

R Schnittstellen - Reproducible Research

Jan-Philipp Kolb

8 Mai 2017

Präsentation von Daten - Reproducible Research

CRAN Taskview zu reproducible research

CRAN Task View: Reproducible Research

Maintainer: Max Kuhn

Contact: max.kuhn at pfizer.com

Version: 2015-12-03

URL: <https://CRAN.R-project.org/view=ReproducibleResearch>

The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified. Packages in R for this purpose can be split into groups for: literate programming, package reproducibility, code/data formatting tools, format converters, and object caching.

Figure 1:

Reproducible Research with R and RStudio

Christopher Gandrud

Reproducible Research with R and RStudio Second Edition

Figure 2:

- Ein Online Buch mit vielen nützlichen Tipps

A crash course in reproducible research in R

written October 14, 2016 in r, programming tips

A couple of weeks ago, I wrote a post giving you an introduction to [reproducible research](#) in Python. While the principles of reproducibility stay the same no matter the language you are using, there are some specific libraries and tools that R has that differ from Python. In this blog post, I'll fill you in on how I conduct a reproducible analysis in R and, like with Python, you'll see how straightforward it is!

Figure 3:

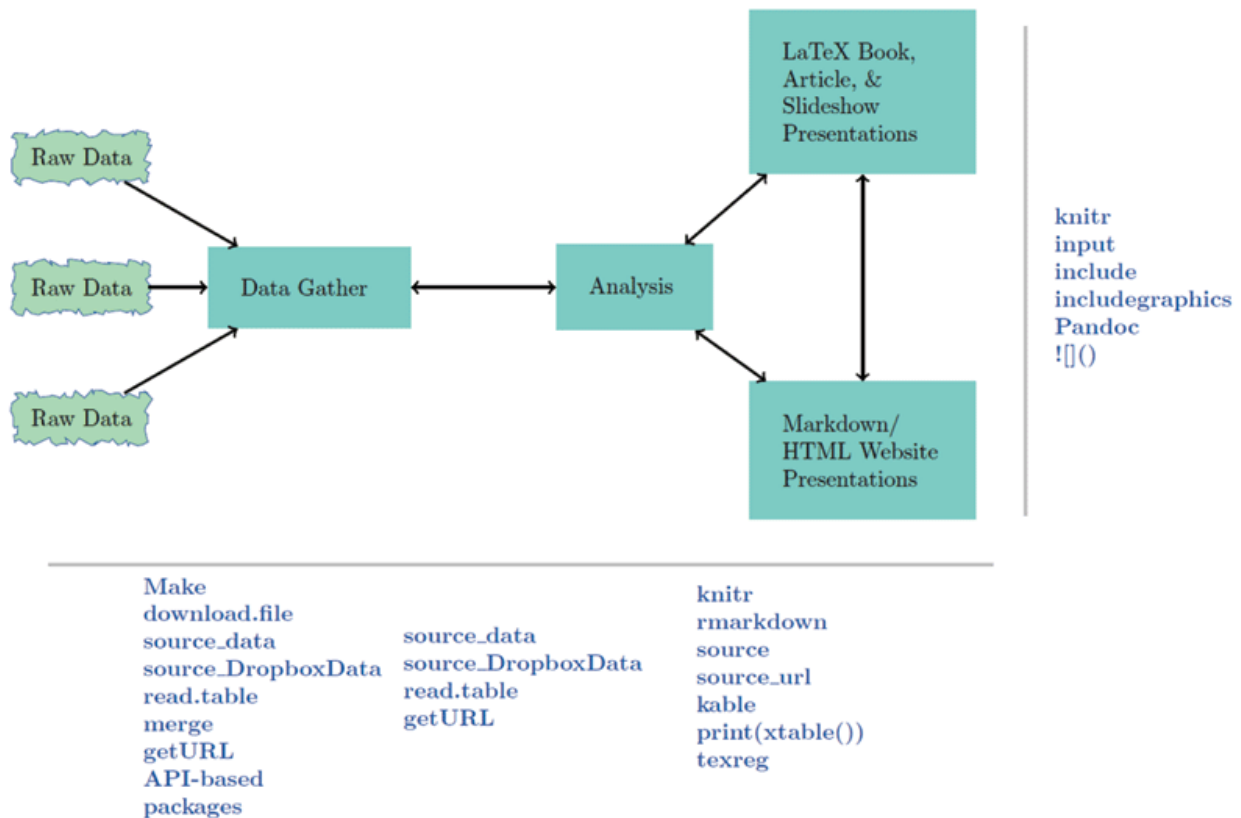


Figure 4:

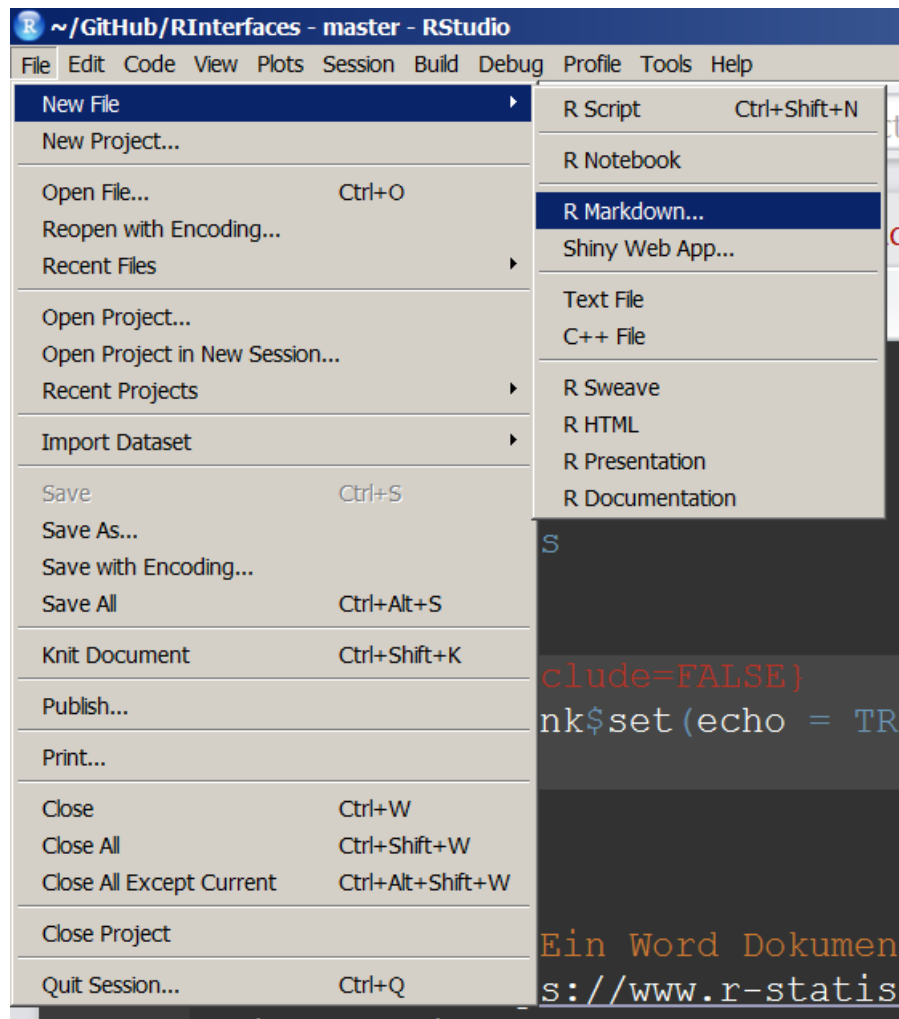


Figure 5:

Ein Crash Kurs in Reproducible Research mit R

Umsetzung in R

Word Dokumente mit R erstellen

Ein Markdown Dokument mit Rstudio erzeugen

Mein erstes mit R erzeugtes Word Dokument

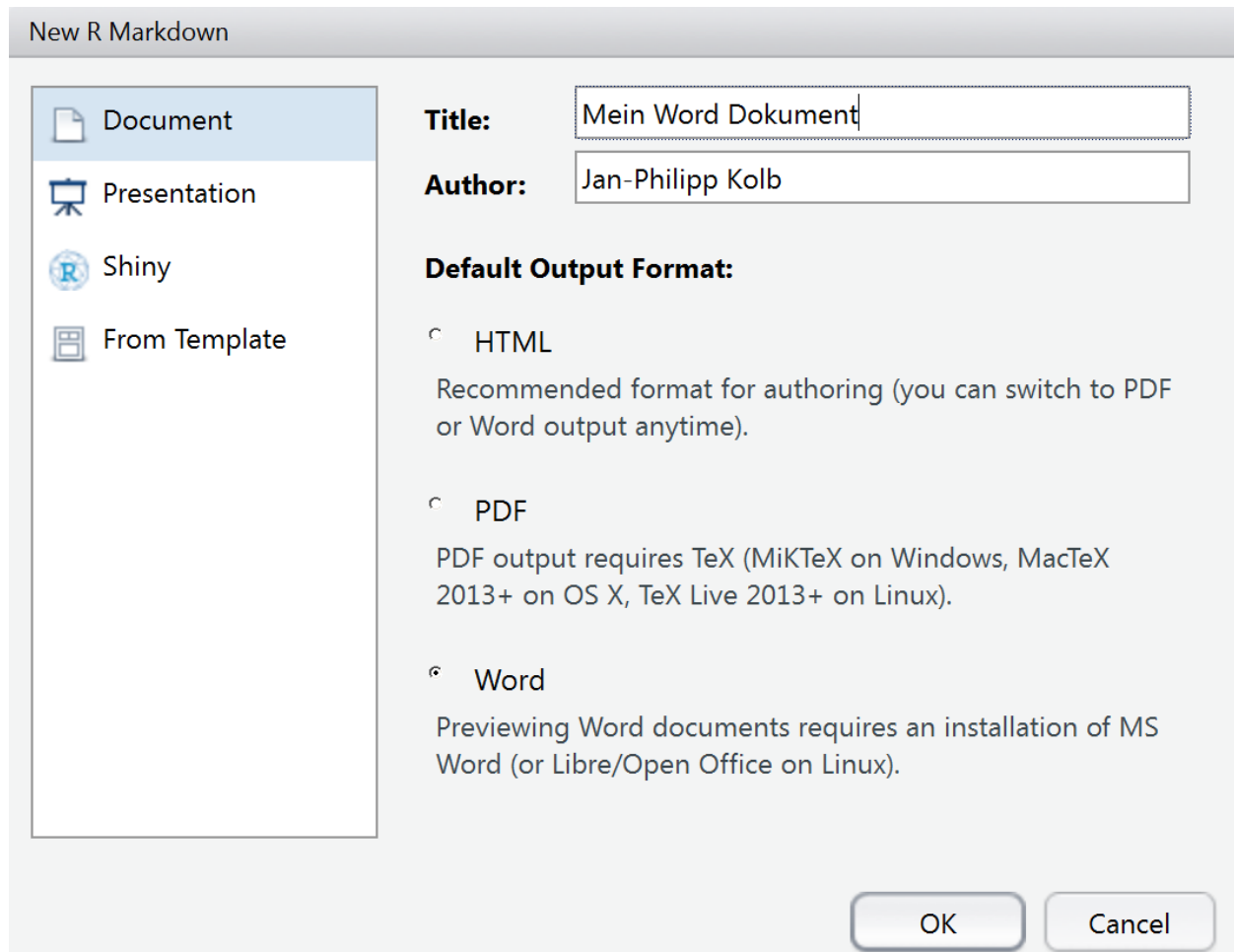


Figure 6:

Erstes Beispiel

Das Arbeiten mit Rmarkdown - erste Schritte

Markdown ist eine sehr einfache Syntax, die es Benutzern erlaubt, aus einfachen Textdateien gut gelayoutete Dokumente zu erstellen.

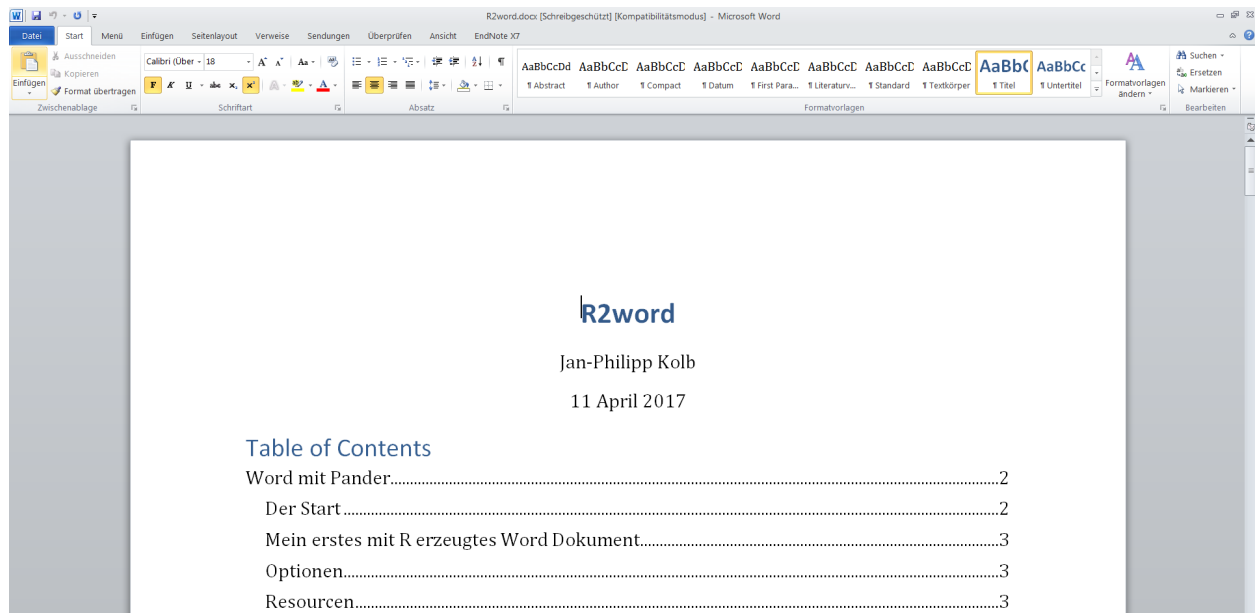


Figure 7:

****fettes Beispiel****
kursives Beispiel
 ~~durchgestrichen~~
 - Aufzählungspunkt

fettes Beispiel

kursives Beispiel

~~durchgestrichen~~

- Aufzählungspunkt

Weitere Markdown Befehle

Überschrift Ebene 3
 #### Überschrift Ebene 4
 [Meine Github Seite](https://github.com/Japhilko)

Überschrift Ebene 3

Überschrift Ebene 4

Meine Github Seite

Weitere Markdown Befehle

- So kann man Bilder einbinden:
- Man kann entweder einen Link angeben:

![BSP](http://e-scientifics.de/content/example_kinderbild.jpg)

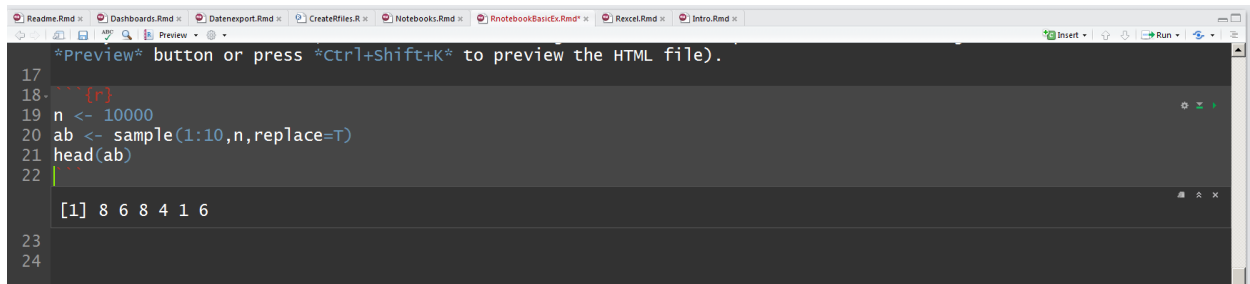
- oder einen (Unterordner) in dem das Bild liegt:

![BSP 2](figure/example.png)

- in den eckigen Klammern steht die Bildunterschrift
- alle gängigen Formate (.png, .jpeg, .gif) können so eingebunden werden
- Man kann auch noch weitere Optionen spezifizieren (Größe, Breite etc.) - dazu später mehr

Chunks - Erste Schritte

- Es lassen sich so genannte Chunks einfügen
- In diesen Chunks wird ganz normaler R-code geschrieben



```

17 *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).
18 {r}
19 n <- 10000
20 ab <- sample(1:10,n,replace=T)
21 head(ab)
22
[1] 8 6 8 4 1 6
23
24

```

Figure 8:

Button um Chunks einzufügen

- Die default Version eines Chunks ist R
- Man hat aber auch die Möglichkeit andere Programmiersprachen einzubinden

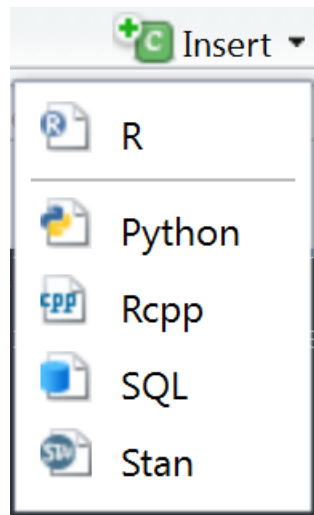


Figure 9:

```
n = 100

# Ein inline Codeblock: `r n`
```

Figure 10:

Inline Code

```
n=100
```

Ein inline Codeblock: 100

Chunk Optionen

- Man kann den Chunks Optionen mitgeben:

Argument	Beschreibung
eval	Soll Rcode evaluiert werden?
warning	Sollen Warnings angezeigt werden?
cache	Soll der Output gespeichert werden?

- Bei eval kann ein logischer Wert angegeben werden oder eine/mehrere Nummer(n)

Optionen

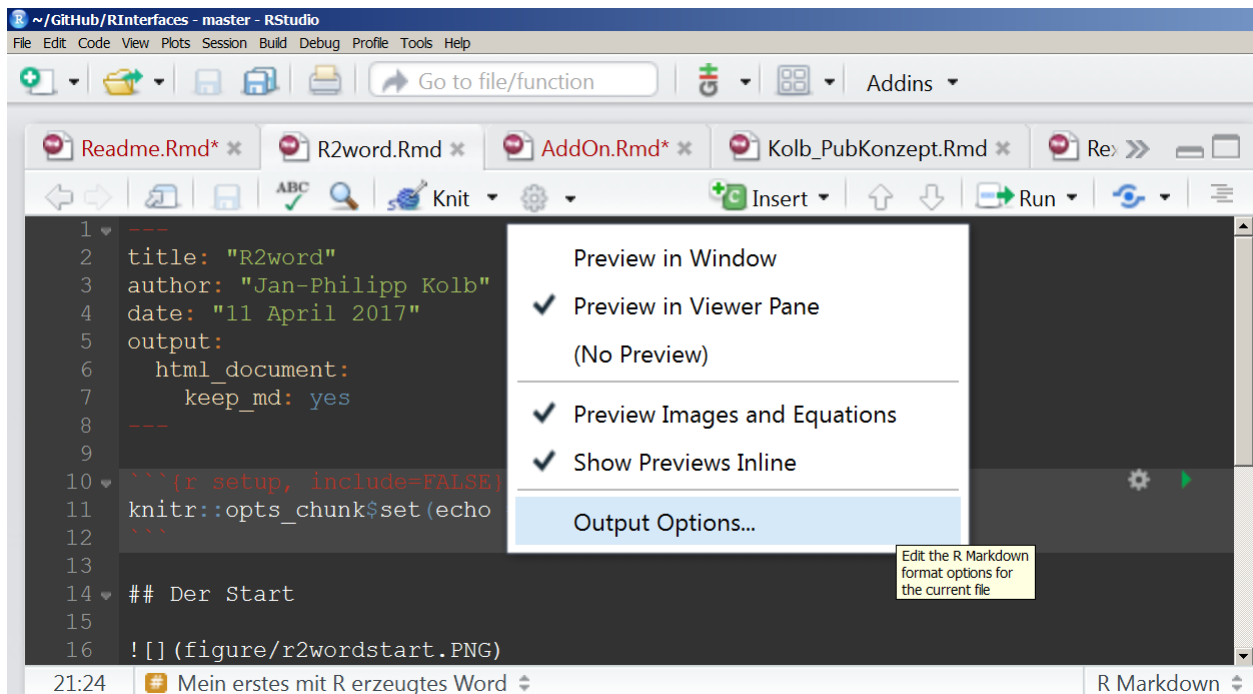


Figure 11:

Optionen für Word Output

Code Hervorhebung

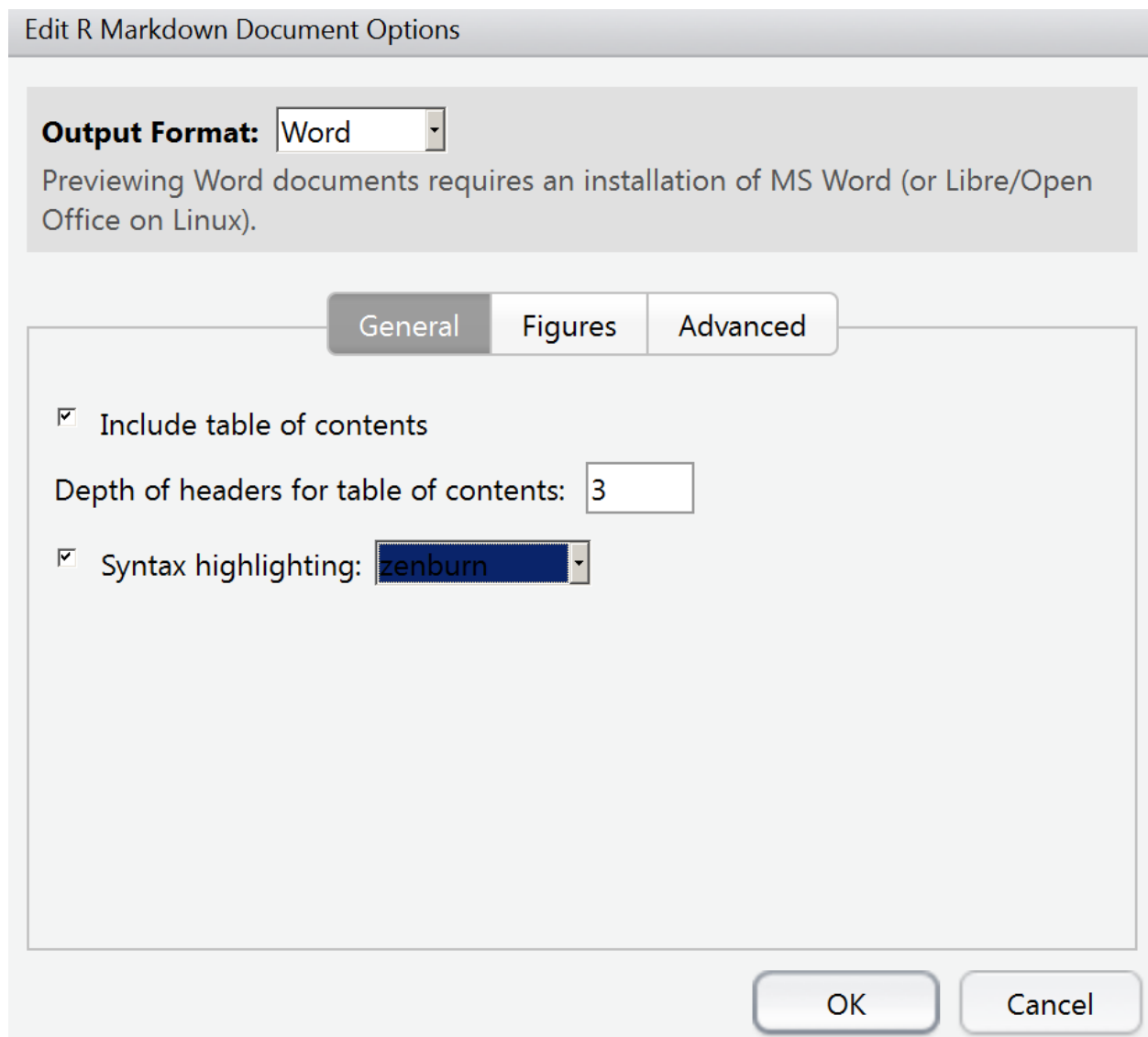


Figure 12:

```
# Beispiel für Code
ab <- sample(1:10,5,replace=T)
summary(ab)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	2.0	3.0	5.0	5.4	8.0	9.0

Figure 13:

- tango

```
# Beispiel für Code
ab <- sample(1:10,5,replace=T)
summary(ab)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      5.0     6.0     6.0     6.6     7.0     9.0
```

Figure 14:

Das Paket knitr

```
install.packages("knitr")
```

```
library("knitr")
```

- Das Paket knitr enthält zahlreiche wichtige Funktionen
- Beispiel: Befehl `kable` um Tabellen zu erzeugen

Eine Tabelle mit `kable` erzeugen

```
a <- runif(10)
b <- rnorm(10)
ab <- cbind(a,b)
kable(ab)
```

a	b
0.1672997	0.9847657
0.9587024	1.8510870
0.4651757	0.1385755
0.4557838	-0.1517748
0.7588403	0.6232395
0.5408735	1.3232444
0.7837682	0.1718908
0.4346141	1.0392511
0.2576911	-0.3837067
0.7098376	0.8976535

Vorlagen verwenden

- Formatvorlagen können verändert werden
1. Ein Word Dokument mit Rmarkdown erstellen
 2. Das Dokument in Word öffnen und Format verändern
 3. Vorlage als Referenz angeben

```

1 ---
2 title: "R2word"
3 author: "Jan-Philipp Kolb"
4 date: "11 April 2017"
5 output:
6   word_document:
7     reference_docx: RefDoc.docx
8     highlight: zenburn
9     toc: yes
10 ---

```

Figure 15:

Immer das aktuelle Datum im Kopf

date: "08 Mai, 2017"

```

1 ---
2 title: "RPostgreSQL"
3 author: "Jan-Philipp Kolb"
4 date: "`r format(Sys.time(), '%d %B, %Y')`"
5 output:
6   slidy_presentation:
7     keep_md: yes
8 ---

```

Figure 16:

Ein Schummelzettel

Resourcen

- Interview - Ein Word Dokument mit wenig Aufwand schreiben
- pander: Ein R Pandoc Wrapper
- Einführung in Markdown
- Warum TeX besser als Word ist

RPostgreSQL

Jan-Philipp Kolb

06 Mai, 2017

Figure 17:

R Markdown

Schummelzettel

Mehr auf rmarkdown.rstudio.com

rmarkdown 0.2.50 Update: 8/14

1. Arbeitsablauf

Die Auszeichnungssprache „R Markdown“ erlaubt die Erstellung von reproduzierbaren und dynamisch anpassbaren Protokollen in R. R-Quellcode und dessen Ergebnisse können in Präsentationen, PDF-Dateien, HTML-Seiten, Word-Dateien etc. eingebettet werden. Um solch ein Protokoll zu erstellen:

- i. Öffnen – Eine Datei mit .Rmd Extension öffnen oder erstellen.
- ii. Schreiben – Inhalt mittels einfach zu bedienender „R Markdown“-Syntax schreiben.
- iii. Einbetten – R-Quellcode erstellen, der ausgebaute Ergebnisse erzeugt die in das Protokoll eingebettet werden.
- iv. Rendern – Quellcode durch seine Ergebnisse ersetzen und das Protokoll in eine Präsentation, PDF-, HTML- oder Word-Datei umwandeln.

2. Datei erstellen

Zuerst wird eine Textdatei mit der Extension .Rmd erstellt oder ein RStudio Rmd-Template geöffnet.

- Zum Erstellen eines „R Markdown“ Protokolls im Menü folgendes auswählen: **Datei > Neue Datei > R Markdown...** Oder im englischen Menü: **File > New File > R Markdown...**
- Im neuen Fenster den Typ des Dokuments auswählen, das mittels .Rmd erstellt werden soll. Das entsprechende Optionfeld für die Dateiart anklicken (kann nachträglich geändert werden).
- OK klicken.

3. Markdown schreiben

Anschließend wird das Protokoll in Klartext geschrieben. Zur Formatierung wird Markdown-Syntax verwendet.

Syntax

Klartext
Zeile mit 2 Leerzeichen beendet um neuen Absatz zu beginnen.
kursiv und _kursiv_
fett und **fett**
hochgestellt: ²

Hyperlink

Überschrift 1
Überschrift 2
Überschrift 3
Überschrift 4
Überschrift 5
Überschrift 6

Strichstrich (engl. endash): --
Gedankenstrich (engl. emdash): ---
Auslassungspunkte (engl. ellipsis): ...
einseitige Formel: $\Delta x = r \cdot \Delta \theta$
Bild:

ungepreschter Strich:

> Blockzitat

* ungeordnete Liste
* Punkt 2
 + Unterpunkt 1
 + Unterpunkt 2

1. geordnete Liste
2. Punkt 2
 + Unterpunkt 1
 + Unterpunkt 2

Tabellenüberschrift	Zweite Überschrift
Tabellenelement	Element 2
Element 3	Element 4

Resultat

Klartext
Zeile mit 2 Leerzeichen beendet um neuen Absatz zu beginnen.
Ausdruck *Ausdruck*
fett und **fett**
^{hochgestellt}

[Hyperlink](http://www.rstudio.com)

Überschrift 1
Überschrift 2
Überschrift 3
Überschrift 4
Überschrift 5
Überschrift 6

Strichstrich (engl. endash): --
Gedankenstrich (engl. emdash): ---
Auslassungspunkte (engl. ellipsis): ...
einseitige Formel: $\Delta x = r \cdot \Delta \theta$
Bild:

ungepreschter Strich:

> Blockzitat

* ungeordnete Liste
* Punkt 2
 + Unterpunkt 1
 + Unterpunkt 2

1. geordnete Liste
2. Punkt 2
 + Unterpunkt 1
 + Unterpunkt 2

Tabellenüberschrift	Zweite Überschrift
Tabellenelement	Element 2
Element 3	Element 4

4. Ausgabeformat auswählen

Eine YAML Kopfzeile wird erstellt. Sie verdeutlicht, welchen Dokumenttyp die R Markdown Datei erzeugen soll.

YAML
Eine YAML Kopfzeile besteht aus Schlüssel-Wert-Paaren am Beginn der Datei. Drei Bindestriche (---) bilden Anfang und Ende dieser Kopfzeile.

```
---
title: "Überschrift"
author: "Schummelzettel"
output: html_document
---
```

Wie ist der Anfang des Protokolls. Der obige Text enthält die Metadaten der YAML Kopfzeile.

Die RStudio-Vorlage schneidet automatisch die entsprechende YAML Kopfzeile.

Anhand dieser .Rmd Datei kreiert R eine neue Datei für das Protokoll. Der Wert von output_ bestimmt die Dateiart (siehe auch 8.), z. B.

output: html_document HTML-Datei (Webseite)

output: pdf_document PDF-Datei

output: word_document Microsoft Word-Datei (DOCX)

output: beamer_presentation beamer Slideshow (PDF)

output: ioslides_presentation ioslides Slideshow (HTML)

RStudio ist ein eingetragenes Warenzeichen von RStudio, Inc. • © 2017 RStudio • info@rstudio.com • 844-449-1313 • rstudio.com
Übersetzt von Lucia Quintana • luc@quintana.org

Figure 18:

PDF Dokumente und Präsentationen mit LaTeX, Beamer und Sweave

Präsentationen mit Rmarkdown - beamer Präsentationen

The image shows a Beamer presentation slide. The title bar at the top is dark blue with the text 'Import csv' in white. The main content area is white and contains two lines of R code in a monospaced font: `url <- "https://raw.githubusercontent.com/Japhilko/GeoData/master/2015/data/whcSites.csv"` and `whcSites <- read.csv(url)`. The footer consists of two parts: a black bar on the left with the text 'Jan-Philipp Kolb' in white, and a light blue bar on the right with the text 'Datenimport' in black.

```
url <- "https://raw.githubusercontent.com/Japhilko/GeoData/master/2015/data/whcSites.csv"

whcSites <- read.csv(url)
```

Jan-Philipp Kolb Datenimport

Figure 19:

Beamer Optionen

Beamer Themen

Chunks einfügen

- Auch hier lassen sich natürlich Chunks einfügen
- Wenn `cache=T` angegeben ist, wird das Ergebnis des Chunks abgespeichert
- Es ist sinnvoll die Chunks zu benennen, dann findet man auch das Ergebnis einfacher

Edit R Markdown Presentation Options

Output Format: PDF (Beamer) ▼

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

General

Figures

Advanced

☐ Include a table of contents

☐ Render slide bullets incrementally

Theme: default ▼

Color theme: default ▼

Font theme: default ▼

☒ Syntax highlighting: default ▼

OK

Cancel

Figure 20:

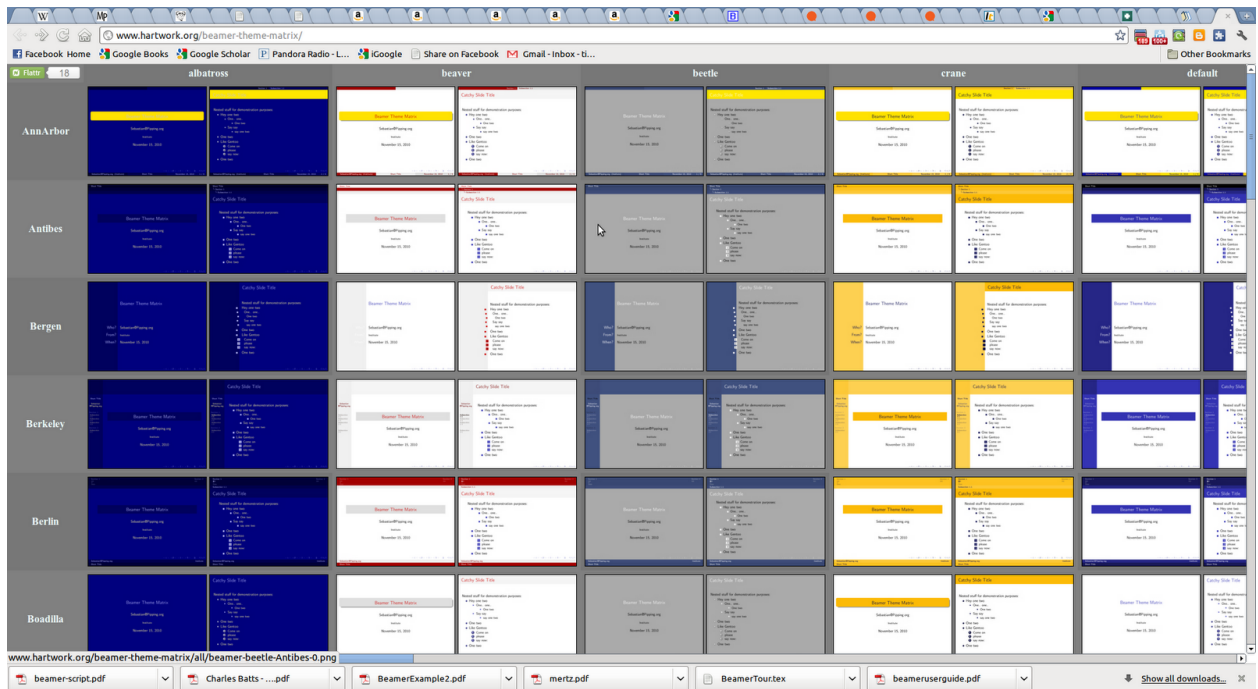


Figure 21:

```

```{r Zufallszahlen, cache=TRUE}
ab <- runif(1000)
```

```

Figure 22:

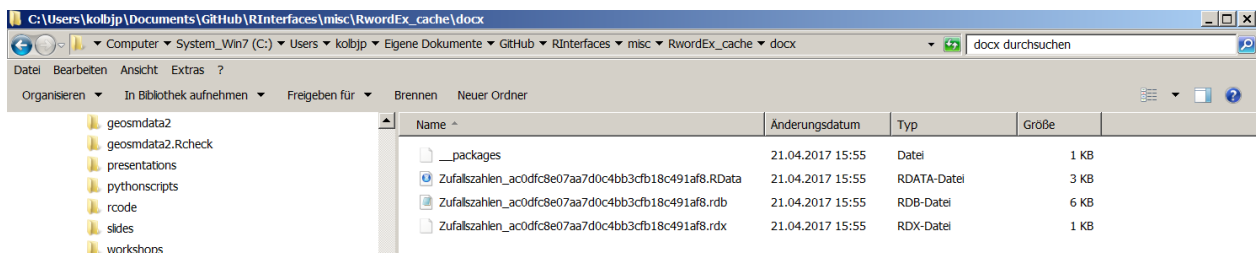


Figure 23:

Ergebnis - Cache

Wie man das im Header des Dokuments angibt

```
---
title: "Intro - Erste Schritte"
author: "Jan-Philipp Kolb"
date: "10 April 2017"
output:
  beamer_presentation:
    colortheme: beaver
    theme: CambridgeUS
---
```

Inhaltsverzeichnis I

```
---
output:
  beamer_presentation:
    toc: true
---
```

Figure 24:

```
output:
  beamer_presentation:
    toc: yes
```

Optionen für die Graphikeinbindung

- *fig_caption: false*, wenn man keine Bildunterschriften haben möchte

```
---
title: "Habits"
output:
  beamer_presentation:
    fig_width: 7
    fig_height: 6
    fig_caption: true
---
```

Figure 25:

Präsentationen mit Sweave

- Das Dokument erstellen

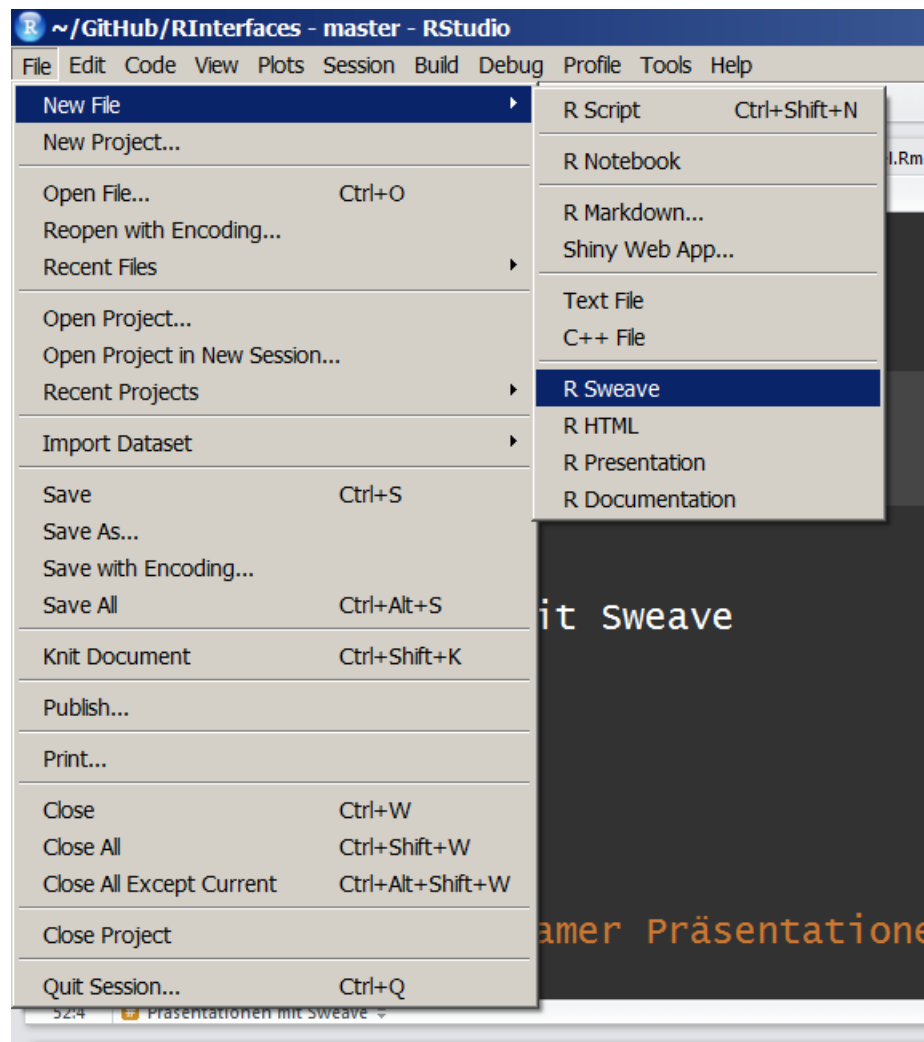


Figure 26:

Sweave Präsentation

- Ganz normaler LaTeX Code wird verwendet

```
1 \documentclass{beamer}
2
3 \begin{document}
4
5 \begin{frame}
6 \frametitle{Erste Folie}
7 \end{frame}
8
9
10 \end{document}
```

Figure 27:

Chunks bei Sweave

- Auch hier kann R-code verwendet werden

```
9 \begin{frame}
10 \frametitle{Zweite Folie}
11
12 <<=>
13 ab <- sample(1:10,8)
14 @
15
16 \end{frame}
```

Figure 28:

Chunk Optionen

- Auch bei Sweave Chunks können Optionen mitgegeben werden

Inline Code

- Manchmal braucht man das Ergebnis direkt auf der Folie

`\Sexpr{}`

| | |
|---------------------------|----------------------------------------------|
| <code>echo=false</code> | R-Code ist nicht sichtbar |
| <code>results=tex</code> | die Ergebnisse werden wie TeX-Code behandelt |
| <code>results=hide</code> | die Ergebnisse sind nicht sichtbar |
| <code>fig=true</code> | Grafiken werden eingebunden |
| <code>width=</code> | Breite der Grafik in Inch (z. B. 4) |
| <code>height=</code> | Höhe der Grafik in Inch |

Figure 29:

```

5  \begin{frame}
6  ♥ <<>>=
7  CRANmirror <- "http://cran.revolutionanalytics.com"
8
9  cran <- contrib.url(repos = CRANmirror,type = "source")
10
11 info <- available.packages(contriburl = cran, type = x)
12 @
13
14 Es gibt aktuell \Sexpr{nrow(info)} Pakete auf CRAN.
15 \end{frame}

```

Figure 30:

```

CRANmirror <- "http://cran.revolutionanalytics.com"
cran <- contrib.url(repos = CRANmirror,type = "source")

info <- available.packages(contriburl = cran, type = x)

```

Es gibt aktuell 10423 Pakete auf CRAN.

Figure 31:

Inline Code - das Ergebnis

```
CRANmirror <- "http://cran.revolutionanalytics.com"
cran <- contrib.url(repos = CRANmirror, type = "source")
info <- available.packages(contriburl = cran, type = x)
nrow(info)
```

```
## [1] 10512
```

PDF Paper mit R

- Mit R ist es möglich Berichte oder Paper zu erzeugen
- Dies eignet sich besonders gut, wenn man viel Code hat oder einen Bericht sehr oft erzeugen muss
- Literatur lässt sich am Besten mit einem bibtex file einbauen

Jabref

- Literaturverwaltungssystem

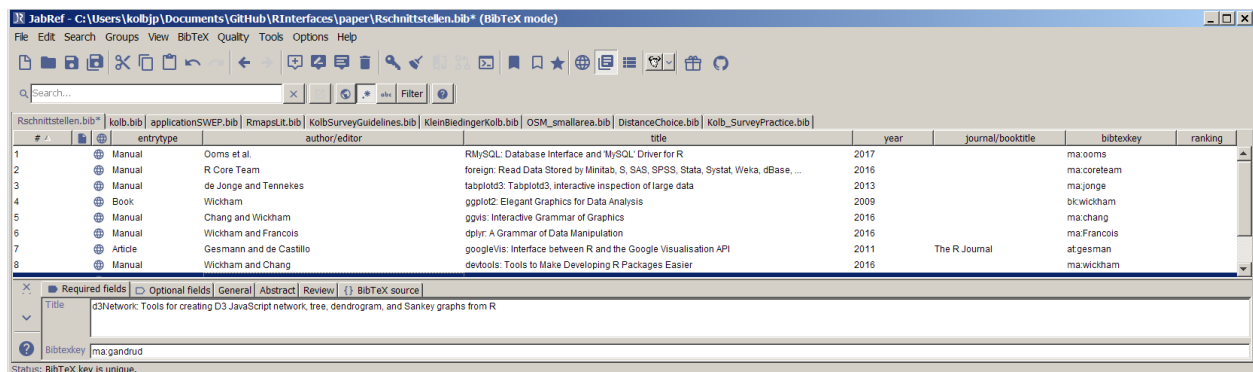


Figure 32:

Referenz mit R bekommen

- Mit dem Befehl `citation()` bekommt man sehr schnell die Referenz

```
install.packages("RMySQL")
```

```
citation("RMySQL")
```

```
##
## To cite package 'RMySQL' in publications use:
##
## Jeroen Ooms, David James, Saikat DebRoy, Hadley Wickham and
## Jeffrey Horner (2017). RMySQL: Database Interface and 'MySQL'
## Driver for R. R package version 0.10.11.
## https://CRAN.R-project.org/package=RMySQL
##
## A BibTeX entry for LaTeX users is
##
```

```
## @Manual{,
##   title = {RMySQL: Database Interface and 'MySQL' Driver for R},
##   author = {Jeroen Ooms and David James and Saikat DebRoy and Hadley Wickham and Jeffrey Horner},
##   year = {2017},
##   note = {R package version 0.10.11},
##   url = {https://CRAN.R-project.org/package=RMySQL},
## }
```

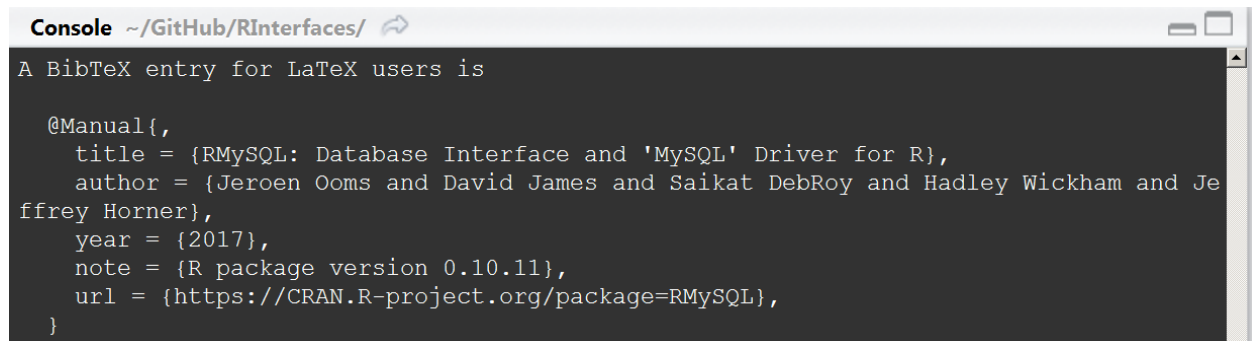


Figure 33:

Das bibtex file einbinden I

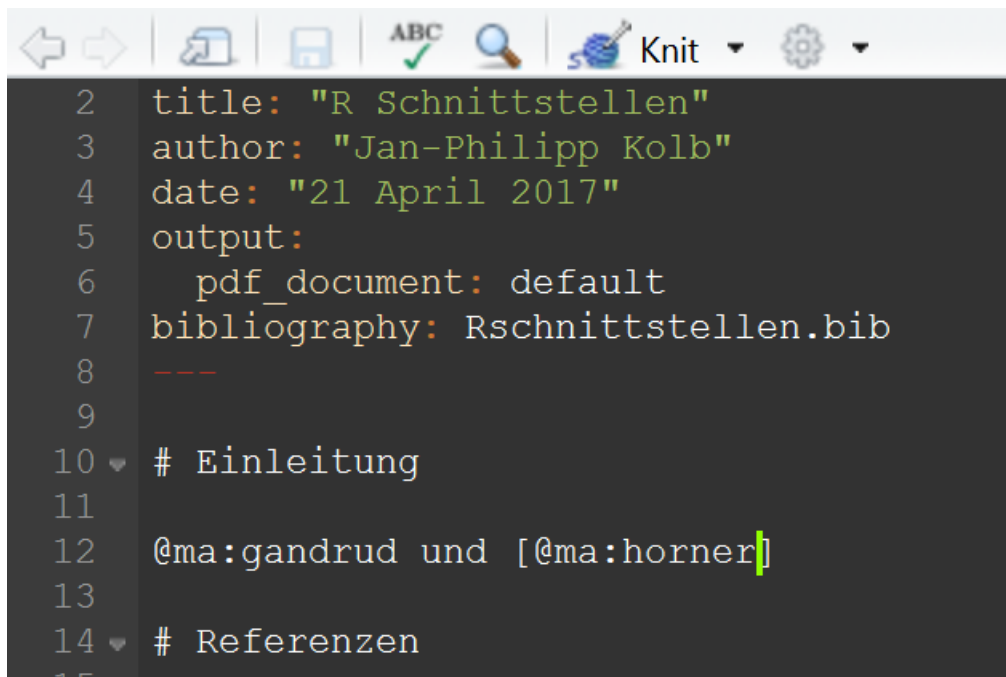


Figure 34:

Das bibtex file einbinden II

```
title: "R Schnittstellen"
author: "Jan-Philipp Kolb"
date: "21 April 2017"
output:
  pdf_document: default
bibliography: Rschnittstellen.bib
---
```

Das Ergebnis

R Schnittstellen

Jan-Philipp Kolb

21 April 2017

Einleitung

Gandrud (2015) und (Horner 2014)

Referenzen

Gandrud, Christopher. 2015. *D3Network: Tools for Creating D3 Javascript Network, Tree, Dendrogram, and Sankey Graphs from R*. <https://CRAN.R-project.org/package=d3Network>.

Horner, Jeffrey. 2014. *Rook: Rook - a Web Server Interface for R*. <https://CRAN.R-project.org/package=Rook>.

Figure 35:

Links

- Optionen für Beamer Präsentationen
- Wie R und LaTeX zusammen funktionieren