BINF 5003: Data Mining, Modeling, and Biostatistics

Week 2

Module 2 – Data Wrangling

Overview

- Working with data
  - Best practices and formats

- Data wrangling and tidying data

Refresher

- R is a programming language, RStudio is the program we use to write and run code
- R is an object-oriented language
- Functions are commands that act on objects

Working with data

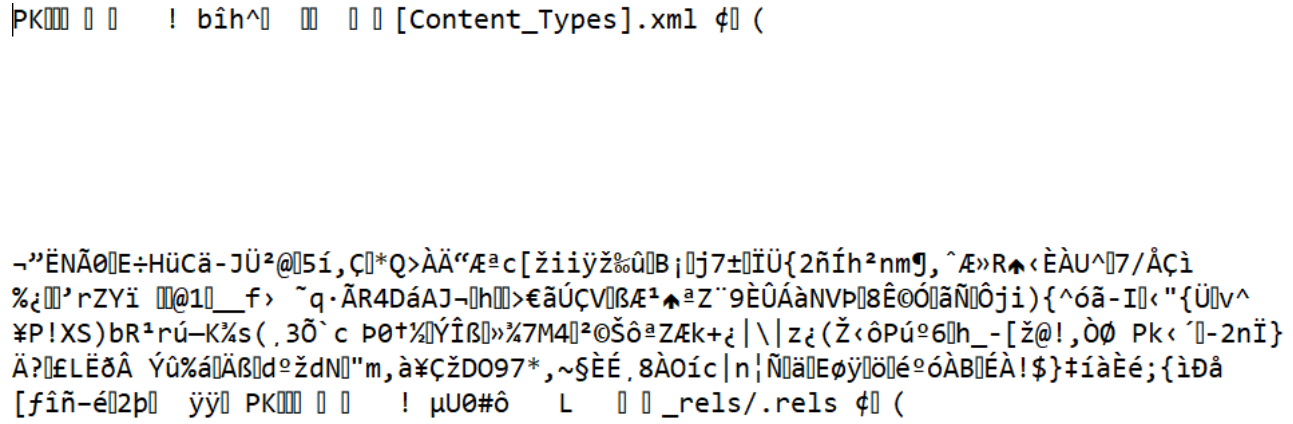
- Data can be created in R or read in from external files

- More likely, you will collect data from a different source and **import** it into R for analysis

Why not use Excel?

- Uses a lot of memory in working with tables
- Uses more space when storing .xlsx files
- Multiple tabs can make accessing data difficult
- Equations are not open-source, more restricted in customization
- Rich text format

	A	B	C	D	E	F	G	H
1	country	iso2	iso3	year	new_sp_r	new_sp_r	new_sp_r	new_sp_r
2	Afghanist	AF	AFG	1980	NA	NA	NA	NA
3	Afghanist	AF	AFG	1981	NA	NA	NA	NA
4	Afghanist	AF	AFG	1982	NA	NA	NA	NA
5	Afghanist	AF	AFG	1983	NA	NA	NA	NA
6	Afghanist	AF	AFG	1984	NA	NA	NA	NA
7	Afghanist	AF	AFG	1985	NA	NA	NA	NA
8	Afghanist	AF	AFG	1986	NA	NA	NA	NA



Comma separated values, .csv files

- Rows are separated by hard enters
- Columns are separated by commas
- Be careful to not use special characters in your column names or data collection!

```
"country","iso2","iso3","year","new_sp_m014","new_sp_m1524","new
_sp_m2534","new_sp_m3544","new_sp_m4554","new_sp_m5564","new_sp_
m65","new_sp_f014","new_sp_f1524","new_sp_f2534","new_sp_f3544",
"new_sp_f4554","new_sp_f5564","new_sp_f65","new_sn_m014","new_sn
_m1524","new_sn_m2534","new_sn_m3544","new_sn_m4554","new_sn_m55
64","new_sn_m65","new_sn_f014","new_sn_f1524","new_sn_f2534","ne
w_sn_f3544","new_sn_f4554","new_sn_f5564","new_sn_f65","new_ep_m
014","new_ep_m1524","new_ep_m2534","new_ep_m3544","new_ep_m4554"
,"new_ep_m5564","new_ep_m65","new_ep_f014","new_ep_f1524","new_e
p_f2534","new_ep_f3544","new_ep_f4554","new_ep_f5564","new_ep_f6
5","newrel_m014","newrel_m1524","newrel_m2534","newrel_m3544","n
ewrel_m4554","newrel_m5564","newrel_m65","newrel_f014","newrel_f
1524","newrel_f2534","newrel_f3544","newrel_f4554","newrel_f5564
","newrel_f65"
"Afghanistan","AF","AFG",1980,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,N
A,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA
,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA
```

Importing data

- R can access lots of different data types, plain text formats are preferred for reliability and consistency

```
abalone <- read.csv("abalone.csv", header = TRUE)
```

Object — **Function** — **Parameter: File path** — **Parameter: Column names**

Remember, learning R is learning to problem solve and trouble shoot.

It is not about memorizing all available functions or parameters – reference the documentation whenever you want!

Data structures within R

- Vectors are 1 dimensional
- Each item has a position
- Class can be numeric, character, or logical

- Data frames are 2 dimensional
- Positions specify a row and column
- Each column is like a vector, holds one type of data

```
> fruits <- c("orange", "apple", "banana", "grapefruit",
 "starfruit")
> fruits
[1] "orange"     "apple"      "banana"     "grapefruit"
[5] "starfruit"
```

```
> firstDF
  first5      fruits
1      1      orange
2      2       apple
3      3      banana
4      4  grapefruit
5      5   starfruit
```

Accessing data

- Also called indexing
- Use square brackets to specify position

- One number indicates the position

```
> fruits <- c("orange", "apple", "banana", "grapefruit",
 "starfruit")
> fruits
[1] "orange"     "apple"      "banana"     "grapefruit"
[5] "starfruit"
> fruits[3]
[1] "banana"
```

- Two numbers specify the row and then column position

```
> firstDF
  first5     fruits
1      1     orange
2      2      apple
3      3     banana
4      4 grapefruit
5      5  starfruit
> firstDF[3, 2]
[1] "banana"
```

Is there a relationship between head size and brain weight?

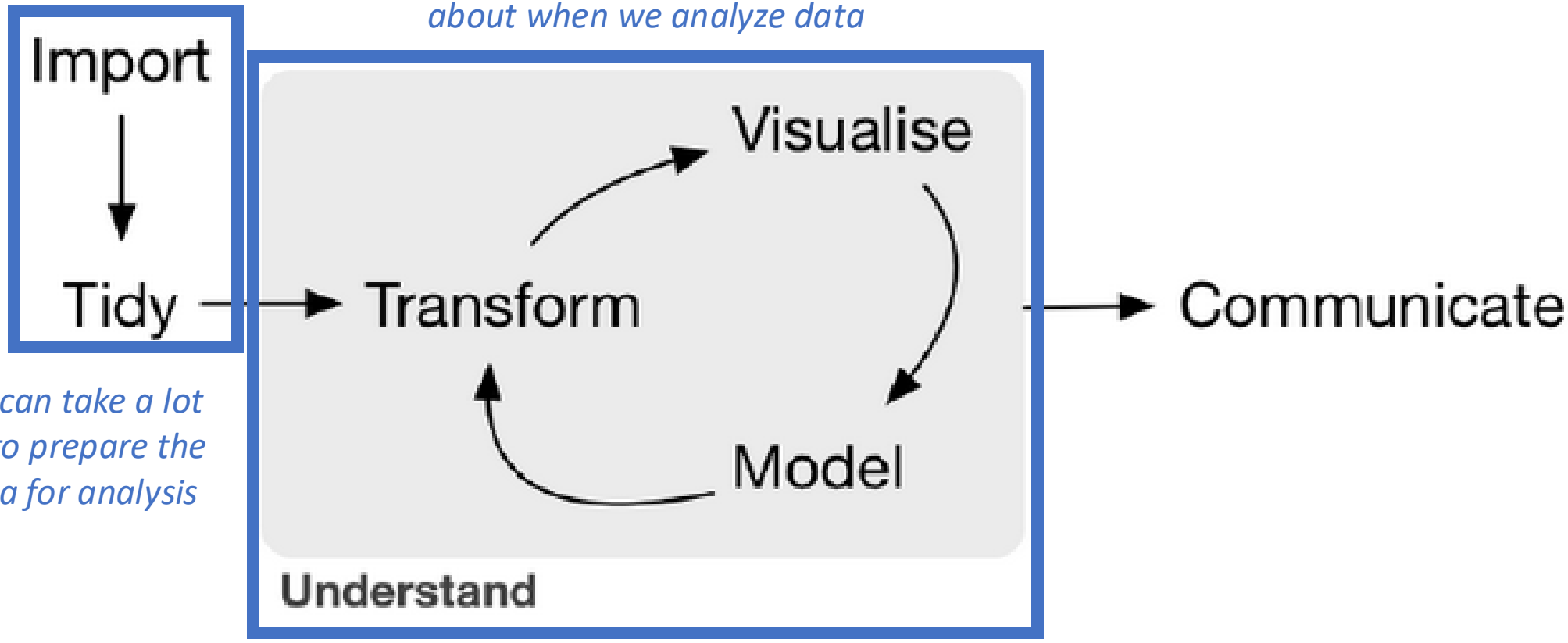
Need computational tools to analyze large datasets!

Head Size(cm^3)	Brain Weight(grams)
4512	1530
3738	1297
4261	1335
3777	1282
4177	1590
3585	1300
3785	1400
3559	1255
3613	1355
3982	1375
3443	1340
3993	1380
.	.
.	.
.	.

https://www.injurymap.com/free-human-anatomy-illustrations

Finding patterns in data is fun!

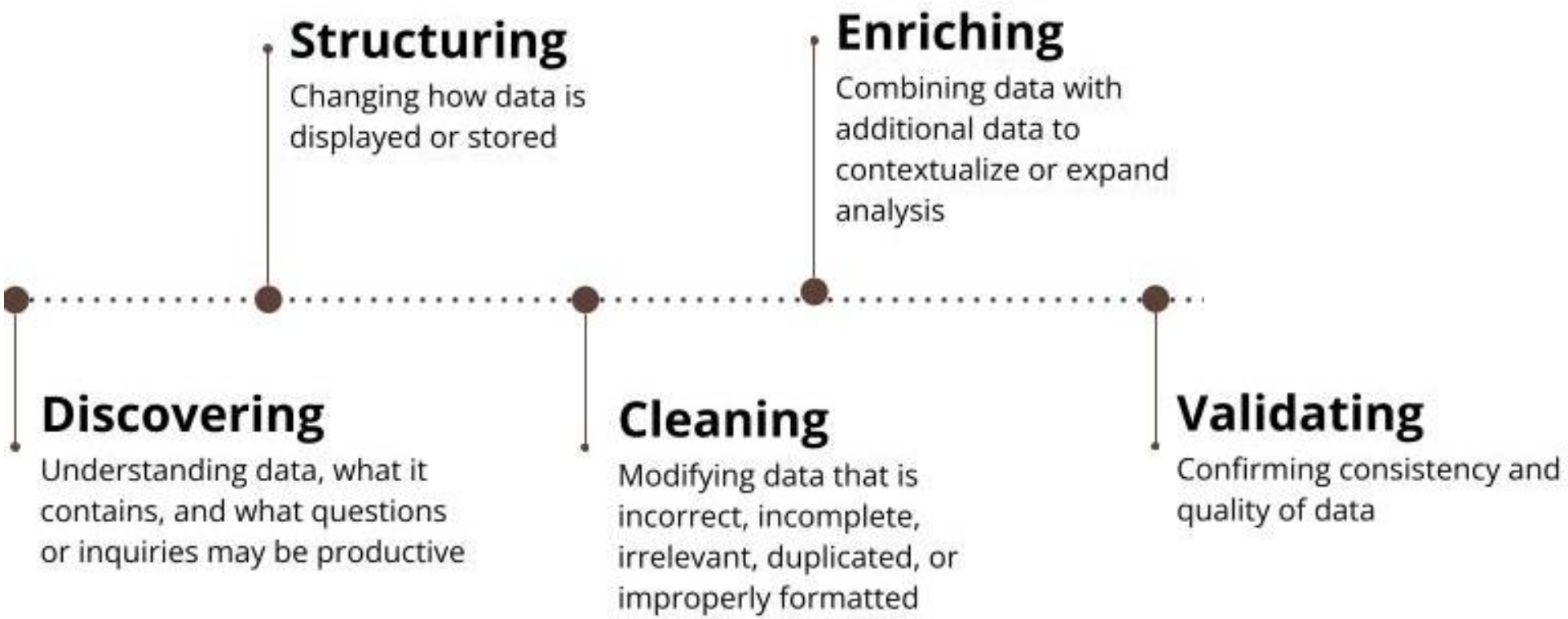
This is the part we normally think about when we analyze data



… but it can take a lot of time to prepare the raw data for analysis

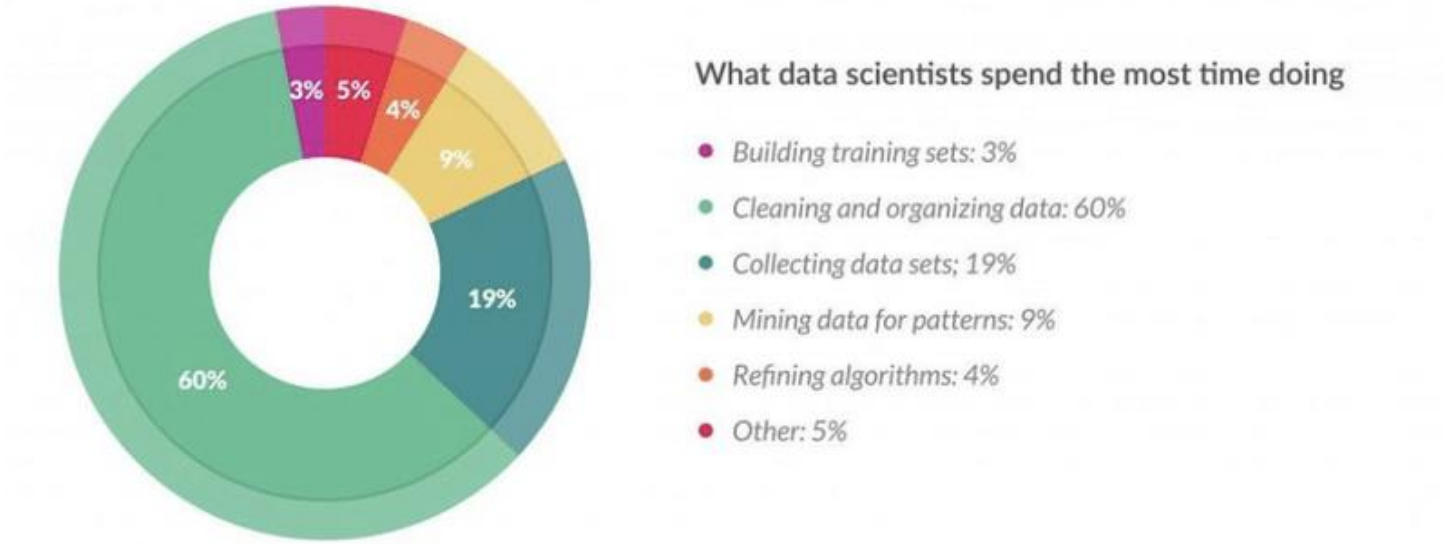
"Playing the whole game": A data collection and analysis exercise with Google Calendar

Data Wrangling Workflow



Data wrangling can often be a large component of the total analysis

Data preparation *accounts for about 80% of the work of data scientists*

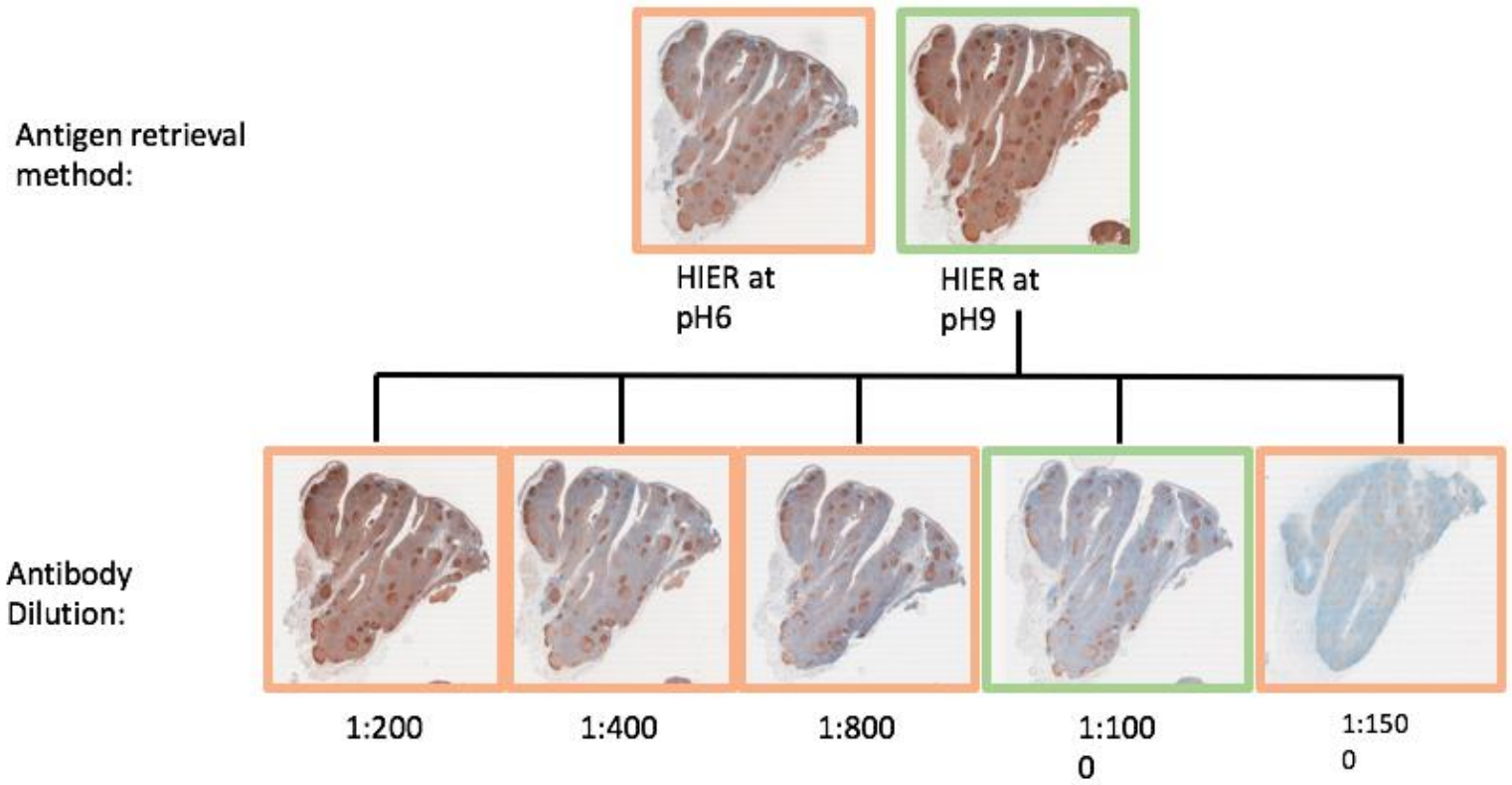


"Playing the whole game": A data collection and analysis exercise with Google Calendar

Data wrangling can be frustrating

- There is no set formula for data wrangling
  - Depends on how the data is collect, what tools you want to use for the analysis
- Often is a very time intensive and iterative process
- Much of the work will not go into the “final product”
  - Ex. often in the supplemental figures rather than main figures of publications

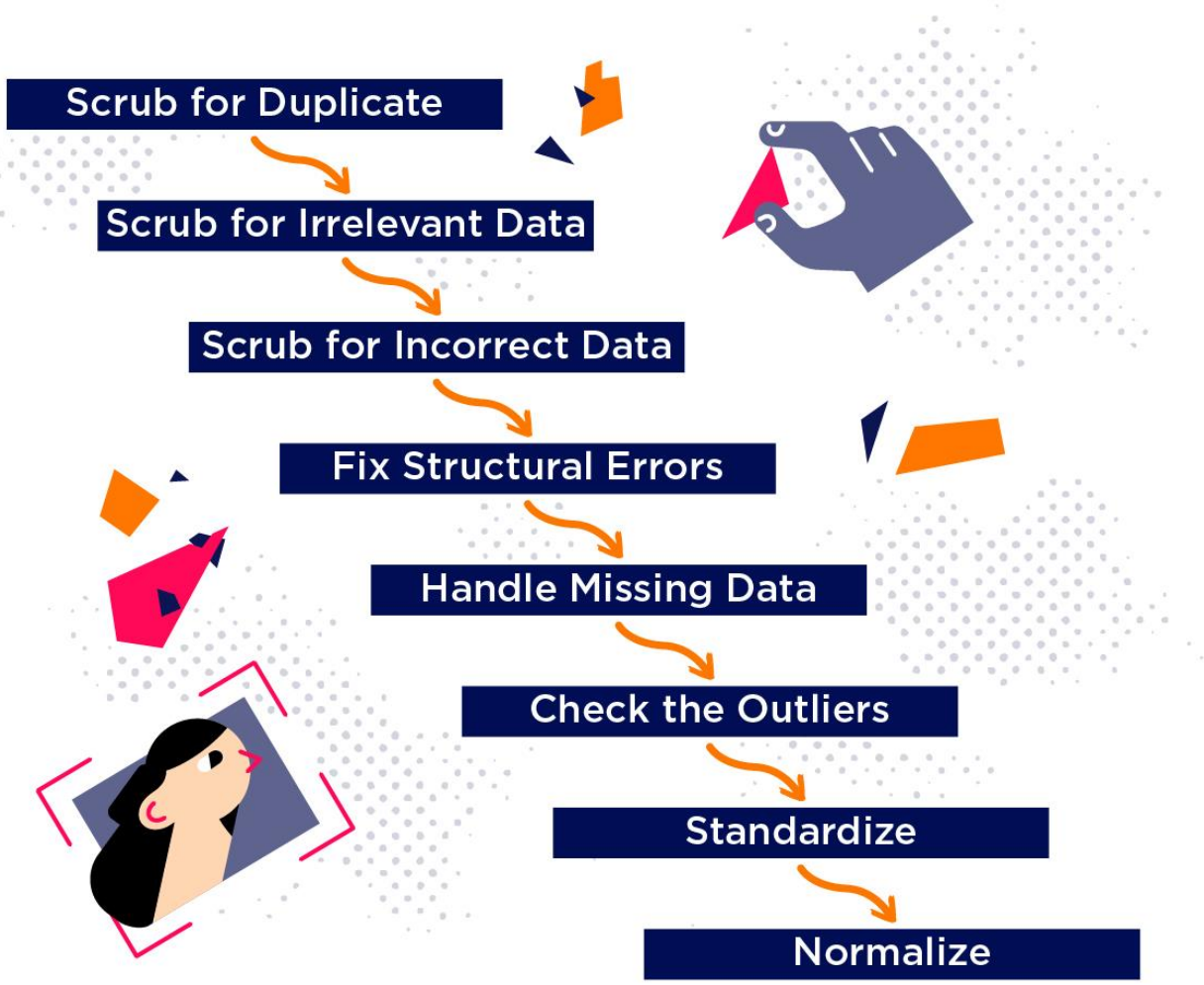
Optimizing your workflow



Optimisation of chromogenic IHC for the Activation-Induced Cytidine Deaminase (AID) Antibody on Tonsil Tissue

https://www.cancer.ox.ac.uk/support/THL/Ab-list

What to look for in your dataset



https://www.iteratorshq.com/blog/data-cleaning-in-5-easy-steps/

Structure of tidy data (for computers!)

1. Variables make up the columns
2. Observations make up the rows
3. Values go into cells
4. Reduce redundancy

Wrap up

- Plain text formats are most reliable and preferred
  - Ex. “.csv” files with “read.csv()”
- Data analysis is to find patterns and trends, often for prediction
- Data wrangling is reshaping and cleaning the data to prepare it for analysis
  - Structure of tidy data

General check in

- You're learning many different languages right now – this can be difficult!

- Some concepts will overlap and be reinforced in multiple classes
  - R and Python both have a working directory the respective program is looking at for reading and writing files to your computer

- Other concepts will be different and can make it more difficult to remember
  - The functions for recalling or specifying the working directory are different
  - R indexes from 1, Python indexes from 0