

# RMarkdown und R Basics

# Reproduzierbare Datenanalyse

# Reproduzierbare Datenanalyse

## Was bedeutet Reproduzierbarkeit?

Kurzfristig:

- + Können die Tabelle und Schaubilder aus dem zur Verfügung stehenden Code und den Daten erstellt werden?
- + Wird klar beschrieben was und **warum** etwas gemacht wurde?
- + Sind alle Schritte der Analyse nachvollziehbar dokumentiert?

Langfristig:

- + Können Elemente des Codes für andere Projekte wiederverwendet werden?

# Programme in diesem Kurs

Quelle: [datasciencebox.org](https://datasciencebox.org).

- + Programmiersprache -> R und RStudio
- + "Literate Programmierung" (alles in einem Ort, d.h. Code, Text und Output) -> RMarkdown
- + Versionierung -> Git/Github

# R und RStudio

# Was ist R und RStudio?

- + R ist eine Programmiersprache
- + RStudio ist ein Interface für R

Wie können Sie sich das vorstellen?

**R: Engine**



**RStudio: Dashboard**



# Lernen Sie R und RStudio kennen

Erstes Problem Set:

- + R als Taschenrechner
- + Kleine Grafiken erzeugen
- + Datensätze in R einlesen

# tidyverse



## tidyverse

- + Zusammenstellung verschiedener Pakete zur Datenanalyse
- + Alle Pakete verbindet eine gemeinsame Philosophie und Struktur
- + Hauptautor: Hadley Wickham

Markdown

# Einführung

- + Sehr einfache Syntax ohne komplexe Formatierung
- + Sie können sich voll auf das Schreiben konzentrieren
- + Plattformunabhängig (Kann zwischen verschiedenen Geräten geteilt werden)
- + Besonders gut für Readmes, Tutorials, Reports, deskriptive Analysen, Blogs, Journal Artikels ...
- + Einfache Möglichkeit PDFs, Word-Dateien oder HTMLs zu erstellen
  - + PDFs können Sie mit dem Paket `pandoc` erzeugen, vorausgesetzt sie haben Latex installiert

# Überschriften

Es sind bis zu sechs Gliederungsebenen in RMarkdown möglich:

- + Überschrift 1 wird so erreicht: `# Überschrift 1`
- + Überschrift 4 wird so erreicht: `### Überschrift 4`

Durch das voranstellen eines weiteren Hashtags (#) gelangen Sie jeweils eine Gliederungsebene tiefer

# Links

In Markdown können Sie auch Links zu externen Dokumenten setzen:

## + Möglichkeit 1: Lokale Links

- + Zu Verlinkender Text in eckige Klammern gesetzt (`[ ]`) und der Link danach in runde Klammern (`( ( ) )`)

- + Beispiel (`[Beispiel] (https://www.markdowntutorial.com/)`).

## + Möglichkeit 2: Globale Links

- + Es gibt auch die Möglichkeit Links global zu setzen

- + Markdown Tutorial (`[Markdown Tutorial] [Tutorial]`).

- + Später im Text, oder am Ende:

  - `[Tutorial]: https://www.markdowntutorial.com/`

- + Vorteilhaft bei mehrmaligem Verwenden des Links

# Bilder

- + Funktioniert ähnlich wie Links
- + Bildunterschrift in eckigen Klammern, Link in runden Klammern

- + Beispiel:

```
![RMarkdown Logo] (https://www.rstudio.com/wp-content/uploads/2014/04/rmarkdown-400x464.png)
```

# Bilder

+ Ausgeführt sieht dies dann folgendermaßen aus:  RMarkdown Logo

# Formeln

- + Möglichkeit Formeln in Latex zu setzen
- + Inline Formel möglich:

$$R = \alpha + \beta * \pi^2 + \epsilon$$

(\$R = \alpha + \beta \* \pi^2 + \epsilon\$)

- + Oder in einer Formelumgebung:

```
\begin{equation}
\mathbb{E}[Y] = \beta_0 + \beta_1x
\end{equation}
```

$$\mathbb{E}[Y] = \beta_0 + \beta_1x$$

# R Markdown

# Einführung

- + R Markdown ist eine Erweiterung von Markdown mit sogenannten `Chunks`
- + R Code kann direkt in R Markdown ausgeführt werden
- + Resultate aus R werden direkt in das Markdown-Dokument eingefügt
- + Einfaches Erstellen von HTML-Seiten mit integrierten Tabellen, Grafiken, Code
- + Mit `knitr` können R Markdown Dateien kompiliert und in *normale* Markdown Dateien umgewandelt werden
- + Hilfe zu R Markdown gibt es unter `?rmarkdown`
- + Das R Markdown [Cheatsheet](#) kann oft sehr hilfreich sein

# knitr

Um `knitr` zu verwenden klicken Sie den folgenden Button in RStudio:

 Dieses Bild zeigt wo der Knit-button zu finden ist

# Einbetten von Code

Es gibt drei Arten, wie Sie ihren Code in R Markdown Dokumenten so verpacken, dass er beim "knitten" auch verarbeitet wird.

- + Fassen Sie den Code in Blöcke: Geben Sie `` `` {r}` beim Start des Blocks und wenn der Block zu Ende ist `` ``` ein
- + Benutzen Sie die Tastaturkombination **Strg + Alt + I** (OS X: **Cmd + Option + I**)
- + Gehen Sie auf "Code" -> "Insert Chunk" in der Funktionsleiste

# Einbetten von Code

- + Chunks sind eingebettete Code-Blöcke in R Markdown
- + Auf der folgenden Folie wird die Funktionsweise von Chunks am Beispieldatensatz `economics` aus dem `tidyverse` Paket demonstriert
  - + Der Datensatz `economics` beinhaltet Daten zur Arbeitslosigkeit in den USA seit 1967
- + Im folgenden Beispiel wollen wir zuerst das Paket `tidyverse` laden und anschließend deskriptive Analyse mit zwei verschiedenen Befehlen durchführen
- + Durch das `knitten` in HTML wird sowohl der Code, als auch dessen Output angezeigt

# Einbetten von Code

```
library(tidyverse)
summary(economics)
```

```
      date          pce          pop          psavert
Min.   :1967-07-01  Min.   : 506.7  Min.   :198712  Min.   : 2.200
1st Qu.:1979-06-08  1st Qu.: 1578.3  1st Qu.:224896  1st Qu.: 6.400
Median :1991-05-16  Median : 3936.8  Median :253060  Median : 8.400
Mean   :1991-05-17  Mean   : 4820.1  Mean   :257160  Mean   : 8.567
3rd Qu.:2003-04-23  3rd Qu.: 7626.3  3rd Qu.:290291  3rd Qu.:11.100
Max.   :2015-04-01  Max.   :12193.8  Max.   :320402  Max.   :17.300

      uempmed          unemploy
Min.   : 4.000  Min.   : 2685
1st Qu.: 6.000  1st Qu.: 6284
Median : 7.500  Median : 7494
Mean   : 8.609  Mean   : 7771
3rd Qu.: 9.100  3rd Qu.: 8686
Max.   :25.200  Max.   :15352
```

# Einbetten von Code

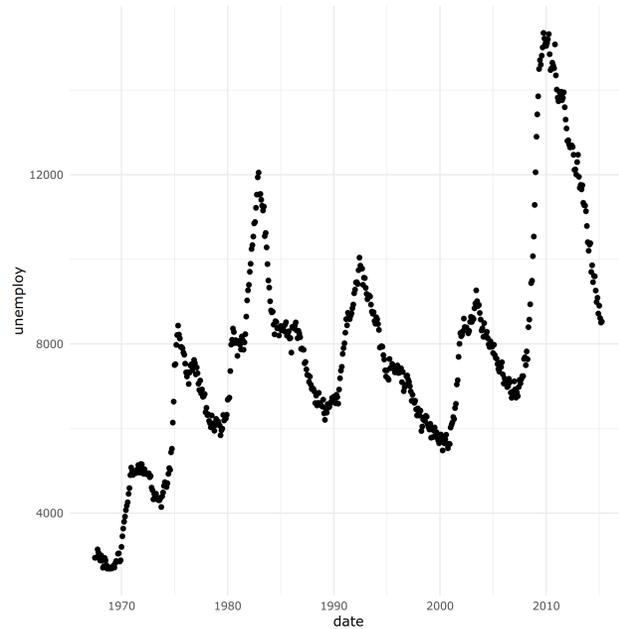
```
glimpse(economics)
```

```
Rows: 574  
Columns: 6  
$ date      <date> 1967-07-01, 1967-08-01, 1967-09-01, 1967-10-01, 1967-11-01, ...  
$ pce       <dbl> 506.7, 509.8, 515.6, 512.2, 517.4, 525.1, 530.9, 533.6, 544.3...  
$ pop       <dbl> 198712, 198911, 199113, 199311, 199498, 199657, 199808, 19992...  
$ psavert   <dbl> 12.6, 12.6, 11.9, 12.9, 12.8, 11.8, 11.7, 12.3, 11.7, 12.3, 1...  
$ uempmed   <dbl> 4.5, 4.7, 4.6, 4.9, 4.7, 4.8, 5.1, 4.5, 4.1, 4.6, 4.4, 4.4, 4...  
$ unemploy  <dbl> 2944, 2945, 2958, 3143, 3066, 3018, 2878, 3001, 2877, 2709, 2...
```

# Schaubilder

- + Sie können auch Schaubilder direkt in R Markdown erstellen lassen und einbinden
- + Beispiel: Scatter-Plot der Anzahl der Arbeitslosen in den USA seit 1967

```
qplot(date, unemploy, data=economics)
```



# Schaubilder

**Aufgabe:** Lesen Sie die Dokumentation des Economics Datensatzes mittels `?economics` und erstellen Sie einen Scatter-Plot, welcher das Datum auf der x-Achse und die *Sparquote* auf der y-Achse darstellt.

# Tabellen

- + Tabellen können Sie in Markdown durch den Spaltentrenner | und den Zeilentrenner - - - erstellen.
- + Linksbündig ausgerichtet
- + Durch Doppelpunkte auch mittige oder rechte Ausrichtung möglich

Hier ein Beispiel:

```
A | B | C
---:|---|---
1 | 2 | 3
1 | 2 | 3
1 | 2 | 3
```

Wird in Markdown wie folgt dargestellt:

A	B	C
1	2	3
1	2	3
1	2	3

# Tabellen mit Pander

- ✚ In R erstellte Tabellen durch Paket `pander()` direkt darstellen
- ✚ R Chunk zusätzlich den Parameter `results='asis'` übergeben (`````{r, result='asis'}````), damit es korrekt interpretiert wird
- ✚ Pander noch informieren, dass es sich um ein R Markdown Dokument handelt mit dem Parameterstil (`style="rmarkdown"`)
- ✚ Beispiel: Ersten 5 Zeilen für die ersten 4 Spalten aus dem `economics` Datensatz als Markdown Tabelle ausgegeben:

```
#install.packages("pander")
library(pander)
library(tidyverse)
pander(economics[1:5, 1:4], style = "rmarkdown")
```

date	pce	pop	psavert
1967-07-01	506.7	198712	12.6
1967-08-01	509.8	198911	12.6
1967-09-01	515.6	199113	11.9
1967-10-01	512.2	199311	12.9
1967-11-01	517.4	199498	12.8

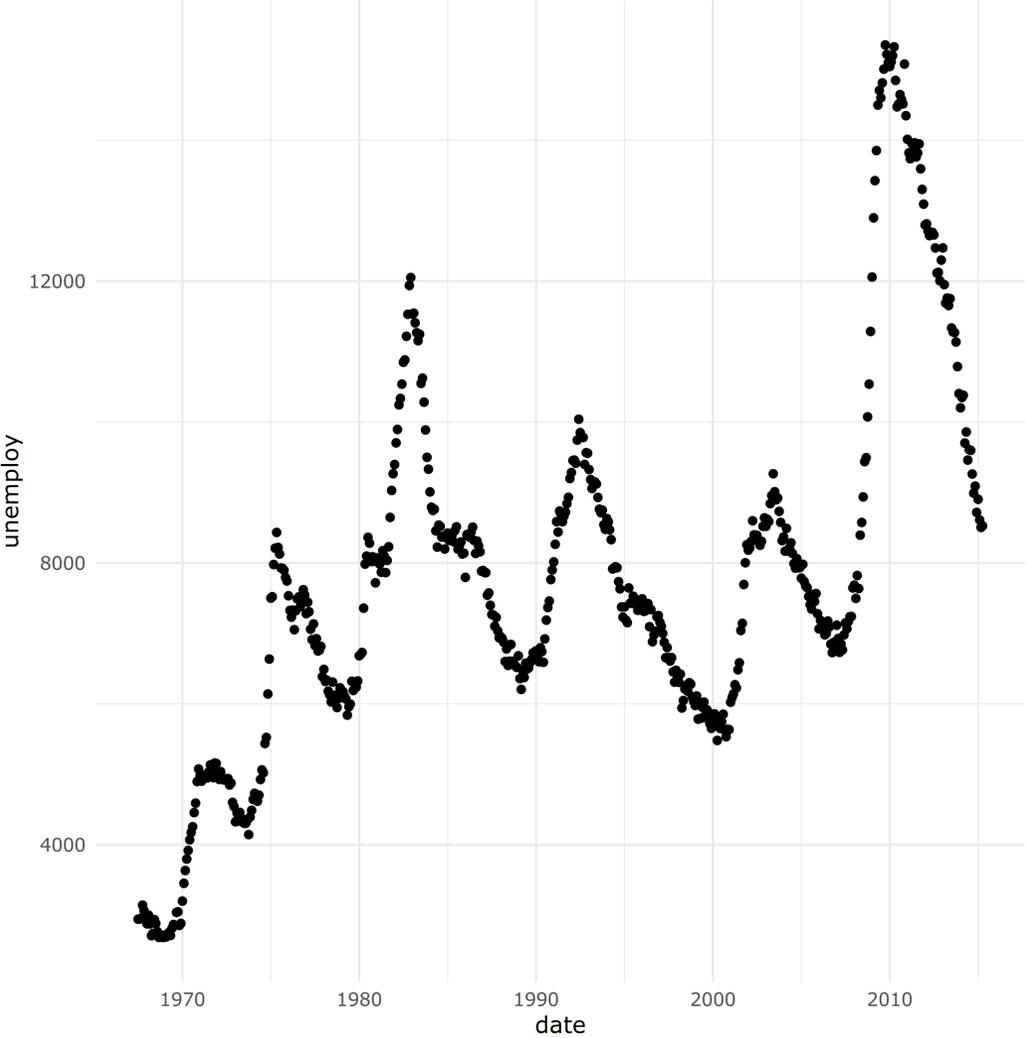
# Der Cache

- + Bei großen Dokumenten kann das "knitten" sehr lange dauern
- + Möglichkeit Ergebnisse zu `cache`n, d.h. Ergebnisse zwischenspeichern
- + Option `cache = TRUE` nach der Einführung des Code Blocks ````{r, cache = TRUE}`
- + Wenn sich in den Chunks mit den *gecachten* Informationen jedoch etwas ändert muss die Option `cache=TRUE` entfernt werden, ansonsten werden die Änderungen nicht in ihr Dokument aufgenommen

# Anzeigen von Chunks

- ✚ Nicht immer wünschenswert, dass der Code-Chunk mit angezeigt wird
- ✚ Beispielsweise sollen Sie in ihren Projektarbeiten die Chunks immer ausblenden und nur die Ergebnisse zeigen
- ✚ Ausschalten der Option durch `echo=FALSE` möglich (````{r, echo=FALSE}`)
- ✚ Beispielsweise unser Scatter-Plot von vorhin:

# Anzeigen von Chunks



# Anzeigen von Chunks

- ✚ Soll nur der Code Chunk angezeigt werden, jedoch kein Output, dann müssen Sie ein `eval=FALSE` voranstellen  
(```{r,eval=FALSE})

```
qplot(date, unemploy, data=economics)
```

# Typische Optionen

Im Chunk haben Sie mehrere Optionen, wie dieser von R interpretiert werden soll.

Output:

- + results: "asis"/"hide" (Output wie er vom Chunk kommt anzeigen/nicht zeigen)
- + echo: "TRUE"/"FALSE" (Code aus Chunk zeigen/nicht zeigen)
- + eval: "TRUE"/"FALSE" (Chunk nicht beachten/beachten)
- + include: "TRUE"/"FALSE" (Code Output zeigen/nicht zeigen)
- + message: "TRUE"/"FALSE" (Benachrichtigungen anzeigen/nicht anzeigen)
- + warnings: "TRUE"/"FALSE" (Warnmeldungen anzeigen/nicht anzeigen)
- + error: "TRUE"/"FALSE" (Fehlermeldungen anzeigen/nicht anzeigen)
- + cache: "TRUE"/"FALSE" (Zuvor gespeicherte Inhalte anzeigen/ Immer neu berechnen)

# Typische Optionen

Schaubilder:

- + `fig.height`: Nummer (Höhe eines Schaubilds festlegen)
- + `fig.width`: Nummer (Breite eines Schaubilds festlegen)
- + `out.width`: Nummer (Breite des Outputs, kann auch in '%' angegeben werden)

# Code extrahieren

Zusammenfassen des R Code einer `.Rmd` Datei möglich?

- + Der Code kann durch `purl()` aus einer `.Rmd`-Datei separat abgespeichert werden
- + Hier ein Beispiel mit allen Befehlen, welche bisher gebraucht wurden, abgespeichert in einem "Einfuehrung-in-RMarkdown.R" Dokument im aktuellen Arbeitsverzeichnis.

```
library(knitr)
purl(input = "v2_RMarkdown.Rmd", output="Einfuehrung-in-RMarkdown.R", documentation = 0)
```

# RMarkdown Hilfe

## RMarkdown cheat sheet

The image shows a screenshot of the RStudio RMarkdown Cheat Sheet. It is divided into several sections:

- What is markdown?**: Explains the purpose of RMarkdown, including creating files, dynamic documents, reproducible research, and workflow.
- Workflow**: A 5-step process: 1. Open a new Rmd file, 2. Embed code in chunks, 3. Write text and add tables, figures, images, and citations, 4. Set output format and options, 5. Save and render the whole document.
- Embed Code with knitr**: Includes sections for CODE CHUNKS (with options like echo, eval, include, warning, results, fig.align, fig.width, fig.height, fig.cap, fig.out.cap, fig.ext, fig.retina, fig.asp, fig.dpi, fig.align, fig.out.cap, fig.ext, fig.retina, fig.asp, fig.dpi), GLOBAL OPTIONS, and INLINE CODE.
- Insert Citations**: Details on creating citations, building a bibliography, and inserting citations into the document.
- Insert Tables**: Shows how to create tables using the `knitr::kable()` function.

## Markdown Guide

### Help -> Markdown Quick Reference

The image shows a screenshot of the 'Markdown Quick Reference' help page in RStudio. It provides a comprehensive list of Markdown syntax rules:

- Emphasis**: `*italic*`, `**bold**`, `_italic_`, `__bold__`
- Headers**: `# Header 1`, `## Header 2`, `### Header 3`
- Lists**: Unordered list (`* Item 1`), Ordered list (`1. Item 1`), and nested lists (`+ Item 2a`).
- Manual Line Breaks**: `End a line with two or more spaces: Roses are red, Violets are blue.`
- Links**: `Use a plain http address or add a link to a phrase: http://www.r-project.org`

# Wofür nutzen wir RMarkdown

- + Alle Vorlesungsfolien/`RTutor` Problem Sets/Projekte etc. sind in RMarkdown
- + Sie starten immer mit einem RMarkdown Template in ihre Projekte
- + Die Vorgaben in den Templates werden im Laufe des Semesters geringer
  - + `RTutor` Problem Sets ist noch sehr genau wie Sie zu einem Ergebnis kommen
  - + In den Projekten können Sie frei coden