



# Data Handling: Import, Cleaning and Visualisation

Lecture 8:

Data Preparation

Prof. Dr. Ulrich Matter

19/11/2020

Updates

# Next week: Guest lecture by Dr. Thomas Spycher (Novalytica AG)!



*Dr. Thomas Spycher (Novalytica AG).*

- In Zoom!
- 45-60 minutes talk.
- Q&A.

## Additional online material

- Hint: [tidyverse!](#)
- Hint: [R for Data Science Online Learning Community!](#)

# Mandatory reading

- See the [syllabus](#).
- **NOTE:** The chapter numbers for WiGr refer to the online version of the book (<http://r4ds.had.co.nz/>).

**Recap: Data Import**

# Sources/formats in economics

- CSV (typical for rectangular/table-like data)
- Variants of CSV (tab-delimited, fix length etc.)
- XML and JSON (useful for complex/high-dimensional data sets)
- HTML (a markup language to define the structure and layout of webpages)
- Unstructured text

# Sources/formats in economics

- Excel spreadsheets (`.xls`)
- Formats specific to statistical software packages (SPSS: `.sav`, STATA: `.dat`, etc.)
- Built-in R datasets
- Binary formats



# A Template/Blueprint

```
#####  
# Data Handling Course: Example Script for Data Gathering and Import  
#  
# Imports data from ...  
# Input: links to data sources (data comes in ... format)  
# Output: cleaned data as CSV  
#  
# U. Matter, St.Gallen, 2019  
#####  
  
# SET UP -----  
# load packages  
library(tidyverse)  
  
# set fix variables  
INPUT_PATH <- "/rawdata"  
OUTPUT_FILE <- "/final_data/datafile.csv"
```

# Script sections

Finally we add sections with the actual code (in the case of a data import script, maybe one section per data source)

```
#####  
# Project XY: Data Gathering and Import  
#  
# This script is the first part of the data pipeline of project XY.  
# It imports data from ...  
# Input: links to data sources (data comes in ... format)  
# Output: cleaned data as CSV  
#  
# U. Matter, St.Gallen, 2019  
#####  
  
# SET UP -----  
# load packages  
library(tidyverse)  
  
# set fix variables  
INPUT_PATH <- "/rawdata"  
OUTPUT_FILE <- "/final_data/datafile.csv"  
  
# IMPORT RAW DATA FROM CSVs -----
```

# Parsing CSVs

Recognizing columns and rows is one thing...

```
swiss
```

```
## # A tibble: 47 x 7
##   District      Fertility Agriculture Examination Education Catholic Infant.Mortality
##   <chr>          <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 Courtelary      80.2           17           15           12           9.96           2
## 2 Delemont        83.1           45.1          6            9           84.8           2
## 3 Franches-Mnt    92.5           39.7          5            5           93.4           2
## 4 Moutier         85.8           36.5          12           7           33.8           2
## 5 Neuveville      76.9           43.5          17           15           5.16           2
## 6 Porrentruy      76.1           35.3          9            7           90.6           2
## 7 Broye           83.8           70.2          16           7           92.8           2
## 8 Glane           92.4           67.8          14           8           97.2           2
## 9 Gruyere         82.4           53.3          12           7           97.7           2
## 10 Sarine         82.9           45.2          16           13           91.4           2
## # ... with 37 more rows
```

What else did `read_csv()` recognize?

# Parsing CSVs

- Recall the introduction to data structures and data types in R
- How does R represent data in RAM?
  - **Structure**: `data.frame/tibble`, etc.
  - **Types**: `character, numeric`, etc.
- Parsers in `read_csv()` guess the data **types**.

# Parsing CSV-columns

```
library(readr)
```

```
read_csv('A,B  
         12:00, 12:00  
         14:30, midnight  
         20:01, noon')
```

```
## # A tibble: 3 x 2  
##   A      B  
##   <time> <chr>  
## 1 12:00 12:00  
## 2 14:30 midnight  
## 3 20:01 noon
```

# Parsing CSV-columns: guess types

Under the hood `read_csv()` used the `guess_parser()` - function to determine which type the two vectors likely contain:

```
guess_parser(c("12:00", "midnight", "noon"))
```

```
## [1] "character"
```

```
guess_parser(c("12:00", "14:30", "20:01"))
```

```
## [1] "time"
```

Data Preparation/Munging/Wrangling

# The dataset is imported, now what?

- In practice: still a long way to go.
- Parsable, but messy data: Inconsistencies, data types, missing observations, wide format.



# The dataset is imported, now what?

- In practice: still a long way to go.
- Parsable, but messy data: Inconsistencies, data types, missing observations, wide format.
- **Goal** of data preparation: Dataset is ready for analysis.
- **Key conditions:**
  1. Data values are consistent/clean within each variable.
  2. Variables are of proper data types.
  3. Dataset is in 'tidy' (in long format)!

# Some vocabulary

Following Wickham (2014):

- **Dataset**: Collection of **values** (numbers and strings).
- Every value belongs to a **variable** and an **observation**.
- **Variable**: Contains all values that measure the same underlying attribute across units.
- **Observation**: Contains all values measured on the same unit (e.g., a person).

# Tidy data

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	8666	20595360
Brazil	1999	34737	172006362
Brazil	2000	80488	174604898
China	1999	214258	1272915272
China	2000	216766	1286425583

variables

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	8666	20595360
Brazil	1999	34737	172006362
Brazil	2000	80488	174604898
China	1999	214258	1272915272
China	2000	216766	1286425583

observations

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	8666	20595360
Brazil	1999	34737	172006362
Brazil	2000	80488	174604898
China	1999	214258	1272915272
China	2000	216766	1286425583

values

Tidy data. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States](https://creativecommons.org/licenses/by-sa/3.0/) license.

# Data preparation in R (`tidyverse`)

Q&A

# References

Wickham, Hadley. 2014. "Tidy Data." **Journal of Statistical Software** 59 (10): 1–23.  
<https://doi.org/10.18637/jss.v059.i10>.

Wickham, Hadley, and Garrett Grolemund. 2017. Sebastopol, CA: O'Reilly. <http://r4ds.had.co.nz/>.