Grundlagen in R Markdown

Martin Berger

Wintersemester 18/19

R Markdown ist ein Dateiformat, welches erlaubt in einer Datei R Code, Outputs (z. B. Plots) und Kommentare zu kombinieren und diese als formatierte html, pdf oder Worddatei zu exportieren. Typische Anwendungen von R Markdown umfassen das Schreiben von wissenschaftlichen Artikeln, erstellen regelmässiger Geschäftsberichte, shiny apps oder Websites und Blogs. Das folgende Skript gibt eine kurze Einführung in R Markdown und basiert dabei hauptsächlich auf (Xie 2018) und (Yihui Xie 2018).

1 Installation und Grundlagen

Um R Markdown zu verwenden, muss das zugehörige Package installiert werden:

```
install.packages('rmarkdown')
```

Um zudem PDF Dateien erstellen zu können, muss LaTeX installiert sein. Obiger Befehl installiert insbesondere das *Knitr* package. Knitr ist der Grundbaustein, der es erlaubt Code mit Text zu vermischen. Wenn Knitr eine Datei kompiliert, extrahiert es die vorhandenen Code Chunks und führt diese aus. Anschliessend "knitted" es den ausgeführten Code mit den Textbausteinen zu einem Markdown file. Dieses wird anschliessend mittels *Pandoc* in das gewünschte Output Format (z. B. html) kompiliert. Das Package rmarkdown automatisiert obigen Prozess. Da Pandoc kein R Package ist, muss man es prinzipiell manuell installieren. Wird jedoch RStudio verwendet, ist dies nicht notwendig, da RStudio Pandoc beinhaltet. Knitr kann auch auf andere Programmiersprachen angewandt werden, z. B. Python, C++ usw. Wir beschränken uns hierbei allerdings auf R.

Wir werden zudem im Folgenden *Bookdown* verwenden. Bookdown ist eine Erweiterung von R Markdown, welche es insbesondere ermöglicht einfach auf Tabellen und Bilder zu referenzieren.

```
install.packages('bookdown')
```

Zum Kompilieren eines R Markdowns Files kann der Shortcut `Ctrl+Shift+k` oder die Schaltfläche `Knit` verwendet werden.

2 Dokumentenaufbau und Syntax

In RStudio kann ein neues R Markdown file mittels `File -> New File -> R Markdown` erstellt werden. Alternativ speichert man eine Datei mit der Endung `.Rmd`. Ein R Markdown Dokument besteht aus drei Teilen: der Metadata, Text und Code.

2.1 Metadata

Den Anfang macht immer die Metadata. Diese besteht aus Informationen wie Titel, Autor oder Ähnlichem und dem gewünschten Outputformat und wird zwischen einem Paar von drei Strichen `---` definiert. Die Syntax für die Metadata ist *YAML*. Hier ein Beispiel:

```
---
title: "Grundlagen in R Markdown"
```

```
author: "Martin Berger"
date: "Wintersemester 18/19"
output:
  bookdown::pdf_document2:
    toc: false
lang: de
bibliography: ref.bib
---
```

Obiger Code erzeugt ein PDF Dokument. Wird anstelle von bookdown nur das rmarkdown Package verwendet, reicht es nur `output: pdf_document` zu schreiben. Da wir bookdown zum Referenzieren verwenden wollen, benötigen wir obige Syntax. Alternativ kann auch `html_document2` oder `word_document2` verwendet werden. Die Option `toc` legt fest, ob ein Inhaltsverzeichnis erstellt werden soll. Setzt man `lang` auf `de` werden deutsche Begriffe, anstelle von englischen verwendet, z. B. Abbildung anstelle von Figure, und in `Bibliography` schreibt man den Namen des `.bib` Files, siehe Abschnitt 4. Noch eine kurze Warnung, wie in Python werden Einrückungen in YAML beachtet.

2.2 Code

Der Metadata folgen Text und Code. R Code kann auf zwei Arten eingefügt werden. Inline Code beginnt mit `` `r `` und endet mit erneut mit einem Backtick `` ` ``. Code Chunks beginnen mit drei Backticks ```` ``` ```` und der Angabe der verwendeten Sprache und enden wiederum mit drei Backticks. Alternativ kann der Shortcut `Ctrl + Alt + I` bzw. `Cmd + Option + I` (bei macOS) verwendet werden. Beispielsweise erzeugt der Code Chunk

````
```{r, echo = TRUE}
sin(pi)
```
````

den Output

```
sin(pi)
```

```
## [1] 1.224606e-16
```

Innerhalb von Code Chunks kann in gewohnter Weise programmiert werden. Der allgemeine Aufbau eines Chunks ist `{r chunk label, options}`. Das Chunk Label wird für spätere Referenzen benötigt und mittels der Chunk Optionen können bestimmte Eigenschaften festgelegt werden. Der Befehl `echo = TRUE` ist eine ebensolche Chunk Option und bewirkt, dass der Source Code ebenso ausgegeben wird. Ebenso nützlich ist die Option `eval`, diese legt fest ob der Code ausgeführt werden soll oder nicht. Für weitere Optionen siehe (Yihui Xie 2018).

2.3 Text

Die Syntax für Text ist *Pandoc's Markdown*. Wir besprechen kurz ein paar der wichtigsten Befehle. Um Text *kursiv* zu schreiben verwendet man `*text*` oder `_text_`. Will man Text **fett** schreiben, umgibt man ihn mit `**`, also `**text**`. Indizies werden von `~` umgeben und Exponenten von `^`. Hyperlinks werden mittels `[text](link)` eingefügt, zum Beispiel erzeugt `[UIBK](https://www.uibk.ac.at/)` den Link UIBK. Überschriften werden mittels `#` eingeleitet, wobei die Anzahl der `#` angibt, welche Stufe verwendet wird, also zum Beispiel mittels

```
# Oberste Stufe

## Zweite Stufe

### Dritte Stufe {-}
```

wobei `{-}` angibt, dass keine Nummerierung vorgenommen werden soll. Aufzählungszeichen erstellt man mittels `*`,`-` oder `+`, Nummerierungen mit Zahlen und durch Einrückungen können untergeordnete Listen erstellt werden, beispielsweise erzeugt

```
1. item
2. item
    - item
    - item
```

die Aufzählung

1. item
2. item
    - item
    - item

Zitate werden mittels `>` geschrieben. Zum Beispiel,

```
> The chase...is better than the catch
> Transforming the Tunes
> We need your support
> If you've got the breath back
> It's the first page of the second chapter
>
> --- H.P. Baxxter
```

und als Ausgabe erhält man

> The chase... is better than the catch
> Transforming the Tunes
> We need your support
> If you've got the breath back
> It's the first page of the second chapter
>
> — H.P. Baxxter

Beachte, um Zeilenumbrüche innerhalb des Zitats zu erhalten, müssen am Ende der Zeile zwei Leerzeichen eingefügt werden.

2.3.1 LaTeX

Um mathematische Ausdrücke zu schreiben, kann weiterhin LaTeX Syntax verwendet werden. Um LaTeX inline zu verwenden schreibt man den Ausdruck zwischen zwei `$` Zeichen, zum Beispiel schreibt man `$f(x) = x^2$` für $f(x) = x^2$. Um Display Style zu verwenden, werden jeweils zwei `$$` benötigt. Schauen wir uns wieder ein Beispiel an, `$$\int f'(x)\operatorname{d}\!x = f(x)+c$$` erzeugt

$$\int f'(x)\operatorname{d}\!x = f(x)+c\,.$$

3 Referenzieren mittels Bookdown

Um auf Bilder zu referenzieren verwenden wir das Paket *Bookdown*. Bilder können prinzpiell einfach mittels html Code eingefügt werden. Will man allerdings auf Grafiken mittels Bookdown referenzieren, müssen diese innerhalb von Code Chunks eingefügt werden. Dazu muss dem Chunk ein Label gegeben werden und anschließend eine Caption. Danach kann mittels `\@ref(fig:chunklabel)` auf die Grafik verwiesen werden. Betrachten wir folgendes Beispiel:

```
```{r ecdf, echo = F, fig.height = 3, fig.width = 4,
fig.cap = "Empirische Verteilungsfunktion LaTeX einer normalverteilten Stichprobe."}
```
```

```
x = rnorm(10)
plot(ecdf(x), main = "Empirische Verteilungsfunktion")
```

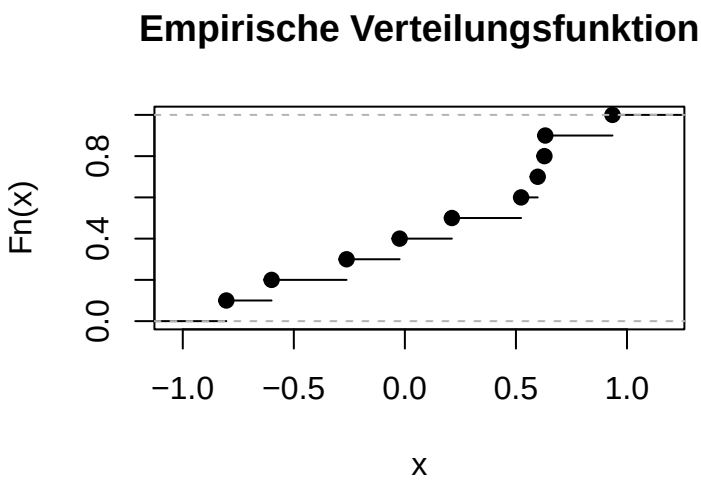

Abbildung 1: Empirisiche Verteilungsfunktion einer normalverteilten Stichprobe.

Schreiben wir nun `Abbildung \@ref(fig:ecdf)`, so erzeugt dies: Abbildung 1.

Soll auf eine nicht von R generierte Grafik verwiesen werden, können wir diese innerhalb eines R Code Chunks mit dem Befehl `knitr::include_graphics(filename)` einfügen. Betrachten wir wiederum ein Beispiel. Zuerst definieren wir mittels

`(ref:normaldist) Quelle: https://en.m.wikipedia.org/wiki/File:NormalDist.png`

eine neue Caption, indem wir obige Zeile in den Fließtext schreiben, wobei darüber und darunter eine Leerzeile sein muss. Danach können wir einen Code Chunk erstellen, in welchem wir auf `(ref:normaldist)` verweisen. Weiters verwenden wir nun die Optionen `fig.align`, um das Bild zu zentrieren und `out.width`, was die Breite des Bildes reguliert:

```
```{r normaldist, echo = FALSE,
fig.cap = "Quelle: \url{https://en.m.wikipedia.org/wiki/File:NormalDist.png}",
out.width = "120px",fig.align = 'center'}
knitr::include_graphics('normaldist.png')
```
```

Dies erzeugt Abbildung 2. Man beachte zudem, dass innerhalb des Code Chunks automatisch der Befehl `\url{}` verwendet wird. Zudem könnten wir auch die Chunk Option `fig.cap` direkt anstelle von `(ref:normaldist)` verwenden, diese erlaubt allerdings keine Unterstriche, sofern Pdf als Output gesetzt ist. Das Referenzieren auf Gleichungen, Tabellen, Theoreme usw. funktionert ähnlich. Es sei dazu erneut auf (Xie 2018) verwiesen.

4 Literaturverzeichnis

Als letztes beschäftigen wir uns mit der Bibliographie. Um eine solche zu erstellen, muss ein `.bib` File erstellt und in der Metadata die Option `bibliography: FILENAME.bib` gesetzt werden. Das bib File für dieses Dokument hat die folgende Form:

```
@manual{bookdown,
```

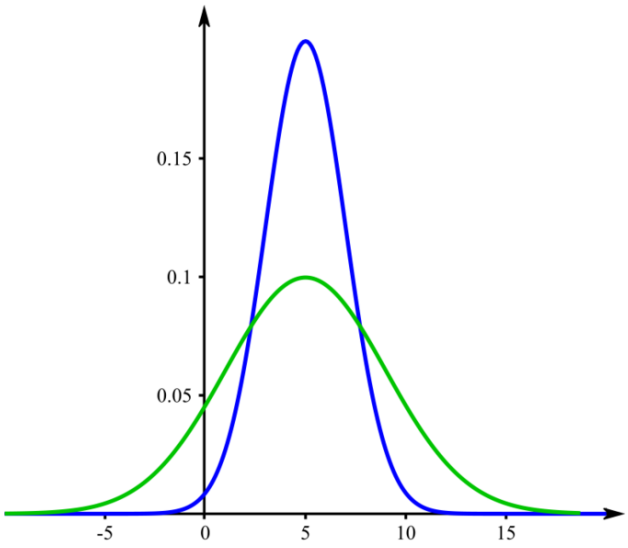

Abbildung 2: Quelle: https://en.m.wikipedia.org/wiki/File:NormalDist.png

```
author =   {Yihui Xie},
title =    {bookdown: Authoring Books and Technical Documents with R Markdown
},
howpublished = {\url{https://bookdown.org/yihui/bookdown/}},
year = {2018}
}

@manual{rmarkdown,
author =   {Yihui Xie, J. J. Allaire, Garrett Grolemund},
title =    {R Markdown: The Definitive Guide
},
howpublished = {\url{https://bookdown.org/yihui/rmarkdown}},
year = {2018}
}
```

Jeder Eintrag der Bibliographie beginnt mit `@type{`, typischerweise `article`, `book`, `manual` oder `misc`. Dieser wird vom citation key gefolgt, welchen man verwendet, um mittels `[@citationkey]` den entsprechenden Eintrag zu zitieren. Anschließend können diverse Angaben wie Autor, Titel usw. gemacht werden. Die Bibliographie wird immer zum Schluss des Dokumentes eingefügt, wobei hierbei standardmäßig nur jene Einträge des `.bib` Files verwendet werden, welche auch innerhalb des R Markdown Files zitiert werden.

Bibliographie

Xie, Yihui. 2018. *bookdown: Authoring Books and Technical Documents with R Markdown*. https://bookdown.org/yihui/bookdown/.

Yihui Xie, Garrett Grolemund, J. J. Allaire. 2018. *R Markdown: The Definitive Guide*. https://bookdown.org/yihui/rmarkdown.