Getting and Cleaning Data Week 1

Course Description

Before you can work with data you have to get some. This course will cover the basic ways that data can be obtained. The course will cover obtaining data from the web, from APIs, and from colleagues in various formats including raw text files, binary files, and databases. It will also cover the basics of data cleaning and how to make data tidy. Tidy data dramatically speed downstream data analysis tasks. The course will also cover the components of a complete data set including raw data, processing instructions, codebooks, and processed data. The course will cover the basics needed for collecting, cleaning, and sharing data.

- Data Collection
  - Raw files (.csv, .xlsx)
  - Databases (mySQL)
  - APIs

- Data Formats
  - Flat files (.csv, .txt)
  - XML
  - JSON

- Making Data Tidy

- Distributing data

- Scripting for data cleaning

Obtaining Data Motivation

- Basic ideas behind getting data ready for analysis
  - Finding and extracting raw data
  - tidy data principles and how to make data tidy
  - Practical implementation through a range of R packages

- Prerequisite courses:
  - The Data Scientist's Toolbox
  - R Programming

- Other useful courses:
  - Exploratory Analysis
  - Reporting Data and Reproducible Research

Goal of this course:
Focus on the first three stages:
raw data -> **Processing script** -> **tidy data** -> data analysis -> data communication

#Raw and Processed Data
Raw data can be different, according to who you're speaking to.
"Data are values of qualitative or quantitative variables, belonging to a set of items."

Raw Data

- Original source of the data
- Often hard to use for data analyses
- Data analysis *includes* processing
- Raw data may only need to be processed once

Processed Data

- Data that is ready for analysis
- Processing can include merging, subsetting, transforming, etc.
- There may be standards for processing
- **All steps should be recorded.**

Components of tidy data

Four things you should have when you finished going from the raw data to a tidy data set:

- the raw data.
- a tidy data set
- a code book describing each variable and its values in the tidy data set (often called metadata)
- an explicit and exact recipe you used to go from 1 to 2 and 3. (in this case it will be recorded as an R script)

The Raw Data should be the rawest form of the data that you had access to.

- You ran no software on the data.
- You did not manipulate any numbers in the data set
- you did not remove any data from the data set
- you did not summarisa the data set in any way

The tidy data is your objective

- each variable you measure is in one column
- each observation should be in a different row

- there should be one table for each “kind” of variable (eg: data from twitter, fb, etc, one table for each)
- if there are multiple tables, they should include a column in the table that allows them to be linked together.

include variable names if possible, and make them human-readable. in genera, data should be saved in one file per table

The code book

- Information about the variables in the data set, not contained in the tidy data.
- information about the summary choices made.
- information about the experimental study design used.

often written in Word or text file
include a section called “Study design”that has a thorough description of how you collected the data
include a section called “code book” that describes all variables and units

Instruction list

- ideally a computer script (in R or python)
- the input for the script is the raw data
- the output is the processed, tidy data
- there are no parameters to the script -> exact recipe
  Needs to be reproducible

In some cases, it will not be possible to script every step. In that case you should provide instructions like:

- Step 1: take the raw file, run version 3.1.2 of *summarize software* with parameters a=1, b=2, c=3
- step 2: run software separately for each sample
- setp 3 - take column three of outputfile.out for each sample and that is the corresponding row in the output data set.
  very detailed instructions. **REPRODUCIBILITY**

Downloading files with R

You might use R to download files, so that the downloading process is included in the processing script.

- A basic component of working with data is knowing your working directory.
- the two main commands are `getwd()` and `setwd()`.
- Be aware of the relative versus absolute paths:
  – **relative** `setwd("../")`
  – **absolute** `setwd("C:/Users/RudyR/OneDrive/Desktop")`
- Important difference in Windows: `setwd(C:\\users\\RudyR\\Desktop)`

Checking for and creating directories

- `file.exists("directoryName")`
- `dir.create("directoryName")`

```
setwd("C:/Users/RudyR/Desktop/Data Science Course/Rudys_First_Project")
if(!file.exists("data")) {
    dir.create("data")
}  ## creates a folder called data, if it doesn't already exist.
```

Getting data from the internet

- `download.file()` command.
- Even if you could do this by hand, using the command helps with the reproducibility.
- important parameters are `url`, `destfile` and `method`.
- useful for downloading tab-delimited, csv and other files.

```
fileURL <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.csv?accessType=DOWNLOAD"
download.file(fileURL, destfile = "./data/cameras.csv", method = "curl")
list.files("./data")
dateDownloaded <- date()
dateDownloaded
```

- method `curl` needs to be specified for https websites when downloading on a Mac. Not needed on windows.
- it's important to keep track of the date in which it was downloaded, as the file online could be updated.

Notes

- if the url starts with *http*, you can use download.file()
- if it starts with *https*, on windows you may be ok.
- if it starts with *https*, on Mac you need to set `method = "curl"`
- if the file is big, this might take a while.
- be sure to record the date when you downloaded it.