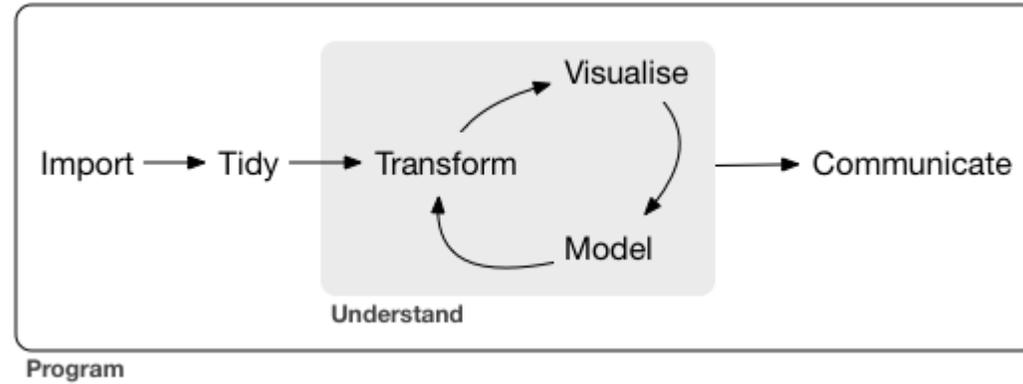


# Importieren von Daten



Quelle: Wickham, H., & Golemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data." O'Reilly Media, Inc.

# Einführung

Bisher hatten Sie sich mit bestehenden Datensätzen in Form von R Objekten beschäftigt

In den Projekten sind Dinge wichtig wie:

- + Zusammenfügen von Informationen aus verschiedenen Datenquellen
- + Bereinigen des Datensatzes (falsch ausgefüllte Fragebögen, Codierung von Zeitreihen in Datastream, ...)
- + Konsistenzchecks (Outlier, doppelte Beobachtungen, ...)

Dies wollen wir nun lernen.

# Daten einlesen

- + Es ist selten, dass Sie in ihrer Analyse auf bereits bearbeitete Datensätze stoßen
- + *Meistens*: Daten müssen aus Excel, Text, Datenbank, API, HTML ... importiert werden
  - + Sie können sich Excel- und Textdateien aus den meisten Datenbanken generieren lassen (DAFNE, Datastream, Bloomberg ...)
  - + Pakete `readr` and `readxl` können uns bei Excel und Textdateien helfen
  - + Paket `rvest` kann uns bei HTMLs helfen
  - + Paket `haven` kann uns bei anderen Formaten wie z.B. Stata-Dateien helfen

# Daten einlesen

- + Sie sollten Datensätze grundsätzlich *nicht* in Excel abspeichern
- + Vorteil von ".csv" (comma-separated value) oder ".txt" (tab-separated value) Dateien:
  - + Sie sind plattformunabhängig lesbar (UNIX/Windows/MAC)
  - + Sie benötigen kein lizenziertes Programm um den Datensatz öffnen zu können
  - + Der Datensatz wird im ASCII Format abgespeichert, wodurch er in jedem Texteditor begutachtet werden kann
  - + Reproduzierbarkeit der Analysen durch Datengrundlage gegeben

# Daten einlesen

- + Sie sollten Datensätze grundsätzlich *nicht* in Excel abspeichern
- + Vorteil von ".csv" (comma-separated value) oder ".txt" (tab-separated value) Dateien:
  - + Sie sind plattformunabhängig lesbar (UNIX/Windows/MAC)
  - + Sie benötigen kein lizenziertes Programm um den Datensatz öffnen zu können
  - + Der Datensatz wird im ASCII Format abgespeichert, wodurch er in jedem Texteditor begutachtet werden kann
  - + Reproduzierbarkeit der Analysen durch Datengrundlage gegeben

Deshalb gilt: Datensätze bitte **immer** in ".csv" oder ".txt"-Format abspeichern!

**Außnahme:** Sie arbeiten nur mit anderen R-Nutzern zusammen, dann können die Daten in .Rds abgespeichert werden.

# Daten einlesen

Um Dateien einzulesen sollten Sie drei Dinge wissen:

- + Wo befinden Sie sich aktuell in ihrem System?
  - + Aktuelles *Arbeitsverzeichnis* mit `getwd`
- + Wo befindet sich die einzulesende Datei?
  - + Pfad zur Datei mit `file.path`
- + Welches Format hat die Datei?
  - + ".csv", ".txt", ".xls(x)", ".dta" ...

# Das Arbeitsverzeichnis

- ✚ Wo befinden Sie sich aktuell und wie kann das *Arbeitsverzeichnis* geändert werden

```
getwd()
```

Laden Sie das Git-Repository mit den Vorlesungsunterlagen herunter. Anschließend wechseln Sie ihr Arbeitsverzeichnis in R zu dem Ordner `wrangling`.

Wechseln Sie in den Ordner `wrangling` mit Hilfe des Befehls `setwd()`

```
setwd("/Pfad/zum/neuen/Arbeitsverzeichnis") # Achten Sie auf die Anführungszeichen und Slashes!
```

# Das Arbeitsverzeichnis

- ✚ Wo befinden Sie sich aktuell und wie kann das *Arbeitsverzeichnis* geändert werden

```
getwd()
```

Laden Sie das Git-Repository mit den Vorlesungsunterlagen herunter. Anschließend wechseln Sie ihr Arbeitsverzeichnis in R zu dem Ordner `wrangling`.

Wechseln Sie in den Ordner `wrangling` mit Hilfe des Befehls `setwd()`

```
setwd("/Pfad/zum/neuen/Arbeitsverzeichnis") # Achten Sie auf die Anführungszeichen und Slashes!
```

```
setwd("/home/riever/pkds/wrangling/") # Pfad bei UNIX  
setwd("C:/Users/riever/Desktop/pkds/wrangling/") # Pfad bei Windows  
setwd("/Users/riever/Desktop/pkds/wrangling/") # Pfad bei Mac
```

```
#Check des Pfades  
getwd()
```

# Beispieldatensätze herunterladen und einlesen

Welche Datensätze befinden sich in dem Unterordner *data*?

# Beispieldatensätze herunterladen und einlesen

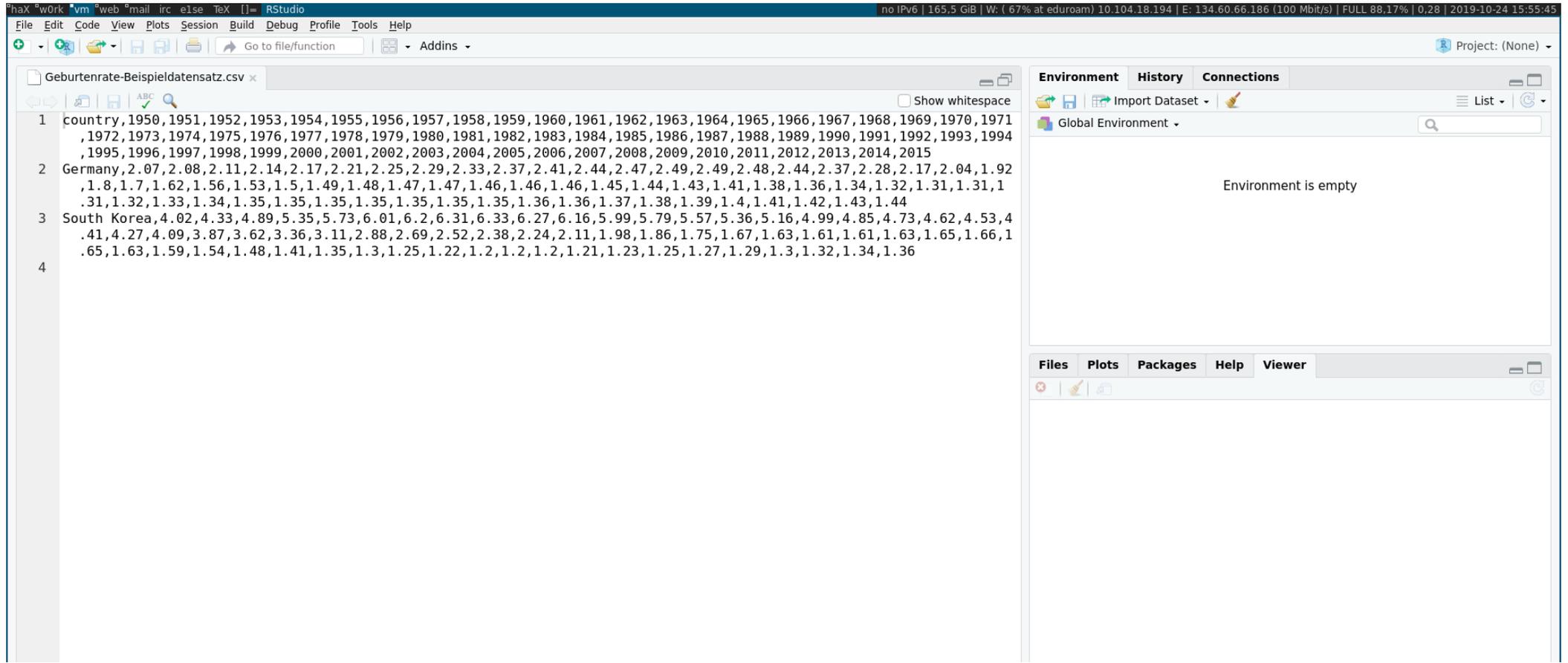
Welche Datensätze befinden sich in dem Unterordner *data*?

```
list.files("data/")
```

```
[1] "balls_percentile_cis.rds"  
[2] "bip.xls"  
[3] "evals.csv"  
[4] "Gapminder_1800-2020.rds"  
[5] "gapminder_life.rds"  
[6] "Geburtenrate-Beispieldatensatz.csv"  
[7] "Geburtenrate-Lebenserwartung_Beispiel.xlsx"  
[8] "Kindersterblichkeit.dta"  
[9] "pwt_2014.rds"
```

# Spreadsheets einlesen

- + Ein Großteil aller Datensätze werden in Spreadsheets abgespeichert
  - + Ein solches Spreadsheet ist im Grund eine Datei in Data Frame-Format



The screenshot shows the RStudio interface with a CSV file named 'Geburtenrate-BeispielDATENSATZ.csv' open in the editor. The data is displayed as a data frame with 4 rows and many columns. The first row lists years from 1950 to 2015. The second row is labeled 'Germany' and the third row is labeled 'South Korea'. The fourth row is empty. The right-hand pane shows the Environment tab, which is currently empty, indicating that the data has not yet been loaded into the R environment.

```
1 country, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015
2 Germany, 2.07, 2.08, 2.11, 2.14, 2.17, 2.21, 2.25, 2.29, 2.33, 2.37, 2.41, 2.44, 2.47, 2.49, 2.49, 2.48, 2.44, 2.37, 2.28, 2.17, 2.04, 1.92, 1.8, 1.7, 1.62, 1.56, 1.53, 1.5, 1.49, 1.48, 1.47, 1.47, 1.46, 1.46, 1.46, 1.45, 1.44, 1.43, 1.41, 1.38, 1.36, 1.34, 1.32, 1.31, 1.31, 1.31, 1.32, 1.33, 1.34, 1.35, 1.35, 1.35, 1.35, 1.35, 1.35, 1.35, 1.35, 1.36, 1.36, 1.37, 1.38, 1.39, 1.4, 1.41, 1.42, 1.43, 1.44
3 South Korea, 4.02, 4.33, 4.89, 5.35, 5.73, 6.01, 6.2, 6.31, 6.33, 6.27, 6.16, 5.99, 5.79, 5.57, 5.36, 5.16, 4.99, 4.85, 4.73, 4.62, 4.53, 4.41, 4.27, 4.09, 3.87, 3.62, 3.36, 3.11, 2.88, 2.69, 2.52, 2.38, 2.24, 2.11, 1.98, 1.86, 1.75, 1.67, 1.63, 1.61, 1.61, 1.63, 1.65, 1.66, 1.65, 1.63, 1.59, 1.54, 1.48, 1.41, 1.35, 1.3, 1.25, 1.22, 1.2, 1.2, 1.2, 1.21, 1.23, 1.25, 1.27, 1.29, 1.3, 1.32, 1.34, 1.36
4
```

# Spreadsheets einlesen

- + Enthält die Datei eine *Kopfzeile* in der die Variablennamen definiert werden?
  - + Datei sollte vor dem Einlesen betrachtet werden
  - + Mit einem Editor
  - + In RStudio direkt (Rechtsklick auf Datei -> Anschauen mit "Rstudio")
- + Einige Spreadsheet können nicht mit einem Texteditor geöffnet werden
  - + z.B. Excel-Dateien
  - + Dateiformat wird häufig verwendet
  - + **keine** eigenen Datensätze darin abspeichern
  - + *jedoch* dazu in der Lage sein Excel-Dateien in R einzulesen (mit `readxl` Paket)

readr und readxl



# readr und readxl

Mit den Paketen `readr` und `readxl` können verschiedene Datensätze eingelesen werden.

Für alle Datensätze, welche mit einem Texteditor geöffnet werden können, das `readr` Paket:

- + `read_table`, `read_csv`, `read_csv2`, `read_tsv`, `read_delim`
- + Beim Einlesen erhalten Sie eine Nachricht, welcher Datentyp pro Spalte erkannt wurde
- + Funktionen aus dem Pakt `readr` sind deutlich schneller die build-in Funktionen von R
  - + *Nicht benutzen*: `read.table`, `read.csv`, `read.delim`

```
library(readr)
geburtenrate <- read_csv("data/Geburtenrate-Beispieldatensatz.csv")
```

# readr und readxl

Für Excel Dateien gibt es das Paket `readxl` mit den Funktionen:

- + `read_excel`, `read_xls`, `read_xlsx`
- + Mit `excel_sheets` erfahren Sie welche Tabellenblätter die Datei beinhaltet
- + Hier können durch `sheet` einzelne Tabellenblätter angesprochen werden

```
library(readxl)
excel_sheets("data/Geburtenrate-Lebenserwartung_Beispiel.xlsx")
```

```
[1] "Lebenserwartung_Geburtenrate" "Erklärung"
```

```
leben_und_geburt <- read_xlsx("data/Geburtenrate-Lebenserwartung_Beispiel.xlsx", sheet="Lebenserwartung_Geburtenrate")
```

# readr und readxl

Sowohl `readr` als auch `readxl` Datensätze werden als `tibble` (eine aktualisierte Form eines Data Frame) eingelesen

```
geburtenrate %>%  
  select(1:4) %>%  
  head(4)
```

```
# A tibble: 2 × 4  
  country      `1950` `1951` `1952`  
  <chr>      <dbl> <dbl> <dbl>  
1 Germany      2.07  2.08  2.11  
2 South Korea  4.02  4.33  4.89
```

```
leben_und_geburt %>%  
  select(1:4) %>%  
  head(4)
```

```
# A tibble: 4 × 4  
  country `1950_life_expectancy` `1951_life_e  
  <chr>      <chr>      <chr>  
1 Brazil  50.33      50.59  
2 Canada  68.26      68.53  
3 China   41.04      41.98  
4 Germany 66.91      67.08
```

# Unterschied zwischen `readr`, `readxl` und Base R

## `readr` und `readxl`

- Die von `readr` eingelesenen Daten werden als `tibble` abgespeichert-
- `readr` erkennt automatisch Faktorvariablen und kann String- und Faktorvariablen unterscheiden
- Datum und Zeit wird durch das `readr` Paket direkt erkannt und in ein R Datum umgewandelt
- Das Einlesen durch `readr` ist ~10 mal schneller als in den Basisfunktionen

## Base R

- Die Basisfunktionen (`read.csv`, `read.table` oder `read.delim`) speichern die Daten als Data Frame
- Die Basisfunktionen lesen String-Variablen als Faktorvariablen ein
- Datum und Zeit werden nicht erkannt und müssen manuell umgeformt werden

# Das haven Paket

- + Neben Excel und R wird in der Wirtschaft und Wissenschaft oft Stata, SPSS und SAS eingesetzt
- + Durch das `haven` Paket können auch diese Datensätze eingelesen werden
- + Das `haven` Paket bringt Flexibilität, denn hierdurch können Sie:
  - + mit Personen kooperieren, welche Stata verwenden
  - + Stata-Datensätze einlesen, welche oft mit Artikeln in Fachzeitschriften veröffentlicht werden

```
library(haven)
kindersterblichkeit <- read_dta("data/Kindersterblichkeit.dta")
head(kindersterblichkeit, 4)
```

# Das haven Paket

- + Neben Excel und R wird in der Wirtschaft und Wissenschaft oft Stata, SPSS und SAS eingesetzt
- + Durch das `haven` Paket können auch diese Datensätze eingelesen werden
- + Das `haven` Paket bringt Flexibilität, denn hierdurch können Sie:
  - + mit Personen kooperieren, welche Stata verwenden
  - + Stata-Datensätze einlesen, welche oft mit Artikeln in Fachzeitschriften veröffentlicht werden

```
library(haven)
kindersterblichkeit <- read_dta("data/Kindersterblichkeit.dta")

head(kindersterblichkeit, 4)
```

```
# A tibble: 4 × 3
  Country      Year Mortality
  <chr>      <dbl>   <dbl>
1 Afghanistan 1950     435.
2 Afghanistan 1951     432.
3 Afghanistan 1957     376.
4 Afghanistan 1958     370.
```

# Probleme beim Einlesen von Daten

Wenn Sie Daten in R einlesen kann einiges schief gehen.

Hier einige Beispiele:

- + Datensätze können mehrere Kopfzeilen enthalten
- + Datensätze können in einem ungünstigen Format abgespeichert sein
- + Zellen können leer sein
- + Die Kodierung kann anders sein als erwartet
  - + Bzgl. der Kodierung, insbesondere im Hinblick auf Unicode ist [dieser Blogeintrag](#) sehr interessant