**About this course**

* Covers the basic ideas of getting data ready for analysis
  + - Finding and extracting raw data
    - Tidy data principles
    - Practical implementation
* What would be useful
  + - Exploratory analysis
    - Reporting data and reproducible research

**What you wish data looked like**

* A matrix, with rows and columns
* However raw data looks completely different

**Where is data?**

* mySQL
* MongoDB
* Literally everywhere
* <https://data.baltimorecity.gov>

**The goal of this course**

Raw data 🡺 processing script 🡺 tidy data 🡺 data analysis 🡺 data communication

**Raw and Processed Data**

* **Definition of data**

“Data are values of qualitative or quantitative variables, belonging to a set of items.”

* **Raw vs. Processed data**
  + - **Raw data**
      * The original source of data
      * Hard to use
      * Data analysis includes processing
      * Raw data may only need to be processed once
    - **Processed data**
      * Data is ready for analysis
      * Processing include

Merging

Subsetting

Transforming

Etc.

* + - * There may be standards for processing
      * All steps should be recorded

**Components of Tidy Data**

* **The 4 things you should have**
  + - The raw data
    - A tidy dataset
    - A code book describing each variable and its values in the tidy dataset
    - An explicit and exact recipe you used to go from 1 🡺 3
* **The raw data?** 
  + - Ran no software on the data
    - Did not manipulate anything
    - Did not remove anything
    - Did not summarize the data in anyway
* **The tidy data**
  + - Each variable measured should be in ONE column
    - Each different observation of that variable should be in a different row
    - There should be 1 table for each “kind” of variable
    - If you have multiple tables, they should include a column in the table that allows them to be linked
* **The code book?**
  + - Information about the variables in the datasets not contained in the tidy data
    - Information about the summary choices you made
    - Information about the experimental study design you used
  + *Others*
    - Should be word/txt file
    - Should have “study design”, describe how you collected the data
    - Must have “code book” explaining each variables and its units
* **The instruction list**
  + - A computer script
    - The input is raw data
    - The output is the tidy data
    - No parameters to the script

**Downloading files**

* **Get/set your working directory**
  + - Getwd() and setwd()
      * Relative – setwd(“./Data”), setwd(“../”)
      * Absolute – setwd(“/users/doant/Data”)
* **Checking for and creating directories**
  + - File.exist(“directoryName”) check to see if the directory exists
    - Dir.create(“directoryName”) will create a directory if it doesn’t exist
* **Getting data from the internet**
  + - **download.file()**
    - url, destfile, method
    - Should use this so that it is reproducible

fileUrl <- “https://vudoanthe1910.gq”

download.file(fileUrl, destfile = “./data/camera.csv”, method = “curl”)

list.files(“./data”)

dateDownloaded <- date()

**Reading local (flat) files**

* **read.table()** 
  + - related: read.csv()
    - parameters: file, header, sep, row.names, nrows

cameraData <- read.table(“./data/camera.csv”, sep = “,”, header = TRUE)

head(cameraData)

* Some other important parameters
  + - quote
    - na.strings
    - nrows
    - skip

**Reading Excel files**

fileUrl <- “https://vudoanthe1910.gq”

download.file(fileUrl, destfile = “./data/camera.xls”, method = “curl”0

list.files(“./data”)

dateDownloaded <- date()

library(xlsx)

cameraData <- read.xlsx(“./data/cameras.xlsx”, sheetIndex = 1, header = TRUE)

* **Further notes**
  + - XLConnect package
    - XLConnect vignette

**Reading XML**

* **XML**
  + - Extensible markup language
    - Frequently used to store structured data
    - Particularly widely used in internet applications
    - Extracting XML is the basis for most web scrapping
    - Components
      * Markup – labels that give the text structure
      * Content – the actual text of the document
* **Tags, elements and attributes**
  + - **Tags correspond to general labels**
      * Start tags <section>
      * End tags </section>
      * Empty tags <line-break/>
    - **Elements are specific examples of tags**
      * <greeting> Hello World! </greeting>
    - **Attributes** 
      * <img src = “vu.jpg” alt = “student”>
      * <step number = “3”> Connect A to B. </step>
* **Read the file into R**

Library(XML)

fileUrl <- <https://vudoanthe1910.gq>

doc <- xmlTreeParse(fileUrl, useInternal = TRUE)

rootNode <- xmlRoot(doc)

xmlName(rootNode)

* **Directly access parts of the XML document**

rootNode[1]

* **Extract parts of the file**

xmlSapply(rootNode, xmlValue)

* **XPath** 
  + - /node
    - //node
    - Node[@attr-name]
    - Node[@attr-name = “bob”]

**Reading JSON**

* **JSON**
  + - Javascript Object Notation
    - Lightweight data storage
    - Common for applications (APIs)
    - Similar to XML, but different format
    - **Data stored as**
      * Numbers
      * Strings
      * Boolean
      * Array
      * Object
* **Reading from JSON**

Library(jsonlite)

jsonData <- fromJSON(<https://dadada>)

names(jsonData)

names(jsonData$owner)

jsonData$owner$login

myjson <- toJSON(iris, pretty = TRUE)

cat(myjson)

iris2 <- fromJSON(myjson)

head(iris2)

**The data.table Package**

* **Data.table** 
  + - Inherits from data.frame
      * All functions that accept data.frame work on data.table
    - Faster

Library(data.table)

DF = data.frame(x = rnorm(9), y = rep(c(“a”, “b”, “c”), each = 3), z = rnorm(9))

head(DF, 3)

* **See all the data tables in memory**

Table()

* **Subsetting rows**

DT[2, ]

DT[, c(2, 3)]

* **Column subsetting in data.table**
* **Calculating values for variables with expressions**

DT[ , list(mean(x), sum(z))]

DT[ , table(y)]

* **Adding new columns**

DT2 <- DT

DT[ , y := 2]

* **Multiple operations**

DT[ , a := x > 0]

* **plyr like operations**

DT[, b := mean(x + w), by = a]

* **Special variables**

.N – an integer length 1 containing the number

Set.seed(123)

DT <- data.table(x = sample(letters[1:3], 1E5, TRUE)

DT[, .N, by = x]

* **Keys**

DT <- data.table(x = rep(c(“a”, “b”, “c”), each = 100), y = rnorm(300))

Setkey(DT, x)

DT[“a”]

* **Joins**

DT1 <- data.table()

DT2 <- data.table

Setkey(DT1, x)

Setkey(DT2, x)

Merge(DT1, DT2)

* **Fast reading**

Big\_df <- data.frame()

File <- tempfile()

Write.table(big\_df, file = file, row.names = FALSE, col.names = TRUE, sep = “\t”, quote = FALSE)

System.time(fread(file))