

Markdown und Git

Moderne Methoden der Wissenschaft

ifes/iwp | FOM | Sebastian Sauer

Gliederung

1. Reproduzierbarkeit
2. Markdown
3. Git
4. Fazit

Ihre Erwartungen an diesen Workshop



Lernziele

TEIL *EINFÜHRUNG*

1. Relevanz von Reproduzierbarkeit vermitteln

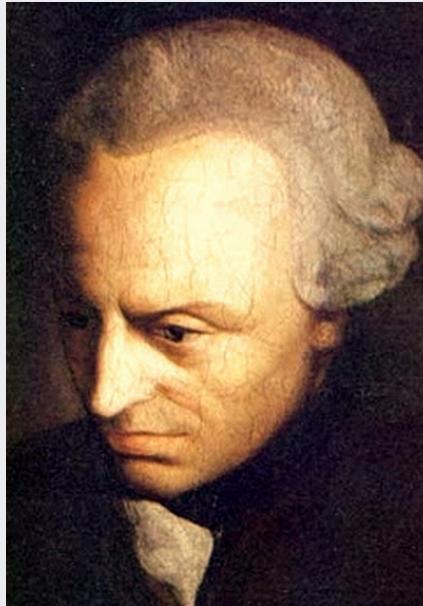
TEIL *MARKDOWN*

1. Nützlichkeit/Anwendungsfälle von Markdown vermitteln
2. Kompetenz für typische Anwendungsfälle (für Markdown) vermitteln

TEIL *GIT*

1. Nützlichkeit/Anwendungsfälle von Git vermitteln
2. Grundkompetenz (von Git) vermitteln

Was ist das Wesensmerkmal der Wissenschaft?



"Habe den Mut,
dich deines eigenen
Verstandes zu
bedienen".

~~Das~~ Ein Wesensmerkmal der Wissenschaft ist *Transparenz*.

Wissenschaft heißt zeigen, nicht glauben.

Transparenz ist notwendig für Reproduzierbarkeit

reproducibility refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator. [GFI16].

Ein Forschungsbericht sollte wie ein Kochrezept sein: Alle Informationen sind enthalten, um das Gericht jeweils (ausreichend) gleich nachzukochen.

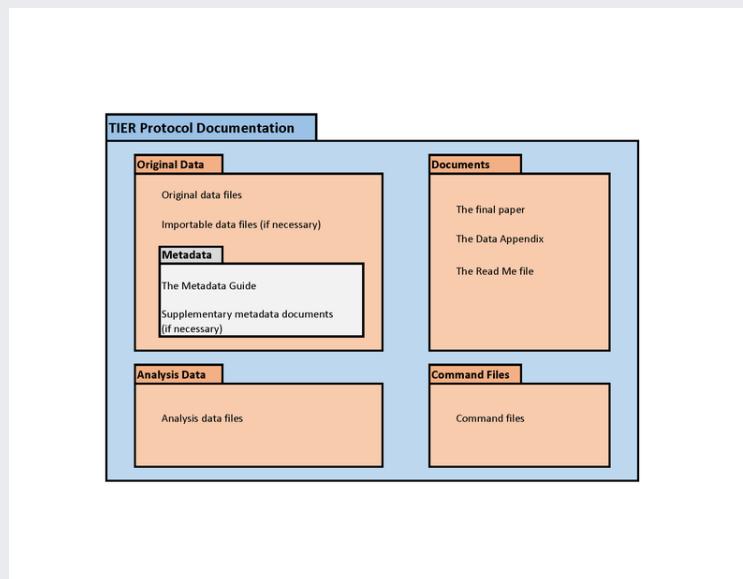
⚡ Ist ein Forschungsergebnis nicht reproduzierbar, sind wir zum (Nicht-)Glauben gezwungen. Das ist keine Wissenschaft.

Open Science: Open data, open materials, open code, literate programming ...

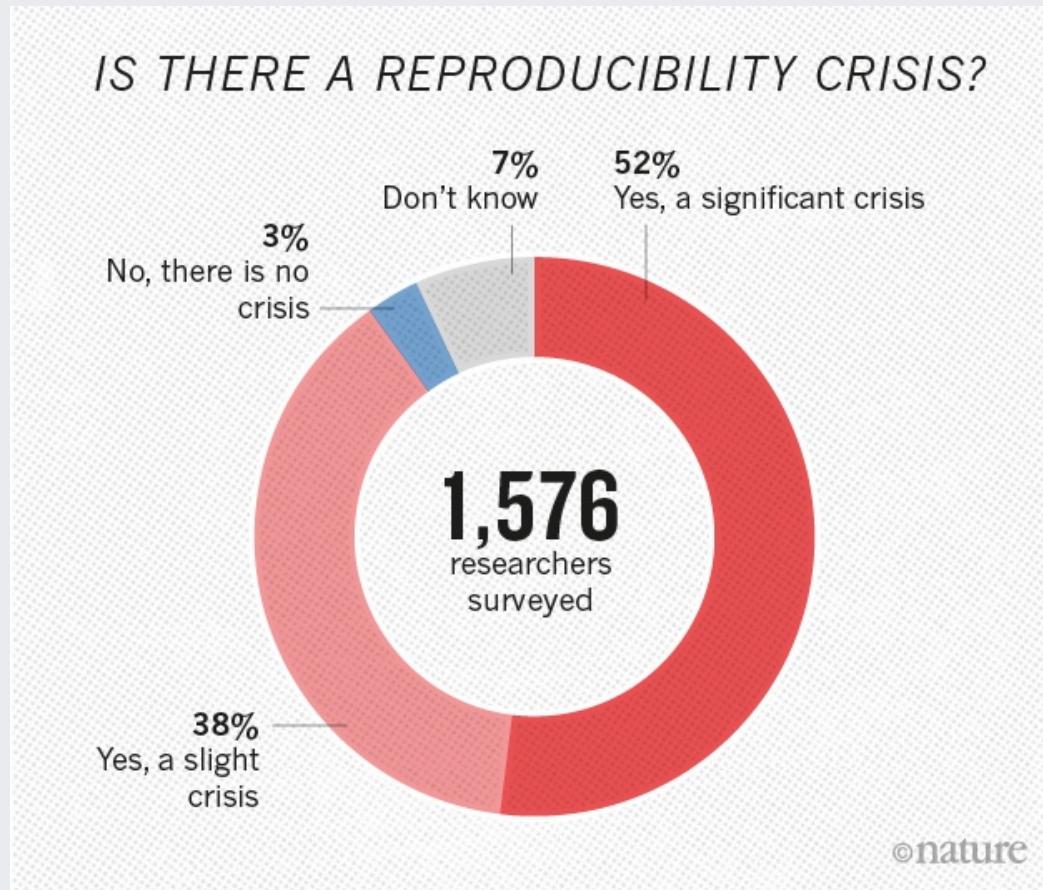
Beispiel: Projekt TIER

Project TIER (Teaching Integrity in Empirical Research) promotes the integration of principles and practices related to transparency and replicability in the research training of social scientists.

Our goal is to reach a day in which training in research transparency becomes standard and ubiquitous in the education of social scientists.



Aber haben wir ein Problem mit geringer Reproduzierbarkeit?

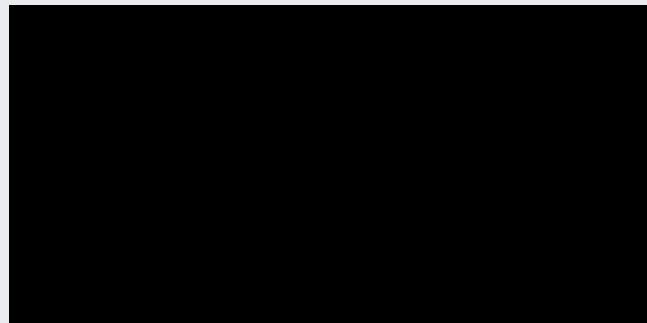


[Bak16]

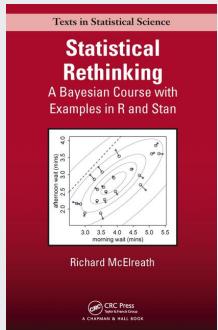
Was sind die Ursachen geringer Reproduzierbarkeit?

- *Fehlende Informationen*
- Fehlende Nachvollziehbarkeit
- Copy-Paste-Fehler
- Keine Syntax
- Keine frei nutzbaren Messinstrumente
- Software-Updates
- Verwendete Software nicht mehr verfügbar
- Verwendete Werkzeuge zu teuer
- ...

Reproduzierbarkeit in typischer Forschungsarbeit



Ein prominenter Forscher zu "Point-and-Click"



[The command line] also saves you time and fulfills ethical obligations. With command scripts, each analysis documents itself, so that years from now you can come back to your analysis and replicate it exactly. You can re-use your old files and send them to colleagues. Pointing and clicking, however, leaves no trail of breadcrumbs. [...] Once you get in the habit of planning, running and preserving your statistical analysis in this way, it pays for itself many times over. With point-and-click, you pay down the road, rather than only up front. [McEl6]

Wäre es nicht schön ... 1/2

"Die mittlere Reaktionszeit betrug [hier automatisch das Ergebnis der Berechnung einfügen] Sekunden."



Wäre es nicht schön ... 2/2

Bla bla bla ...

```
x1 = rechne(Input_Data)  
x2 = rechne_mehr(x1)  
x3 = irgendwas(x2)  
x4 = wildes_zeugs(x3)
```

Die mittlere Reaktionszeit betrug
x4 Sekunden.

Bla bla bla ...

Bla bla bla ...

Die mittlere Reaktionszeit betrug
3.141 Sekunden.

Bla bla bla ...

Gliederung

1. Reproduzierbarkeit

2. Markdown

3. Git

4. Fazit

Es geht: Z.B. mit Markdown

Ich bin ein Titel.

Ich bin ein Satz.

Eine Liste beginnt.

- ja
- nein
- vielleicht

Eine nummerierte Liste:

1. 3.4145
2. 2.7182
3. 42

Ich bin ein Titel.

Ich bin ein Satz.

Eine Liste beginnt.

- ja
- nein
- vielleicht

Eine nummerierte Liste:

1. 3.4145
2. 2.7182
3. 42

Markdown ist fast WYSIWYG

__fett__,

kursiv,

~~durchgestrichen~~

fett,

kursiv,

~~durchgestrichen~~

Noch mehr Markdown

```
# Title level 1
```

.pull_right[

Title level 1

```
## Title level 2
```

Title level 2

Link: [\[my blog\]](#)([Citation: \[@Xie2018\]](https://data-se</p></div><div data-bbox=)

Link: [my blog](#).

Image: ![\[my image\]](#)([imgs/R-logo_s](#)Citation: (Xie & Allaire, 2018)

Image: 

Bilder einfügen

- via Markdown:

```
![my image](imgs/R-logo_small.png)
```

- via HTML:

```

```

- via R:

```
knitr::include_images("path_to_image")
```

Tipp: [RStudio-Snippets](#) anlegen für häufige Idiome.

Markdown ist eine Markup-Sprache

Markup-Sprachen nutzen Auszeichnungen zur Formatierung von Text. Sie basieren auf Textdateien.

Bekannte Beispiele:

- HTML
- LaTeX
- XML
- Word (ist XML)
- Powerpoint (ist XML)
- ...

Markdown ist eine sehr einfache Markup-Sprache (daher "Mark-Down").



Übung: Erstellen Sie ein Markdown-Dokument in RStudio!

1. In RStudio: File > New File > R Markdown > ... OK
2. Löschen Sie alles, was in der Datei schon steht
3. Schreiben Sie ein paar Zeilen in Markdown
4. Klicken Sie auf "Knit" (engl. für stricken) oder drücken Sie Strg-Shift-K

Erklärvideo

Markdown is nicht Latex

```
\makeatletter
\newenvironment{kframe}{%
\medskip{}}
\setlength{\fboxsep}{.8em}
\def\at@end@of@kframe{%
\ifinner\ifhmode%
\def\at@end@of@kframe{\end{minipage}}%
\begin{minipage}{\columnwidth}%
\fi\fi%
\def\FrameCommand##1{\hspace{\totalleftmargin}\hspace{-\fboxsep}
\colorbox{shadecolor}{##1}\hspace{-\fboxsep}
% There is no \totalrightmargin, so:
\hspace{-\ linewidth}\hspace{-\totalleftmargin}\hspace{\columnwidth}}%
\par\unskip\endMakeFramed%
\at@end@of@kframe}
\makeatother
```

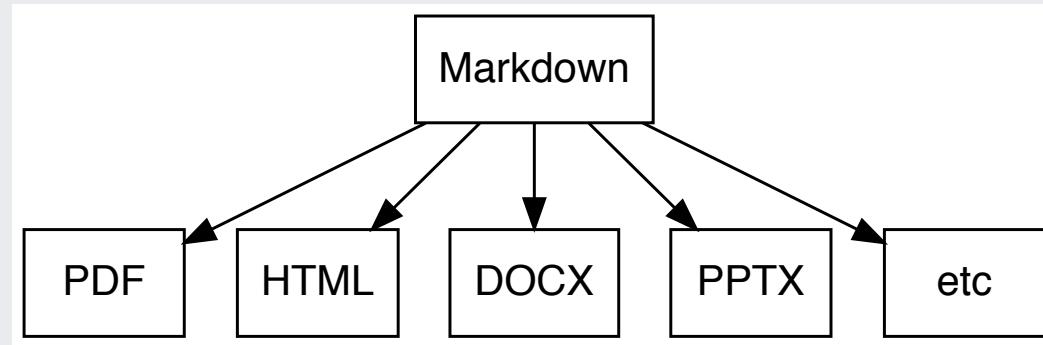
Zum Glück.

Ich, als ich neulich kurz was in meiner Latex-Präambel ändern wollte



Markdown als *lingua franca* der Markup-Sprachen

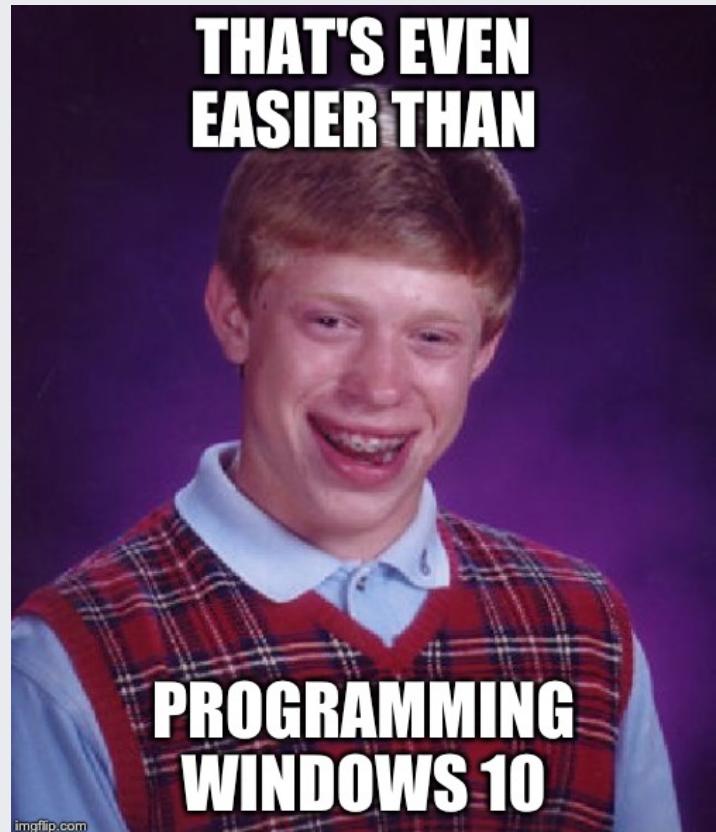
Mit dem Programm **Pandoc** lassen sich Markup-Sprachen ineinander übersetzen.



Kurz mal Pause ...

Wir haben uns *nicht* mit Formatierung des Textes beschäftigt, nur mit seinem Inhalt.

Jemand (etwas) hat uns die Formatierung abgenommen.



R + Markdown = RMarkdown



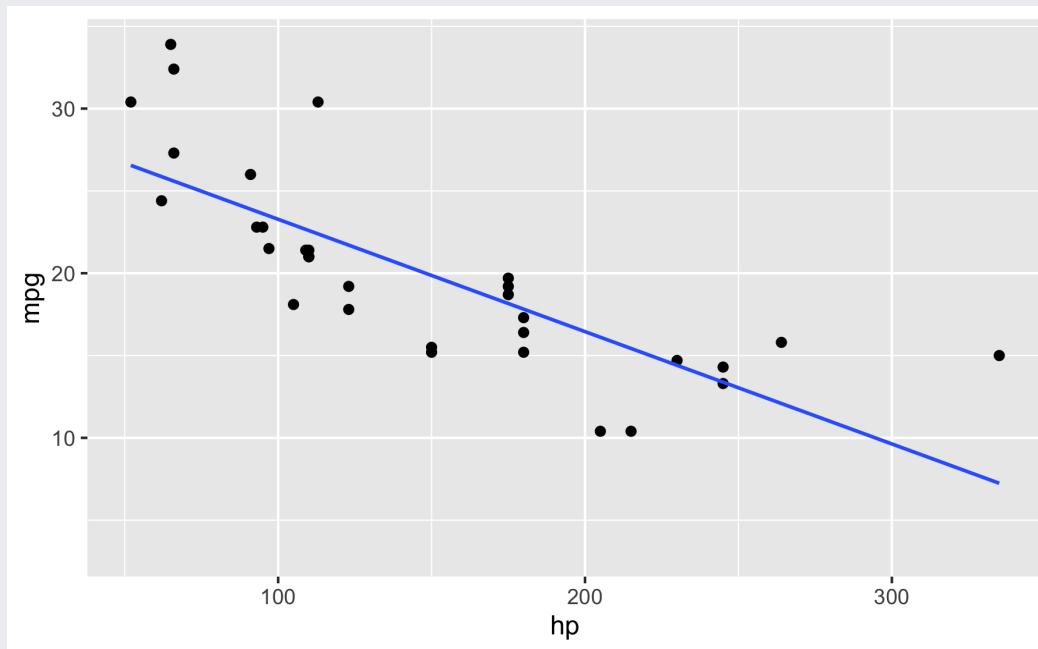
```
# Überschrift Ebene 1  
Bla bla bla.  
__Fettdruck, weil dicke Hose__  
```{r}  
x <- 2^10
```
```

Die Lösung ist `r x`.

R-Diagramme

Auf dem üblichen Weg:

```
gf_point(mpg ~ hp, data = mtcars) %>%  
  gf_lm()
```





Übung: Erstellen Sie ein RMarkdown-Dokument in RStudio!

Sie können mit diesem Code beginnen und dann nach eigenem Belieben anpassen:

```
# Überschrift Ebene 1  
  
Bla bla bla.  
__Fettdruck, weil dicke Hose__  
  
```{r}  
x <- 2^10
```
```

Die Lösung ist `r x`.

Markdown ist schon ziemlich cool



Was sind die Metadaten eines Dokuments?

Metadaten sind Daten über Daten (hier: über ein Dokument), z.B.

- Titel
- Autor
- Papiergröße
- ...

Metadaten in RMarkdown

Metadaten werden in RMarkdown in einem Bereich, der mit `---` vorne und hinten abgegrenzt ist, definiert:

```
---
title: "Das ist der Titel des Dokuments"
author: "Hier steht der Name des Authors"
date: 2019-09-11 # Man kann auch R-Inline-Befehle einflechten
---
```

Das Format ist i.d.R. **Variable = Wert**.

Genauer gesagt wird das ([YAML](#)-)Format verwendet.

Meist steht dieser Bereich zu Beginn des Dokuments (daher "YAML-Header").



Übung: Fügen Sie Ihrem Rmd-Dokument einen YAML-Header hinzu!

```
---
```

```
title: "Das ist der Titel des Dokuments"
author: "Hier steht der Name des Authors"
date: `r Sys.Date()`
```

```
--
```

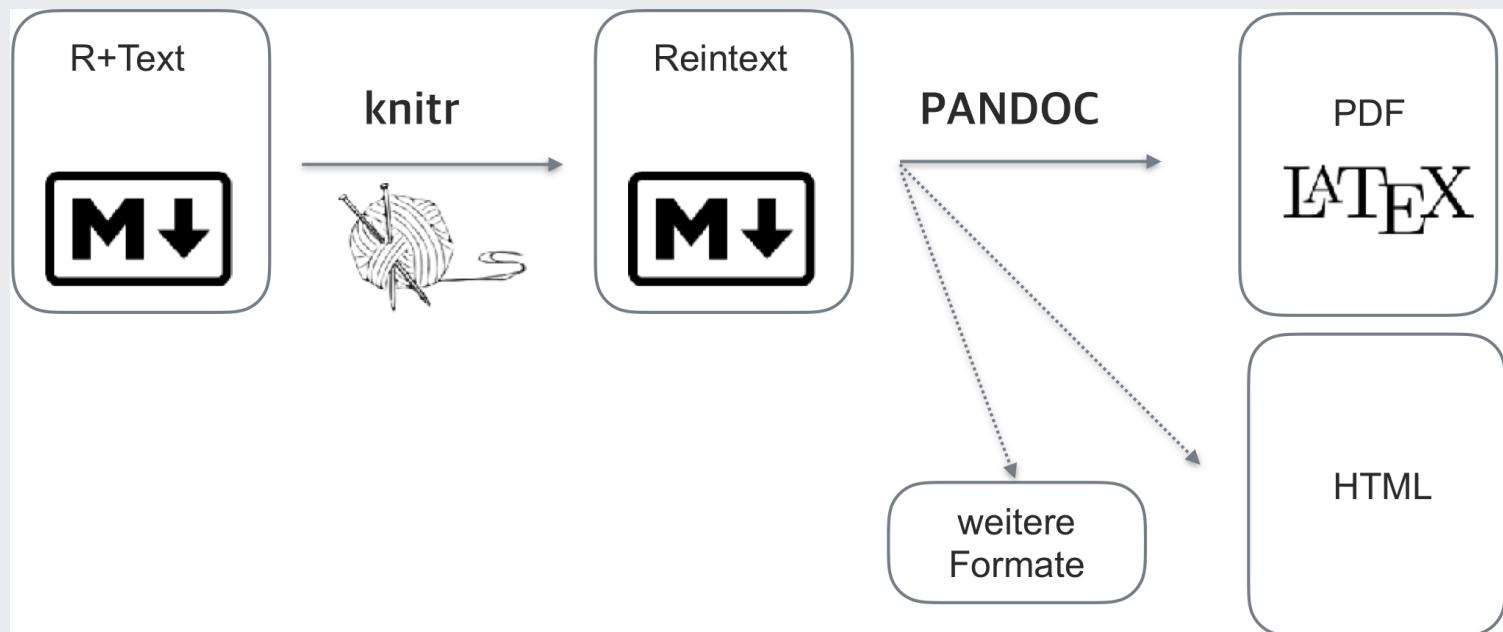
```
# Überschrift Ebene 1
```

```
Bla bla bla.  
__Fettdruck, weil dicke Hose__
```

```
```{r}
x <- 2^10
```
```

```
Die Lösung ist `r x`.
```

RMarkdown-Ablauf



Wofür kann man (R)Markdown alles
verwenden?

Websites/Blogs

Berichte

Analysis of flights from NYC in 2013;
QUARTER: 3



Background and Objective

This report analyses the number of flights with origin one of the airports of NYC.

Main figures

In the present quarter (**Quarter: 3**) there were a total of **84059** flights. From **JFK** started **28415** flights. From **LGA** started **26012** flights. From **EWR** started **29632** flights.

Non eram nescius, Brute, cum, quae summis ingenis exquisitaque doctrina philosophi Graeco sermone tractavissent, ea Latinis litteris mandaremus, fore ut hic noster labor in varias reprehensiones incurreret.

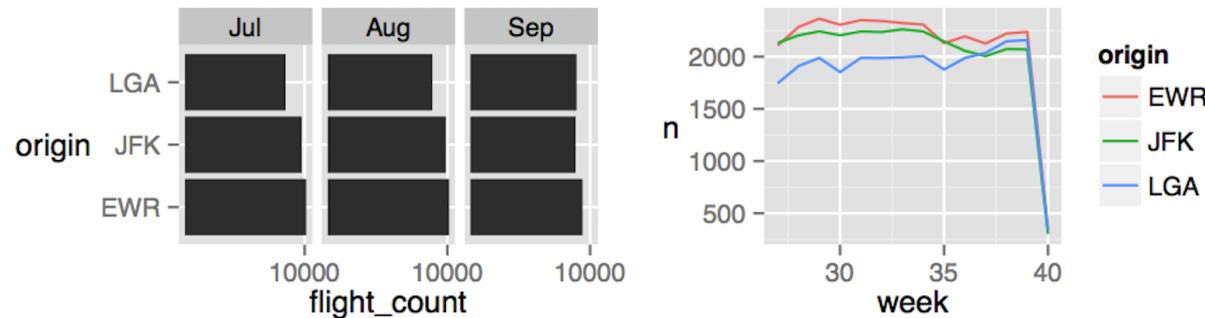


Figure 1: Number of flights per month (left panel) and week (right panel) from each NYC origin airport

Bücher

Moderne Datenanalyse mit R

Die Kaufempfehlung, die Ihnen ein Webstore ausspricht, die Einschätzung, welcher Kunde kreditwürdig ist, oder die Analyse der Werttreiber von Immobilien – alle diese Beispiele aus dem heutigen Leben sind Ergebnis moderner Verfahren der Datenanalyse. Dieses Buch führt in solche statistische Verfahren anhand der Programmiersprache R ein.

Ziel ist es, Leser mit der Art und Weise vertraut zu machen, wie führende Organisationen und Praktiker angewandte Statistik heute einsetzen. Weil sich mit der Digitalisierung auch die statistischen Verfahren verändert haben, vermittelt der Autor neben klassischen Analysemethoden wie Regressionsanalyse auch moderne Methoden wie Textmining und Random-Forest-Modelle. Dabei sind die Inhalte des Buchs durchgehend so aufbereitet, dass sie auch für Leser ohne umfangreiche mathematische Vorkenntnisse verständlich sind. Anhand von Fallbeispielen und Übungen werden die Leser durch alle Phasen der Datenanalyse geführt: Sie lernen, wie Daten eingelesen, aufbereitet, visualisiert, modelliert und kommuniziert werden können. Dabei wird vor allem die Aufbereitung, Umformung und Prüfung der Daten ausführlicher als in anderen Publikationen behandelt, da dieser Teil in der Praxis oft einen wesentlichen Teil des Aufwands ausmacht. Aber auch die Visualisierung bekommt viel Raum, denn gute Diagramme ermöglichen Einblicke, die Zahlen und Worte verbergen.

Mit seinem praxisorientierten Ansatz will das Buch dazu befähigen,

- alle grundlegenden Schritte eines Datenanalyseprojekts durchzuführen,
- Daten kompetent in R zu bearbeiten,
- simulationsbasierte Inferenzstatistik anzuwenden und kritisch zu hinterfragen,
- klassische und moderne Vorhersagemethoden anzuwenden und
- betriebswirtschaftliche Fragestellungen mittels datengetriebener Vorhersagemodelle zu beantworten.

Der Autor

Professor Dr. habil. Sebastian Sauer arbeitet als Hochschullehrer für Wirtschaftspsychologie an der FOM Hochschule für Oekonomie & Management in Nürnberg und versteht sich als Data Scientist. Daten mit R zu analysieren, ist aktuell eines seiner zentralen Interessengebiete. Besonderes Augenmerk legt er auf den Erkenntnisbeitrag, den neue Analyseverfahren leisten. Neben dem „Wie“ der Datenanalyse beschäftigen ihn die Grenzen und Gefahren, die die moderne Datenwissenschaft für den Einzelnen und die Zivilgesellschaft mit sich bringt. Außerdem engagiert er sich für das Thema Open Science und interessiert sich für die Frage, wie die Psychologie zur Klärung von Problemen mit gesellschaftlicher Relevanz beitragen kann. Sein Blog <https://data-se.netlify.com/> dient ihm als Notizbuch sich entwickelnder Gedanken. Data Science für die Wirtschaft bietet er auf <https://www.data-divers.com/> an.



Sauer



Moderne Datenanalyse mit R

FOM-Edition

Sebastian Sauer

Moderne Datenanalyse mit R

Daten einlesen, aufbereiten,
visualisieren, modellieren
und kommunizieren



Springer Gabler

Folien (z.B. mit Xaringan)



Xaringan basiert auf [remark.js](#)

Beispiele für Xaringan-Präsentationen

1. Alison Hill - RStudio Conf 2019
2. Yongfu Liao - Chocolate Theme
3. Yihui Xie - Presentation Ninja
4. Science Communication with Xaringan/R
5. Emi Tanaka - Kunoichi Theme

Papaja: Eine Vorlage für APA-Manuskripte

OOM REVISED (REV. 2)

3

²¹ Observation Oriented Modeling Revised From a Statistical Point of View

²² A dominant procedure in applied statistical reasoning **is** null hypothesis significance

²³ testing (NHST). Althought it is used frequently, voices of criticism have always been present

²⁴ and have gained prominence recently. One alternative is “Observation Oriented Modeling”

²⁵ (OOM), **developed** by Grice (2011). **This method is suitable in some situations in**

²⁶ **the behavioral sciences as it circumvents some of the problems with existing**

²⁷ **inference or modeling techniques. The aim of the present paper is to further**

²⁸ **refine OOM, and present a version that is based on well known classification**

²⁹ **methods. Specifically**, we present a matrix algebra computation approach that is different

³⁰ from the one proposed by **Grice**, and we add an accuracy gain coefficient that **is** novel to

³¹ OOM. The paper is structured as follows: First, we describe some problems with current

³² inference practices, and go on to explain why OOM may be able to alleviate some of these

³³ problems. Next, we describe the original OOM and our revised version. The final part of the

³⁴ manuscript reports a practical application of OOM in some detail as a reanalysis of

³⁵ published data.

yart: Eine Vorlage für Forschungsberichte

The screenshot shows the RStudio interface with two panes. The left pane displays an R Markdown file named 'yart4.Rmd'. The code includes sections for bibliography, author information, and a LaTeX preamble. It also contains a large block of placeholder text from 'Frangilla' and various R Markdown syntax examples. The right pane shows the generated PDF output titled 'yart4.pdf'. The PDF has a title page with the title 'Forschungsbericht', a logo placeholder, and the subtitle 'Seminar: Solutions to All and Nothing'. Below this are sections for 'THIS REPORT HAS A TITLE' and 'AT LEAST IT PRETENDS TO'. The main body of the report contains the placeholder text and the LaTeX header information.

```
11 bibliography: bib.bib
12 lot: yes
13 course: "Seminar: Solutions to All and Nothing"
14 referee: "Referee: Prof. Dr. I. Weiss-Ols"
15 address: "Meine Straße 1, 12345 Hausenhausen"
16 ID: "Matrikelnummer: 12345678"
17 toc: true
18 abstract: !
19 this manuscript demonstrates how to use R Markdown and pandoc to
20 create an APA conform manuscript. pandoc builds on R Markdown,
21 uses pandoc to turn Markdown into PDF or Word documents. The conversion
22 to Word documents currently supports only a limited set of features.
23 output: yart::yart
24
25
26
27 # Section Header 1
28
29 If you use a logo file, images, or other files, make sure the files are located in the folder of this .rmd file.
30
31 ## Section Header 2
32
33
34 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin mollis
35 dolor vitae tristique eleifend. Quisque non ipsum sit amet velit
36 malesuada consectetur. Proesent vel facilisis leo. Sed facilisis
37 varius ut aliquam lorem malesuada non turpis purus etiam
38 fringilla volutpat. Ut aliquam lobortis bibendum nulli. Nulla
39 purus justo, congue egestas ante o. elementum sollicitudin erat. Cras
40 porta augue ligula, vel adipiscing odio ullamcorper eu. In tincidunt
41 nisi sit amet tincidunt tincidunt. Maecenas elementum neque eget dolor
42 [legestas fringilla](http://example.com);
43
44
45 > Nullam eget dapibus quam, sit amet sagittis magna. Nam tincidunt,
46 > erci ac imperdiet ultricies, neque metus ultrices quam, id gravida
47 > augue. Locus ac leo.
48
49 Vestibulum id sodales lectus, sed scelerisque quam. Nullam euctor mi
50 et feugiat commodo. Duis interdum imperdiet nulla, vitae bibendum eros
51 placerat non. Cross ornare, risus in faucibus malesuada, libero sem
52 fringilla quam, ut luctus enim sapien eget dolor.
53
54 - Aufzählungen (nummerniert oder nicht) sind möglich.
55 - Sonderzeichen werden unterstützt: äüß.
56
57 - \\LaTeX wird unterstützt.
58
59 - Und damit auch "schöne" Formeln: $e^{ln(e)}=e$ (stimmt das?).
60
61 - Ein Überblick zur **Markdown-Syntax** findet sich hier.
62
63
64
65
66
67 - Ein paar Gimicks: H-2-O, This --is deleted text--, feasible, not feasible, long---ganz lang.
68
69 - Fußnoten sind erlaubt[1].
70
71 - Zitationen sind möglich, im beliebigen Format, z.B. APA6. Das Format wird über die Variable 'cls' definiert
    (im Kopfteil oben). Die entsprechende Datei muss im gleichen Ordner liegen wie diese Rmd-Datei. Die Datei mit
```

THIS REPORT: HAS A TITLE
AT LEAST IT PRETENDS TO

Seminar: Solutions to All and Nothing
Sophism

Sebastian Sauer
Matrikelnummer: 12345679
Meine Straße 1, 12345 Hausenhausen

13. Mai 2016
Referee: Prof. Dr. I. Weiss-Ols

Poster mit posterdown



Using posterdown to generate reproducible conference posters via RMarkdown > Knitr > Markdown > Pandoc > HTML/CSS > PDF workflow

W. Brent Thorne¹, William B. Thorne¹

¹Department of Earth Science, Brock University

1 Introduction

Welcome to `posterdown`! This is my attempt to provide a semi-smooth workflow for those who wish to take their `RMarkdown` skills to the conference world. Many creature comforts from `RMarkdown` are available in this package such as `Markdown` section notation, figure captioning, and even citations like this one (Allaire, Xie, McPherson, et al. 2018). The rest of this example poster will show how you can insert typical conference poster features into your own document.

1.1 Study Site

Here is a map made to show the study site using `ggplot2`, `ggspatial`, and `sf` and you can even reference with a hyperlink, this will take you to [Figure 1.1](#) (Dunnington, n.d.; Arnold, n.d.; Pebesma 2018).

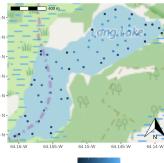


Figure 1.1: Map of Long Lake example from the `ggspatial` package.

1.2 Objectives

1. Easy to use reproducible poster design.
2. Integration with `RMarkdown`.
3. Easy transition from `posterdown` to `thesistdown` or `articles` (Allaire, Xie, R Foundation, et al. 2018; Solomon 2019).

2 Methods

This package uses the same workflow approach as the `posterdown` you know and love. Basically it goes from `RMarkdown` > `Knitr` > `Markdown` > `Pandoc` > `HTML/CSS` > `PDF`. You can even use the bibliography the same way (Turner et al. 2014).

Lore ipsum dolor sit amet, consectetur adipiscing elit. Quisque auctor lorem ex. Pellentesque tempus accumsan nulla, nec eleifend erat rhoncus vel. Ut commodo metus ipsum, sed accumsan risus placerat ut.

```
library(ggplot2)
library(ggspatial)
library(sf)
library(knitr)
library(ggposterdown)

ggplot(gapminder, aes(gdpPercap,
  lifeExp,
  size = pop,
  colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country.colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~ continent)
# Here comes the gganimate specific bits
transition_time(~ year - (time * year),
  x = "GDP per capita",
  y = "life expectancy") +
  transition_time(~ year) +
  ease_aes("linear")
```

3 Results

Usually you want to have a nice table displaying some important results that you have calculated. In `posterdown` this is as easy as using the `kable` table formatting you are probably used to as per typical `RMarkdown` formatting. I suggesting checking out the `kableExtra` package and its in depth documentation on customizing these tables found [here](#) (Zheng 2019). Hopefully I can make this with an inline reference like, Table 3.1:

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|--------------|-------------|--------------|-------------|
| 5.1 | 3.5 | 1.4 | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| 4.6 | 3.1 | 1.5 | 0.2 |
| 5.0 | 3.6 | 1.4 | 0.2 |

Look at this animation (Pedersen and Robinson 2017)!!!! Figure 3.1.

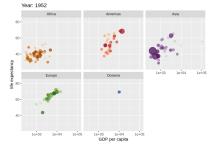


Figure 3.1: WOW THIS IS AN AWESOME GIF!

4 Next Steps

Lore ipsum dolor sit amet, consectetur adipiscing elit. Aliquam placerat augue at velit tincidunt semper. Donec elementum porta posuere. Nullam interdum, odio at tincidunt feugiat, turpis nisi blandit eros, eu posuere risus felis non quam. Nam eget lorem odio. Duis et aliquet orci. Phasellus nec viverra est. Praesent odio orci, mattis vel mauris nec, consectetur fermentum mauris. Etiam eu hendrerit tortor. Donec mi tellus, efficitur et porttitor eu, auctor eu tellus. Quisque faucibus vestibulum sapien vel lacinia. Ut auctor lorem non interdum blandit.

```
library(ggplot2)
library(ggspatial)
library(sf)
library(knitr)
library(ggposterdown)

ggplot(gapminder, aes(gdpPercap,
  lifeExp,
  size = pop,
  colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country.colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~ continent)
# Here comes the gganimate specific bits
transition_time(~ year - (time * year),
  x = "GDP per capita",
  y = "life expectancy") +
  transition_time(~ year) +
  ease_aes("linear")
```

5 Conclusion

Lore ipsum dolor sit amet, consectetur adipiscing elit. Aliquam placerat augue at velit tincidunt semper. Donec elementum porta posuere. Nullam interdum, odio at tincidunt feugiat, turpis nisi blandit eros, eu posuere risus felis non quam. Nam eget lorem odio. Duis et aliquet orci. Phasellus nec viverra est. Praesent odio orci, mattis vel mauris nec, consectetur fermentum mauris. Etiam eu hendrerit tortor. Donec mi tellus, efficitur et porttitor eu, auctor eu tellus. Quisque faucibus vestibulum sapien vel lacinia. Ut auctor lorem non interdum blandit.



Figure 5.1: Here is a leaflet figure which will run as expected online, when printed it will take the last state it is left in before choosing to print.

References

Allaire, J., Yihui Xie, Jonathan McPherson, Kevin Ushey, Anne Konda, Hadley Wickham, Joe Cheng, Wilson Chang, and Richard Iannone. 2018. *Reproducible Documents for R*. <https://rmarkdown.rstudio.com>.

Allaire, J., Yihui Xie, R. Foundation, Hadley Wickham, J. H. Wickham, and R Core Team. 2014. *knitr: A General Framework for Dynamic Documents for R*. <https://CRAN.R-project.org/packag...>.

Arnold, Jeffrey B. n.d. *Ggplot2: Spatial Data Framework for Ggplot2*. <https://rdrr.io/cran/ggplot2/man/index.html>.

Dunnington, Dwyer, n.d. *ggspatial: Spatial Data Framework for Ggplot2*. <https://rdrr.io/cran/ggspatial/man/index.html>.

Pedersen, Thomas Lin, and David Robinson. 2017. *Gganimate: A Grammar of Animation*. <https://CRAN.R-project.org/packag...>.

Solomon, Nick. 2019. *Thesistdown: An Updated R Markdown Thesis Template Using the Bookdown Package*.

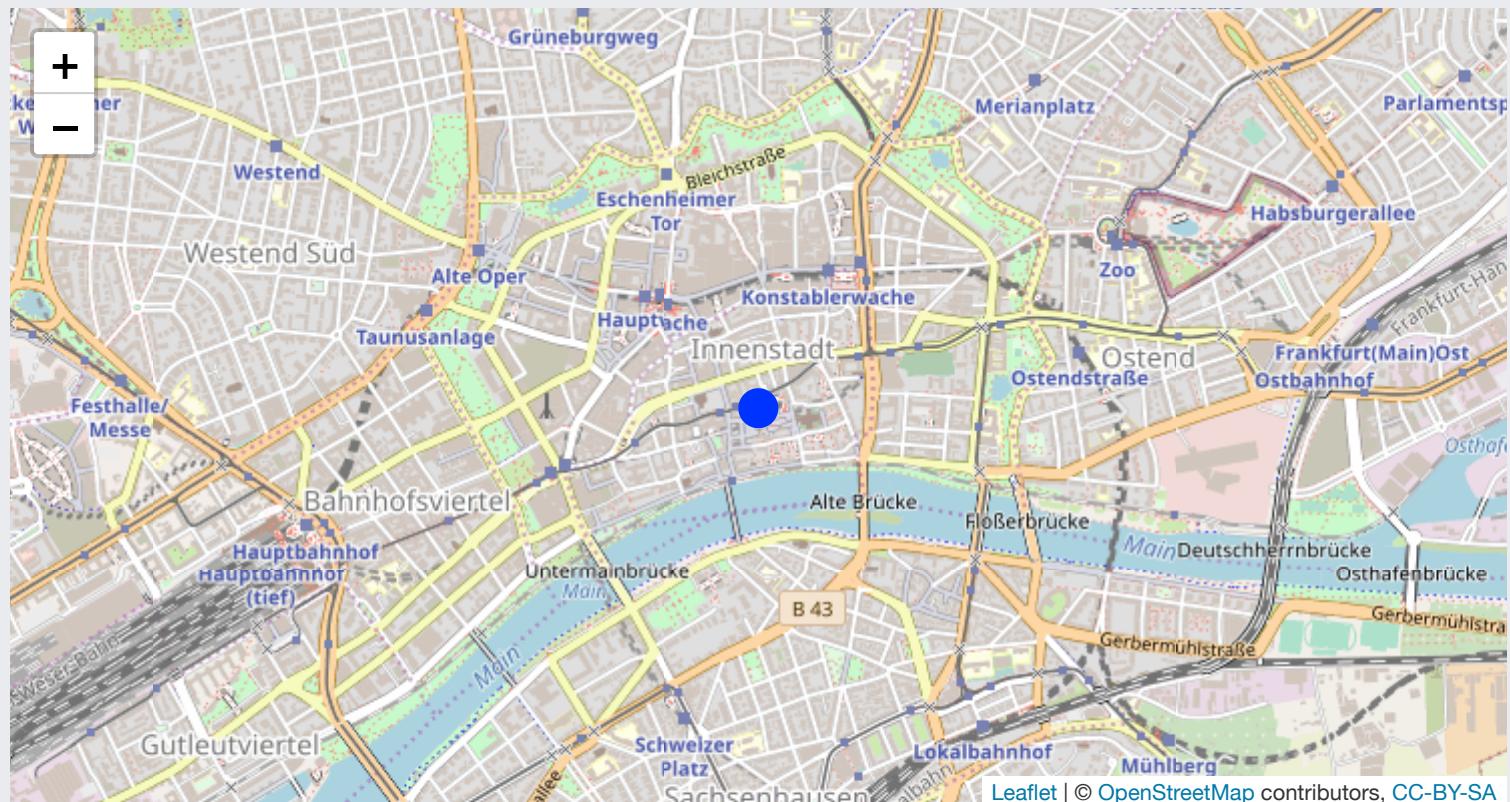
Thorne, W. Brent. 2018. *Posterdown: An R Package Built to Generate Reproducible Conference Posters*. <https://CRAN.R-project.org/packag...>.

Turner, Kevin W., Darrin D. Whittle, Thomas W. D. Edwards, Trevor C. Lantz, Roland J. Hall, and Gillianne Lavoie. 2014. "Controls on Water Balance of Shallow Thermokarst Lakes in the Mackenzie River Basin, Northwest Territories, Canada: A Comparison of Two Assimilated Based on Water Isotope Traces and Remote Sensing in Old Crow Flats, Yukon (Canada)." *Global Change Biology* 20 (3): 1385–1393. <https://doi.org/10.1111/j.1365-2486.2013.03082.x>.

Zhu, Han. 2019. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. <https://CRAN.R-project.org/packag...>.

41 / 107

Interaktive Diagramme in HTML



Ja, aber - Sollte *ich* Markdown
verwenden?

Markdown hat mehr Power



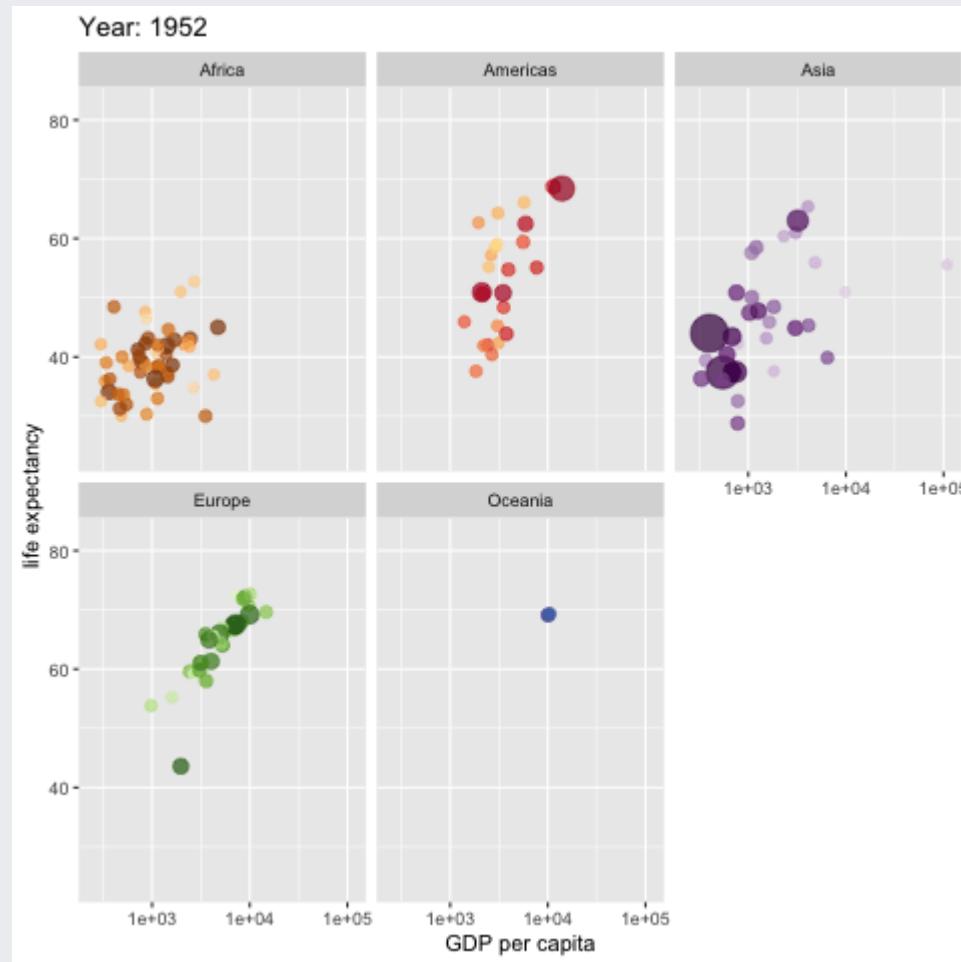
Ja, aber - ich liebe MS Word ...

Eine Seite mit Word, noch ein paar Seiten, und irgendwann ...



Markdown verkraftet auch große Dokumente.

Animationen



Buch "Moderne Datenanalyse mit R" (MODAR)

- ca. 500 Seiten
- ca. 200 Abbildungen
- ca. 200 R-Codes

Relativ großes Dokument.

Keine Schnapp-Atmung beim Bearbeiten/Erstellen, aber relativ lange Kompilationsdauer (ca. 2 Minuten).



Pro MODAR-Kapitel eine Datei

```
rmd_files:  
[  
  "index.Rmd",  
  "01_Statistik_heute.Rmd",  
  "02_Hallo_R.Rmd",  
  "03_R_starten.Rmd",  
  "04_Erstkontakt.Rmd",  
  # "05_Datenstrukturen.Rmd",  
  "06_Datenimport_und_export.Rmd",  
  ...  
]
```

MODAR-Metadaten-Datei

```
title: "Moderne Datenanalyse mit R"  
subtitle: "Entwurf"  
author: "Sebastian Sauer"  
lang: de-De  
documentclass: book  
classoption: a4paper  
fontsize: 11pt  
fontfamily: mathpazo
```

...

MODAR-Beispieleseite

9.2. DATENANOMALIEN

109

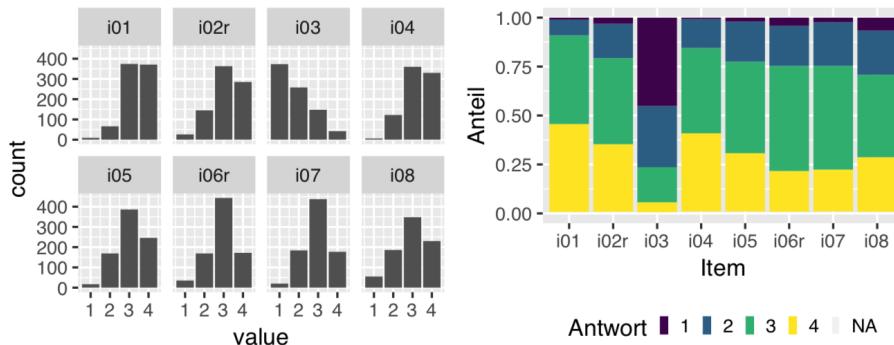


Abbildung 9.2: Anomalien identifizieren

```
stats_test %>%
  dplyr::count(interest, sort = TRUE)

stats_test %>%
  dplyr::count(interest, sort = TRUE) %>%
  ggplot(aes(x = factor(interest), y = n)) +
  geom_point()
```

Schauen wir uns noch die Verteilungen der Item-Antworten an (s. Abbildung 9.2); sehen die plausibel aus? Einige Verteilungen sind ordentlich schief. Man könnte räsonieren, dass man sich damit in guter Gesellschaft befindet - mit Blick auf andere Umfragen. Aber die ungleiche Fallzahl pro Itemantwort und die Abweichung von der Normalverteilung könnten später Probleme bereiten. Es würde sich anbieten, die schiefen Items zu überarbeiten.



Übung: HTML-Stylesheets

Vorab sind 13 HTML-Formatvorlagen (Stylesheets, Themes) installiert:
“cerulean”, “cosmo”, “flatly”, “journal”, “lumen”, “paper”, “readable”,
“sandstone”, “simplex”, “spacelab”, “united”, und “yeti”. [Hier](#) einige Beispiele.

Probieren Sie verschiedene aus!

```
---
```

```
title: "test"
output:
  html_document:
    theme: united
---
```

Syntax-Highlighting

Es gibt auch verschiedene Syntax-Highlight-Stile:

highlight specifies the syntax highlighting style. Supported styles include default, tango, pygments, kate, monochrome, espresso, zenburn, haddock, and textmate. Pass null to prevent syntax highlighting.

Überblick

Passen Sie Ihren YAML-Header entsprechend an:

```
---
title: "test"
output:
  html_document:
    # theme: united
    highlight: tango
---
```



Übung: Prettydoc-Stylesheets (für HTML)

1. Installieren Sie das R-Paket **prettydoc** mit
`install.packages("prettydoc")`.
2. Öffnen Sie eine neue Rmd-Datei "from Template"
3. Wählen Sie die **prettydoc**-Vorlage
4. Basteln Sie nach Herzenslust daran herum.
5. Wählen Sie weitere Vorlagen etc.

[Projektseite](#)



Übung: Chunk-Options

R-Chunks kann man mit Optionen versehen, um die Ausgabe zu steuern.

Entschlüsseln Sie die Bedeutung der Chunk-Optionen!

```
```{r demo-plot, out.width="70%", fig.align = "center",
 echo = FALSE, eval = FALSE}
knitr::include_graphics("imgs/modar-sample1.png")
```
```

Lesen Sie [hier](#) mehr über Chunk-Optionen!

Ein Credo zum Textschreiben

Ich glaube, dass man beim Schreiben ...

1. nur schreiben soll, sich nicht schon mit der Formatierung beschäftigen soll.
2. immer nur eine Sache zu einer Zeit tun soll. Man beginne mit der Gliederung, es folgen stichpunktartige Notizen , schließlich formuliere man Sätze , danach erst beginne man mit Formatierung.
3. reichlich von Metatext Gebrauch machen soll, in Sinne von Kommentaren über seinen Text (als Gedankenstütze).
4. zum Schluss auf anspruchsvolle Typografie achten soll (Ästhetik und Lesbarkeit). Allerdings verzichte man auf Schnörkel.
5. nicht einen 500-Seiten-Haufen schreiben soll, sondern intensiv und auf mehreren Ebenen gliedern soll (Ordner, Dateien, Dateien für Bilder oder Daten, Kapitel, Absätze, ...).

Grundlagen des Zitierens

- Zitations-Infos sollten im Bib-Format in einer Datei abgelegt sein (s. rechte Seite).
- Im YAML-Header wird der Dateiname mit den Zitationsinfos festgelegt.
- Außerdem wird im YAML-Header etwaige Zitationsstile (z.B. APA6) definiert.
- Im Text kann dann mit `[@ID]` zitiert werden, z.B.
`[@baker_1500_2016]`

```
@article{baker_1500_2016,  
  title = {1,500 Scientists Lift ...},  
  volume = {533},  
  issn = {0028-0836, 1476-4687},  
  language = {en},  
  number = {7604},  
  journal = {Nature},  
  doi = {10.1038/533452a},  
  author = {Baker, Monya},  
  month = may,  
  year = {2016},  
  pages = {452-454}  
}
```



Übung: Zitieren

1. Erstellen Sie eine Datei namens **Literatur.bib** im Verzeichnis Ihrer Rmd-Datei.
2. Kopieren Sie einen Zitationseintrag im Bib-Format (s. vorherige Seite) in Ihre Literatur-Datei (oder nehmen Sie [diese](#)).
3. Fügen Sie den Namen ihrer Literaturdatei in den YAML-Header ein mit dem Parameter **bibliography: <name.bib>**.

Wenn Sie keinen Pfad (zu Ihrer Literaturdatei) angeben, so muss diese im gleichen Verzeichnis wie die Rmd-Datei liegen.

```
---
```

blablabla

bibliography: Literatur.bib

blablabla

```
---
```

[Hier](#) finden Sie einen Lösungsvorschlag.

Zitationsstil wählen

Böse Zungen behaupten, es gäbe mehr Zitationsstile als Journals ...

Eine gängige Variante zur Definition von Zitationsstilen ist das **CSL**-Format.

Um einen Zitationsstil zur Formatierung Ihrer Zitationen anzuwenden, müssen Sie:

- Den Zitationsstil als CSL-Datei herunterladen (z.B. von [hier](#) oder von [hier, schon als APA6](#))
- Die CSL-Datei in Ihr Verzeichnis stellen (dort, wo sich auch die Rmd-Datei befindet)
- Im YAML-Header mit dem Parameter **csl: <csl-datei.csl>** auf die Datei verweisen.

Tabellen 1

Es gibt viele Optionen für Tabellen in Markdown via R, z.B. `knitr::kable()`

```
knitr::kable(head(iris), format = "html")
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

Tabellen 2

```
DT::datatable(  
  head(iris, 5),  
  fillContainer = FALSE, options = list(pageLength = 8)  
)
```

| Show 8 entries | | | | | Search: <input type="text"/> |
|----------------|--------------|-------------|--------------|-------------|------------------------------|
| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 5 | 3.6 | 1.4 | 0.2 | setosa |

Showing 1 to 5 of 5 entries

Previous 1 Next

Referenzieren von Abschnitten

```
# Wichtiges Kapitel {#wichtig}
```

blablabla

```
# Unwichtiges Kapitel
```

Man lese im Kapitel \@ref(wichtig).

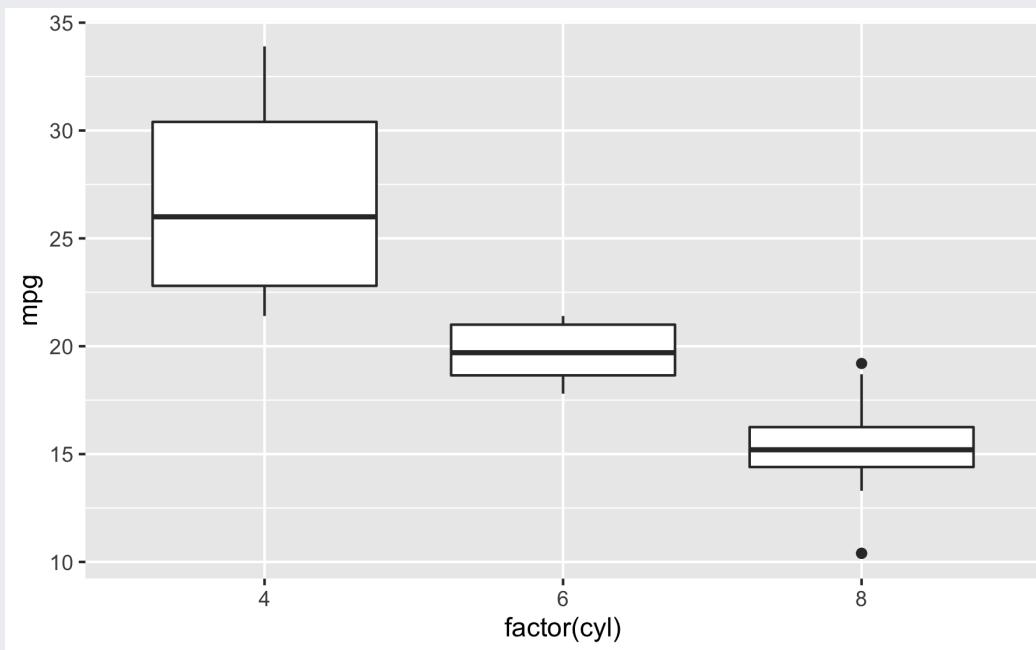
Das Referenzieren von Kapiteln ist [hier](#) nachzulesen.

! Um Referenzier-Funktionen der Art `@ref(id)` nutzen zu können, passen Sie bitte Ihren YAML-Header wie folgt an:

```
blablabla
output:
  bookdown::pdf_document2
    <weitere-output-angaben>
```

blablabla

Referenzieren von Abbildungen



Ein Beispiel

Wie in Abbildung \@ref(fig:demo-plot) ersichtlich ...

Referenzieren von Tabellen

```
mtcars %>%  
  head(3) %>%  
  knitr::kable(caption = "Tabellen-Beschriftung", format = "html")
```

Tabellen-Beschriftung

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|---------------|------|-----|------|-----|------|-------|-------|----|----|------|------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |

Wie in Tabelle \@ref(tab:demo-tab) ersichtlich ...

🏋️ Übung: Referenzieren

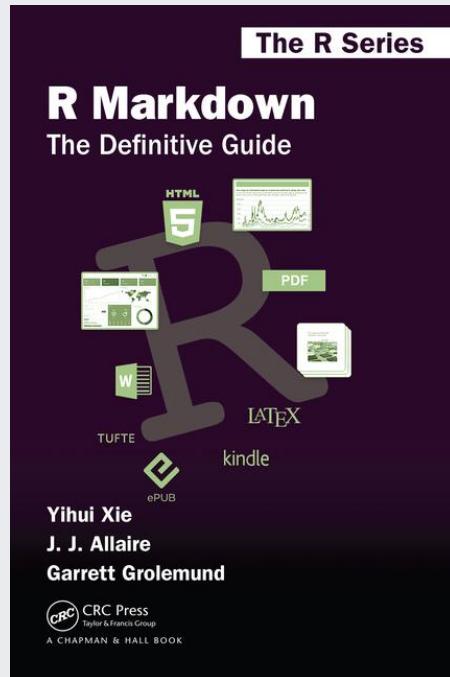
Erstellen Sie eine Dokument, in dem Sie (Kapitel, Abbildungen und/oder Tabellen) referenzieren!
Anstelle von `pdf_document2` ist auch `html_document2` möglich.

Hier finden Sie einen Lösungsvorschlag.

Weitere Formatierungsoptionen

- Im Buch [RMarkdown - The Definitive Guide](#) finden sich (in Kapitel 3.1) weitere Hinweise zum Formatieren von HTML-Dokumenten mit RMarkdown.
- Dieses [Cheatsheet "Rmarkdown 2.0"](#) gibt einen guten Überblick über Optionen mit RMarkdown.
- RStudio bietet einige gute Tutorials, z.B. [hier](#).

Buch zum Einstieg



Dieses ist auch gut



Gliederung

1. Reproduzierbarkeit

2. Markdown

3. **Git**

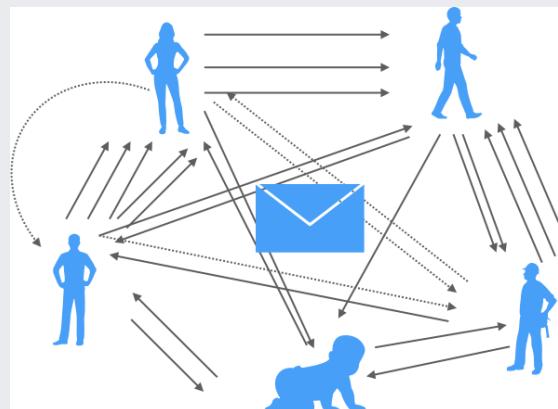
4. Fazit

Jungwissenschaftler Sebastian S. -- Ein Drama in 3 Akten (1/3)



Jung- Forscher
Sebastian S.

- Unser Jungwissenschaftler schreibt als brave Fleißbiene 🐝 einen Entwurf für einen Fachartikel 📜.
- Den Text (als Word) schickt er an seine Betreuer 🧑 (und weitere wichtige Menschen 🙏👩‍🔬🦸‍♂️💻, die auf dem Paper als Coautoren erscheinen möchten). Insgesamt an fünf Menschen.

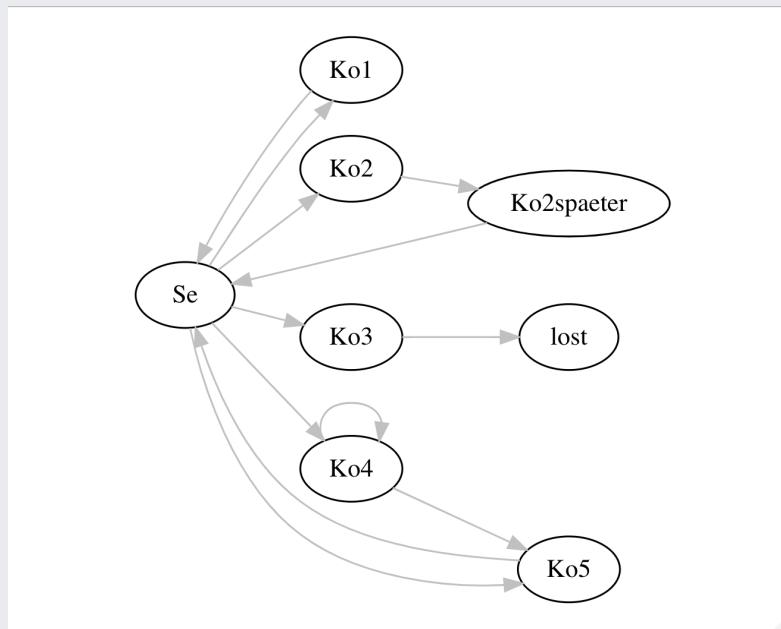


2. Akt



Jung- Forscher
Sebastian S.

- Die wichtigen Menschen tun, was Sie immer tun, und manche schicken irgendwann irgend etwas an den Jungwissenschaftler zurück.
- Der Jungwissenschaftler versucht, die fünf Beiträge mit seinem Entwurf zusammenzufassen.



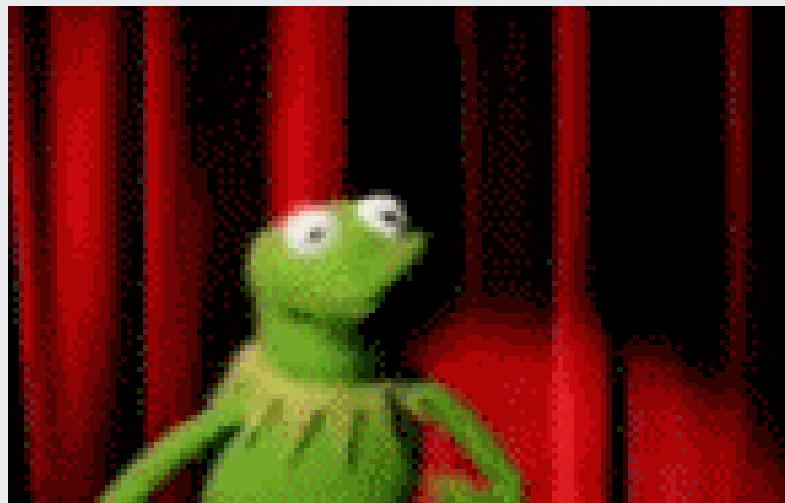
3. Akt



- Beim Jungwissenschaftler ist eine nervöse Selbstauflösung zu beobachten.

Jung- Forscher
Sebastian S.

läuft AMOK

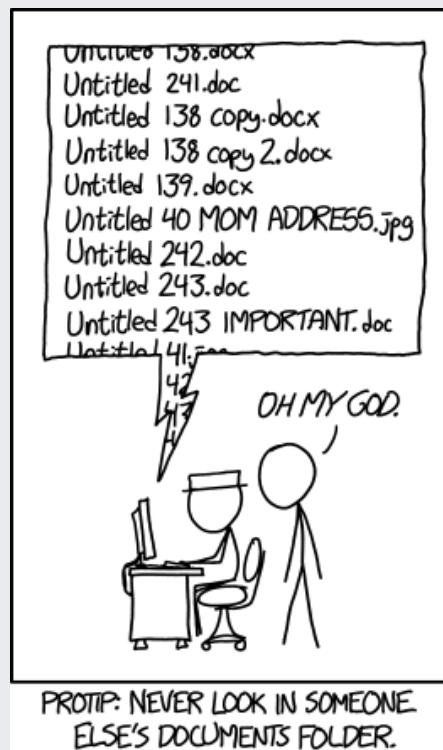


Wie können wir den Jungwissenschaftler glücklich machen?

Desiderata der Änderungsnachverfolgung beim Texte schreiben

| Nr | Desideratum | Word | Git |
|----|--|------|------|
| 1 | Übersichtlichkeit | Nein | Ja |
| 2 | Exakte Kontrolle | Nein | Ja |
| 3 | Schutz vor versehentlichem Überschreiben | Nein | Ja |
| 4 | Mehrpersonentoleranz (Kollaboration) | Nein | Ja |
| 5 | Varianten | Nein | Ja |
| 6 | Robustheit bei großen Dokumenten | Nein | Ja |
| 7 | Technische Ausfallsicherheit | Nein | Ja |
| 8 | Produktunabhängigkeit | Nein | Ja |
| 9 | Flache Lernkurve | Ja | Nein |

Dateien versionieren -- kann helfen



Wäre es nicht schön ... 1/2

Wenn man Änderungen am Text in zusammengehörigen Bündeln schön übersichtlich sehen könnte?

The screenshot shows a list of commits on September 5, 2019, from a user named sebastiansauer. The commits are grouped under the heading "Commits on Sep 5, 2019". Each commit includes the author's name, the message, the timestamp, a "Verified" badge, a copy icon, the commit hash, and a diff icon. The commits are:

- Create Antwort.txt (sebastiansauer committed 3 minutes ago)
- Wichtiges Update (sebastiansauer committed 3 minutes ago)
- Es gibt ein Problem. (sebastiansauer committed 5 minutes ago)
- Frage statt Antwort (sebastiansauer committed 15 minutes ago)
- Create testFile.txt (sebastiansauer committed 15 minutes ago)

At the bottom of the list are "Newer" and "Older" buttons.

Betrachten Sie dieses Beispiel-Repo:

https://github.com/sebastiansauer/test_ses

Wäre es nicht schön ... 2/2

Wenn man dann für jedes "Änderungsbündel" die Änderungen schön übersichtlich sehen könnte?

The screenshot shows a GitHub commit page for a file named 'testFile.txt'. The commit message is 'Wichtiges Update' and it was made by 'sebastiansauer' 4 minutes ago. The commit hash is '76bf7115864b9c558e0bd0539d2ec3f67f5b2e39'. The diff view shows the following changes:

| Line | Old Content | New Content |
|------|------------------------------|--|
| 1 | -Hallo Welt | +Hallo Welten |
| 2 | -Hallo Alle | |
| 3 | Was ist die Frage? | Was ist die Frage? |
| 4 | | |
| 5 | Die Frage hab ich vergessen. | Die Frage hab ich vergessen.
+Macht aber auch nichts. |

Below the diff view, there is a comment section for the commit '76bf711'. It includes a 'Write' button, a preview area, rich text editing tools, and a text input field with placeholder 'Leave a comment'. A note says 'Attach files by dragging & dropping, selecting or pasting them.' At the bottom is a green 'Comment on this commit' button.

Natürlich sollte Autor und Zeitpunkt der Änderung auch dokumentiert sein.

Je größer das Projekt,

... desto wichtiger ist ein gutes Werkzeug zur Änderungskontrolle

The screenshot shows a GitHub commit page for a repository named 'Typos'. The commit was made by 'Treppmann' on 14 Mar. It has 1 parent commit (f18f059) and a commit hash of 69a4f4fdf1d567edce20d6f26a833ee8fdc6ad20. The commit message indicates 7 changed files with 98 additions and 98 deletions. The changes are listed as follows:

- > 92 Inhalte/EDA.Rmd
- > 16 Inhalte/Einführung-R.Rmd
- > 68 Inhalte/Grundlagen-QuantDat.Rmd
- > 14 Inhalte/Grundlagen-Wissenschaft.Rmd
- > 2 Inhalte/Organisatorisches/LizenzVersion.Rmd
- > 2 Inhalte/Organisatorisches/VorbereitungMS.Rmd
- > 2 Inhalte/useR/useR-pre.Rmd

Below the file changes, there is a comment section for the commit 69a4f4d. A user profile picture is shown next to the 'Write' button. The comment area contains a text input field with placeholder text 'Leave a comment' and a note 'Attach files by dragging & dropping, selecting or pasting them.' A green button at the bottom right says 'Comment on this commit'.

Ein "Änderungsbündel" wird in Git als *Commit* bezeichnet. Ein Git-Ordner mit allem Drum und Dran wird als *Repo(sitorium)* bezeichnet.

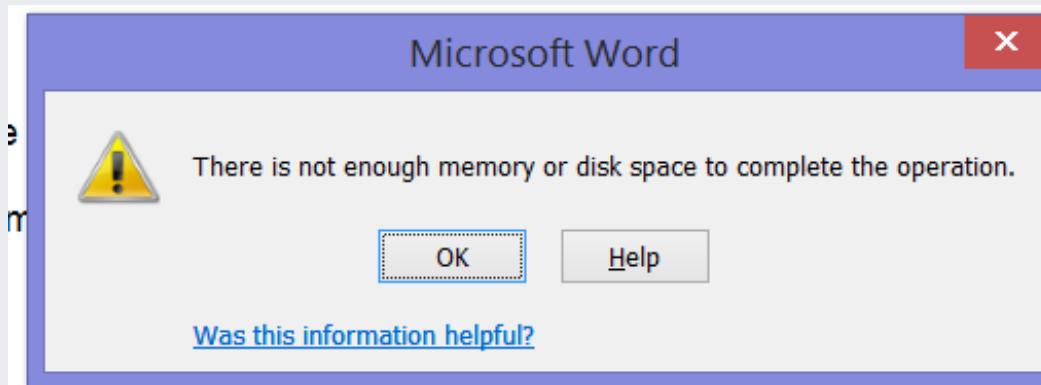
Beispielprojekt: Statistik-Vorlesungen an der FOM

The screenshot shows a GitHub repository page for 'luebby/Vorlesungen'. The repository is private, has 8 issues, 6 projects, and 7 contributors. It contains 1,836 commits, 6 branches, and 14 releases. The latest commit was made yesterday by NMarkgraf. The commit history lists various updates and bug fixes, such as fixing errors in the Formelsammlung and correcting memory management in R. The repository also includes files like 'Inhalte.Rmd', 'MGdWI-Tools/Formelsammlung.Rmd', and 'Uebungsklausuren.Rmd'.

| Commit | Message | Time Ago |
|--------------------------------|--|---------------|
| Inhalte.Rmd | ohne :: und ohne Wiederholung ist cooler. | 6 days ago |
| MGdWI-Tools/Formelsammlung.Rmd | Oops! - Dummer Fehler in der Formelsammlung korrigiert. | 9 months ago |
| PDF.Rmd | Vorschlag - Norman | 2 months ago |
| Schleifen.Rmd | Bugfixe ... Das Speichermanagement in R ist HORROR! | 2 years ago |
| Uebungsklausuren.Rmd | Umstellung ggformula. | 5 months ago |
| Vorlagen.Rmd | Oops! | 11 months ago |
| Vorlesungstermine.Rmd | Rmd für Termine als Vorlage | 6 months ago |
| backup.Rmd | Lösungen zu: | 4 months ago |
| bin | Update pandoc-filter verbessert und getestet. Dabei die aktuelle gefi... | 2 months ago |
| datasets | Überarbeitung WS 2019 | 21 days ago |
| defaults | - NoCache eingebaut! | 2 years ago |
| images | neue Abbildungen | 24 days ago |
| pandoc-filter | Merge branch 'master' of https://github.com/luebby/Vorlesungen | 7 days ago |
| prelude | Umstellung von "makeDozi/makeStudi/makeLsgSkript" auf **eine** Funkti... | 2 days ago |
| private | Updates pandoc-filter | 16 days ago |
| .gitignore | update gitignore | last month |
| Anhang-Datenhandling.Rmd | für die MSBs | last month |

196 Dokumente in 63 Ordnern, 825 Mb

Solch große Projekte sind nicht gut in Word aufgehoben



Dafür ist Word nicht gemacht.

Auszug aus den Commits im Repo Vorlesungen

Quiz mit Word

The screenshot shows a Microsoft Word document with several red and green tracked changes. A vertical timeline on the left indicates the sequence of changes. On the right, a detailed revision history is listed.

Stichprobe von $N=404$ bis $N=446$ für eine gewünschte Schätzgenauigkeit von $p=1$ bei einem 95 %-Konfidenzintervall. Insgesamt liegt eine Stichprobe von $N=416$ gültigen Datensätzen vor. (222 Frauen, 194 Männer). Es wurden nur Versuchspersonen entfernt, die die Studie nicht abgeschlossen hatten; sonst wurden keine Daten entfernt. Die Alterspanne lag zwischen 19 und 63 Jahren ($M=34.9$ Jahre, $SD=10.7$). Die Befragten waren überwiegend Angestellte ohne Leitungsfunktion ($n=298$), leitende Angestellte ($n=53$) oder Beamte ($n=34$) aus unterschiedlichen Branchen.[¶]

Operationalisierung[¶]

Autor hat formatiert: Schriftart: Kursiv
Autor hat formatiert: Schriftart: Kursiv
Autor hat gelöscht: wurde die Befragung
Autor hat gelöscht: Personen
Autor hat gelöscht: weiblich
Autor hat gelöscht: männlich
Autor hat gelöscht: die

1. Wer hat "liegt eine Stichprobe geändert"?
2. Was stand an dieser Stelle davor?
3. Handelt es sich bei der Änderung um eine Ergänzung oder eine Modifikation oder eine Löschung?
4. Ist "Autor" ein Name oder der Urheber des Dokuments?
5. Wann wurden die Änderungen vorgenommen?
6. Wie kann man den Verlauf der Änderungen sehen?

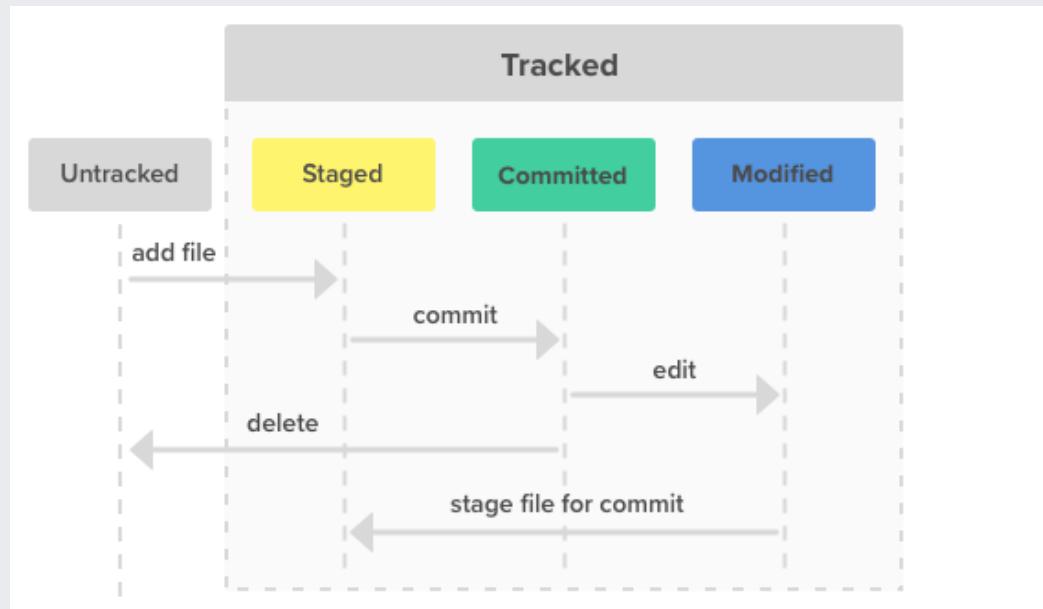
Git-Workflow 1

1. Git starten (Ordner für Git initialisieren)
2. Änderungen vornehmen (ggf. an mehreren Dateien)
3. Dateien auswählen, deren Änderungen dokumentiert werden sollen ("adden" oder "stagen")
4. Änderungen einstellen ("committen")
5. Änderungen zum Server replizieren ("pushen")
6. Ggf. Änderungen prüfen und kommentieren

Komponenten eines Git-Projekts

- *Repository*: Der Container, der alle Dateien und alle Änderungen eines Projekt enthält.
- *Working Tree*: Dateisystem der Repos in aktueller (oder frühere) Version
- *Staging Area (Index)*: Dokumentiert Dateien mit Änderungen, die noch nicht eingestellt sind.
- *Head*: Aktuellste Version des Projekts (neuester Commit) im aktuellen Branch
- *Branch*: Variante des Projekts

Git-Workflow mit Fachbegriffen



Aber lohnt sich Git für mich?

A solo data analyst, working on a single computer, will benefit from adopting version control. But not nearly enough to justify the pain of installation and workflow upheaval. There are much easier ways to get versioned back ups of your files, if that's all you're worried about.

In my opinion, for new users, the pros of Git only outweigh the cons when you factor in the overhead of communicating and collaborating with other people.

<https://happygitwithr.com/big-picture.html>

Github



- Hosting-Anbieter für Git-Repos
- 2018 von Microsoft gekauft (7.5 Mrd. USD)
- Neben reinem Git werden noch ergänzende Funktionen wie *Issues* (Art Ticketsystem) etc. angeboten
- Größte Sammlung von Source Code in der Welt

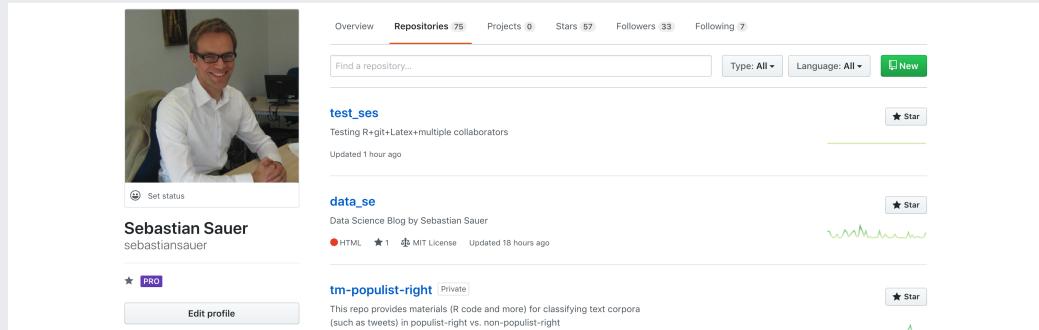
<https://en.wikipedia.org/wiki/GitHub>

Git* installieren

1. Account (kostenlos) bei [Github](#) anlegen
2. RStudio herunterladen bzw. aktualisieren
3. Git herunterladen/installieren [Hinweise](#)
4. Bei Git [anmelden](#)
5. Git-Editor (z.B. [Github Desktop](#)) installieren

Ein erstes Repo anlegen 1/2

1. Loggen Sie sich bei Github ein.
2. Klicken Sie bei "Repositories" auf New (grüner Knopf).



Ein erstes Repo anlegen 2/2

1. Füllen Sie die Maske aus.
2. Grünen Knopf drücken ("Create Repository"). Fertig!

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner **Repository name ***

 sebastiansauer 

Great repository names are short and memorable. Need inspiration? How about [psychic-bassoon](#)?

Description (optional)

blablabla

 **Public**
Anyone can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

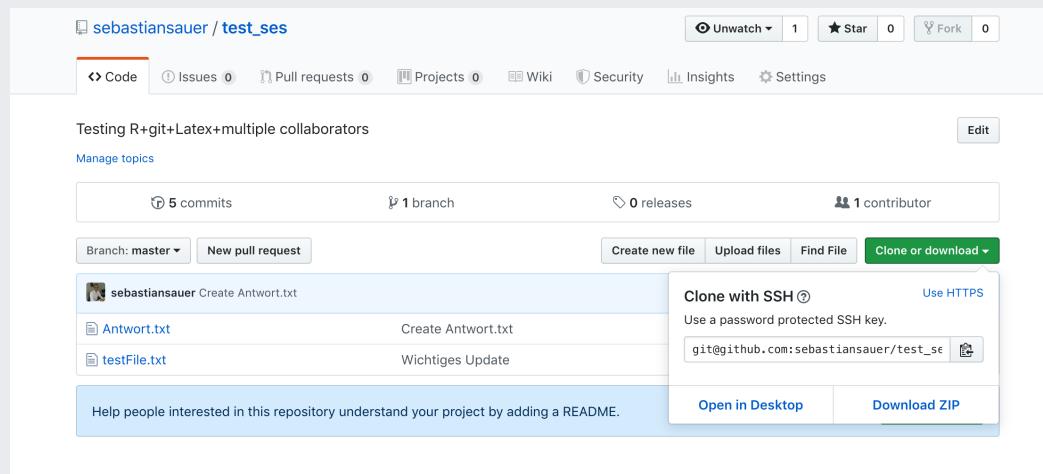
Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None** 

Create repository

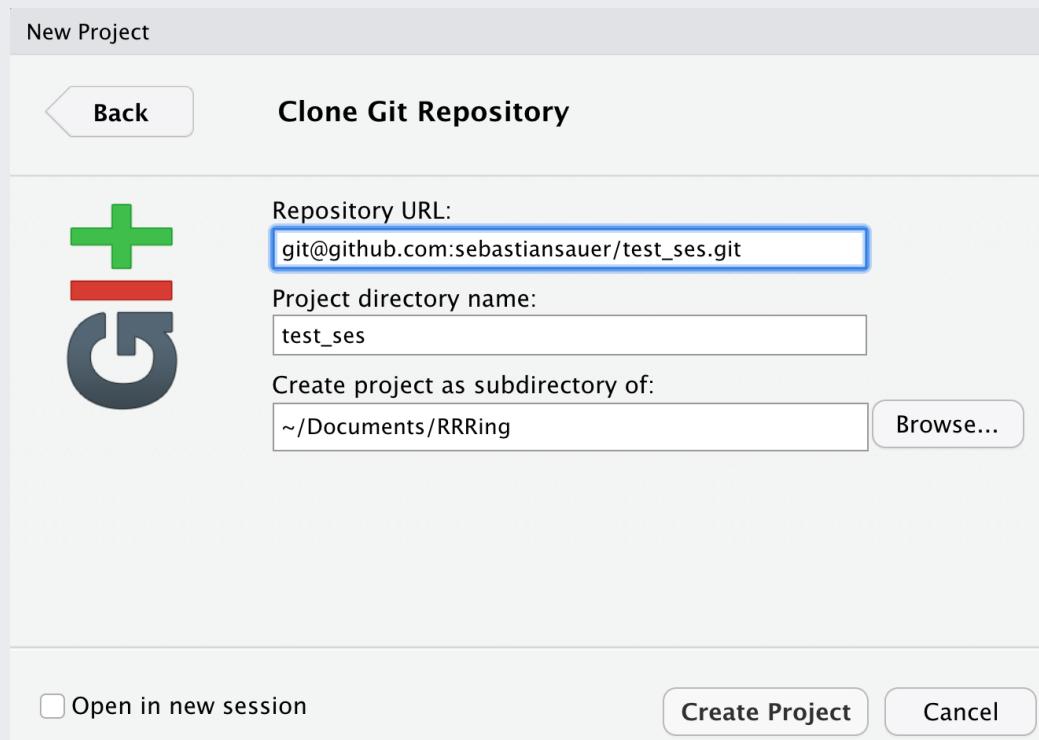
Das Repo "clonen" 1/2

1. Gehen Sie in das Repo.
2. Klicken Sie auf den grünen Knopf ("Clone or Download"), wählen Sie "Clone".
3. Kopieren Sie den Git-Link heraus, z.B. "git@github.com:sebastiansauer/test_ses.git".



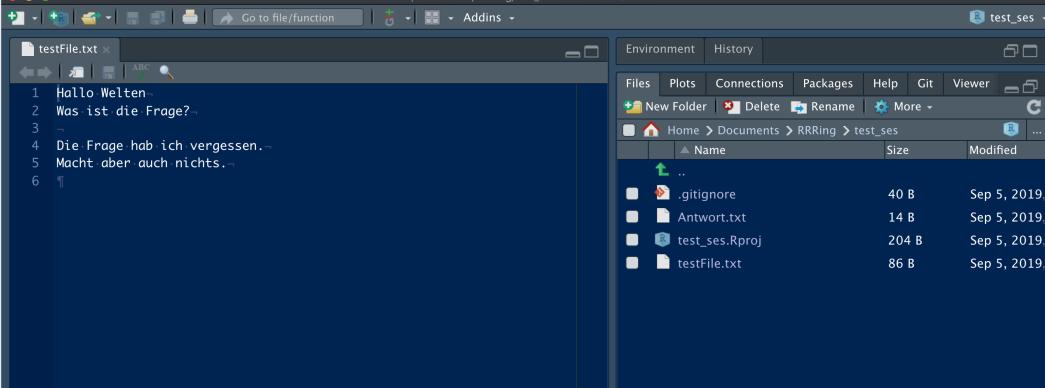
Das Repo "clonen" 2/2

1. Öffnen Sie RStudio
2. Legen Sie ein Git-Projekt an: File > New Project ... > Version Control > Git.
3. Kopieren Sie bei "Repository Url" die Git-Url Ihres Repos ein.
4. Verzeichnisname etc. können Sie nach Belieben wählen.



🏋️ Übung: Legen Sie sich ein Github-Repo in RStudio an

Jetzt haben Sie Ihr Repo in RStudio verfügbar.



The screenshot shows the RStudio interface with a GitHub repository named 'test_ses'. On the left, there's a code editor window titled 'testFile.txt' containing the following text:

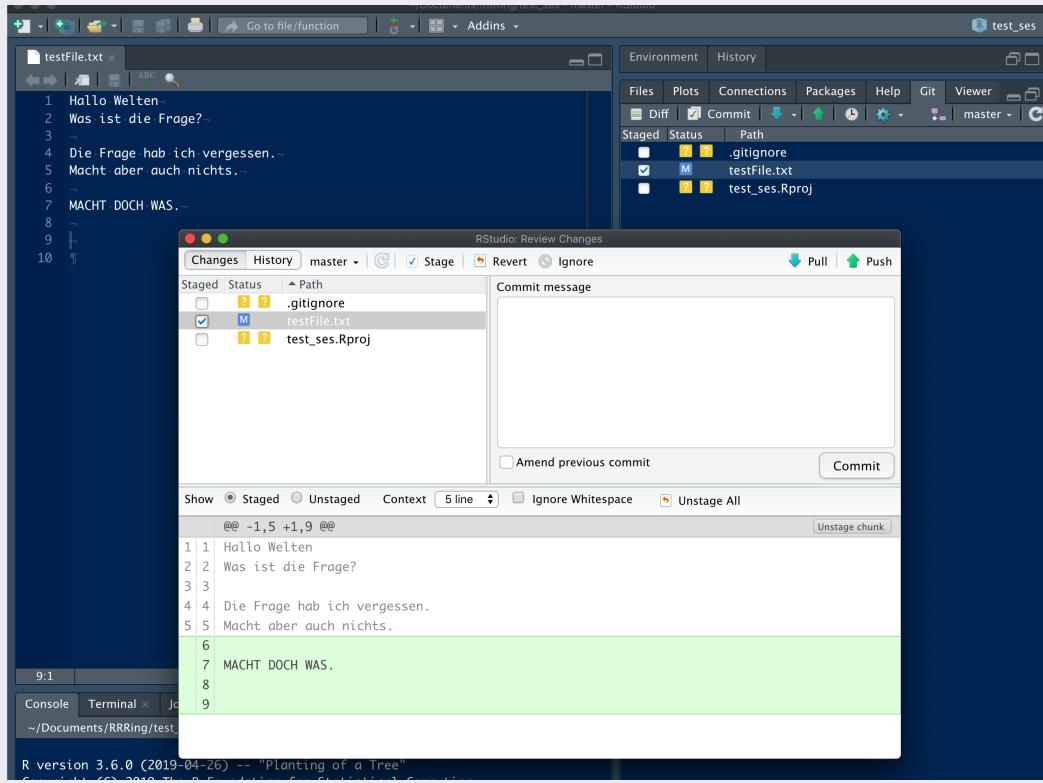
```
1 Hallo Welten.  
2 Was ist die Frage?  
3 ~  
4 Die Frage hab ich vergessen.  
5 Macht aber auch nichts..  
6 |
```

On the right, the 'Files' tab of the GitHub interface is visible, showing the contents of the repository:

| Name | Size | Modified |
|----------------|-------|--------------|
| .. | | |
| .gitignore | 40 B | Sep 5, 2019, |
| Antwort.txt | 14 B | Sep 5, 2019, |
| test_ses.Rproj | 204 B | Sep 5, 2019, |
| testFile.txt | 86 B | Sep 5, 2019, |

Vertiefung: Passwörter-Cache mit Git/Github

🏋️ Übung: Stellen Sie Änderungen in Ihr Repo ein.



Mögliche Probleme

Das Buch von Jenny Bryant *Happy Git with R* gibt eine gute Einführung in das Thema.

Dort finden Sich auch Hinweise bei Problemen.



Übung: Arbeiten Sie im Team

1. Fügen Sie sich in Ihrem Github-Repo gegenseitig als Mitarbeiter hinzu:
Settings > Collaborators.
2. Sie müssen dazu die entsprechenden Nutzernamen wissen.
3. Ändern Sie nun etwas an einem Repo, zu dem Sie eingeladen wurden.

The screenshot shows a GitHub repository settings page for 'sebastiansauer / test_ses'. The 'Collaborators' tab is selected. On the left, there's a sidebar with options like Options, Collaborators (which is selected), Branches, Webhooks, Notifications, Integrations & services, Deploy keys, Moderation, and Interaction limits. The main area displays a search bar with 'lue' typed in, a list of suggestions including 'luebby Karsten Luebke', and a button labeled 'Add collaborator'.

Einige Git-Befehle für die Kommandozeile (Terminal)

- **git init**: Legt in einem Ordner die Git-Infrastruktur an.
- **git status**: Gibt einen Überblick über den Status eines Repos.
- **git add ..**: Fügt alle Dateien dem Index zu (staged die Dateien).
- **git commit -m "meine nachricht"**: Committed die gestageden Dateien und fügt eine (hoffentlich aussagekräftige) Nachricht hinzu
- **git pull**: zieht das Repo vom zentralen Server und mergt es in das lokale Repo.
- **git push**: Lädt das lokale Repo zum zentraler Server hoch.
- **git stash**: Speichert die aktuellen Änderungen (sofern noch nicht committed) temporär.
- **git blame**: Zeigt, wer als letztes eine bestimmte Zeile einer Datei verändert hat.

frühere Zustände betrachten - git checkout <id>

Mit

`git checkout a1e8fb5`

versetzt man den Working Tree in dem Zustand des Commits mit der ID a1e8fb5.

! Man kann jetzt keine Änderungen vornehmen (*detached head*), sondern nur alte Zustände betrachten¹.

Mit `git checkout master` kommt man wieder zum aktuellen Stand auf dem Branch *master*.

1: Aber man kann von diesem Zustand aus einen neuen Branch öffnen mit z.B `git checkout -b my_new_branch`.

Frühere Zustände wiederherstellen 3/3 -

git checkout <commit-hash>

<file>

```
git checkout git checkout bdc88fdf18bce554046b4188e0a8aee0b5f1b0a Tv
```

Damit wird eine Datei/ein Ordner aus dem Commit <commit-hash> im Working Directory wiederhergestellt.

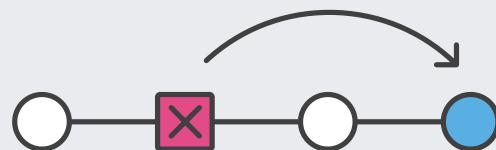
Mit

```
git reset HEAD~2 foo.R
```

Wird die Datei `foo.R` aus dem 2. letzten Commit wiederhergestellt.

<https://www.atlassian.com/git/tutorials/resetting-checking-out-and-reverting>

Frühere Zustände wiederherstellen 2/3 - git revert



Einen früheren Commit wiederherstellen:

```
git revert HEAD
```

Damit wird der Zustand

Anstelle von **HEAD** kann man auch eine Commit-ID angeben.

<https://www.atlassian.com/git/tutorials/undoing-changes/git-revert>

Frühere Zustände wiederherstellen 3/3 - git reset

💣 So widerrufen Sie Änderungen *unwiderruflich*:

```
git reset --hard HEAD~
```

Working Tree und HEAD werden auf den Vorfahren des Heads zurückgesetzt. Der Index ist leer (keine Dateien auf der Staging Area).

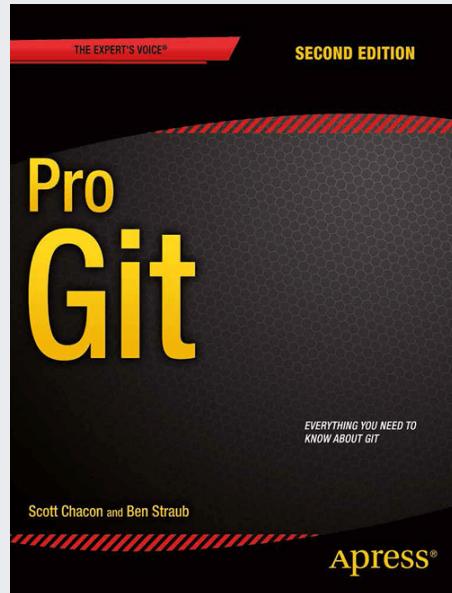
So löscht man Änderungen im Working Directory (d.h. nicht committed)

```
git reset --hard
```

git reset zum Unstagen aller Dateien

Buch *Pro Git* zum Einstieg und Vertiefen

freie Online-Version



Wir brauchen eine Pause.



Gliederung

1. Reproduzierbarkeit

2. Markdown

3. Git

4. Fazit

OK, es kann auch Probleme geben



☞ StackOverflow ist der Retter in Not.

Danke für die Aufmerksamkeit

Sebastian Sauer

 [sebastiansauer](#)

 <https://data-se.netlify.com/>

 ssauer@posteo.de

 [Sebastian Sauer](#)

 Get slides here:

<http://data-se.netlify.com/slides/WS-MD-Git-2019/WS-MD-Git-2019.html>

 Get source code here:

<http://data-se.netlify.com/slides/WS-MD-Git-2019/WS-MD-Git-2019.Rmd>

CC-BY

Literatur

Baker, M. (2016). "1,500 Scientists Lift the Lid on Reproducibility". En. In: *Nature* 533.7604, pp. 452-454. ISSN: 0028-0836, 1476-4687. DOI: [10.1038/533452a](https://doi.org/10.1038/533452a).

Goodman, S. N, D. Fanelli, and J. P. A. Ioannidis (2016). "What Does Research Reproducibility Mean?" In: *Science Translational Medicine* 8.341, pp. 341ps12-341ps12. ISSN: 1946-6234, 1946-6242. DOI: [10.1126/scitranslmed.aaf5027](https://doi.org/10.1126/scitranslmed.aaf5027).

McElreath, R. (2016). *Statistical Rethinking*. New York City, NY: Apple Academic Press Inc.. ISBN: 1-4822-5344-5.

Jetzt sind Sie dran



Reproducibility

- Versions of employed software as of 2019-09-11, running this OS: macOS Mojave 10.14.6.
- Built with R, R version 3.6.0 (2019-04-26), RStudio 1.2.1335, xaringan, on the shoulders of giants
- Icons are from FontAwesome, licenced under CC-BY-4 ([details](#))
- R-Packages used: assertthat_0.2.1, backports_1.1.4, bibtex_0.4.2, brew_1.0-6, broom_0.5.2, cellranger_1.1.0, cli_1.1.0, codetools_0.2-16, colorspace_1.4-1, crayon_1.3.4, crosstalk_1.0.0, curl_3.3, data.table_1.12.2, DiagrammeR_1.0.1, DiagrammeRsvg_0.1, digest_0.6.20, downloader_0.4, dplyr_0.8.3, DT_0.7, emo_0.0.0.9000, evaluate_0.14,forcats_0.4.0, generics_0.0.2, ggdendro_0.1-20, ggformula_0.9.1, ggplot2_3.2.0, ggrepel_0.8.1, ggstance_0.3.2, glue_1.3.1.9000, gridExtra_2.3, gtable_0.3.0, haven_2.1.1, here_0.1, highr_0.8, hms_0.5.0, htmltools_0.3.6, htmlwidgets_1.3, httpuv_1.5.1, httr_1.4.0, icon_0.1.0, igraph_1.2.4.1, influenceR_0.1.0, jsonlite_1.6, knitr_1.23, later_0.8.0, lattice_0.20-38, lazyeval_0.2.2, leaflet_2.0.2, lubridate_1.7.4, magrittr_1.5, MASS_7.3-51.4, Matrix_1.2-17, mime_0.7, modelr_0.1.4, mosaic_1.5.0.9001, mosaicCore_0.6.0, mosaicData_0.17.0, munsell_0.5.0, nlme_3.1-140, pillar_1.4.2, pkgconfig_2.0.2, plotly_4.9.0, plyr_1.8.4, promises_1.0.1, purrr_0.3.2, R6_2.4.0, RColorBrewer_1.1-2, Rcpp_1.0.1, readr_1.3.1, readxl_1.3.1, RefManageR_1.2.12, rgexf_0.15.3, rlang_0.4.0, rmarkdown_1.14, Rook_1.1-1, rprojroot_1.3-2, rstudioapi_0.10, rvest_0.3.4, scales_1.0.0, sessioninfo_1.1.1.9000, shiny_1.3.2, stringi_1.4.3, stringr_1.4.0, tibble_2.1.3, tidyverse_1.2.1, V8_2.3, vctrs_0.2.0, viridis_0.5.1, viridisLite_0.3.0, visNetwork_2.0.7, widgetframe_0.3.1, withr_2.1.2, xaringan_0.11, xaringanthemer_0.2.0, xfun_0.8, XML_3.98-1.20, xml2_1.2.0, xtable_1.8-4, yaml_2.2.0, zeallot_0.1.0
- Last update 2019-09-11 14:30:08