

Teil 2: Dokumentation mit Rmarkdown

Edith Scheifele

01 März 2019

Wie legt man ein Rmd-File an?

- ▶ In RStudio -> File -> New File -> Rmarkdown
- ▶ Wir vergeben einen Titel und schreiben unseren Namen in das Autoren-Feld.
- ▶ Ferner wählen wir *HTML* als Output.
- ▶ Alle 3 Optionen lassen sich im Nachhinein noch anpassen bzw. variieren.
- ▶ Über den Button mit dem blauen Knäul lässt sich das gewünschte Dokument *knitten*.

Aufgabe

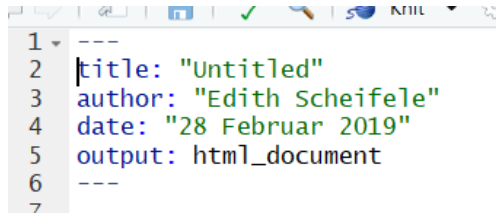
Legt wie oben beschrieben ein Rmd-File an. RStudio gibt Euch ein Dummy-File aus. Knittet dieses!

Aufbau und Bestandteile eines Rmd-Files

Ein Rmd-File besteht im Wesentlichen aus folgenden Bestandteilen

- ▶ YAML-Header
- ▶ Chunks und Inline-Kode
- ▶ Text

YAML-Header

A screenshot of a code editor window. The editor has a toolbar at the top with icons for undo, redo, save, and search. The code is displayed on a light gray background with line numbers on the left. The code is a YAML header for a document.

```
1 ---
2 title: "Untitled"
3 author: "Edith Scheifele"
4 date: "28 Februar 2019"
5 output: html_document
6 ---
7
```

- ▶ der Header beginnt und endet mit drei Dashes (- - -)
- ▶ alle Optionen sind als key-value pairs variierbar
- ▶ **R Markdown:: Cheat Sheet: S. 2 Set render options with YAML**
- ▶ Etwas tricky: Achtung bei der Indentierung; Doppelpunkte zeigen an, dass danach noch ein Wert kommt

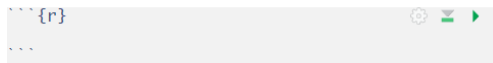
YAML-Header - Aufgabe

1. Variiert das Output-Format
2. Fügt ein Inhaltsverzeichnis dazu
3. Probiert 1 andere Veränderung aus! Z.B. setzt die Linkfarbe auf blau.

Nutzt Google!

Chunks

- ▶ Chunks enthalten Euren Code und werden von R ausgeführt, wenn Ihr das Dokument knittet
- ▶ Einfügen von Chunks:
 - ▶ Windows: STRG+ALT+I
 - ▶ Mac: Cmd+Option+I
 - ▶ Über das Icon *Insert* > R
- ▶ ein Chunk beginnt und endet mit 3 Tickmarks, wobei nach den ersten drei ein `{r}` kommt



Chunks - Aufgabe

1. Platziert in unserem Dummy-Rmd-File 3 Chunks am Ende des Files:
2. Chunk: Summe aus 2 beliebigen Zahlen:
3. Chunk: eine Variable *sum_ab*, die die Summe aus zwei beliebigen Zahlen beinhaltet
4. Chunk: einen R-Kommentar

Knittet das Dokument! Was fällt Euch auf?

Inline-Kode

- ▶ Manchmal möchte man Variablen, die man einem Chunk berechnet hat, im Fließtext verwenden
- ▶ Vorteil: Jedesmal, wenn sich der Wert der Variablen ändert, ändert sich auch der entsprechende Wert im Fließtext
- ▶ Ihr könnt auch innerhalb des Inline-Kodes Berechnungen vornehmen wie in ganz normalen Chunks auch

```
```{r}  
sum_ab <- 2 + 3
```
```

Die Summe von a und b ist `r sum_ab`.

- ▶ Inline-Kode beginnt mit einem Backtick, gefolgt von einem kleinen R und einem schließenden Backtick: ``r und endet mit ``

Inline-Kode - Aufgabe

Probiert es aus!

Local options

- ▶ Das lokale Verhalten von Chunks steuert man über Argumente, die man direkt in den Chunk schreibt

Auswahl aus **R Markdown:: Cheat Sheet**: S. 1 *Important Chunk Options* mit gängigen Werten

- ▶ `comment = NA`
- ▶ `echo = TRUE / FALSE`
- ▶ `eval = TRUE / FALSE`
- ▶ `fig.align, fig.cap, fig.height, fig.width`
- ▶ `include = TRUE / FALSE`
- ▶ `message = TRUE / FALSE`
- ▶ `results = 'asis' / 'hide'`

Local options - Aufgabe

Wir wechseln wieder in unser Dummy-File und verändern nach und nach jedes Argument.

- ▶ `comment = NA` (default: `'###'`)
- ▶ `echo = TRUE / FALSE`
- ▶ `eval = TRUE / FALSE`
- ▶ `fig.align`, `fig.cap`, `fig.height`, `fig.width`
- ▶ `include = TRUE / FALSE`
- ▶ `message = TRUE / FALSE`
- ▶ `results = 'asis' / 'hide'`

Global options

- ▶ Um die Chunk-Optionen für das ganze Dokument zu setzen, verwendet man folgenden Chunk zu Beginn des Dokuments (unterhalb des YAML-Headers)
- ▶ **opts_chunk\$set()**

```
---  
title: "Untitled"  
author: "Edith Scheifele"  
date: "28 Februar 2019"  
output: html_document  
---
```

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(echo = TRUE, comment = NA)
```
```

- ▶ lokale Options überschreiben globale Options

Text (und Formatierung)

- ▶ normalen Text könnt Ihr einfach in das Dokument tippen
- ▶ wichtig: Identierung macht Unterschiede

Formatierungsauswahl: **R Markdown:: Cheat Sheet: S. 2**

Pandoc's Markdown

- ▶ **kursiv**
- ▶ ****fett****
- ▶ Überschriften mit verschiedenen Einbettungen: #, ## usw.
- ▶ Listen mit Bullet-Points: -

Text (und Formatierung) - Aufgabe

Probiert die Formatierungsoptionen aus:

- ▶ kursiv, fett, Überschriften verschiedener Einbettungstiefen, Listen und noch 2 weitere aus der Liste

Tabellen

R Markdown:: Cheat Sheet: S. 2 Table Suggestions

- ▶ je nach Output (html, pdf, Word) sind andere Pakete sinnvoll
- ▶ für html eignet sich das *kable*- bzw. das *kableExtra*-Package, die wie alle anderen Packages installiert und geladen werden müssen

Aufgabe

- ▶ Nehmt das Dataset *cars*, das in R pre-installiert ist, und macht mit daraus in unserem Dokument eine schön formatierte Tabelle.
- ▶ nehmt das *kable*-Package und den entsprechenden *kable*-Befehl
- ▶ Beachtet die *Chunk*-Option

Andere Output-Formate

- ▶ andere Formate: pdf (Latex muss installiert sein), Word (MS Word muss installiert sein), shinyapps, dashboard und weitere
- ▶ Format-Vorlagen

Im **APA**-Format

```
install.packages("papaja")
```

viele weitere:

```
devtools::install_github("rstudio/articles")
```