

R Markdown

Schummelzettel

Mehr auf rmarkdown.rstudio.com

rmarkdown 0.2.50 Update: 8/14



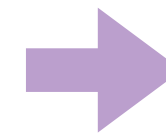
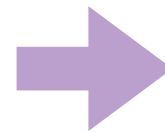
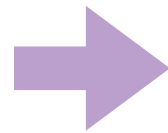
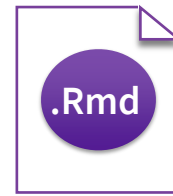
1. Arbeitsablauf Die Auszeichnungssprache „R Markdown“ erlaubt die Erstellung von reproduzierbaren und dynamisch anpassbaren Protokollen in R. R-Quellcode und dessen Ergebnisse können in Präsentationen, PDF-Dateien, HTML-Seiten, Word-Dateien etc. eingebettet werden. Um solch ein Protokoll zu erstellen:

i. **Öffnen** – Eine Datei mit .Rmd Extension öffnen oder erstellen.

ii. **Schreiben** – Inhalt mittels einfach zu bedienender „R Markdown“-Syntax schreiben.

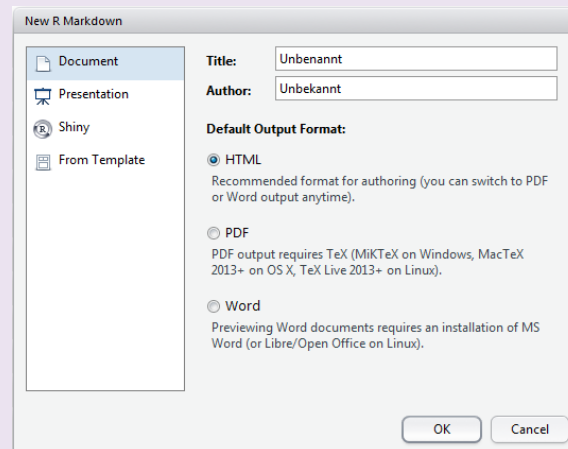
iii. **Einbetten** – R-Quellcode erstellen, der Ausgabewerte/Ergebnisse erzeugt die in das Protokoll eingebettet werden.

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 Optionsfeld für die Dateiart anklicken (kann nachträglich geändert werden).
- OK klicken.



3. Markdown schreiben Anschließend wird das Protokoll in Klartext geschrieben. Zur Formattierung wird Markdown-Syntax verwendet.

Syntax

Klartext

Zeile mit 2 Leerzeichen beenden um neuen Absatz zu beginnen.

kursiv und _kursiv_

fett und __fett__

hochgestellt^2^

~~durchgestrichen~~

[Hyperlink](www.rstudio.com)

Überschrift 1

Überschrift 2

Überschrift 3

Überschrift 4

Überschrift 5

Überschrift 6

Bindestrich (engl. endash): --

Gedankenstrich (engl. emdash): ---

Auslassungspunkte (engl. ellipsis): ...

einzeilige Formel: $A = r^2 * \pi$

Bild:

waagrechtlicher Strich:

> Blockzitat

* ungeordnete Liste

* Punkt 2

+ Unterpunkt 1

+ Unterpunkt 2

1. geordnete Liste

2. Punkt 2

+ Unterpunkt 1

+ Unterpunkt 2

Tabellenüberschrift

Tabellenelement

Element 3

Zweite Überschrift

Element 2

Element 4

Resultat

Klartext

Zeile mit 2 Leerzeichen beenden um neuen Absatz zu beginnen.

kursiv und *kursiv*

fett und **fett**

hochgestellt²

~~durchgestrichen~~

[Hyperlink](#)

Überschrift 1

Überschrift 2

Überschrift 3

Überschrift 4

Überschrift 5

Überschrift 6

Bindestrich (engl. endash): –

Gedankenstrich (engl. emdash): —

Auslassungspunkte (engl. ellipsis): ...

einzeilige Formel: $A = r^2 * \pi$

Bild:

waagrechtlicher Strich:

Blockzitat

• ungeordnete Liste

• Punkt 2

◦ Unterpunkt 1

◦ Unterpunkt 2

1. geordnete Liste

2. Punkt 2

◦ Unterpunkt 1

◦ Unterpunkt 2

Tabellenüberschrift

Tabellenelement

Element 3

Zweite Überschrift

Element 2

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: "Unbenannt"
author: "Unbekannt"
output: html_document
---
```

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

Die RStudio-Vorlage schreibt automatisch die entsprechende YAML Kopfzeile.

Anhand dieser .Rmd Datei kriert R eine neue Datei für das Protokoll. Der Wert von **output** bestimmt die Dateiart (siehe auch 6.), z. B.

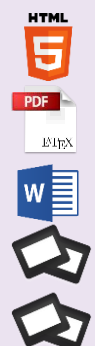
output: html_document HTML-Datei (Webseite)

output: pdf_document PDF-Datei

output: word_document Microsoft Word-Datei (DOCX)

output: beamer_presentation beamer Slideshow (PDF)

output: ioslides_presentation ioslides Slideshow (HTML)



5. Code einbetten

Die „knitr“-Syntax bettet R-Code in das Protokoll ein. R führt den Code aus sobald das Dokument gerendert wird.

Inline-Code

Quellcode wird mit einem Gravis (`) und einem r umschlossen. R wertet den Inline-Code aus und ersetzt ihn durch das Resultat.

Zwei mal drei macht ``r 2 * 3`` widewidewitt und drei macht neune.

➔ Zwei mal drei macht 6 widewidewitt und drei macht neune.

Code-Bereiche

Solch einen Bereich wird mit `````{r}```` gestartet und mit `````` beendet.

Hier ist Code
`````{r}```  
dim(iris)  
`````

Hier ist Code

`dim(iris)`

`## [1] 150 5`

Anzeigeoptionen

„knitr“-Optionen in den geschwungenen Klammern gestalten die Ausgabe eines Code-Bereiches, z. B. nur den Code anzeigen oder nur das Resultat anzeigen.

Hier ist Code
`````{r eval=FALSE}```  
dim(iris)  
`````

➔ Hier ist Code

`dim(iris)`

Hier ist Code
`````{r echo=FALSE}```  
dim(iris)  
`````

➔ Hier ist Code

`## [1] 150 5`

Option Standard Effekt

| | | |
|------------|----------|---|
| eval | TRUE | Code auswerten und Resultat anzeigen? |
| echo | TRUE | Code (nicht nur das Resultat) anzeigen? |
| warning | TRUE | Warnungen anzeigen? |
| error | FALSE | Fehler anzeigen? |
| message | TRUE | Meldungen anzeigen? |
| tidy | FALSE | Code für die Anzeige neu formatieren? |
| results | "markup" | "markup", "asis", "hold", oder "hide" |
| cache | FALSE | Resultate für zukünftiges Rendern cachen/zwischenspeichern? |
| comment | "###" | Symbol(e) am Zeilenbeginn für die Anzeige von Resultaten |
| fig.width | 7 | Breite in Zoll für Grafiken aus Code-Bereichen |
| fig.height | 7 | Höhe in Zoll für Grafiken aus Code-Bereichen |

6. Dokument rendern

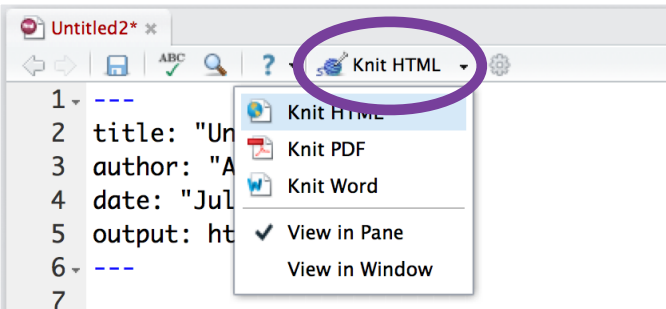
Die .Rmd Datei wird als Bauplan zur Erstellung des endgültigen Protokolls verwendet.

Zwei Möglichkeiten das Dokument zu rendern:

- `rmarkdown::render("<file path>")` ausführen.
- Die Schaltfläche „**Knit HTML**“ in der Kopfleiste anklicken.

Während des Renderns wird R

- jeden eingebetteten Code-Bereich ausführen und die Resultate in das Protokoll einfügen
- eine neue Version des Protokolls im eingestellten Ausgabeformat speichern
- eine Vorschau des Protokolls anzeigen
- das Protokoll im Arbeitsverzeichnis speichern.



7. Dokument interaktiv gestalten

Das Protokoll kann in 3 Schritten in eine interaktive „Shiny Anwendung“ (engl. Shiny app) umgewandelt werden.

1 **runtime: shiny** zur YAML Kopfzeile hinzufügen

```
---
title: "Kurvendiagramm"
output: html_document
runtime: shiny
---
```

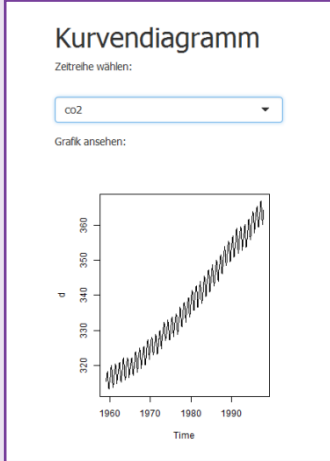
2 In den Code-Bereichen: Shiny **input** Funktionen einfügen um Widgets (Komponenten der Benutzeroberfläche) einzubinden, Shiny **render** Funktionen einfügen um reaktive Ausgabewerte/Ergebnisse zu erzeugen

```
---
title: "Kurvendiagramm"
output: html_document
runtime: shiny
---

Zeitreihe wählen:
`r echo = FALSE`
selectInput("data", "",
  c("co2", "lh"))

Grafik ansehen:
`r echo = FALSE`
renderPlot({
  d <- get(input$data)
  plot(d)
})
```

3 Mittels **rmarkdown::run** rendern oder mittels Klick auf **Run Document** in Rstudio rendern



* Hinweis: das Protokoll wird als „Shiny app“ generiert, d. h. als „output“ muss html gewählt werden, wie z. B. **html_document** (interaktive Webseite) oder **ioslides_presentation** (interaktive HTML-Slideshow).

8. Protokoll publizieren

Das resultierende Protokoll kann online freigegeben und geteilt werden.

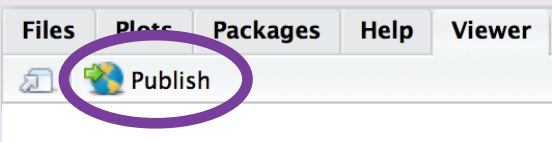
Rpubs.com

Nicht-interaktive Dokumente auf RStudio’s kostenloser „R Markdown“ Webseite www.rpubs.com teilen

ShinyApps.io

Interaktive Dokumente auf RStudio’s Server hosten. Kostenlose und bezahlte Optionen auf www.shinyapps.io

Mit einem einzelnen Klick auf die Schaltfläche „**Publish**“ im RStudio Vorschau-Fenster (engl. preview window) kann das Protokoll auf rpubs.com publiziert werden.



9. Mehr Informationen

Dokumentation und Beispiele - rmarkdown.rstudio.com
 Weitere Artikel - shiny.rstudio.com/articles
 Blog - blog.rstudio.com
 Twitter - @rstudio



RStudio® und Shiny™ sind eingetragene Markenzeichen von RStudio, Inc.
 Übersetzt von Lucia Gjeltema • rpanalysts.org
 CC BY RStudio • info@rstudio.com
 844-448-1212 • rstudio.com