

Einführung in R und RStudio

[Termine 7 & 8]

Miguel Alvarez

10. & 15. November 2022

▶ Termin 1 & 2

- ▶ Grundlagen
- ▶ Datentypen

▶ Termin 3 & 4

- ▶ Objekten
- ▶ Lesen und Schreiben

▶ Termin 5 & 6

- ▶ Statistiken
- ▶ Grafiken

▶ Termin 7 & 8

- ▶ Fortgeschrittenes Programmieren
- ▶ Erstellen von Dokumenten
- ▶ Abschluss

- ▶ Iterationen (Schleifen)
- ▶ Funktionen
- ▶ Quellcode
- ▶ Pakete

```
for (i in 1:5)  
  print(paste("Da wert von i ist", i))
```

```
## [1] "Da wert von i ist 1"  
## [1] "Da wert von i ist 2"  
## [1] "Da wert von i ist 3"  
## [1] "Da wert von i ist 4"  
## [1] "Da wert von i ist 5"
```

```
add_one <- function(x) {  
  return(x + 1)  
}
```

```
add_one(c(1:5))  
## [1] 2 3 4 5 6
```

Iterationen (Schleifen)

```
repeat {Ausdruck}  
while (Bedingung) {Ausdruck}  
for (i in M) {Ausdruck}  
  
next  
break
```

```
for (i in names(Bonn)) {  
  if (is.numeric(Bonn[[i]])) {  
    print(paste(i, mean(Bonn[[i]])))  
  }  
}
```

```
## [1] "BezirkNr 250.258064516129"  
## [1] "Jahr 2020"  
## [1] "Gesamt 5389.98924731183"  
## [1] "DichteKm2 4162.48924731183"  
## [1] "Maenner 2606.31182795699"  
## [1] "Frauen 2783.60752688172"  
## [1] "Zuwanderer 1627.06989247312"  
## [1] "Auslaender 947.693548387097"  
## [1] "FlaecheKm2 2.1936729876851"  
## [1] "X 2578524.77378713"  
## [1] "Y 5620621.4448268"
```

Funktionen

```
foo <- function(Argumente) {Ausdruck}
```

- ▶ Definition
- ▶ Parameter (Argumente)
- ▶ Argumente mit Voreinstellungen
- ▶ ...

```
mean_sd <- function(x) {  
  y <- c(mean(x), sd(x))  
  names(y) <- c("Mittelwert",  
               "Standardabweichung")  
  return(y)  
}
```

```
mean_sd(Bonn$Gesamt)
```

```
##           Mittelwert Standardabweichung  
##           5389.989           2491.407
```

Quellcode

```
source("Pfad")
```

```
source("MeineFunktionen.R", echo = TRUE)
```

```
##
```

```
## > BonnStats <- function(x, ...) {
```

```
## +   if (!is.numeric(x))
```

```
## +       stop("Der Wert 'x' muss numerisch sein
```

```
## +   out <- c(mean(x, ...), sd(x, ...))
```

```
## +   n .... [TRUNCATED]
```

Literarisches Programmieren Paradigma

Reproduzierbare Datenanalyse

- ▶ Sweave
- ▶ knitr
- ▶ Rmarkdown
- ▶ roxygen

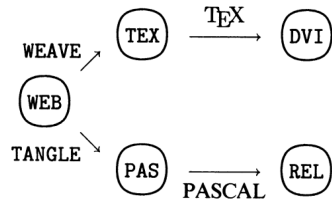


Figure 1. Dual usage of a WEB file.

Knuth (1984)

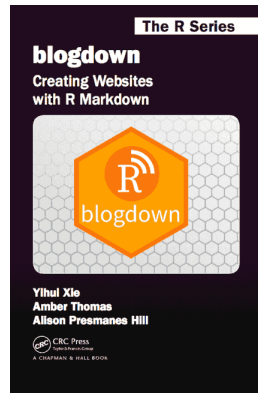
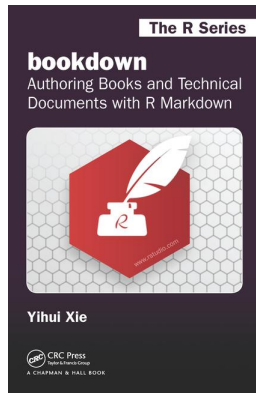
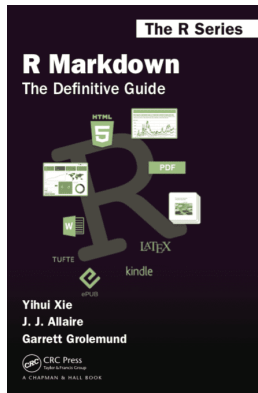
Datei .Rmd

- ▶ Kopf: yaml
- ▶ Inhalt: markdown
- ▶ Code Chunks (Bausteine):
 - ▶ R
 - ▶ SQL
 - ▶ usw.
- ▶ Ausgangsformate
 - ▶ PDF
 - ▶ html
 - ▶ docx (MS Word)
 - ▶ odt (LibreOffice Write)
 - ▶ pptx (MS Power Point)

```

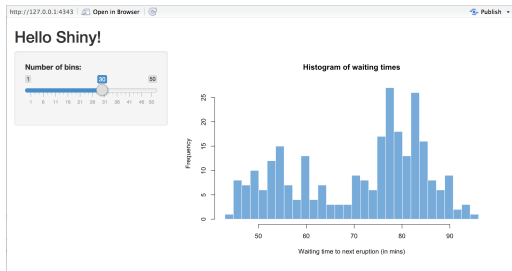
1 ---
2 title: "Untitled"
3 author: "Ich"
4 date: "`r Sys.Date()'"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for
15 authoring HTML, PDF, and MS Word documents. For more details on using
16 R Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the Knit button a document will be generated that includes
19 both content as well as the output of any embedded R code chunks within the
20 document. You can embed an R code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
25
26 ## Including Plots
27
28 You can also embed plots, for example:
29
30 ```{r pressure, echo=FALSE}
31 plot(pressure)
32 ```
33

```

Weitere Themen

- ▶ Interaktive Dokumente
 - ▶ *shiny* Apps
 - ▶ *leaflet* widgets
- ▶ Räumliche Daten (GIS)
- ▶ Text Analyse
- ▶ Bild Animation
- ▶ R Pakete
- ▶ Internet Seiten





Vielen Dank!

```
library(fortunes)
fortune(43)
```

```
##
## My preference goes with the numbering scheme attributed to a tribe on some
## island in the Pacific which consists of a 'factor' with four levels: 'one',
## 'two', 'three', and 'lots'. Hence, I'd go with 'lots of R users'.
##   -- Dirk Eddelbuettel (in a discussion about trying to estimate the number of
##       R users)
##       R-help (April 2004)
```