

# Einführung in R

Grundlagen,

Datenanalyse,

Datenvisualisierung

---

# Einführung in R

## Grundlagen, Datenanalyse, Datenvisualisierung

Pekka Sagner M.Sc.



 @pekkato

 @pekkasagner

 sagner@iwkoeln.de

 iwkoeln.de/.../pekka-sagner

Wintersemester 2021/22

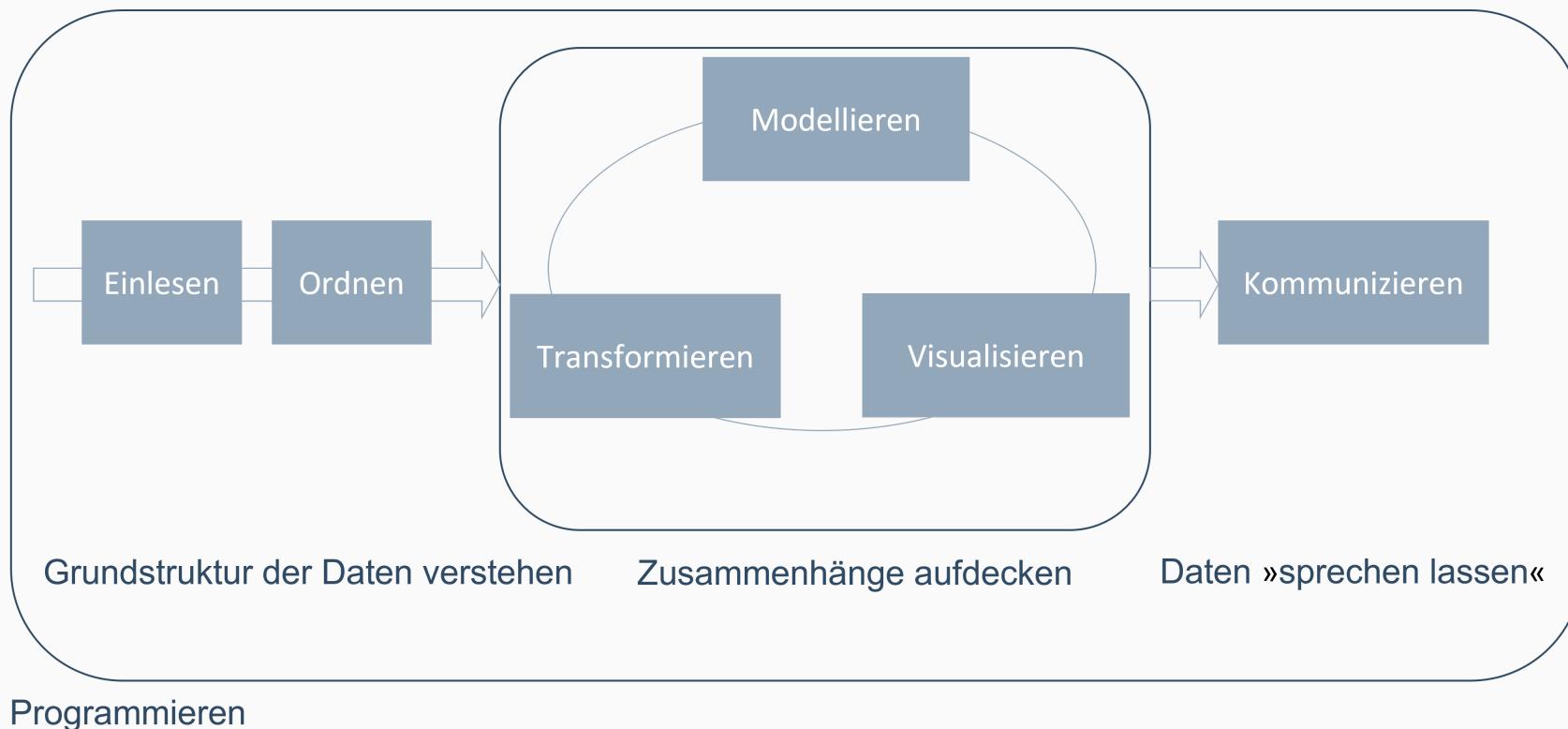
Link zu den Folien:

[pekkasagner.github.io/Einfuehrung\\_in\\_R](https://pekkasagner.github.io/Einfuehrung_in_R)

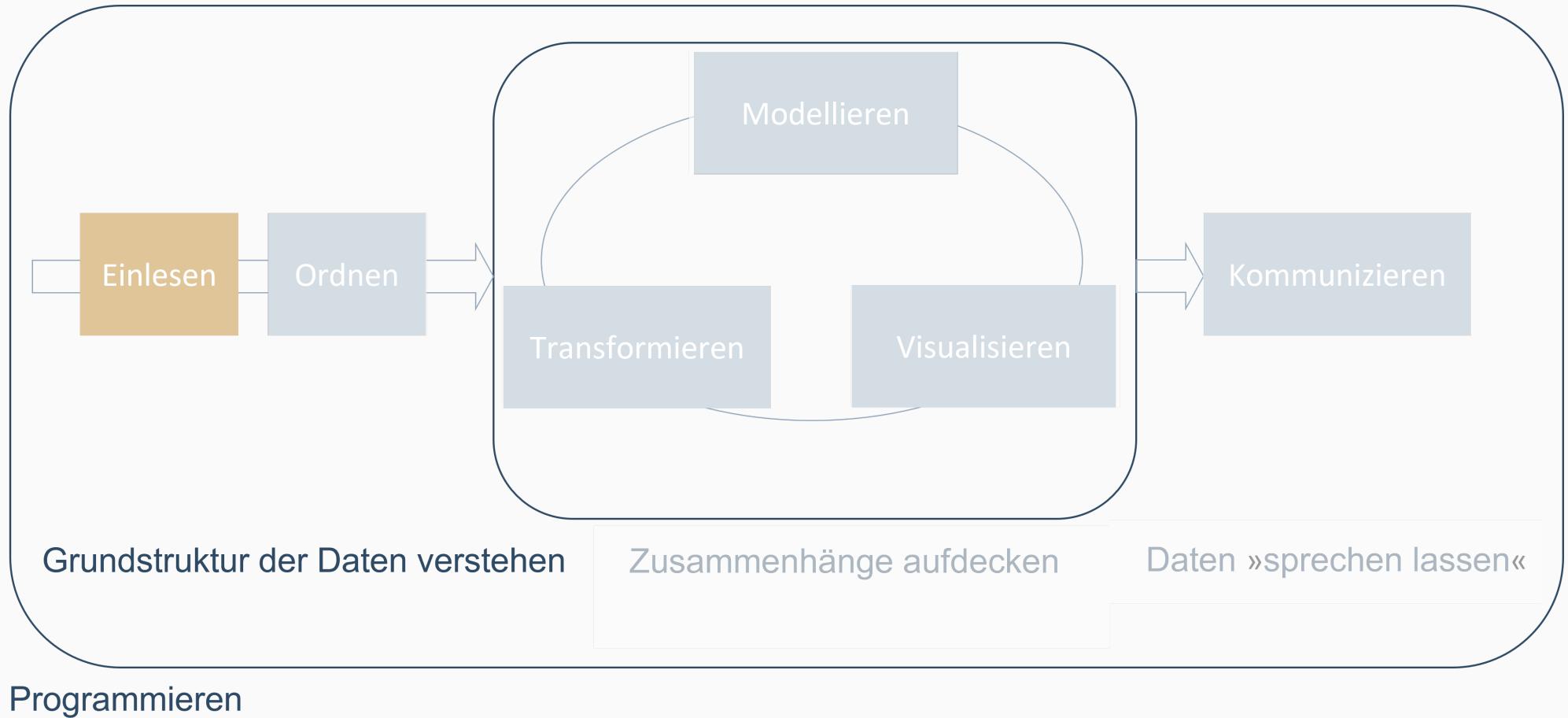
# Was wir im Kurs lernen

# Was wir im Kurs lernen

- Data Science ist ein riesiges Feld. Es ist unmöglich Experte auf jedem Gebiet zu sein.
- Ziel des Kurses ist es, ein solides Fundament zu schaffen, auf dem viele Aufgaben im Bereich Data Science aufbauen.
- Die Arbeit mit Daten ist im Grunde recht simpel aufgebaut:

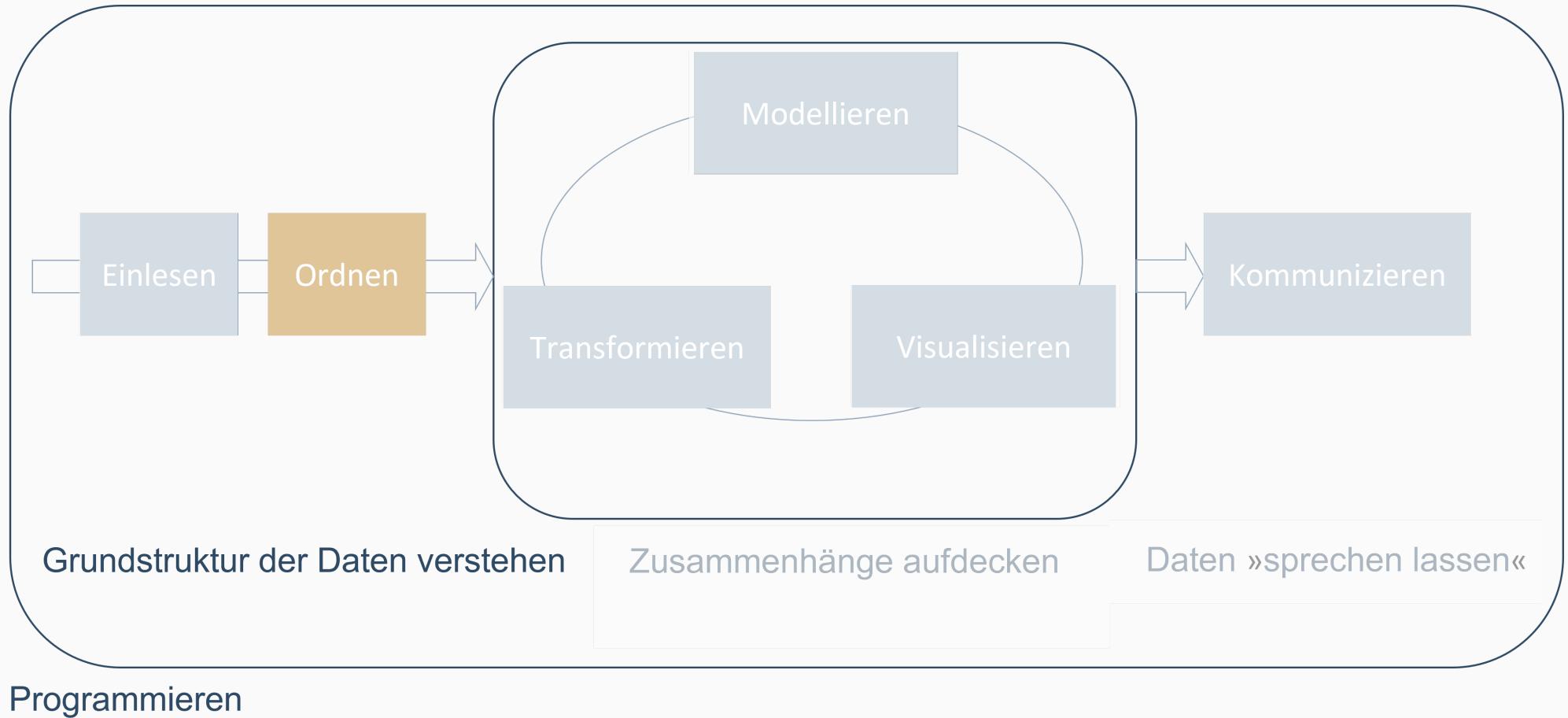


Quelle: R4DS, eigene Darstellung



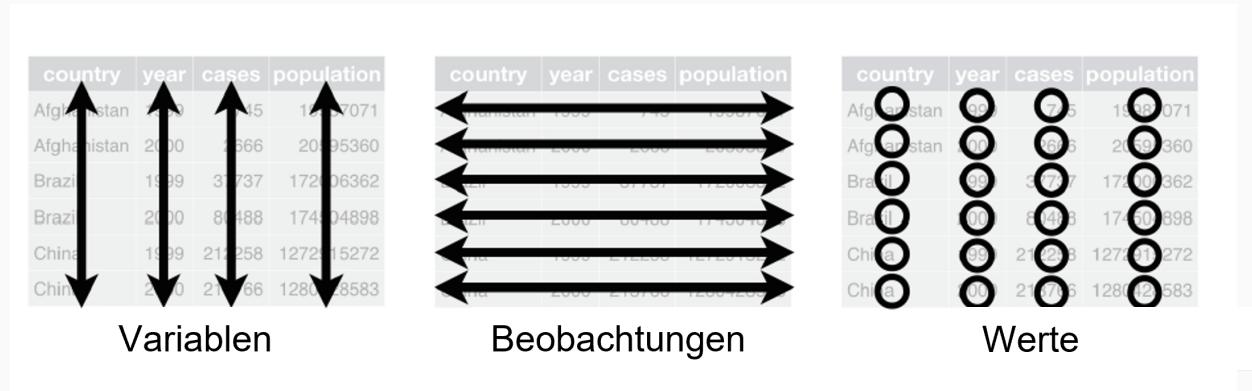
# Daten einlesen

- Typischerweise entweder aus einer Datei, Datenbank oder einer Programmierschnittstelle (API).
- Der erste Schritt ist der wichtigste: Wenn wir unsere Daten nicht in R einlesen können, können wir diese auch nicht analysieren, nicht visualisieren, nicht ...
- Leseempfehlung:
  - [The Tidyverse Cookbook, Kapitel 2 \(Grolemund, 2020\)](#)
  - [R4DS, Kapitel 11 \(Wickham, 2017\)](#)
- Ressourcen:
  - [Vortrag von Hadley Wickham \(Video\)](#) zum Einlesen verschiedener Datentypen.



# Daten ordnen

- **Ordentliche Daten** sind Daten, die in einem konsistenten Format vorliegen und zur Natur der Daten passen.
- Das Konzept der **Tidy Data** ist für die Arbeit mit Daten essenziell.
- Struktur »ordentlicher« Daten:

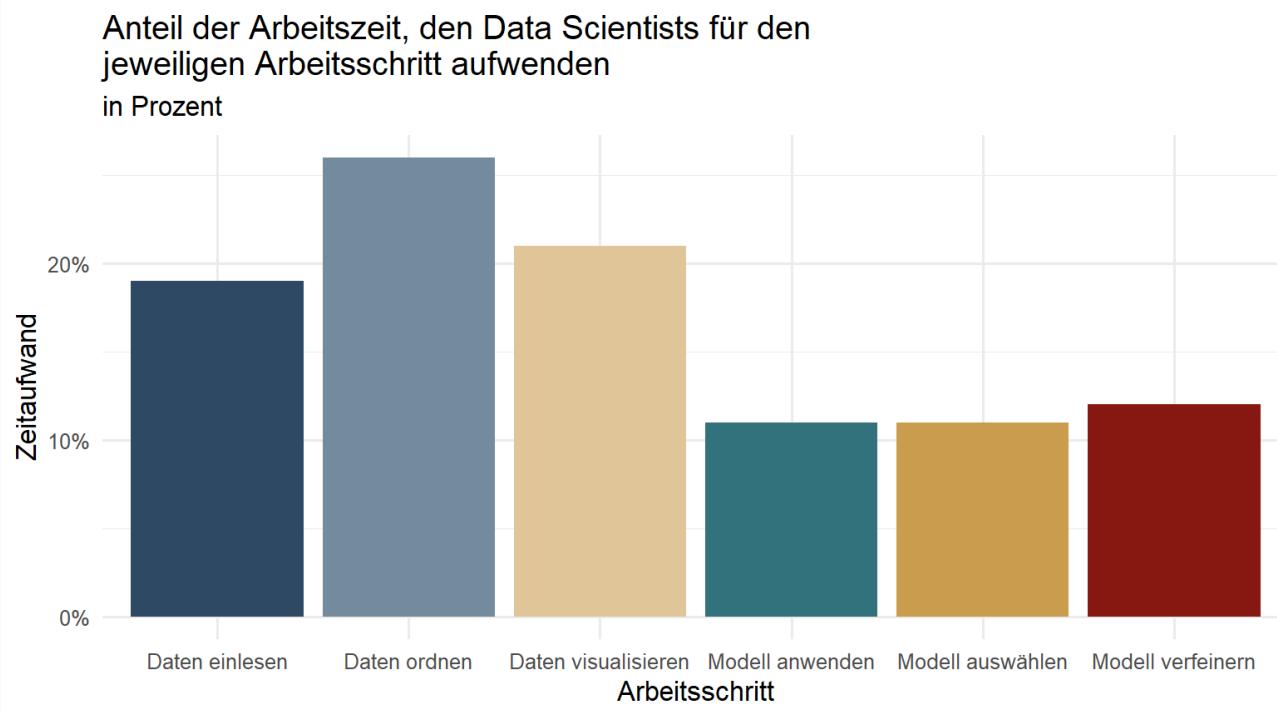


- Leseempfehlung: [Tidy Data \(Wickham, 2014\)](#)
- Die beiden ersten Schritte sind eng miteinander verknüpft.
  - Durch das richtige Einlesen der Daten kann man sich häufig viel Zeit beim Daten ordnen sparen.

# »Garbage in, garbage out«

- Der Zeitaufwand, den die ersten Schritte bei einem Datenprojekt in Anspruch nehmen, sollte nicht unterschätzt werden:

Plot R-Code



Quelle: [Anaconda.com](#), eigene Darstellung

# »Garbage in, garbage out«

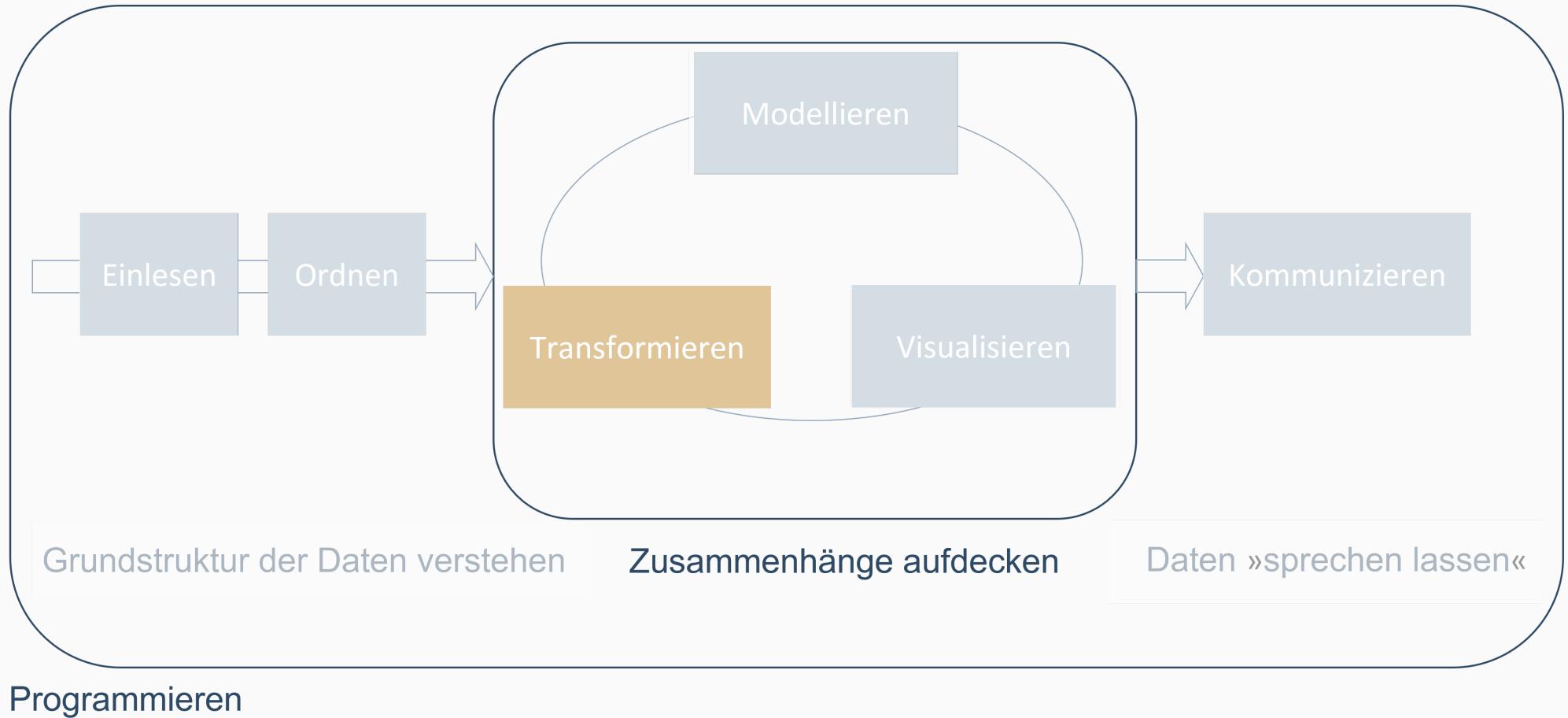
- Der Zeitaufwand, den die ersten Schritte bei einem Datenprojekt in Anspruch nehmen, sollte nicht unterschätzt werden:

Plot      R-Code

```
data <- tibble(Arbeitsschritt = c("Daten einlesen", "Daten ordnen", "Daten visualisieren",
                                 "Modell auswählen", "Modell verfeinern", "Modell anwenden"),
               Zeitaufwand = c(0.19, 0.26, 0.21, 0.11, 0.12, 0.11))

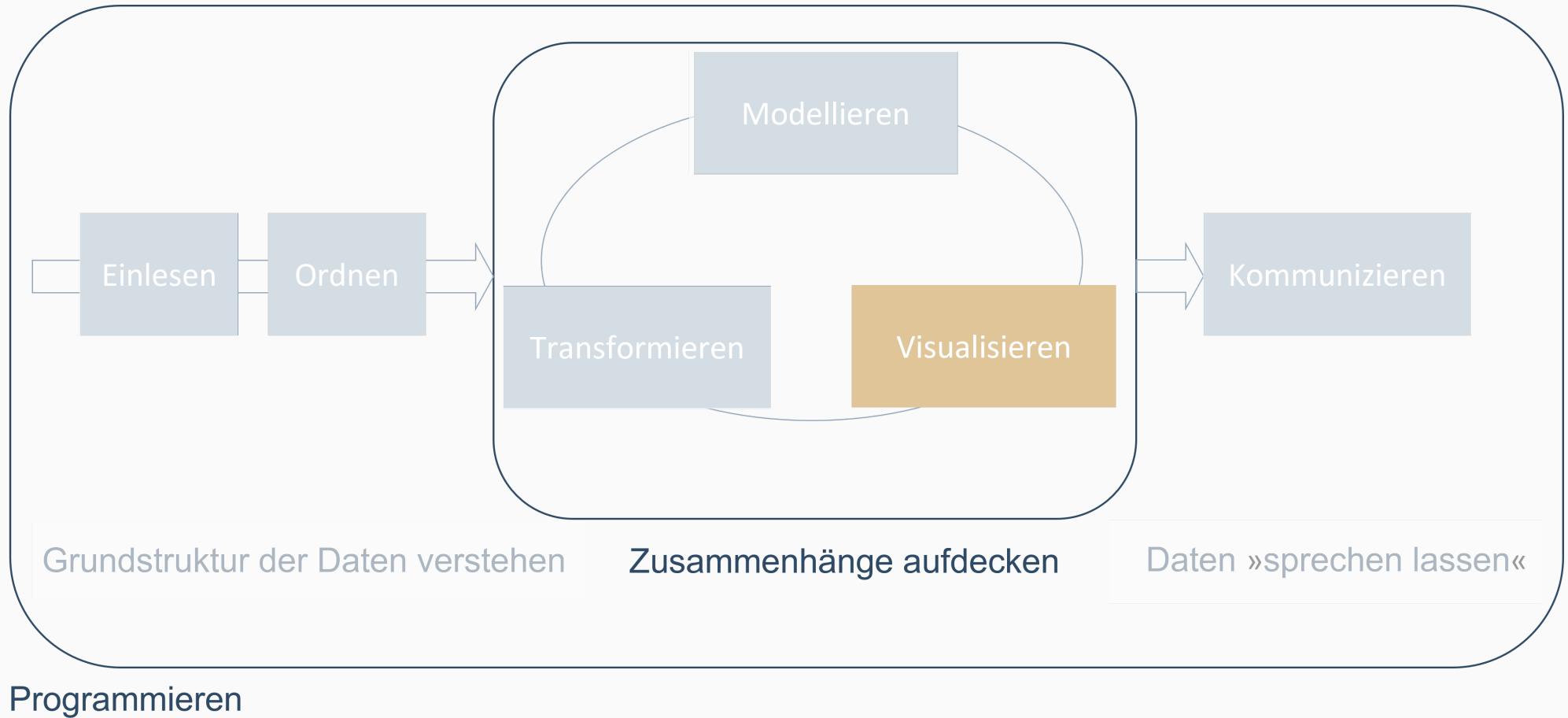
data %>%
  ggplot(aes(x = Arbeitsschritt, y = Zeitaufwand, fill = Arbeitsschritt)) +
  geom_col() +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal() +
  IWcolorRs::scale_fill_IW() +
  theme(legend.position = "none") +
  labs(title = "Anteil der Arbeitszeit, den Data Scientists für den\\njeweiligen Arbeitsschritt aufwenden",
       subtitle = "in Prozent")
```

Quelle: [Anaconda.com](#), eigene Darstellung



# Daten transformieren

- In den seltensten Fällen kommen wir ohne die Transformation von Daten aus. (Ein Beispiel ist die vorherige Abbildung.)
- In der Regel ist es unsere Aufgabe, aus Daten schlau zu werden.
- Daten können auf verschiedenste Weise transformiert werden:
  - **Fokussieren auf Beobachtungen von Interesse:** zum Beispiel Menschen, die in einem bestimmten Land wohnen oder Daten aus einem speziellen Jahr.
  - **Erstellen neuer Variablen aus bestehenden:** zum Beispiel BIP pro Kopf aus dem gesamten Bruttoinlandsprodukt eines Landes geteilt durch die Anzahl an Einwohnern.
  - **Berechnen zusammenfassender Statistiken:** zum Beispiel Durchschnitte, Medianwerte oder Anteile.
- Das ordnen und das Transformieren von Daten nennt man auch **Data Wrangling**.
- Dies ist eine Anspielung darauf, dass es ein Kampf sein kann, bis die Daten im »richtigen« Format sind.



# Daten visualisieren I/II

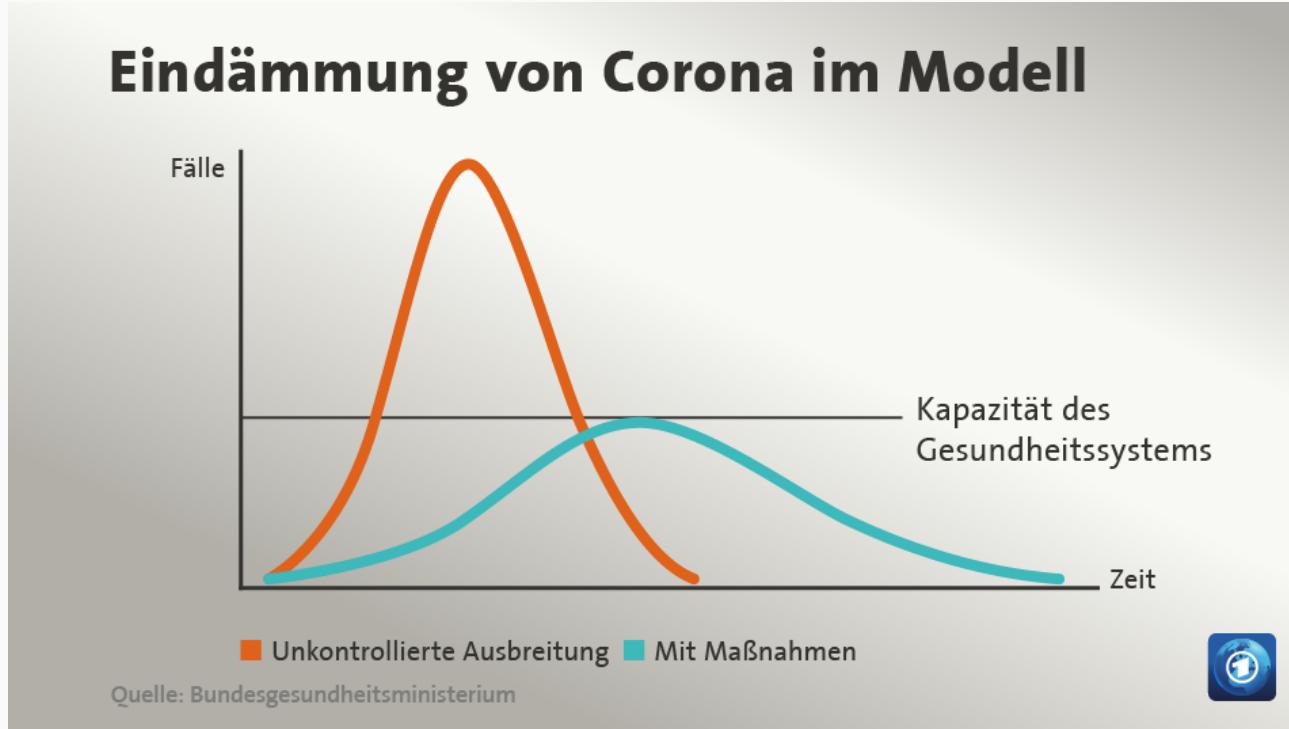
- Sind die Daten im richtigen Format, gibt es zwei wesentliche Vorgehensweisen, um daraus neues Wissen zu schaffen: **Visualisierung** und **Modellierung**. Die beiden Vorgehensweisen sind komplementär.
- »Ein Bild sagt mehr als tausend Worte.«
- Eine gute Visualisierung zeigt möglicherweise unerwartete Dinge.
- Visualisierungen sind auf dem Weg zum besseren Verständnis von Daten ein unerlässliches Instrument.
- Oft zeigen uns Abbildungen, dass »irgendetwas in den Daten nicht stimmt.«
- Sie sind deshalb ein mächtiges Instrument und sollten Teil der explorativen Datenanalyse sein.
- R ist wegen seiner guten Visualisierungsmöglichkeiten, die auf einer **Grammar of Graphics** basieren, sehr beliebt.
- Leseempfehlung:
  - R4DS, Kapitel 3 (Wickham, 2017)
  - A Layered Grammar of Graphics (Wickham, 2010)
  - ggplot2: Elegant Graphics for Data Analysis (Wickham, 2021)
  - R Graphics Coobook (Chang, 2021)

# Daten visualisieren II/II

Gute Visualisierung

Schlechte Visualisierung

Alternative zur schlechten Visualisierung

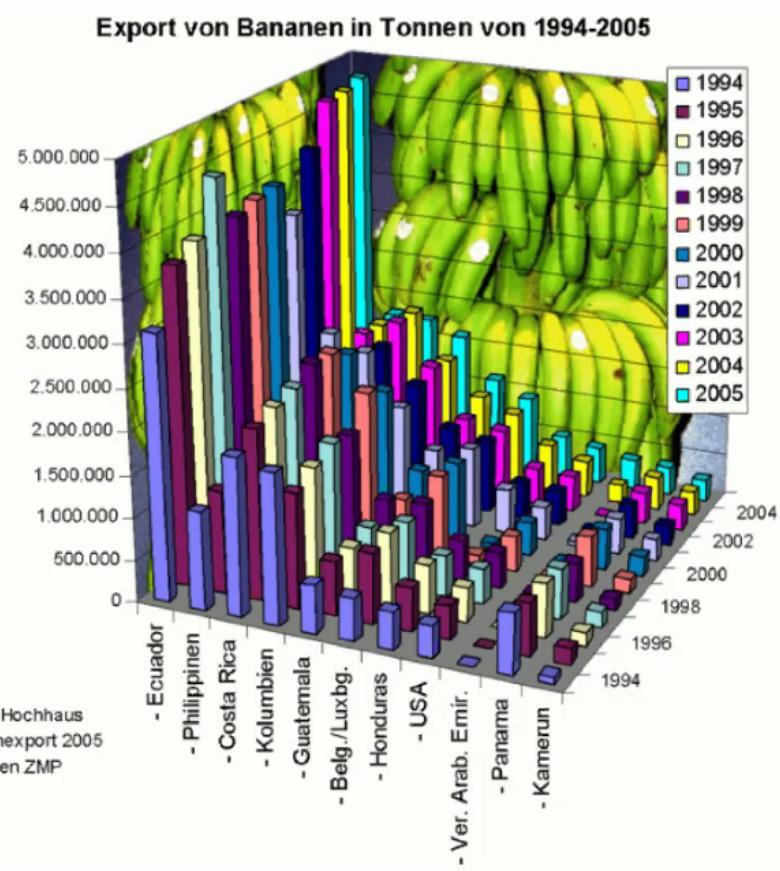


# Daten visualisieren II/II

Gute Visualisierung

Schlechte Visualisierung

Alternative zur schlechten Visualisierung

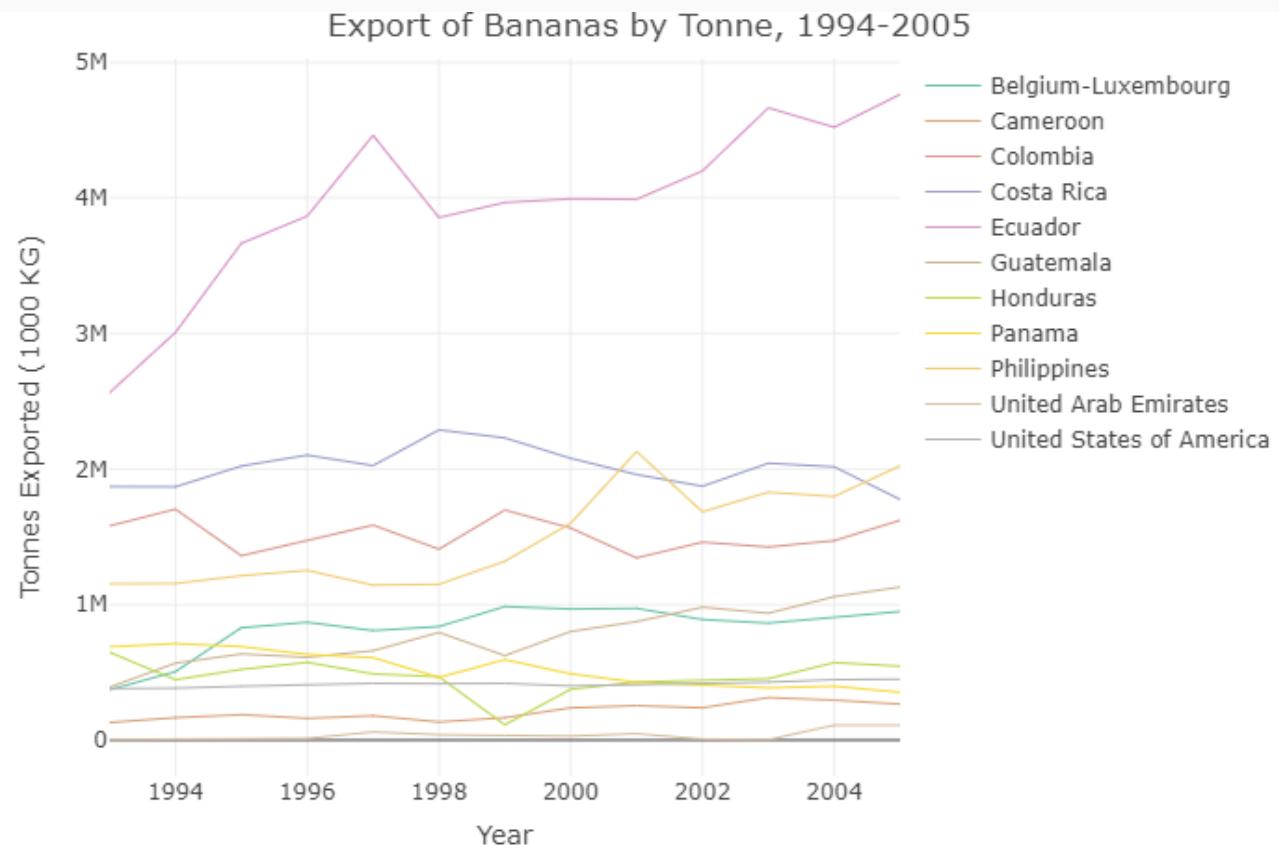


# Daten visualisieren II/II

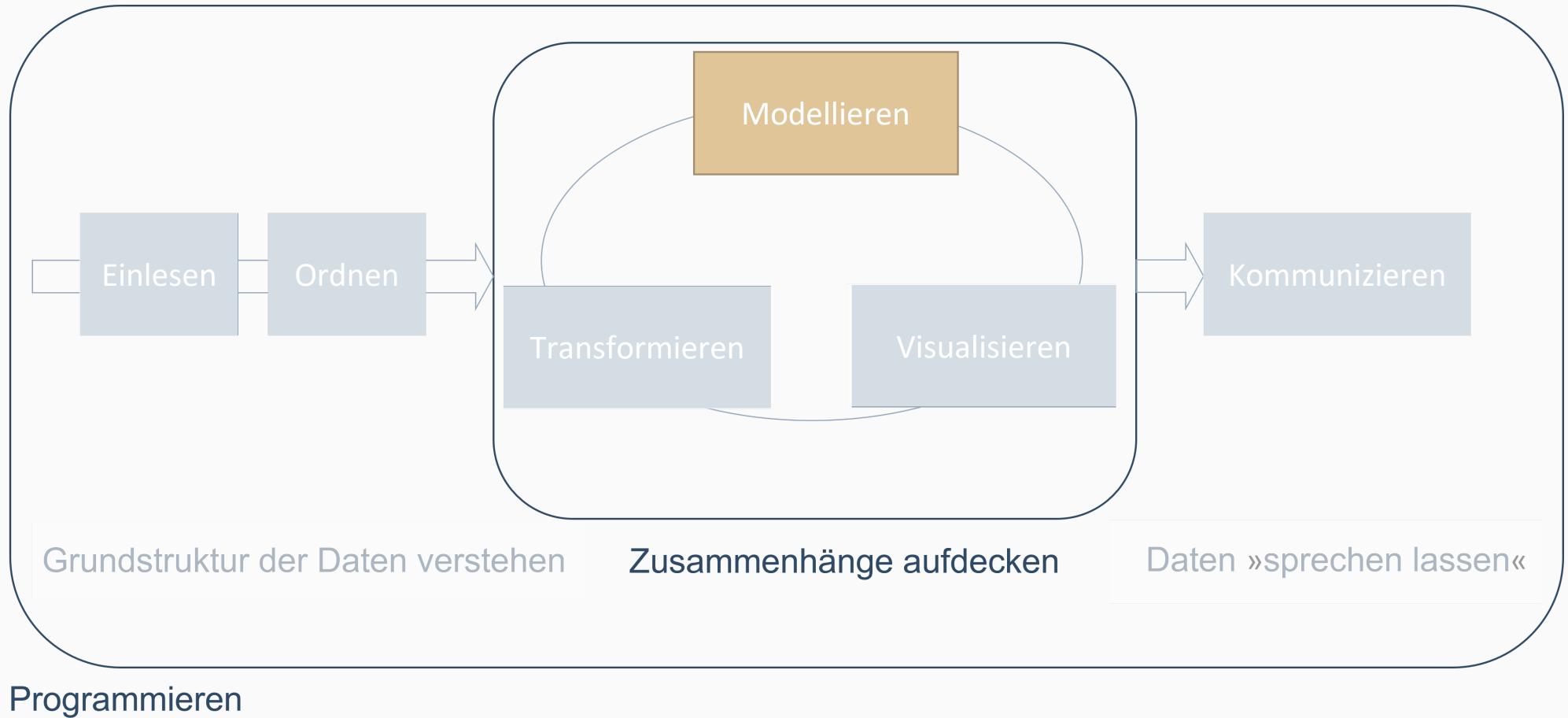
Gute Visualisierung

Schlechte Visualisierung

Alternative zur schlechten Visualisierung

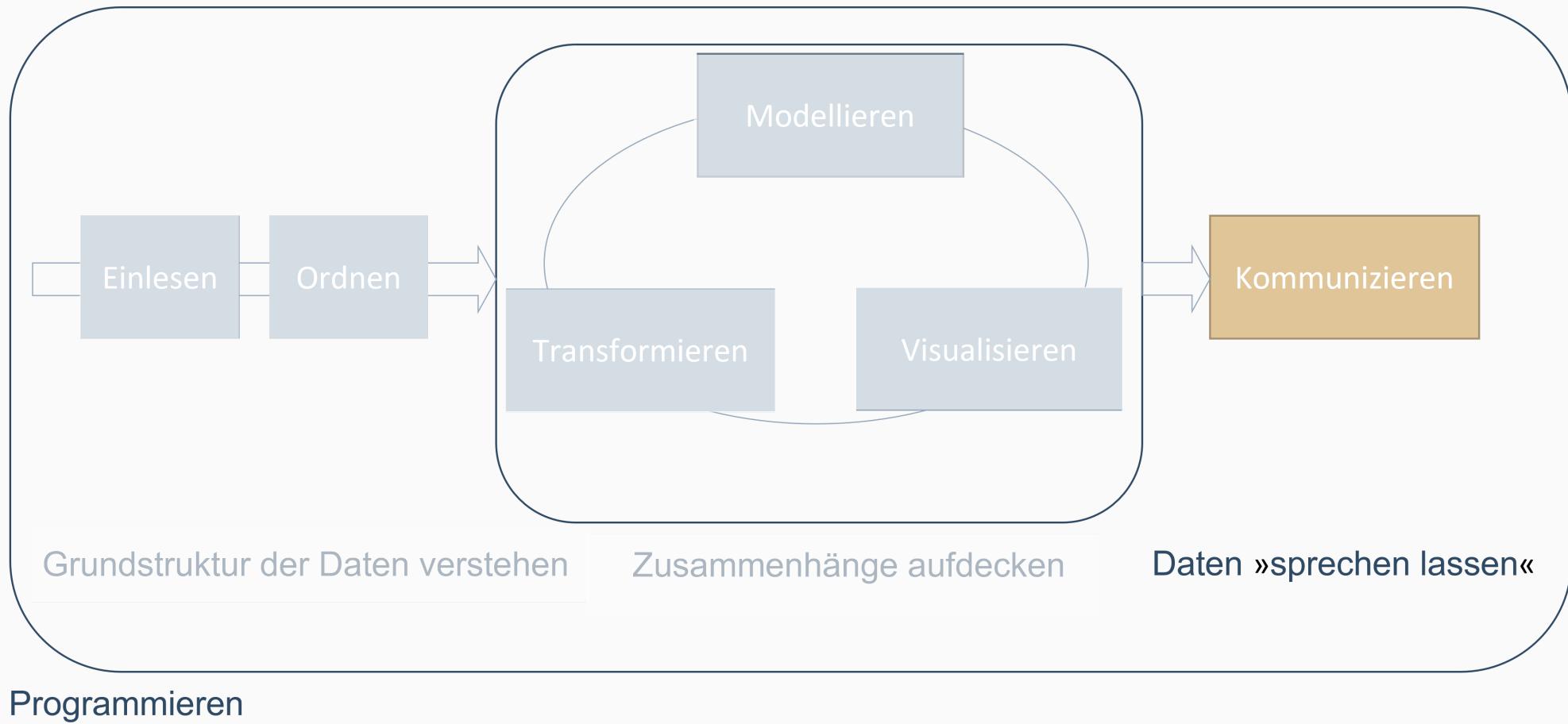


Quelle: [Clarkdatalabs](#)



# Daten modellieren

- Modelle sind komplementär zu Visualisierungen.
- Modelle abstrahieren:
  - Ein Modell verwendet häufig die Formulierung »Wir nehmen an.«
  - Wir vereinfachen in einem Modell die Realität, um uns auf bestimmte Fragen zu konzentrieren.
- Modelle sind ein wichtiger Teil der (empirischen) Wirtschaftsforschung.
- Im Kurs schneiden wir die ökonometrische Modellierung mit R nur an.
- Leseempfehlung:
  - [R4DS, Kapitel 22-25 \(Wickham, 2017\)](#)
- Weiterführende Literatur:
  - [Tidy Modeling with R \(Kuhn & Silge, 2021\)](#)



# Daten und Ergebnisse kommunizieren I/II

- Die Kommunikation von Ergebnissen einer Datenanalyse ist mindestens so wichtig wie alle Schritte zuvor.
- Die Datenanalyse mag bis zum Punkt der Kommunikation perfekt gewesen sein, am Ende gilt es jedoch, anderen die Ergebnisse zu vermitteln.
- Wie man die Daten kommuniziert lässt sich nicht pauschal beantworten.
- Es gibt jedoch fast immer Alternativen, um dieselben Ergebnisse zu kommunizieren.
- Leseempfehlung: [R4DS, Kapitel 26-30](#)

# Daten und Ergebnisse kommunizieren II/II

Klassische Kommunikation

Alternative Kommunikation

Code für beide Alternativen

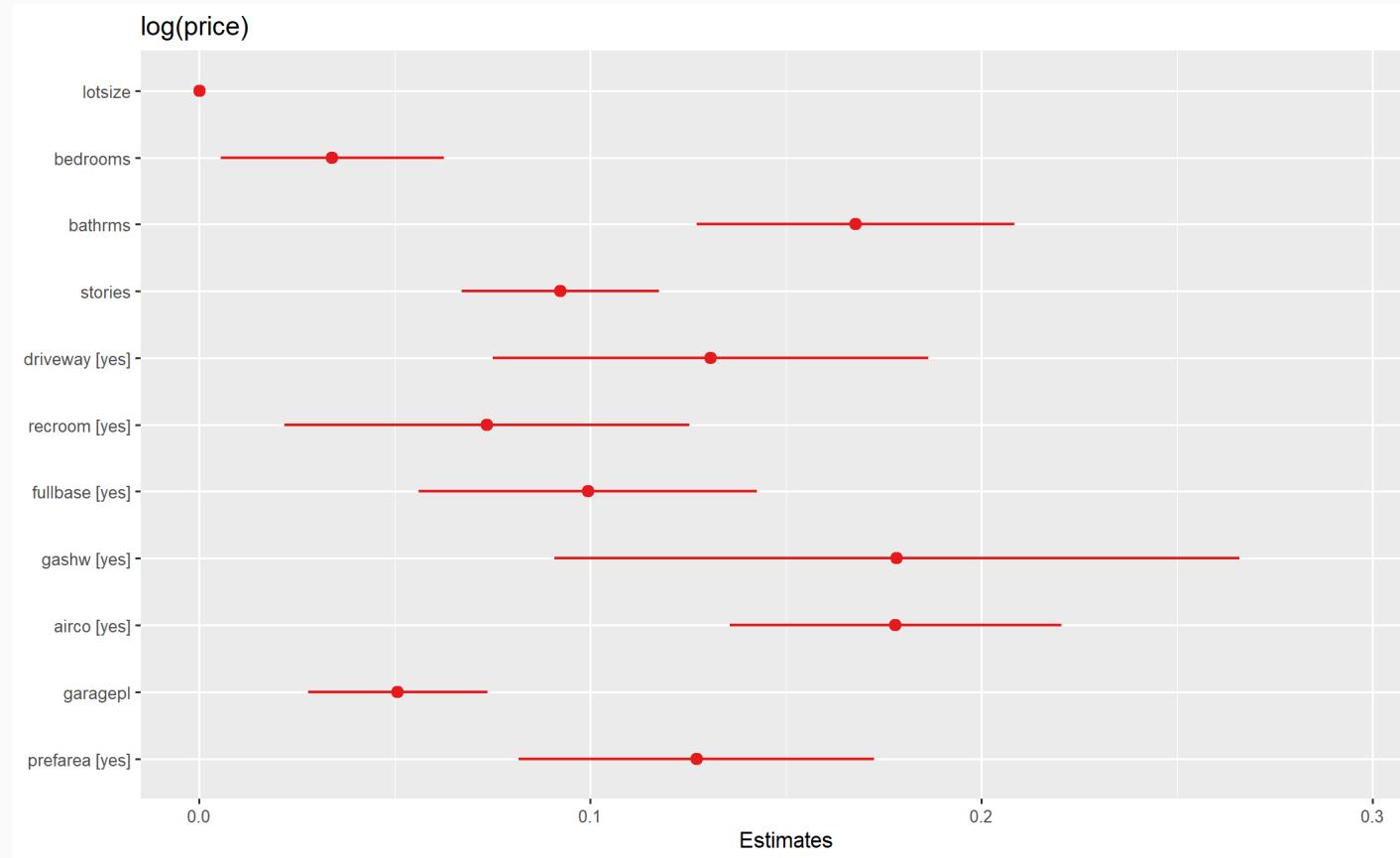
log(price)			
Predictors	Estimates	CI	p
(Intercept)	10.03	9.93 – 10.12	<0.001
lotsize	0.00	0.00 – 0.00	<0.001
bedrooms	0.03	0.01 – 0.06	0.019
bathrms	0.17	0.13 – 0.21	<0.001
stories	0.09	0.07 – 0.12	<0.001
driveway [yes]	0.13	0.07 – 0.19	<0.001
recroom [yes]	0.07	0.02 – 0.13	0.005
fullbase [yes]	0.10	0.06 – 0.14	<0.001
gashw [yes]	0.18	0.09 – 0.27	<0.001
airco [yes]	0.18	0.14 – 0.22	<0.001
garagepl	0.05	0.03 – 0.07	<0.001
prefarea [yes]	0.13	0.08 – 0.17	<0.001
Observations	546		
R <sup>2</sup> / R <sup>2</sup> adjusted	0.677 / 0.670		

# Daten und Ergebnisse kommunizieren II/II

Klassische Kommunikation

Alternative Kommunikation

Code für beide Alternativen



# Daten und Ergebnisse kommunizieren II/II

Klassische Kommunikation

Alternative Kommunikation

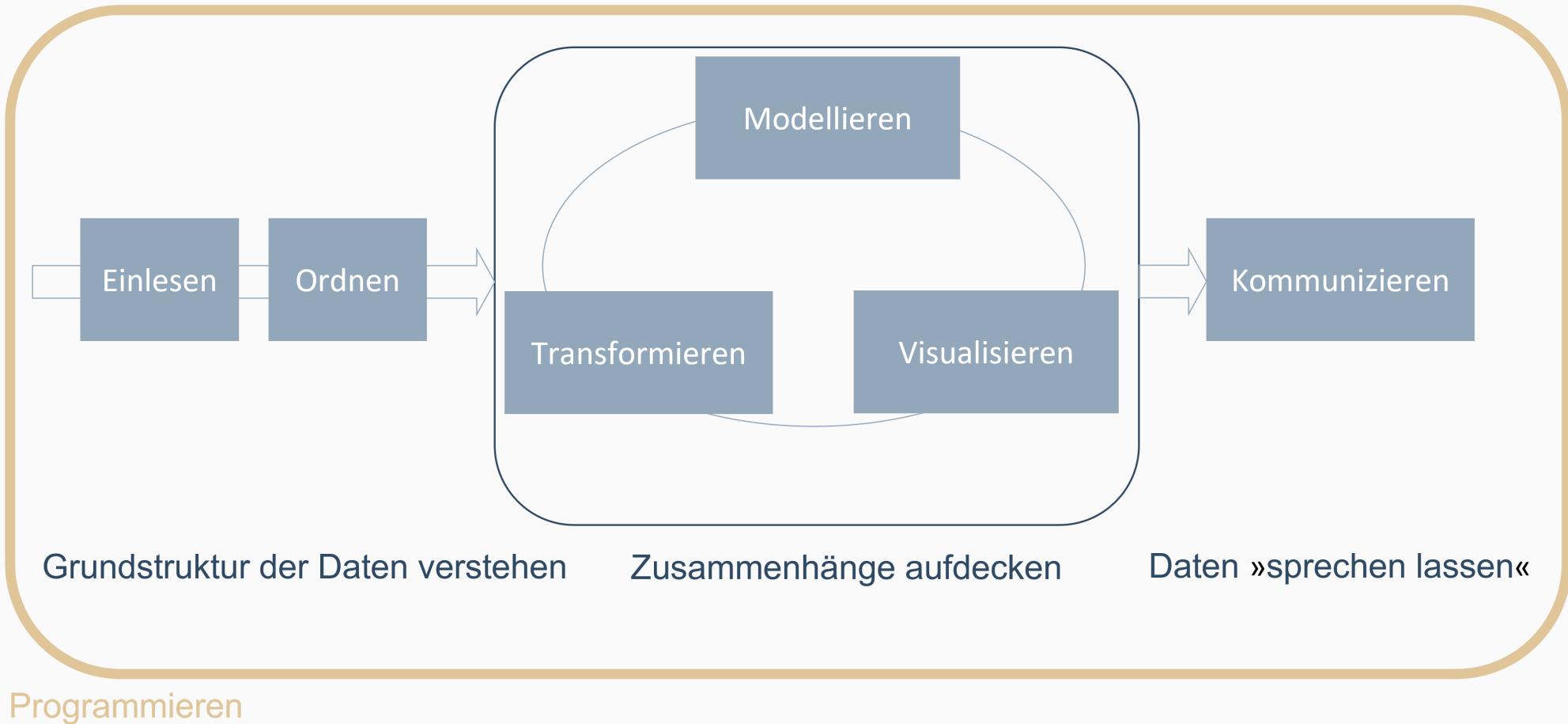
Code für beide Alternativen

```
#Daten laden
library(tidyverse)
library(Ecdat)
data(Housing)

#Modell bestimmen
model1 ← lm(log(price) ~ ., data = Housing)

#Klassische Kommunikation als Regressionstabelle
model1 %>%
  sjPlot::tab_model()

#Alternative Kommunikation als Koeffizientenplot
model1 %>%
  sjPlot::plot_model()
```



# Programmieren

- Grundsätzlich muss die Datenanalyse nicht zwingend durch das Programmieren erfolgen. Man hat die Wahl zwischen dem Erlernen einer **Grafischen Benutzeroberfläche** (GUI - Graphical User Interface) oder einer **Programmiersprache**.
- **Programmieren** bietet jedoch Vorteile gegenüber einer GUI:
  - **Reproduzierbarkeit**: Die Möglichkeit, eine Analyse nachzuvollziehen und zu reproduzieren ist die Grundlage guter wissenschaftlicher Praxis.
  - **Automatisierung**: Befähigt uns Datenanalyseschritte schnell zu wiederholen, falls sich etwas an der Datengrundlage ändert.
  - **Kommunikation**: Programme sind Code und Code ist Text. Text lässt sich einfach kommunizieren und teilen. Auch der Lernprozess wird dadurch vereinfacht.
- Leseempfehlung:
  - Hands-On Programming with R (Grolmund, 2014)
  - R4DS, Kapitel 17-21 (Wickham, 2017)



# Was ist R? I/II

- R ist eine **Programmiersprache** und **kostenfreie** Software.
- Es ist eine Umgebung, in der statistische Methoden implementiert sind.
- R bietet standardmäßige Funktionalitäten → »Base-R«
- Es ist mit **Packages** erweiterbar.
  - Ein Package ist eine **Sammlung von Funktionen**.
  - Die Funktionen erweitern die Basisfunktionalität.
  - Das Paket `readxl` zum Beispiel bietet eine Sammlung von Funktionen, um Excel-Dateien einzulesen, die nicht standardmäßig in R enthalten sind.



Quelle: Mitchell O'Hara-Wild

# Was ist R? II/II

- Im Kern ist R ein schicker *Taschenrechner*.

`1+2`

```
#> [1] 3
```

- Alle Rechenoperationen sind möglich:

`(1+2) / (3-4) * (5^6)`

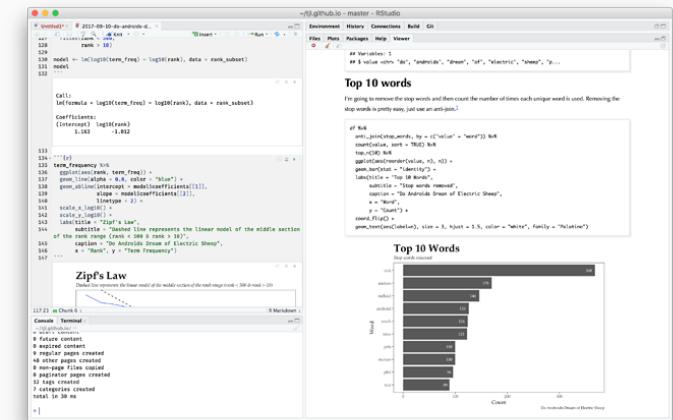
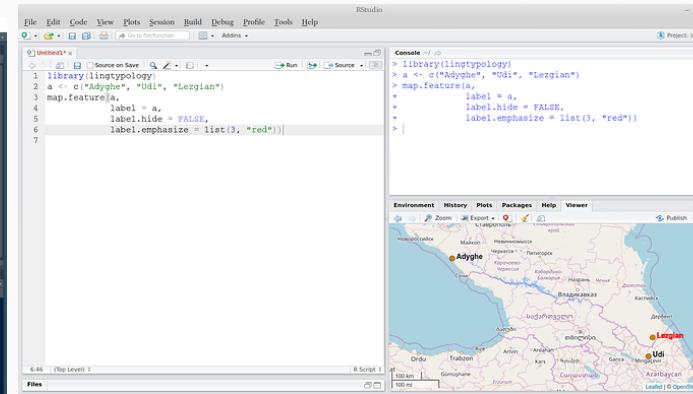
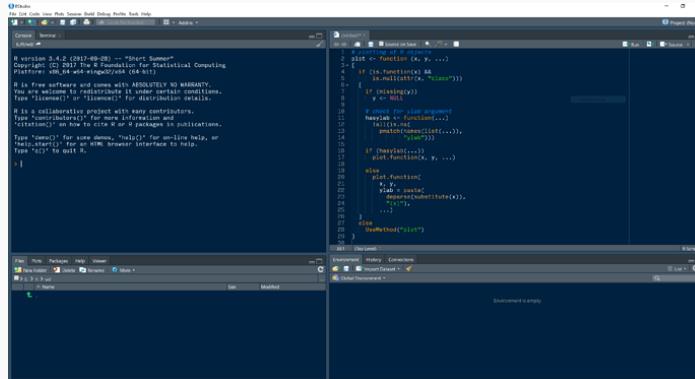
```
#> [1] -46875
```

- Das kann aber eben auch jeder Taschenrechner. In der Praxis arbeiten wir häufig mit Datensätzen als Input, verändern diese und generieren irgendeine Form von Output.



# Was ist R Studio?

- RStudio ist das **Cockpit**, R ist der **Motor**.
- Wir lernen und arbeiten mit R und nutzen dafür RStudio.
- RStudio bietet eine benutzerfreundliche **Bedienoberfläche**, um in R zu »coden«.
- RStudio ist grundsätzlich nicht zwingend erforderlich, macht das Arbeiten mit R aber wesentlich einfacher.
- Die Benutzeroberfläche lässt sich recht frei, je nach persönlicher Präferenz, anpassen.



Quelle: RStudio Community

# Allgemeine Ressourcen

Die Zahl an Ressourcen, um R zu lernen ist groß. Viele Ressourcen sind kostenfrei.

Für den Einstieg:

- **R for Data Science (R4DS)**, (Wickham, 2017)
  - Idealer Einstieg zum Selbstlernen, viele Beispiele.
  - Weite Teile des Kurses basieren auf diesem Buch.
  - Nutzt das **Tidyverse**.

Für den Einstieg und darüber hinaus:

- **Modern Data Science with R** (Baumer et al., 2021)
  - Führt ebenfalls die wichtigsten Konzepte ein.
  - Darüber hinaus Einführung in statistische Grundlagen und Modellierung.
  - Guter Einblick in verschiedene Datentypen, z.B. Textdaten, Geodaten.
- **Modern Statistics with R** (Thulin, 2021)
  - Gute Einführung in die Grundlagen.
  - Fokus auf der Illustration und Anwendung statistischer Methoden mit R.

# Thematische Ressourcen

- Die Themen und Probleme mit denen sich Data Science beschäftigt sind schier endlos.
- Für die meisten gibt es Lösungen in R.
- »**Das einzige Lesezeichen, das man braucht**«: Big Book of R (Baruffa, laufend aktualisiert)
  - Thematische Sammlung einer Vielzahl an Büchern und Tutorials zu R.
  - Die Themen reichen von
    - »R-Internas« (vgl. Kapitel 7, 8, 19, 20, 21),
    - über Data Science und Statistik (vgl. Kapitel 9, 25),
    - bis zu einer Sammlung von Sammlungen (vgl. Kapitel 31).
- Empfehlung: Bei der Suche nach Ressourcen sollte darauf geachtet werden, dass diese sich der **modernen Tidyverse- und Dplyr-Syntax** bedienen.
- Leseempfehlung: Welcome to the Tidyverse (Wickham et al., 2019)

# Die wichtigste Ressource überhaupt



- ...und zwar am besten auf **Englisch**.
- Die meisten Fragen hat sich auch schon mal jemand anderes gestellt...
- ...und wieder jemand anderes beantwortet.
- Für alles, was mit Code zu tun hat, landet man schnell bei [stackoverflow](#).
- Eine Community, in der vom Mitgliedern Fragen gestellt und beantwortet werden.
- Je präziser man das Problem erkannt hat, desto eher findet man die passende Lösung:
  - Googeln von "weighted mean in R".
  - Googeln von "r weighted mean by group dplyr".

Quelle: [tenor.com](#)

# Kursaufbau

- Der Kurs erfordert einen Leistungsaufwand von 2 SWS.
- Der Kurs teilt sich in Vorlesungs- und Übungszeiten und selbständige Eigenleistung in Form von Übungsaufgaben.
- Im Vorlesungsteil werden Konzepte und Strukturen theoretisch eingeführt, die in den praktischen Übungsteilen vertieft werden.
- Die Philosophie des Kurses ist »Hands-On«. Ein Großteil der Zeit wird deshalb auf die praktischen Teile verwendet werden.
- Leistungsnachweise:
  - Präsenz
  - und Bearbeiten der Übungsaufgaben.
- Leistungsvoraussetzungen:
  - Keine.