**principles of tidy data**

* standard/consistent way to organize data in datasets
* facilitate data exploration
* principles ([Wickham](https://vita.had.co.nz/papers/tidy-data.pdf))
  1. Each variable must have its own column.
  2. Each observation must have its own row.
  3. Each value must have its own cell.
* what are mess datasets?

**reproducibility/transparency**

* why do we write code?
* code should be "self evident" but you should comment on "the why" and to break into smaller units/chunks of code
* reproducibility: why?
  + Can you really trust the results?
  + Are the results accurate?
  + What assumptions were made along the way?
  + Can you audit the report/study?
  + Was it done efficiently?
* Add README.md file
* Share code/results in dynamic web-based documents, eg Github

**workflow basics**

* Use of R projects in R studio
* Use of here
* Human readable file structure
  + Do not dump your scripts into a folder
* Integration with github
* Don't use absolute paths (avoid use of setwd() and rm(lists=ls()))
* never modify raw data
* consistent, intuitive, and logical folder (not to mention variable naming (lower/snake case/ camel case ))
* break out long code into multiple scripts
* separate out code, data, and other documents
* Add README.md documents to explain purpose/work/context/nuances/other info AND keep it up to date!

**data visualizing (basics)**

* benefits of visualizing (e.g. Anscombe's quartet)
* value of seeing data while working with it, understand distributions and outliers
* ggplot() syntax/template
* different types of plots
* aesthetics
* faceting/small multiples

**data transformation**

* working with "verbs"
* pseudo coding exercise
* from R4DS:
  + Pick observations by their values (filter()).
  + Reorder the rows (arrange()).
  + Pick variables by their names (select()).
  + Create new variables with functions of existing variables (mutate()).
  + Collapse many values down to a single summary (summarise()).

**conditions - if/ifelse statements**

* logical operators
* using ifelse() vs case\_when()
* if statements (where condition y is met do x)

**summarizing data**

- using `group\_by()`

- `summarise()` & `summarise\_at()`

- working with NAs

**reshaping data**

- melt v cast

- visualizing the changes being made

- id, value

**merging datasets**

- relational data

- keys

- join types

- benefits of `anti\_join()`

- join problems

**iterating**

- automate the boring stuff

- within console, ALT+SHIFT

- loops

- functions

- taking your loops to the next level with `purrr`

**working with string data**

- `stringr` package

- `paste()` & `str\_c()`

- basics of regex [regular expressions] and matching patterns

- `to\_lower()`

- `str\_replace()`

- `select()` opitons of `starts\_with()`, `ends\_with()`, `contains()`

**importing/exporting/data formats**

- `readr` and `readxl` packages

- `Sys.glob()`

- column types

- skip

- reading from GitHub

- exporting NAs

- reminder to never touch raw data