship - David Donoho

# Das (theoretische) Reproduzierbarkeitsspektrum



*Das Reproduzierbarkeitsspektrum. Abbildung aus Peng (2011)*

# Die (reale) Verteilung auf diesem Spektrum I



**Figure 3.** All 1324 values were checked for reproducibility as a function of article and value type (n = count/proportion; ci = confidence interval; misc = miscellaneous; M = mean/median; df = degrees of freedom; es = effect size; test = test statistic; p = *p*-value; sd/se = standard deviation/standard error). Bold red X marks indicate non-reproducible values (major errors) and grey circles indicate reproducible values. Symbol size represents the number of values. Both axes are ordered by an increasing number of errors towards the graph origin. The article colours represent the overall outcome: not fully reproducible despite author assistance (red), reproducible with author assistance (orange) and reproducible without author assistance (green). For articles marked within asterisks (\*), the analysis could not be completed and there was insufficient information to determine whether original conclusions were affected. In all other cases, it is unlikely that original conclusions were affected.

*Aus Hardwicke et al. (2018)*

# Die (reale) Verteilung auf diesem Spektrum II

- Ergebnisse aus *Stockemer, Koehler & Lentz (2018)*:
  - Drei Political-Science Journale (71 Artikel)
    - 32 exakt replizierbare Analysen
    - 19 »minor errors«
    - 3 »significantly different«
    - 16 »replication was not possible«
- Ergebnisse aus *Stodden, Seiler & Ma (2018)*:
  - Zwei Jahrgänge Science (204 Artikel)
    - In 13% Informationen zum Bezug von Daten oder Code
    - In 44% Artefakte beziehbar
    - In 26% Reproduzierbarkeit

R-spezifische Werkzeuge

# “Bewildering Technology Soup” *(Lapp, 2015)*



*Bewildering Technology Soup (Lapp 2015). Eigene Darstellung.*



The plot shows the following tools categorized by their entry barrier and associated features:

- Low Entry Barrier (Bottom Left):** R, R Studio (code editor icon).
- Low to Medium Entry Barrier (Bottom Middle):** haven (database icon).
- Medium Entry Barrier (Middle):** R, knitr, markdown (linking icon).
- Medium to High Entry Barrier (Top Middle):** R, knitr, markdown, packrat (settings icon).
- High Entry Barrier (Top Right):** R, knitr, markdown, Shiny, jupyter, LaTeX, GitHub (communication icon).



# R und RStudio =

```
# Dokumentation der Analysen zum Manuskript "XYZ" #####  
  
## Import der nicht verfügbaren Daten ##  
library(haven)  
meine_daten <- read_spss("MEINE_und_nur_MEINE_vom_Hiwi_gecleanten_Daten.sav")  
  
## Stichprobenbeschreibung  
nrow(meine_daten)  
table(meine_daten$sex)  
  
## Instrumente  
library(lavaan)  
...
```

# Daten einlesen in R = +

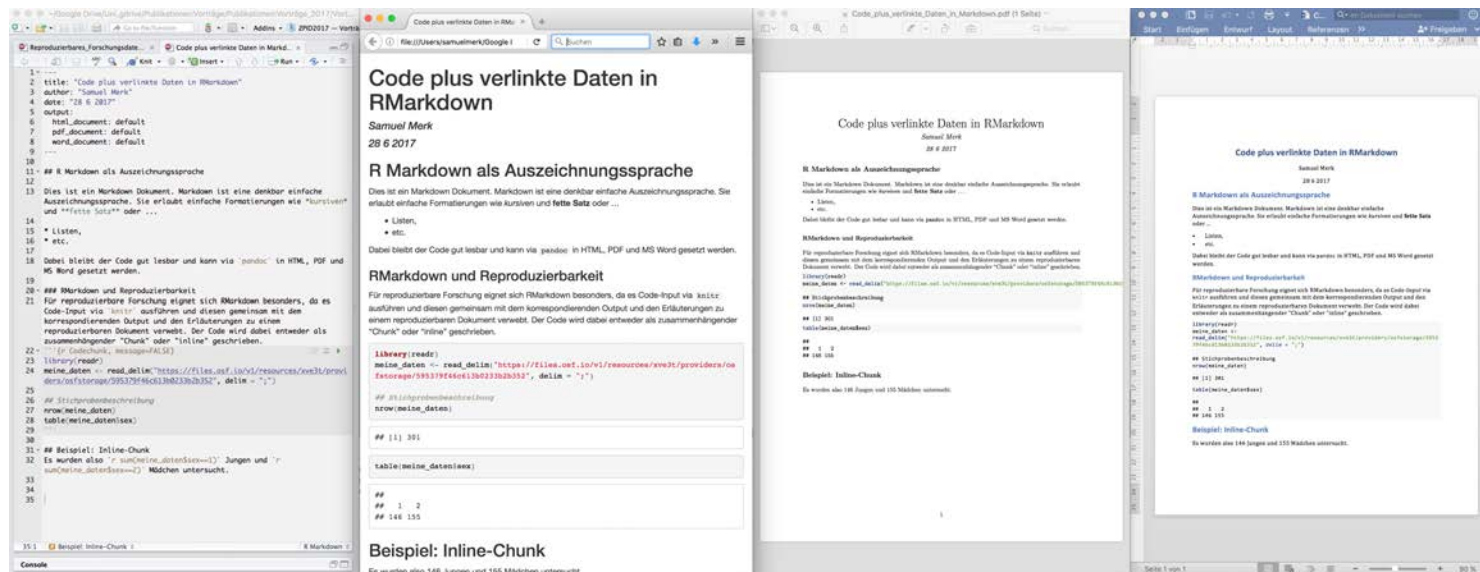
- Mittlerweile liegen hervorragende Pakete für das Einlesen fremder/proprietärer Datenformate vor:
  - SAS, SPSS, Stata `{haven}` (Wickham & Miller, 2018)
  - Excel `{readxl}` (Wickham & Bryan, 2017)
  - Text `{readr}` (Wickham, Hester & Francois, 2017)
- Das Lesen sehr großer Datensätze zeigt gute Performance mit den entsprechenden Paketen (z.B. `{data.table}` (Dowle & Srinivasan, 2017), `{feather}` (Wickham, 2016)).
- Arbeit mit Remote-Datenbanken komfortabel möglich `{dbplyr}` (Wickham, 2017).

# Beispiel für +

```
# Dokumentation der Analysen zum Manuskript "XYZ" #####  
  
## Import der bei PsychData hinterlegten Daten ##  
library(haven)  
meine_daten <- read_spss("PsychDatafile.sav")  
  
## Stichprobenbeschreibung  
nrow(meine_daten)  
table(meine_daten$sex)  
  
## Instrumente  
library(lavaan)  
...
```

# RMarkdown + knitr = + +

- Grundidee: `knitr` (Xie, 2015) verwebt
  - Text (formatiert durch die maximal einfache Auszeichnungssprache (markup language) RMarkdown (Allaire et al., 2017)),
  - ausführbaren Code und dessen Output
  - via `pandoc` (siehe Gandrud, 2014) zu .pdf .html .docx etc.



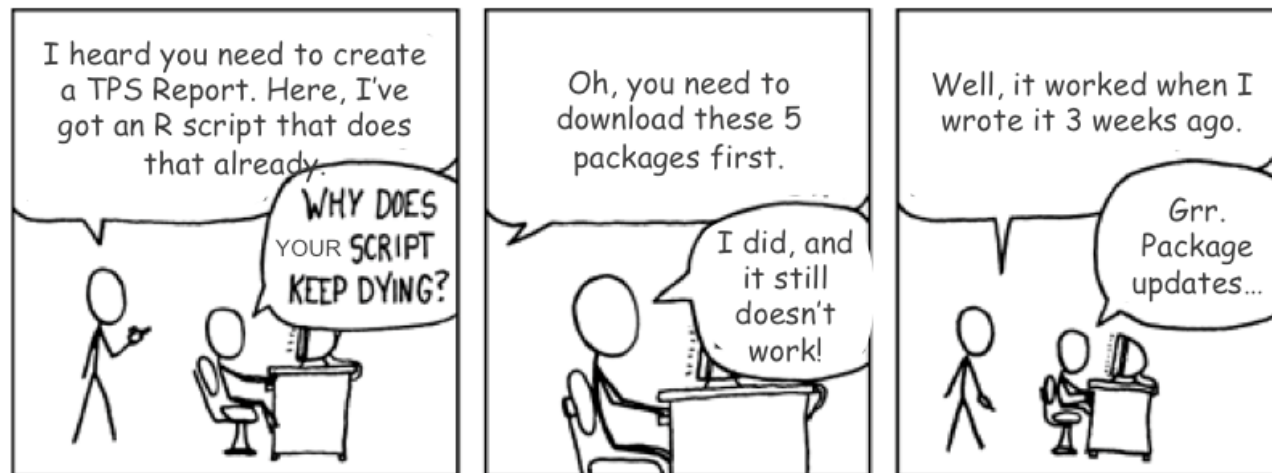
Screenshot: RMarkdown und dessen Outputformate

Beispiel für  +  + 

Code-Beispiel

# RMarkdown + Packrat/renv = + + +

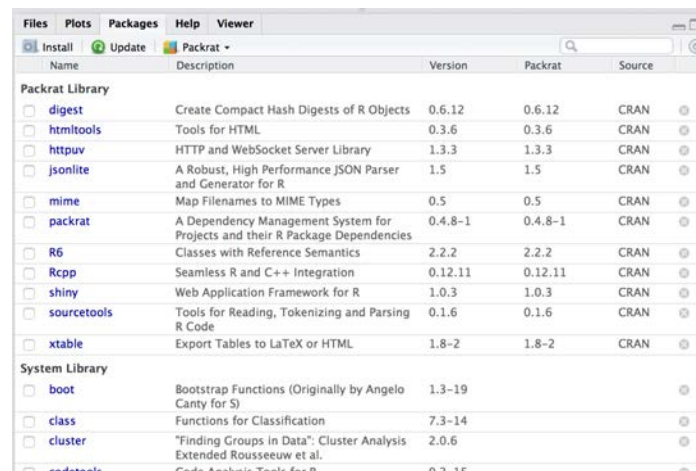
- R-Skripte erzielen bereits ausführbaren Code mit verlinkten Daten.
- RMarkdown erlaubt zusätzlich sog. “Literate Programming” mit einfachsten Mitteln.
- Dennoch bleiben Probleme der Nicht-Reproduzierbarkeit:



Package-Updates. Abbildung: <https://xkcd.com/234/> CC BY-NC 2.5

# Packrat

- Packrat stellt ein Dependency-Management-Werkzeug zur Verfügung, welches eine eigenständige Bibliothek von R-Paketen und deren Versionen erstellt.
- ⇒ Konservierung weiterer Teile des “Computational Environment”.
- Packrat ist unabhängig von RStudio konzipiert - mit RStudio aber deutlich “komfortabler” bedienbar.



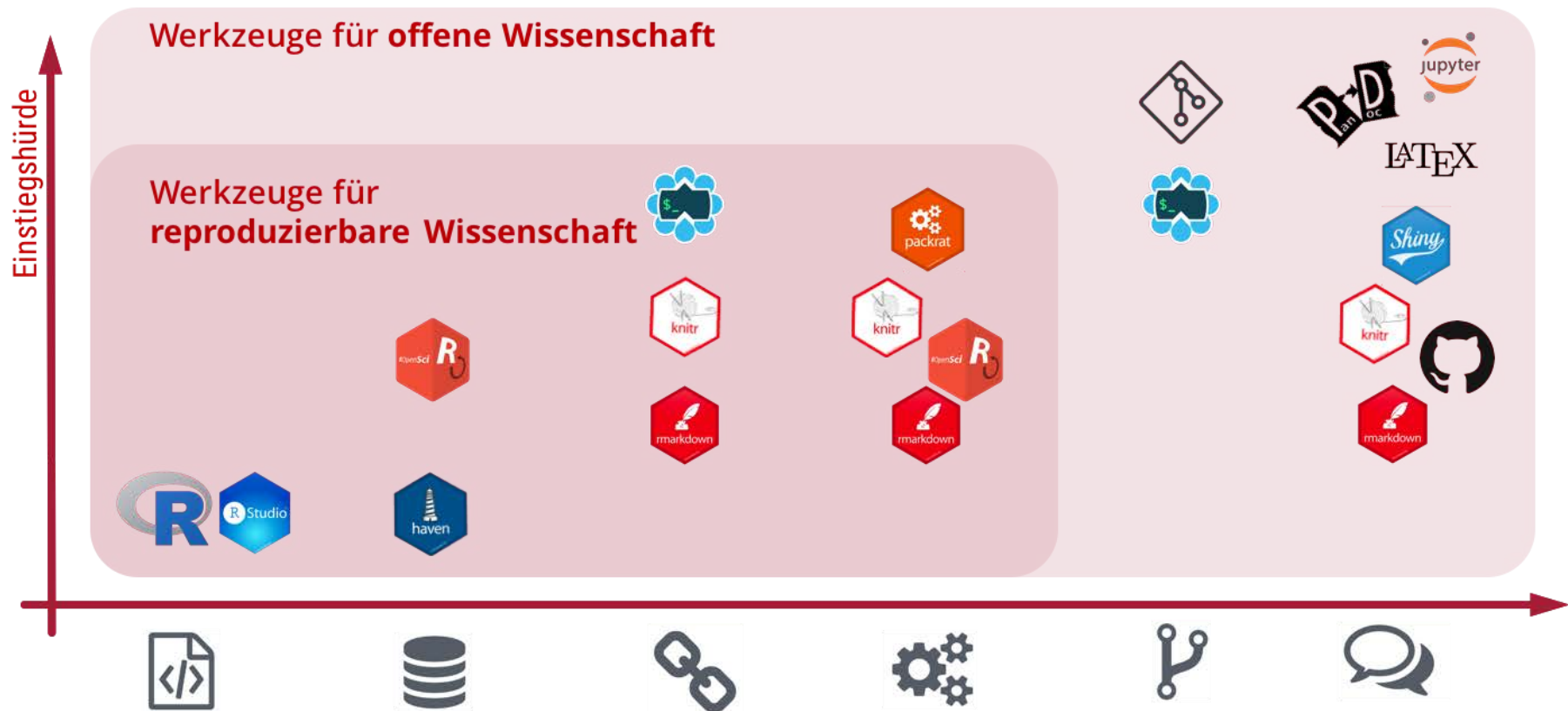
The screenshot shows the Packrat pane in RStudio. The pane has a menu bar with 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'. Below the menu bar are buttons for 'Install', 'Update', and 'Packrat'. The main area displays a table of packages, categorized into 'Packrat Library' and 'System Library'. Each package entry includes a checkbox, the package name, a description, the current version, the Packrat version, and the source (CRAN). The 'packrat' package is highlighted in blue.

	Name	Description	Version	Packrat	Source
<b>Packrat Library</b>					
<input type="checkbox"/>	digest	Create Compact Hash Digests of R Objects	0.6.12	0.6.12	CRAN
<input type="checkbox"/>	htmltools	Tools for HTML	0.3.6	0.3.6	CRAN
<input type="checkbox"/>	httpuv	HTTP and WebSocket Server Library	1.3.3	1.3.3	CRAN
<input type="checkbox"/>	jsonlite	A Robust, High Performance JSON Parser and Generator for R	1.5	1.5	CRAN
<input type="checkbox"/>	mime	Map Filenames to MIME Types	0.5	0.5	CRAN
<input type="checkbox"/>	packrat	A Dependency Management System for Projects and their R Package Dependencies	0.4.8-1	0.4.8-1	CRAN
<input type="checkbox"/>	R6	Classes with Reference Semantics	2.2.2	2.2.2	CRAN
<input type="checkbox"/>	Rcpp	Seamless R and C++ Integration	0.12.11	0.12.11	CRAN
<input type="checkbox"/>	shiny	Web Application Framework for R	1.0.3	1.0.3	CRAN
<input type="checkbox"/>	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6	0.1.6	CRAN
<input type="checkbox"/>	xtable	Export Tables to LaTeX or HTML	1.8-2	1.8-2	CRAN
<b>System Library</b>					
<input type="checkbox"/>	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-19		
<input type="checkbox"/>	class	Functions for Classification	7.3-14		
<input type="checkbox"/>	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.6		
<input type="checkbox"/>	crude	Crude Analytic Tools for R	0.2-15		

*Packrat-Pane in RStudio. Eigene Darstellung*



# Werkzeuge für offene Wissenschaft



Werkzeuge für offene Wissenschaft. Eigene Darstellung.

# Versionsmanagement und Kollaboration mit Git & Github

- Git stellt ein sehr anspruchsvolles und sehr mächtiges Versionsmanagementwerkzeug dar:
  - Aufzeichnung des Verlaufs von Dateiversionen
  - Vergleich von Dateiversionen
  - Branches und Merges
- Github, Bitbucket etc. erlauben Onlinedeposition und -kollaboration.
- Git & Github sind sehr gut in RStudio integriert.

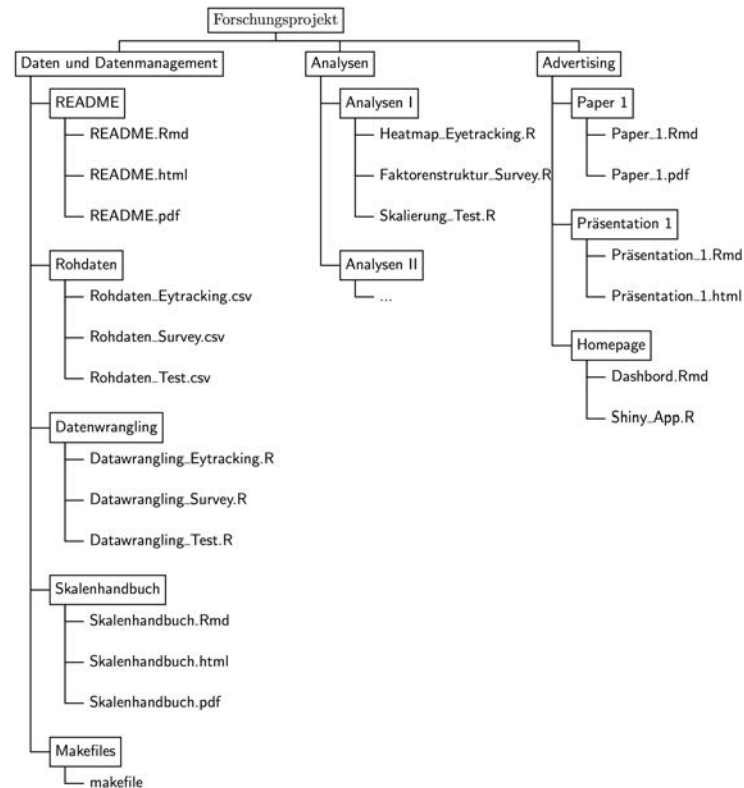
# RMarkdown + Packrat + Shiny = + + + +

- RStudio hat eine One-Button-Publikations Funktion für **statische** R-Skripte und via RMarkdown generierte html-Files.
- Einen **interaktiven** öffentlichen Zugang zu Forschungsergebnissen/Forschungsdaten erlauben \* Notebooks (Adressaten: Forscher/innen, Werkzeuge: {RStudio-Notebooks}, Jupyter) und \* Shiny-Apps (Adressat: Öffentlich, Werkzeug: {shiny}).
- Notebooks erlauben den interaktiven Umgang von **Code** und Output.
- Shiny (*Chang, Cheng, Allaire, Xie & McPherson, 2016*) erstellt html5-Formate die **per Klick** R-Input erzeugen und den Output rendern
  - Webpages
  - Dashboards
  - e-Books

Beispiele für  +  +  +  + 

Code-Beispiele

# Big Picture: Reproduzierbare und offene Forschung mit R und RStudio



*Big Picture. Eigene Darstellung.*

# Wie einsteigen?

- Erfahrungsgemäß stellt die Verwendung von R die größte Hürde dar.
- `RMarkdown`, `knitr` und `pandoc` sind dank RStudio wenigen Stunden lernbar (Halbtagesworkshop).
- Alltägliche Verwendung
  - Lehre
    - Hausaufgaben als gerenderte RMarkdown-Files ausgeben und einsammeln.
    - Veranstaltungen (halb-)öffentlich und interaktiv dokumentieren.
  - Forschung
    - **“Never touch your rawdata & avoid human interaction”**-Maxime motiviert die Verwendung von RMarkdown, knitr und pandoc enorm.

# Literatur

- Allaire, J.J., Cheng, J., Xie, Y., McPherson, J., Chang, W., Allen, J. et al. (2017). Rmarkdown: Dynamic Documents for R.
- Chang, W., Cheng, J., Allaire, J.J., Xie, Y. & McPherson, J. (2016). Shiny: Web Application Framework for R.
- Dowle, M. & Srinivasan, A. (2017). Data.Table: Extension of 'data.Frame'.
- Gandrud, C. (2014). Reproducible Research with R and RStudio. Chapman & Hall.
- Hardwicke, T.E., Mathur, M.B., MacDonald, K., Nilsson, G., Banks, G.C., Kidwell, M.C. et al. (2018). Data Availability, Reusability, and Analytic Reproducibility: Evaluating the Impact of a Mandatory Open Data Policy at the Journal Cognition. Royal Society Open Science, 5 (8), 180448. doi:[10.1098/rsos.180448](https://doi.org/10.1098/rsos.180448)
- Lapp, H. (2015). A Curriculum for Teaching Reproducible Computational Science Bootcamps. Dublin: BOSC 2015, Duke University.
- Peng, R.D. (2011). Reproducible Research in Computing Science. Science, 334 (6060), 1226–1227. doi:[10.1126/science.1213847](https://doi.org/10.1126/science.1213847). **Reproducible**
- Stockemer, D., Koehler, S. & Lentz, T. (2018). Data Access, Transparency, and Replication: New Insights from the Political Behavior Literature. PS: Political Science & Politics, 51 (4), 799–803. doi:[10.1017/S1049096518000926](https://doi.org/10.1017/S1049096518000926)
- Stodden, V., Seiler, J. & Ma, Z. (2018). An Empirical Analysis of Journal Policy Effectiveness for Computational Reproducibility. Proceedings of the National Academy of Sciences, 115 (11), 2584–2589. doi:[10.1073/pnas.1708290115](https://doi.org/10.1073/pnas.1708290115)
- Wickham, H. (2016). Feather: R Bindings to the Feather 'API'.
- Wickham, H. (2017). Dbplyr: A 'dplyr' Back End for Databases.
- Wickham, H. & Bryan, J. (2017). Readxl: Read Excel Files.
- Wickham, H., Hester, J. & Francois, R. (2017). Readr: Read Rectangular Text Data.
- Wickham, H. & Miller, E. (2018). Haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files. haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files. <https://cran.r-project.org/package=haven>.
- Xie, Y. (2015). Dynamic Documents with R and Knitr (Second., Band 29). Boca Raton, FL: CRC Press.