

# R Markdown

## 1 Zielsetzung von R Markdown

Erstellen eines Dokuments, in dem sowohl Text, wie auch Code sein kann:

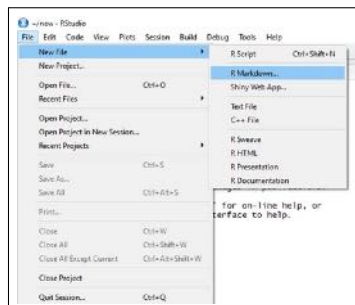
- Forschung wird reproduzierbar.
- Leichterer Austausch von Analyseergebnissen und der zugehörigen Dokumentation.
- Beim Erstellen (“knit”) wird der Code ausgeführt und mit den Beschreibungen zusammen angezeigt.
- Nur ein Dokument, was keine Warnungen/Fehler erzeugt, kann erstellt werden.
- Das kombinierte Dokument wird als Word-, HTML- oder PDF-Datei erstellt.

## 2 Anlegen und Erstellen eines Dokuments

Nachdem RStudio gestartet ist, sind lediglich die folgenden Klicks notwendig, um einen beispielhaften Rumpf einer Datei für R Markdown zu erzeugen.

Open RStudio:

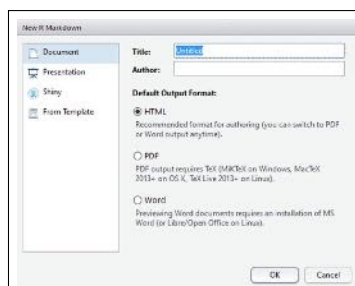
File → New File → R Markdown ...



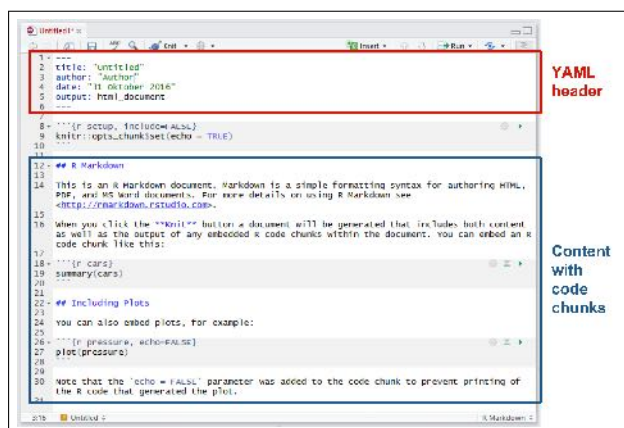
Oben: Titel des Dokuments angeben

Unten: Ausgabeformat wählen

Dialog abschließen mit OK



Die obige Klick-Reihenfolge erstellt nun ein beispielhafte Datei für R Markdown. Der Kommandoblock zu Beginn der Datei (der sogenannte YAML Header) legt fest, wie der Title des Dokumentes lautet, wer als Autor angegeben wird und in welchem Format die Ausgabe erfolgt. In dem erzeugten Dateirumpf findet sich auch der ausgewählte Dokumententitel (`title: [Untitled]`) sowie der Autor wieder. Weitere Optionen in dem YAML Header erlauben es später, Inhaltsverzeichnisse erzeugen zu lassen, das Layout zu variieren, usw.



**Speichern:** Das Dokument kann jetzt gespeichert werden. Die herkömmliche Dateiendung lautet \*.Rmd.

### 3 Mögliche Ausgabeformate

Innerhalb des YAML Headers wird auch das Ausgabeformat festgelegt. Die folgende Tabelle gibt einen Überblick über die wichtigsten Dokumententypen.

| Typ                     | Format | Option im YAML Header                      |
|-------------------------|--------|--|
| Webseite                | HTML   | <code>output: html_document</code>         |
| Dokument                | PDF    | <code>output: pdf_document</code>          |
| Dokument (Word)         | RTF    | <code>output: word_document</code>         |
| Präsentation (beamer)   | PDF    | <code>output: beamer_presentation</code>   |
| Präsentation (ioslides) | HTML   | <code>output: ioslides_presentation</code> |

**Hinweis:** Als Default ist die Ausgabe von PDFs ausgeschaltet, da dies der Installation weiterer Software bedarf. Hier ist nämlich eine  $\text{\LaTeX}$ -Distribution wie MikTeX oder TeXlive notwendig.

### 4 Knitten einer R-Markdown-Datei

Die nun so erstellte R-Markdown-Datei kann nun in das gewünschte Ausgabeformat konvertiert werden. Dabei wird der gesamte Programmcode ausgeführt und sowohl der Code als auch die zugehörige Ausgabe mit aufgenommen. Dieser Prozess heißt Knit und wird über das entsprechende Menü aufgerufen.




## 5 Programmcode


Innerhalb des eigentlichen Dokuments kann nun beschreibender Text als auch Code angelegt werden. Der Code wird in Blöcken eingefügt, sogenannten Chunks. Diese sehen wie folgt aus:

```

'''{r}
3 + 4
# additional code
'''

```

Tipp: Das Tastatur-Shortcut für das Einfügen eines leeren Chunks lautet “Strg + Alt + I”. Alternativ kann auch der Button  gedrückt werden.


Chunks können darüber hinaus auch Namen erhalten. Dies ist im Zusammenspiel mit langen Dokumenten hilfreich, wo es sich lohnt eine Gliederung der Chunks und Kapitel via  einzublenden. Zusätzlich können Optionen zur Ausgabe festgelegt werden. Beispielsweise ist der folgende Chunk mit `sqrt2` benannt, während `echo=FALSE` festlegt, dass ausschließlich die Ausgabe und nicht der angezeigte Code wird.

```

'''{r sqrt2, echo=FALSE}
sqrt(2)
'''


```

Code Chunks werden nicht nur beim Erstellen des gesamten Dokuments ausgewertet, sondern können auch manuell ausgeführt werden:

- Der Tastatur-Shortcut “Strg + R” führt die aktuelle Zeile aus.
- Der grüne Pfeil  in der rechten oberen Ecke des Chunks führt den gesamten Code eines Chunks aus. Alternativ funktioniert der Shortcut “Strg + Shift + Enter”.

## 6 Markos zum Formatieren des Dokumentes

Die Formatierungen innerhalb von R Markdown werden über Markos angegeben. Die entsprechenden Stellen werden beim Knit-Prozess dann entsprechend formatiert. Grundsätzlich gilt: Text wird als Text interpretiert – solange dieser nicht in Code-Chunks oder dem YAML Header auftaucht. Absätze werden durch eine Leerzeile im Dokument von einander getrennt. Darüberhinaus gibt es noch weitere Markos:

| Syntax  | Ausgabe   |
|---|---|
| <b>Textformatierungen und Bilder</b>                  |   |
| Text  | Text  |
| <i>*kursiv*</i> <u>__kursiv__</u>                     | <i>kursiv</i>   |
| <b>**bold**</b> <u>__bold__</u>                       | <b>bold</b>   |
| <code>'verbatim code'</code>                          | verbatim code   |
| <del>~~strikethrough~~</del>                          | <del>strikethrough</del>  |
| <code>[link] (www.is.uni-freiburg.de)</code>          | <a href="#">link</a>  |
| Equation: <code>\$A = \setminus pi r^2\$</code>       | Equation: $A = \pi * r^2$   |
| Image: <code>![] (path/to/image.png)</code>           | Image:  |
| Horizontal rule: <code>***</code>                     | <hr/>   |
| <code>&gt; block quote</code>                         | <div>block quote</div>  |
| <b>Überschriften</b>                                  |   |
| <code># Header 1</code>                               | Header 1  |
| <code>## Header 2</code>                              | Header 2  |
| <code>### Header 3</code>                             | Header 3  |
| <code>#### Header 4</code>                            | Header 4  |
| <code>##### Header 5</code>                           | Header 5  |
| <b>Aufzählungen</b>                                   |   |
| <code>* unordered list</code>                         | • unordered list  |
| <code>* item 2</code>                                 | • item 2  |
| <code>+ sub-item 1</code>                             | - sub-item 1  |
| <code>+ sub-item 2</code>                             | - sub-item 2  |
| <code>1. ordered list</code>                          | 1. ordered list   |
| <code>2. item 2</code>                                | 2. item 2   |
| <code>+ sub-item 1</code>                             | - sub-item 1  |
| <code>+ sub-item 2</code>                             | - sub-item 2  |
| Einrückungen erfolgen jeweils durch 4 Leerzeichen     |   |
| <b>Tabellen</b>                                       |   |
| <code>Table Header</code>   <code>Table Header</code> | <b>Table Header</b> <b>Table Header</b>   |
| -----   -----   | -----   -----   |
| <code>Cell 1</code>   <code>Cell 2</code>             | Cell 1    Cell 2  |
| <code>Cell 3</code>   <code>Cell 4</code>             | Cell 3    Cell 4  |

## 7 Weitere Informationen

Gerade bei komplexeren Arbeiten lohnt sich der Einsatz von R Markdown, da damit das Working Directory automatisch auf den aktuellen Ordner gesetzt wird. Damit werden Aufrufe zu `setwd(path)` überflüssig.

Die folgenden Materialien bieten weitergehenden Informationen an und erleichtern den Einstieg:

- Die offizielle Webseite zu R Markdown: <http://rmarkdown.rstudio.com/>
- Eine kurze Einführung als Video samt detaillierter Erklärungen:  
<http://rmarkdown.rstudio.com/lesson-1.html>
- Eine Kurzübersicht der wichtigsten Befehle in Form eines Cheat Sheet:  
<http://rmarkdown.rstudio.com/lesson-15.html>