Welcome to Week 3!

1. Make sure you have the session 3 practice materials downloaded from the webpage

   https://ucb-psychology-quack.github.io/site/summer_bootcamp/bootcamp

2. Open up `s3_starter_coder.R` in Rstudio and get started with the warm-up

Week 3: Data processing

Elena & Willa

7/20/2021

Some housekeeping

Chat feature guidelines:

Group messages: Use group chat to participate in the discussion, share thoughts on the content, and ask *content* questions.

Private DMs: DM whichever of us is not presenting if you run into errors. Please don't message the whole group, it is distracting to others.

Unsolved errors: If we don't answer your DM or can't solve your error just focus on the demo so you don't miss anything. Stay in the main room at the beginning of group time and we will help you then.

Other questions: If you have other questions not related to the demo save them for after the demo or after the session. Try not to ask them during the demo or you might miss important info.

Today's agenda

- Warm-up
- What is tidy data
- Demo - Introducing tidyverse and the pipe operator
- Group work - Data organization and processing
- Discussion

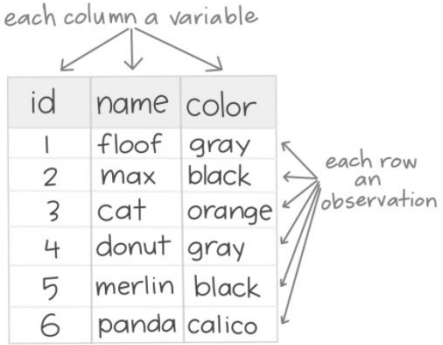
What is “tidy data”?

> “TIDY DATA is a standard way of mapping the meaning of a dataset to its structure.”
>
> –HADLEY WICKHAM

Organizing a dataset this way makes it easy to interpret

In tidy data:

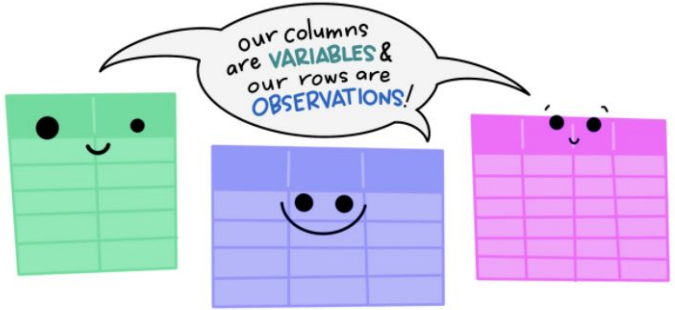
- each variable forms a column
- each observation forms a row
- each cell is a single measurement



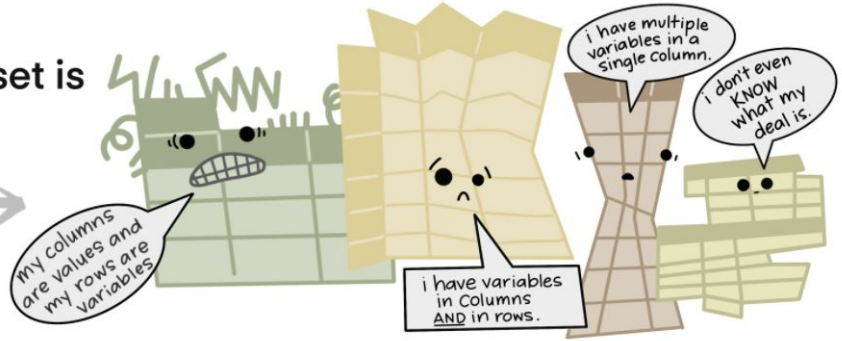
Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

https://github.com/allisonhorst/stats-illustrations

The standard structure of tidy data means that "tidy datasets are all alike..."



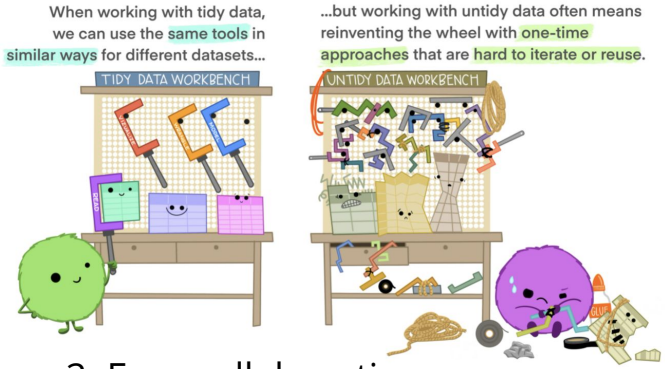
"...but every messy dataset is messy in its own way."
–HADLEY WICKHAM



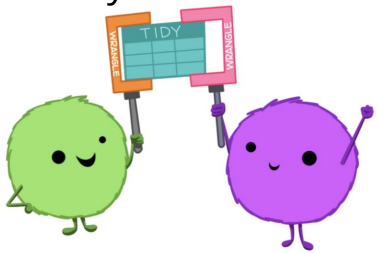
https://github.com/allisonhorst/stats-illustrations

Why do we want tidy data?

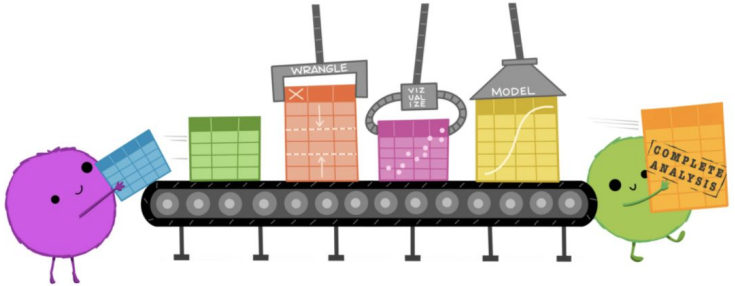
1. Reproducible code (fewer errors)!



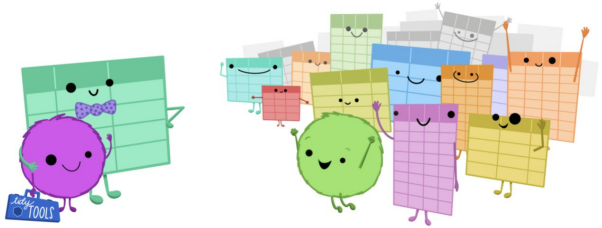
2. Easy collaboration



3. Automated pipelines (efficient and consistent!)



4. Data sharing (easy to interpret and combine with other data)



https://github.com/allisonhorst/stats-illustrations

Is our data tidy?

Open `penguins.csv`

Check the basic structure

- Is every column a variable?
- Is every row an observation?
- Is every cell one value?

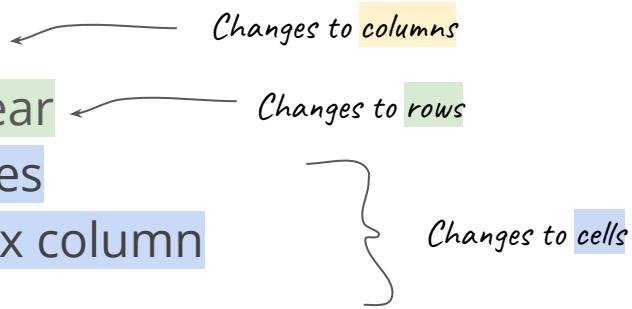
An observation might mean something different for different data! Here each penguin is an observation.

We're in good shape but there is still more processing to do to get the data we want for our analyses.

Open penguins_cleaned.csv

What are some of the differences between these two dataframes?

1. Selected only a few of the variables
2. Filtered observations by a specific year
3. Removed subjects with missing values
4. Changed the values of cells in the sex column



It's often useful to follow this hierarchy when *removing* data

There is an easy way to do all this in R!

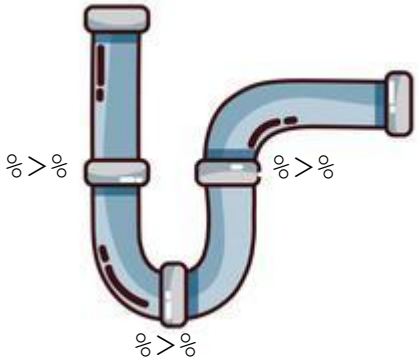
Introducing our favorite library: Tidyverse!

Blast off into the...

- A *library* is an organized collection of code and functions written by other members of the R community.
- `Tidyverse` is a library created specifically for organizing and processing your data
  - *Includes dplyr, ggplot etc*
- Install `tidyverse` and unlock a whole new world of functions and commands.

A new operator: Pipes %>%

- Once you have installed tidyverse you have access to a new symbol: %>%
- The pipe operator (%>%) allows you to string together many functions on the same data frame.
- This lets you make a workflow of tasks that you perform sequentially on a dataframe.



Let's remember the steps we want to perform on the penguins dataset

1. Select only a few of the variables
2. Filter observations by a specific year
3. Remove NAs
4. Change the values of cells in the gender columns.

In R we can combine these steps using the %>% operator and save it all as a new dataframe.

```
penguins <- read.csv("penguins.csv")

penguins_final <- penguins %>%
  "Select certain columns" %>%
  "Filter by a specific year" %>%
  "Remove missing values" %>%
  "Change the cell values for the gender variable"
```

New df is original df

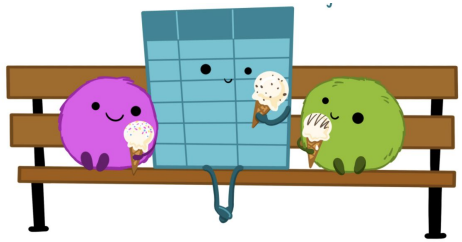
then

note: this is called "*pseudo code*". We'll replace the highlighted sections with real tidyverse commands in R

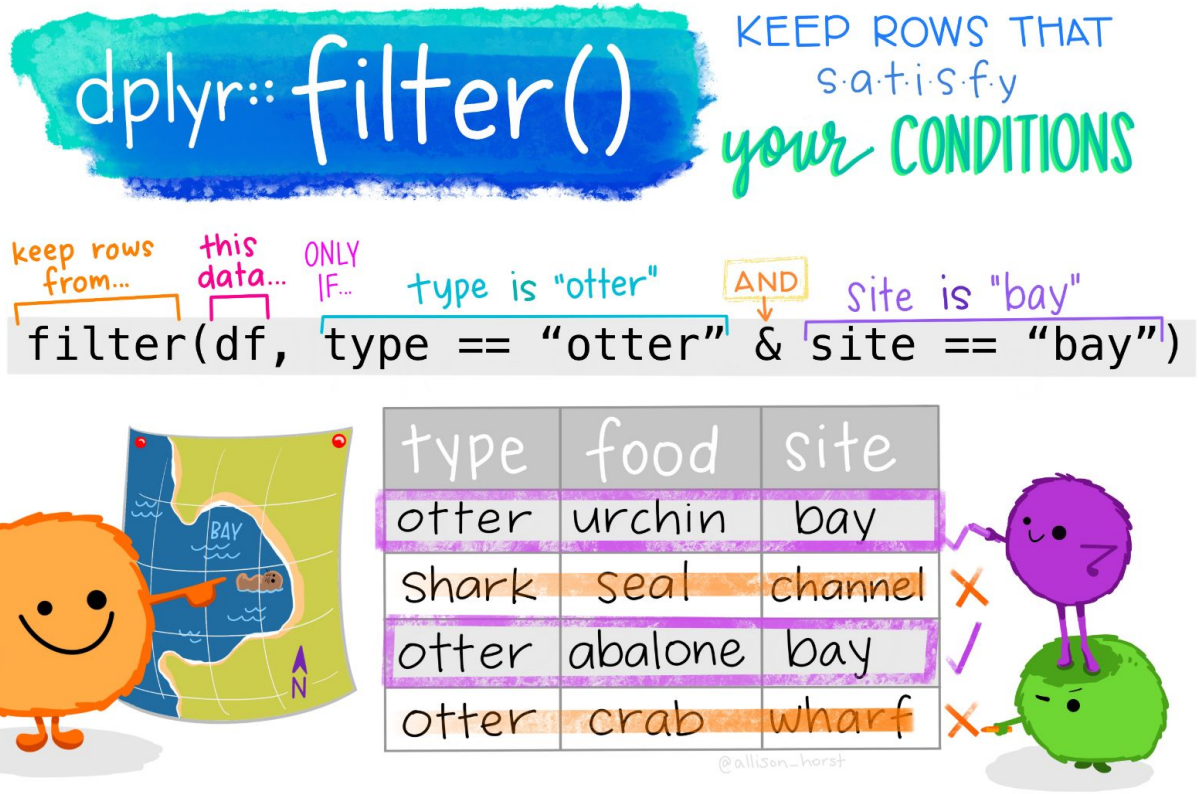
Pipes help make your code:

- *Reproducible*
- *Readable*
- *Easy to automate*

Tidy data and happy collaborators



Now lets venture into the tidyverse…



Allison Horst

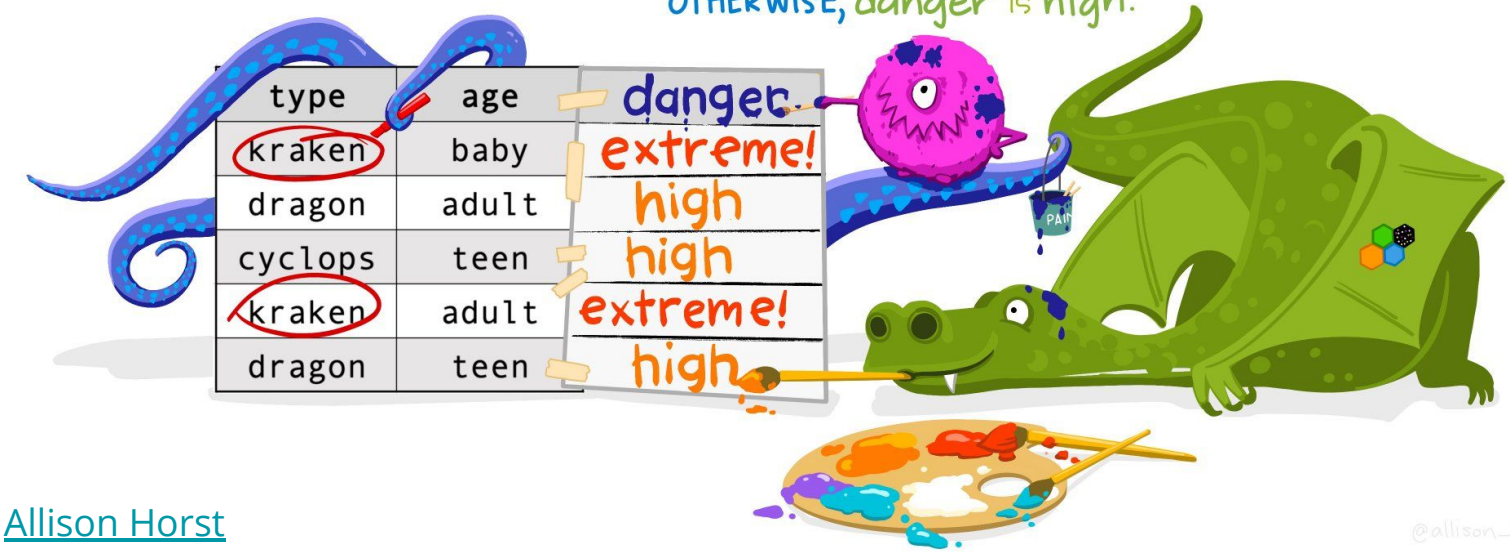
Allison Horst

dplyr::case_when()

IF ELSE...
(but you love it?)

```
df %>%                                  # ADD COLUMN 'danger'
  mutate(danger = case_when(type == "kraken" ~ "extreme!",   # IF type is kraken THEN danger is extreme!
                            T ~ "high"))                     # OTHERWISE, danger is high.
```

type	age	danger
kraken	baby	extreme!
dragon	adult	high
cyclops	teen	high
kraken	adult	extreme!
dragon	teen	high



Allison Horst

Group Activity

Activity:
https://docs.google.com/document/d/1QsnNf7t5AqbQtJwNzQ_eIYQf1Ws-3O1mCuzdRJDDdXg/edit?usp=sharing

Groups:
https://docs.google.com/spreadsheets/d/1QrJg_aJkbehIQXNdAX10GCc2s0cmlySKZMYx4UL7vjA/edit?usp=sharing

Help sheet:
https://docs.google.com/document/d/1cx-Ilgynv4U7PsobJPuVtDPZNXUH-vUr7-KejTYZ6ms/edit