

Data basics

ME 447/547 Visualizing Data

Richard Layton

March 2020

Rose-Hulman Institute of Technology

Preparing data for graphs starts with four basic skills

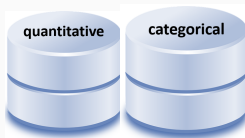
Obtain the raw data



Read raw data into R
and examine it



Identify the structure
of your data



Tidy the data and
write to file

| country | year | cases | population | country | year | cases | population |
|---------|------|--------|------------|---------|------|-------|------------|
| Algeria | 2019 | 1005 | 410071 | | | | |
| Algeria | 2020 | 1006 | 4100360 | | | | |
| Brazil | 1999 | 30017 | 17206302 | | | | |
| Brazil | 2000 | 80066 | 17404898 | | | | |
| China | 1999 | 210058 | 127205272 | | | | |
| China | 2000 | 200000 | 128000000 | | | | |

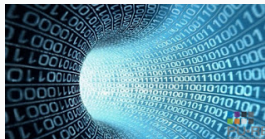
↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ← → ← → ← → ← → ← →
variables observations

Data are everywhere

Data are provided in base R

Data are provided in R packages

Online sources are ubiquitous



- FiveThirtyEight <https://data.fivethirtyeight.com/>
- US government <https://www.data.gov/>
- NOAA climate data <https://www.ncdc.noaa.gov/cdo-web/>
- Publications for which code and/or data are available
<https://reproducible-research.net/reproducible-material/>

You may even have data of your own from prior courses or research

For practice, use data in **base R**

`data()` to list data sets in base R

```
#> AirPassengers    Monthly Airline Passenger Numbers
#> BJsales           Sales Data with Leading Indicator
#> BOD               Biochemical Oxygen Demand
#> CO2               Carbon Dioxide Uptake in Grass Plants
#> Formaldehyde     Determination of Formaldehyde
etc.
```

For practice, use data in R packages

`data(package = "dplyr")` to list data sets in package dplyr

```
#> band_instruments    Band membership
#> band_instruments2   Band membership
#> band_members        Band membership
#> nasa                 NASA spatio-temporal data
#> starwars            Starwars characters
#> storms              Storm tracks data
```

See the help page for a data set by typing `? data_name`

```
library("graphclassmate")  
data(package = "graphclassmate")  
? metro_pop
```

metro_pop {graphclassmate}

R Documentation

Population in the NY metro area

Description

A data set of population in the New York metropolitan area by county and race/ethnicity from the 2000 census.

Usage

```
metro_pop
```

Format

A tidy data frame (tibble) with 60 observations and 3 variables. An observation is the population in a county by race/ethnicity.

race

Race or ethnicity

county

Name of county

population

Number of residents from the 2000 US census

Data in base R and in R packages are automatically loaded



Launching R loads all data sets in base-R



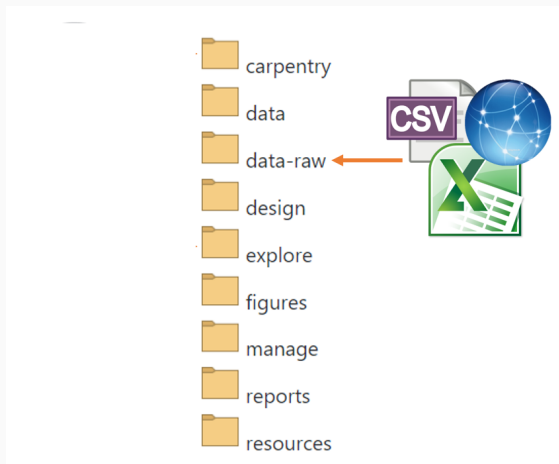
Loading a package with `library()` loads all the data sets in the package



All other data files have to be read or web-scraped

Save original data files in the **data-raw** directory

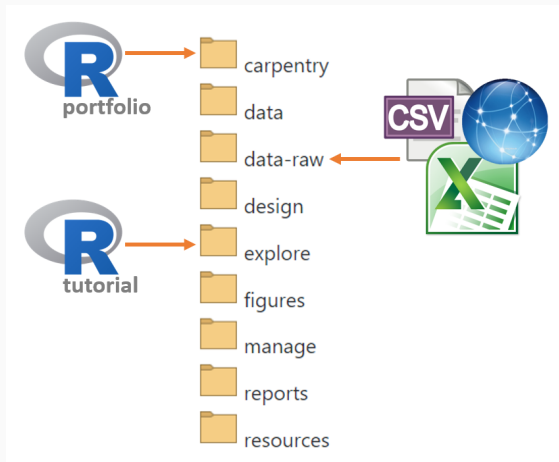
Data in their original form are never edited manually



We work with **file manangement** in detail during the **data studio**.

Read the raw data with R scripts

R scripts are saved in the **carpentry** or **explore** directories



We work with **file management** in detail during the **data studio**.

Suppose `data-raw/` contains data in an **Excel** file

`readxl` is the package (you will have to install the package)

`read_excel()` is the function

Row 1 contains the names of the variables

Row 2 starts the data, one observation per row

Name of the sheet

| | A | B | C | D | E |
|---|-------------|------|------------|-----------|---|
| 1 | country | year | type | count | |
| 2 | Afghanistan | 1999 | cases | 745 | |
| 3 | Afghanistan | 1999 | population | 19987071 | |
| 4 | Afghanistan | 2000 | cases | 2666 | |
| 5 | Afghanistan | 2000 | population | 20595360 | |
| 6 | Brazil | 1999 | cases | 37737 | |
| 7 | Brazil | 1999 | population | 172006362 | |
| 8 | Brazil | 2000 | cases | 80488 | |
| 9 | Brazil | 2000 | population | 174504898 | |

DSR-table1

Ready

We work with **data tidying** in detail during the **data studio**.

read_excel() to read an Excel file

```
library("readxl")  
tidy_data <- read_excel(path = "data-raw/DSR-table1.xlsx",  
                        sheet = "DSR-table1")
```

We can pretty-print the data using `knitr::kable()`

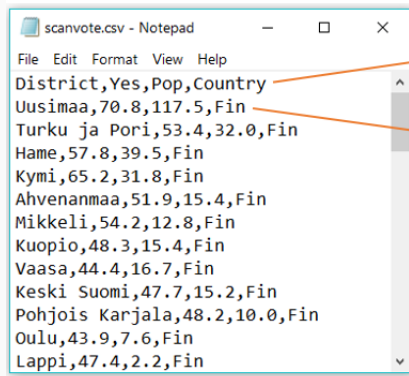
```
library("knitr")  
kable(tidy_data)
```

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 745 | 19987071 |
| Afghanistan | 2000 | 2666 | 20595360 |
| Brazil | 1999 | 37737 | 172006362 |
| Brazil | 2000 | 80488 | 174504898 |
| China | 1999 | 212258 | 1272915272 |
| China | 2000 | 213766 | 1280428583 |

Suppose `data-raw/` contains data in a **CSV** file

`readr` is the package (part of the tidyverse)

`read_csv()` is the function



```
scanvote.csv - Notepad
File Edit Format View Help
District,Yes,Pop,Country
Uusimaa,70.8,117.5,Fin
Turku ja Pori,53.4,32.0,Fin
Hame,57.8,39.5,Fin
Kymi,65.2,31.8,Fin
Ahvenanmaa,51.9,15.4,Fin
Mikkeli,54.2,12.8,Fin
Kuopio,48.3,15.4,Fin
Vaasa,44.4,16.7,Fin
Keski Suomi,47.7,15.2,Fin
Pohjois Karjala,48.2,10.0,Fin
Oulu,43.9,7.6,Fin
Lappi,47.4,2.2,Fin
```

Row 1 contains the names of the variables

Row 2 starts the data, one observation per row

We work with **data tidying** in detail during the **data studio**.

read_csv() to read a CSV file

```
library("tidyverse") # loads the readr package  
tidy_data_2 <- read_csv(file = "data-raw/scanvote.csv")
```

We can pretty-print the top n rows with head()

```
tidy_data_2 %>%  
  head(., n = 5L) %>%  
  kable()
```

| District | Yes | Pop | Country |
|---------------|------|-------|---------|
| Uusimaa | 70.8 | 117.5 | Fin |
| Turku ja Pori | 53.4 | 32.0 | Fin |
| Hame | 57.8 | 39.5 | Fin |
| Kymi | 65.2 | 31.8 | Fin |
| Ahvenanmaa | 51.9 | 15.4 | Fin |

`read_excel()` and `read_csv()` produce tibbles

```
class(tidy_data)
```

```
#> [1] "tbl_df"      "tbl"        "data.frame"
```

```
class(tidy_data_2)
```

```
#> [1] "spec_tbl_df" "tbl_df"      "tbl"        "data.frame"
```


Confine your webscraping (for now) to data in **ASCII** format



INTRODUCTION
METHODOLOGY
DATA EXPLANATION

CANADA

Newfoundland
Prince Edward
Island
Nova Scotia
New Brunswick
Quebec
Ontario
Manitoba
Saskatchewan
Alberta
British Columbia
Northwest
Territories
Yukon

Archives (Life Tables)
Links

Canadian Human Mortality Database

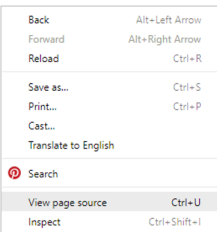
ALBERTA

| CHMD Data 1921-2011 | Age Interval x Year Interval | | | | | |
|---|------------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| | 1 x 1 | 1 x 5 | 1 x 10 | 5 x 1 | 5 x 5 | 5 x 10 |
| Births | View | | | | | |
| Deaths (Lexis triangle) | View | | | View | | |
| Population size | View | | | View | | |
| Exposure-to-risk | View | View | View | View | View | View |
| Death rates | View | View | View | View | View | View |
| Life tables - Male | View | View | View | View | View | View |
| Life tables - Female | View | View | View | View | View | View |
| Life tables - Total | View | View | View | View | View | View |
| Life expectancy at birth | View | | | | | |

On any online data page, right-click > View page source

Canada-Alberta, Population size (1-year) Last modified: 31-Jul-2014, MPv5 (May07)

| Year | Age | Female | Male | Total |
|------|-----|---------|---------|----------|
| 1921 | 0 | 7864.85 | 8133.86 | 15998.71 |
| 1921 | 1 | 7936.45 | 8142.91 | 16079.36 |
| 1921 | 2 | 8024.81 | 8240.34 | 16265.15 |
| 1921 | 3 | 8017.72 | 8244.01 | 16261.73 |
| 1921 | 4 | 7925.59 | 8154.78 | 16080.37 |
| 1921 | 5 | 7760.15 | 7991.74 | 15751.89 |
| 1921 | 6 | 7530.77 | 7768.78 | 15299.55 |
| 1921 | 7 | 7250.42 | 7495.09 | 14745.51 |
| 1921 | 8 | 6926.93 | 7181.14 | 14108.07 |
| 1921 | 9 | 6568.14 | 6829.25 | 13397.39 |
| 1921 | 10 | 6177.87 | 6440.22 | 12618.09 |
| 1921 | 11 | 5832.52 | 6104.80 | 11937.32 |
| 1921 | 12 | 5561.03 | 5857.21 | 11418.24 |
| 1921 | 13 | 5341.23 | 5669.97 | 11011.20 |
| 1921 | 14 | 5112.95 | 5478.27 | 10591.22 |
| 1921 | 15 | 4892.53 | 5302.30 | 10194.83 |
| 1921 | 16 | 4701.44 | 5147.45 | 9848.89 |
| 1921 | 17 | 4557.01 | 5008.00 | 9565.01 |
| 1921 | 18 | 4443.74 | 4893.96 | 9337.70 |
| 1921 | 19 | 4335.37 | 4802.62 | 9137.99 |
| 1921 | 20 | 4242.44 | 4727.01 | 8969.45 |
| 1921 | 21 | 4162.10 | 4645.00 | 8807.10 |



Data formatted in **ASCII** (text) is easily recognized

| Canada-Alberta, Population size (1-year) | | | | Last modified: 31-Jul-2014, MPv5 (May07) |
|--|-----|---------|---------|--|
| Year | Age | Female | Male | Total |
| 1921 | 0 | 7864.85 | 8133.86 | 15998.71 |
| 1921 | 1 | 7936.45 | 8142.91 | 16079.36 |
| 1921 | 2 | 8024.81 | 8240.34 | 16265.15 |
| 1921 | 3 | 8017.72 | 8244.01 | 16261.73 |
| 1921 | 4 | 7925.59 | 8154.78 | 16080.37 |
| 1921 | 5 | 7760.15 | 7991.74 | 15751.89 |
| 1921 | 6 | 7530.77 | 7768.78 | 15299.55 |
| 1921 | 7 | 7250.42 | 7495.09 | 14745.51 |
| 1921 | 8 | 6926.93 | 7181.14 | 14108.07 |
| 1921 | 9 | 6568.14 | 6829.25 | 13397.39 |
| 1921 | 10 | 6177.87 | 6440.22 | 12618.09 |
| 1921 | 11 | 5832.52 | 6104.80 | 11937.32 |
| 1921 | 12 | 5561.03 | 5857.21 | 11418.24 |
| 1921 | 13 | 5341.23 | 5669.97 | 11011.20 |
| 1921 | 14 | 5112.95 | 5478.27 | 10591.22 |
| 1921 | 15 | 4892.53 | 5302.30 | 10194.83 |
| 1921 | 16 | 4701.44 | 5147.45 | 9848.89 |
| 1921 | 17 | 4557.01 | 5008.00 | 9565.01 |

Data formatted in **HTML** is also easily recognized

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 TRANSITIONAL//EN"
2 "http://www.w3.org/TR/REC-html40/loose.dtd">
3 <HTML LANG="en">
4 <head>
5 <script>
6 if (document.layers)
7   WM_scaleFont(initialFontSize, fontUnits);
8 </script>
9 <title>Historical Census of Housing Tables Home Values - Housing Topics - U.S. Census
  Bureau</TITLE>
10
11 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
12
13 <meta name="DC.title" content="US Census Bureau Historical Census of Housing Tables Home
  Values" />
14
15 <meta name="DC.description" content="Selected housing characteristics data from decennial
  census housing files are presented here for the United States and for each state. Trend
  analyses are discussed, with graphic illustration at the national level." />
16
17 <meta name="DC.creator" content="SEHSD Division" />
18
19 <meta name="DC.date.created" scheme="ISO8601" content="2000-06-01" />
20
21 <meta name="DC.date.reviewed" scheme="ISO8601" content="2000-06-01" />
22
23 <meta name="DC.language" scheme="DCTERMS.RFC1766" content="EN-US" />
24
25 <meta name="author" content="US Census Bureau Historical Census of Housing Tables Home
```

With online data in ASCII format, webscraping is easy

`utils` is the package

`read.table()` is the function

```
library("utils")
```

```
url <-
```

```
  "http://www.prdh.umontreal.ca/BDLC/data/alb/Population.txt"
```

```
df <- read.table(url,  
                 skip = 2,  
                 header = TRUE,  
                 stringsAsFactors = FALSE)
```

```
df <- as_tibble(df)
```


When the data are not tidy, ...

VADeaths.xlsx

| | A | B | C | D | E |
|---|-------|-------|-------|-------|-------|
| 1 | | Rural | | Urban | |
| 2 | Group | Men | Women | Men | Women |
| 3 | 50-54 | 11.7 | 8.7 | 15.4 | 8.4 |
| 4 | 55-59 | 18.1 | 11.7 | 24.3 | 13.6 |
| 5 | 60-64 | 26.9 | 20.3 | 37 | 19.3 |
| 6 | 65-69 | 41 | 30.9 | 54.6 | 35.1 |
| 7 | 70-74 | 66 | 54.3 | 71.1 | 50 |

Row 1 has merged cells
variable name information

Row 2 has more variable
name information

Row 3 starts the data

... the read results can be weird.

```
untidy_data <- read_excel(path = "data-raw/VADeaths.xlsx",  
                           sheet = "VADeaths") %>%  
  glimpse()
```

```
#> Observations: 6
```

```
#> Variables: 5
```

```
#> $ ...1 <chr> "Group", "50-54", "55-59", "60-64", "65-69", "
```

```
#> $ Rural <chr> "Men", "11.7", "18.100000000000001", "26.9", "
```

```
#> $ ...3 <chr> "Women", "8.6999999999999993", "11.7", "20.3", "
```

```
#> $ Urban <chr> "Men", "15.4", "24.3", "37", "54.6", "71.099999999999997",
```

```
#> $ ...5 <chr> "Women", "8.4", "13.6", "19.3", "35.1", "50"
```

All the cells have been converted to character data

The result is more easily seen using `knitr::kable()`

```
kable(untidy_data)
```

| ...1 | Rural | ...3 | Urban | ...5 |
|-------|---------------------|--------------------|--------------------|-------|
| Group | Men | Women | Men | Women |
| 50-54 | 11.7 | 8.6999999999999993 | 15.4 | 8.4 |
| 55-59 | 18.1000000000000001 | 11.7 | 24.3 | 13.6 |
| 60-64 | 26.9 | 20.3 | 37 | 19.3 |
| 65-69 | 41 | 30.9 | 54.6 | 35.1 |
| 70-74 | 66 | 54.3 | 71.099999999999994 | 50 |

The first row is not an observation—that's the problem

When reading the file, we need to skip the first row

The data are at least readable but we have lost information

```
untidy_data <- read_excel(path = "data-raw/VADeaths.xlsx",  
                           sheet = "VADeaths",  
                           skip = 1)
```

| Group | Men...2 | Women...3 | Men...4 | Women...5 |
|-------|---------|-----------|---------|-----------|
| 50-54 | 11.7 | 8.7 | 15.4 | 8.4 |
| 55-59 | 18.1 | 11.7 | 24.3 | 13.6 |
| 60-64 | 26.9 | 20.3 | 37.0 | 19.3 |
| 65-69 | 41.0 | 30.9 | 54.6 | 35.1 |
| 70-74 | 66.0 | 54.3 | 71.1 | 50.0 |

Carpentry on untidy data is a large fraction of your effort



It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data.

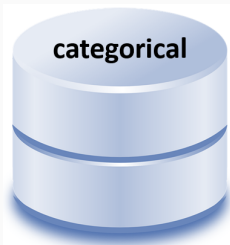
Data preparation is not just a first step, but must be repeated many times over the course of analysis as new problems come to light or new data is collected.

—Hadley Wickham, [Tidy Data](#)

Understanding data structure is necessary for **tidying**



Number of variables?
Continuous or discrete?



Number of variables?
Nominal or ordinal?
Number of levels each?

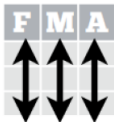
Transforming data to tidy form is necessary for productivity

- how you tidy the data set before graphing
- the graph types that are suitable
- how easy it is to get ggplot to do your bidding
- how productively you spend your time



For graphical productivity, data has to be tidy

In a tidy
data set:



Each **variable** is saved
in its own **column**

&



Each **observation** is
saved in its own **row**

Source: data-wrangling-cheatsheet, <https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf>

Every column is a variable, every row is an observation

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 745 | 19987071 |
| Afghanistan | 2000 | 2666 | 20595360 |
| Brazil | 1999 | 37737 | 172006362 |
| Brazil | 2000 | 80488 | 174504898 |
| China | 1999 | 212258 | 1272915272 |
| China | 2000 | 213766 | 1280428583 |

table1

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 745 | 19987071 |
| Afghanistan | 2000 | 2666 | 20595360 |
| Brazil | 1999 | 37737 | 172006362 |
| Brazil | 2000 | 80488 | 174504898 |
| China | 1999 | 212258 | 1272915272 |
| China | 2000 | 213766 | 1280428583 |

variables

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 745 | 19987071 |
| Afghanistan | 2000 | 2666 | 20595360 |
| Brazil | 1999 | 37737 | 172006362 |
| Brazil | 2000 | 80488 | 174504898 |
| China | 1999 | 212258 | 1272915272 |
| China | 2000 | 213766 | 1280428583 |

observations

Source: Data Science with R by Garrett Grolemund,
<http://garrettgman.github.io/tidying/>

Sadly, untidy data is common

| country | year | key | value |
|-------------|------|------------|------------|
| Afghanistan | 1999 | cases | 745 |
| Afghanistan | 1999 | population | 19987071 |
| Afghanistan | 2000 | cases | 2666 |
| Afghanistan | 2000 | population | 20595360 |
| Brazil | 1999 | cases | 37737 |
| Brazil | 1999 | population | 172006362 |
| Brazil | 2000 | cases | 80488 |
| Brazil | 2000 | population | 174504898 |
| China | 1999 | cases | 212258 |
| China | 1999 | population | 1272915272 |
| China | 2000 | cases | 213766 |
| China | 2000 | population | 1280428583 |

table2

| country | year | key | value |
|-------------|------|------------|------------|
| Afghanistan | 1999 | cases | 745 |
| Afghanistan | 1999 | population | 19987071 |
| Afghanistan | 2000 | cases | 2666 |
| Afghanistan | 2000 | population | 20595360 |
| Brazil | 1999 | cases | 37737 |
| Brazil | 1999 | population | 172006362 |
| Brazil | 2000 | cases | 80488 |
| Brazil | 2000 | population | 174504898 |
| China | 1999 | cases | 212258 |
| China | 1999 | population | 1272915272 |
| China | 2000 | cases | 213766 |
| China | 2000 | population | 1280428583 |

variables

| country | year | key | value |
|-------------|------|------------|------------|
| Afghanistan | 1999 | cases | 745 |
| Afghanistan | 1999 | population | 19987071 |
| Afghanistan | 2000 | cases | 2666 |
| Afghanistan | 2000 | population | 20595360 |
| Brazil | 1999 | cases | 37737 |
| Brazil | 1999 | population | 172006362 |
| Brazil | 2000 | cases | 80488 |
| Brazil | 2000 | population | 174504898 |
| China | 1999 | cases | 212258 |
| China | 1999 | population | 1272915272 |
| China | 2000 | cases | 213766 |
| China | 2000 | population | 1280428583 |

observations

Sadly, untidy data is common

| country | 1999 | 2000 |
|-------------|--------|--------|
| Afghanistan | 745 | 2666 |
| Brazil | 37737 | 80488 |
| China | 212258 | 213766 |

table4

| country | 1999 | 2000 |
|-------------|------------|------------|
| Afghanistan | 19987071 | 20595360 |
| Brazil | 172006362 | 174504898 |
| China | 1272915272 | 1280428583 |

table5

| country | |
|-------------|---------------|
| Afghanistan | 745 2666 |
| Brazil | 37737 80488 |
| China | 212258 213766 |

| country | |
|-------------|-----------------------|
| Afghanistan | 19987071 20595360 |
| Brazil | 172006362 174504898 |
| China | 1272915272 1280428583 |

variables

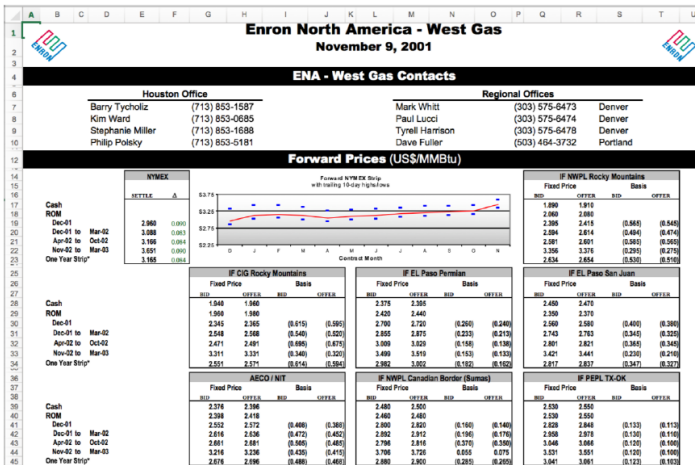
| country | |
|-------------|-----------------------|
| Afghanistan | 19987071 20595360 |
| Brazil | 172006362 174504898 |
| China | 1272915272 1280428583 |

| country | |
|-------------|-----------------------|
| Afghanistan | 19987071 20595360 |
| Brazil | 172006362 174504898 |
| China | 1272915272 1280428583 |

observations

Source: Data Science with R by Garrett Grolemund,
<http://garrettgman.github.io/tidying/>

Some industry or government spreadsheets are horribly untidy



Source: Extract tables from messy spreadsheets with jailbreakr,

<http://blog.revolutionanalytics.com/2016/08/jailbreakr.html>

Data beyond the “raw” stage are written to file

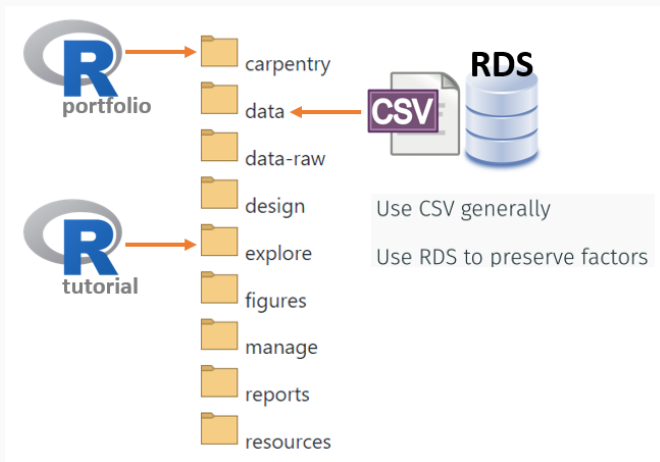
Write functions

```
write_csv() # use CSV generally  
saveRDS()   # use RDS to preserve factors
```

Read functions for further data carpentry

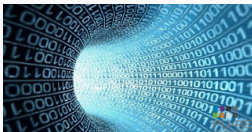
```
read_csv()  
readRDS()
```

Data beyond the “raw” stage reside in the data directory



In the data studio, you'll start practicing the skills we've outlined

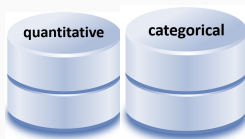
Obtain the raw data



Read raw data into R
and examine it



Identify the structure
of your data



Tidy the data and
write to file

| country | year | cases | population | country | year | cases | population |
|---------|------|-----------|------------|---------|------|-----------|------------|
| Algeria | 2019 | 10000 | 44000000 | Algeria | 2019 | 10000 | 44000000 |
| Algeria | 2020 | 10000 | 44000000 | Algeria | 2020 | 10000 | 44000000 |
| Brazil | 2019 | 17126362 | 212500000 | Brazil | 2019 | 17126362 | 212500000 |
| Brazil | 2020 | 17126362 | 212500000 | Brazil | 2020 | 17126362 | 212500000 |
| China | 2019 | 827215272 | 1400000000 | China | 2019 | 827215272 | 1400000000 |
| China | 2020 | 827215272 | 1400000000 | China | 2020 | 827215272 | 1400000000 |

↑ variables ↓ observations