

Einführung in R Markdown

Benjamin Schlegel

24. Oktober 2018

R Markdown ist eine Sprache mit der PDF-, Word- und HTML-Dateien erstellt werden können. Um R Markdown verwenden zu können muss R und R Studio installiert sein. In [R Grundlagen](#) wird die Installation erklärt. Standardmässig können HTML-Dateien erstellt werden. Um Word-Datei erstellen zu können, muss Word oder ein ähnliches Programm installiert sein. Am häufigsten wird jedoch ein PDF-Datei erstellt, da das Layout dieser Dateien am schönsten ist.

Vorbereitung

Um PDF-Dateien mit R Markdown erstellen zu können, wird eine TeX-Distribution benötigt. Wer LaTeX nur mit R Markdown verwenden will, installiert am besten TinyTeX. Wer LaTeX auch selbstständig nutzen will ohne dafür jedes Mal mit dem Terminal arbeiten zu müssen, greift am besten auf MiKTeX, MacTeX oder TeX Live zurück. In diesem Tutorial wird die Installation von TinyTeX erklärt.

Die Installation von TinyTeX ist Dank der R Bibliothek tinytex ganz einfach.

```
install.packages("tinytex")
tinytex::install_tinytex()
```

Die Installation nimmt einige Zeit in Anspruch. Sobald die Installation fertig ist, können R Markdown zu PDF-Dateien konvertiert werden. Wenn nach einer gewissen Zeit die Fehlermeldung "Remote repository newer than local" erscheint, braucht TinyTeX ein Upgrade. Das geht mit dem folgenden Befehl:

```
tinytex::reinstall_tinytex()
```

R Markdown verwenden

Um eine neue R Markdown-Datei zu erstellen, kann in R Studio unter Datei/Neue Datei R Markdown ausgewählt werden. Es erscheint ein Popup für die Voreinstellungen. Unter Titel wird der Titel der Datei angegeben unter Autor der eigene Name (oder mehrere, falls es eine Gruppenarbeit ist). Unten kann PDF ausgewählt werden, damit beim konvertieren standardmässig ein PDF erstellt wird. Es ist auch möglich Folien zu erstellen, wobei dieses Tutorial nicht darauf eingeht. Anschliessend auf ok klicken. Alle Einstellungen können später auch in der Datei angepasst werden.

In einem ersten Schritt lohnt es sich zu testen, ob das erstellen eines PDFs klappt. Dazu klickt man oberhalb der Datei auf Knit und speichert anschliessend die R Markdown Datei am gewünschten Ort. Anschliessend wird das Beispiel in ein PDF umgewandelt.

Die R Markdown-Datei besteht aus einem Kopf und R Markdown Code. Im Kopfbereich sind zwischen zwei --- die Einstellungen notiert. Dort können sie auch geändert werden. Alles nach dem Kopfbereich kann gelöscht werden, da es nur Beispielcode ist, den R Studio in jeder neuen Datei zur Verfügung stellt.

Grundsätzlich kann in R Markdown normaler Text geschrieben werden. Um einen neuen Absatz zu erstellen werden am Ende einer Zeile zwei Leerzeichen hinzugefügt. Mit einem * am Anfang und Ende kann Text kursiv angezeigt werden,

mit zwei * fett. Anstelle des Sterns kann auch ein _ verwendet werden. Titel können mit # markiert werden, wobei # ein Haupttitel darstellt, ## ein Untertitel, ### ein Unteruntertitel und so weiter bis max. sechs #.

Aufzählungen können mit * erstellt werden, wobei vorher eine leere Zeile und nach dem * ein Leerzeichen vorhanden sein muss. Unterlisten funktionieren mit + und eingerückt. Geordnete Listen können mit 1., 2., etc. erstellt werden, wobei auch hier die Leerzeile davor notwendig ist.

Weitere Formatierungen finden sich im [R Markdown Cheat Sheet](#).

R Code in R Markdown

Das interessante an R Markdown ist, dass einfach R Code hinzugefügt werden kann. Im Dokument wird auch gleich der Output angezeigt. Um R Code einzufügen, wird oben auf Einfügen --> R geklickt. Anschliessend erhält man einen R Code-Block:

```
```{r}
```
```

In diesem Code-Block kann R Code eingefügt werden. Standardmässig wird sowohl der R Code als auch der Output und alle Meldungen und Warnungen, aber keine Fehler angezeigt.

Für den Code Block gibt es viele Einstellungen, welche nach dem r Kommagetrennt aufgelistet werden können. Die Einstellung eval ist standardmässig auf TRUE eingestellt. Setzt man diesen Wert auf FALSE wird der Code zwar angezeigt, aber nicht evaluiert. Wer den Code verbergen will kann echo=FALSE setzen. Mit warning=FALSE können Warnungen verborgen werden. Mit error=TRUE können Fehlermeldungen angezeigt werden. Mit message=FALSE werden Meldungen verborgen.

Um mit stargazer/texreg schöne Tabellen anzeigen zu können, muss results='asis' gesetzt werden. Wichtig dabei ist, dass die Funktion für PDF-Dateien LaTeX-Code zurückgibt, für HTML- und Word-Dateien HTML-Code. Bei results='hide' wird das Resultat verborgen. Im Folgenden ein kurzes Beispiel zum Erstellen einer Tabelle:

```
```{r, echo=FALSE, results='asis'}
data = data.frame(x = rnorm(100), y = 0.5*rnorm(100))
model = lm(y ~ x, data = data)
texreg::texreg(model)
```
```

| | Model 1 |
|---------------------|-----------------|
| (Intercept) | -0.03
(0.05) |
| x | -0.04
(0.06) |
| R ² | 0.00 |
| Adj. R ² | -0.01 |
| Num. obs. | 100 |
| RMSE | 0.55 |

***p < 0.001, **p < 0.01, *p < 0.05

Table 1: Statistical models

Mit ``r 2 + 2`` kann im Text das Resultat eines R Codes angezeigt werden. Dies erlaubt Zahlen automatisch in den Text einzufügen und einen Bericht mit neuen Daten ohne Aufwand mit einem Klick zu erstellen.

Um Daten als Tabellen anzuzeigen, kann `knitr::kable` benutzt werden.

```
```{r}
data = data.frame(x = rnorm(5), y = 0.5*rnorm(5))
knitr::kable(data, caption = 'A caption')
```
```

Weitere Details können im übersichtlichen [R Markdown Cheat Sheet](#) nachgeschaut werden.

Mathematische Formeln

Mathematische Formeln können in R Markdown mit einem Dollar-Zeichen am Anfang und Ende in einen Text eingefügt werden. Mit zwei Dollar-Zeichen am Anfang und Ende wird die Formel zentriert auf einer neuen Zeile dargestellt. Die Formel wird mit LaTeX -Code geschrieben und wird unterhalb des Formel-Blocks direkt angezeigt.

Ein Beispiel:

```
$$
\hat{y}_i = \beta_0 + \beta_1x_{1i} + \beta_2x_{2i} + \beta_3x_{1i}x_{2i}
$$
```

Mehr zu mathematischen Formeln in LaTeX findet man beispielsweise auf [Overleaf](#).