Resources:

* <https://towardsdatascience.com/the-ultimate-guide-to-data-cleaning-3969843991d4>

Packages:

* Tidyverse
  + Dplyr
    - Data transformation
  + Tidyr
    - Reshaping
      * Long to wide
      * Separate or unite columns
  + Readr
    - Import data and parse vectors
  + Forcats
    - Categorical variable manipulation
  + Stringr
    - Regular expressions
  + Tibble
  + Ggplot2
  + Lubridate & hms
* Codebook
* Here
* Tidylog
* labelled

Goals/Reasons:

* Improve data quality
* Change the format of data structures which contain the data to facilitate a current or future analysis
* Improve the quality and documentation of code for sharing and open-access

Concepts/Coding:

* Style guides
  + Variable naming
    - Human and machine readable
    - Anxiety vs anxiety\_likert\_1\_5
      * I would say don’t be too concerned about length of the name, so long as it doesn’t crowd up your code
  + Tidyverse
  + Google
  + Whatever you want just be internally consistent, though following a style guide will likely lead to more sharable end product
  + snake\_case
* Understanding the pipe operator as a way of having more human readable code
* Commenting as you go
* Variable/object naming
* tidy data (a main goal, especially for many analyses and ggplotting)
* data structures (tibble, data.frame, vector, lists)
* Addins:
  + Codebook:: codebook browser
* Cheatsheets:
  + Readr/tidyr
  + Stringr
  + dplyr
  + forcats
* R Projects
  + Avoids path problems and setting working directories
  + More robust code
* Each section needs an additional reading/resources page

Data:

* Find a dataset
  + Possible insert known issues
    - Missing data coded differently
    - Duplicate id values
* Factors:
  + - Specify known levels
    - Detect levels
* Strings:
  + All lower case
  + Remove padding
  + Regular Expression crash course
* Dates:
  + Formats are the trickiest thing to deal with
* Numeric:
  + Check range
  + Check integer
* Rectangling data
* Merging data sets

Tips:

* Read only for the raw data
* Create script which imports the data and cleans it.
* Codebook

Overall Structure:

* Data Quality:
  + Validity:
    - Data-Type Constraints
      * Vectors types:
        + Numeric, character, factor, integer
        + Solution: Read\_csv using cols argument
        + readr::parse\_\*() or as.integer() etc.
    - Range Constrains:
      * Dplyr::filter()
        + between()
        + is.na()
        + !
        + &
      * Colnames as meta data?
    - Mandatory Constraints
    - Unique Constraints
    - Set-Membership Constraints
    - Foreign-key constraints
    - Regular expression patterns
    - Cross-field validation
    - Gender, sex
    - Dates
    - Marital status
    - Height, weight
  + Accuracy:
  + Completeness:
  + Consistency:
  + Uniformity:
* Generate codebook:
  + Template:
    - codebook::new\_codebook\_rmd(filename = “codebook.Rmd”)
  + Variable labels,
  + Metadata (if sharing openly)

Workflow:

* Create project and subdirectories (data, scripts, plots,reports)
  + Ideally you would also setup version control for this project. I use Github, though Rstudio also supports SVN)
* Load data
  + Save file into the data folder
* Inspect data
  + Check for problems that are typically associated with certain types of data
* Clean data
  + Create a “clean\_data.R” script which should import the raw data (as read.only), clean it up and resave it as a new file in both .csv (if appropriate, or .rds). This allows for greater interoperability of your data.
* Helper\_functions.R
  + Possibly include this if you need to create your own functions
* Save\_cleaned data
* Generate codebook with labels

Sections:

1. Outline topics
   1. What is data cleaning?
      1. You are here so you recognize the importance
      2. Sharing data and code for transparency and accountability
      3. Sharing data and code speed up the dissemination of your research and improve your area.
      4. Our results are (at best) as good as the data used to generate them. Garbage in -> Garbage out
   2. Why the Tidyverse?
      1. It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data

-Dasu and Johnson, 2003

* + 1. The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

-www.tidyverse.org

* + 1. There isn’t anything that can be done by the tidyverse that isn’t done in Base R.
    2. It has been my experience that the quality and cleanliness of my code has improved drastically after I learned to use the tidyverse. Also, the speed of my scripting has increased faster than simply mastery would explain.
  1. What do I even look for?
  2. How will I know when it is clean?
  3. Whenever possible I will provide sources and recommended readings.

1. Important Concepts and Coding practices
   1. R Projects
      1. Avoid using working directories
         1. Permits easier sharing of code (absolute file paths aren’t needed)
         2. File paths change depending on operating system
         3. So code becomes more robust in the sense if being more likely to run on another computer (e.g., a reviewer or collaborators computer)
      2. Allows to change between working projects without mixing up results
      3. Organizing your project into subdirectories
         1. data
            1. raw data as read-only
            2. cleaned data for sharing and using in analyses
         2. plots
         3. scripts
            1. cleaning script
            2. analysis script
            3. report generating script
         4. misc
            1. Notes
            2. materials
         5. readme.md
   2. Packages we will focus on/use:
      1. dpyr
         1. filter()
            1. generate a subset of rows
         2. select()
            1. generate a subset of columns
         3. tidy\_selectors are helpful
         4. group\_by()
            1. Partition dataset for subsequent steps
         5. mutate()
            1. transform or create a new variable using existing variables
         6. summarise()
            1. Summarise the data
      2. Readr
         1. read\_csv() etc
      3. Tidyr
         1. gather()
         2. spread()
         3. unite()
         4. separate()
      4. forcats
      5. lubridate
      6. hms
      7. ggplot2
      8. stringr
      9. broom
   3. Piping Operator
      1. Show example
   4. Tidy Data
      1. “Happy families are all alike; every unhappy family is unhappy in its own way.” –– Leo Tolstoy
      2. “Tidy datasets are all alike, but every messy dataset is messy in its own way.” –– Hadley Wickham
   5. Review of base R and data structures
      1. Link to my shiny app
   6. Version Control
      1. Not covered here, but there are many benefits to using version control like Git and Subversion (both of which are integrated into RStudio when using R projects).
   7. Style guides
   8. Variable/object naming conventions
   9. Cheat sheets
2. Diagnosing the problem?
   1. Data profiling – Univariate Constraints
      1. For each variable
         1. Check the type
         2. If numerical, check for a valid range (if applicable)
            1. Are they Likert-type? Integer
            2. Are they a sum-score of 30 items
            3. If there is a know minimum and/or maximum value, then build in a check for that constraint
            4. Are there “logical” constraints?

Height, or weight must be non-zero positive values

* + - * 1. Scale of measurement

Height does a height value of 2.4 make sense

* + - 1. Outliers (more on this later)
         1. Trickier, as the definition for an outlier can vary depending on the source you consult
      2. Check for special values which were meant to be flags (e.g., -99,-999, etc.)
    1. Categorical columns
       1. Check that only specific groups are present (as factors should be used when there are fixed and known groups)
       2. Can check all possible groups
    2. Dates
       1. What format are the dates stored as?
          1. Integers starting from a certain date
          2. Are they character strings?
          3. Are the stored as day/month/year or month/day/year?
          4. Are they ISO8601 format?

<https://en.wikipedia.org/wiki/ISO_8601>

* + - 1. Do you know when the data were collected?
         1. Check that all dates fall inside of that range
    1. Visually
       1. Construct graphs to get broad views of your data
  1. Data Profiling – Multivariate Constraints
     1. Are there certain conditions across multiple columns which need to be met?
        1. A subjects age is 10, but their marital status is married or divorced (not likely but possible)

1. Section 2:
   1. Intro to Tidy data
   2. Spotting untidy data
      1. The Usual Suspects
      2. Variable information stored in column headers
      3. Multiple variables stored in a single column
         1. PPid
            1. 101m25
   3. Tidyr functions
      1. Gather
      2. Spread
      3. Unite
      4. Separate
      5. Pivot\_longer
      6. Pivot\_wider
   4. Sleep Deprivation Example
2. There are three options to address problematic data
   1. Fix it (might be impossible)
      1. Refer to original data collected or codebook to help resolve problem
      2. Contact participants to verify response
      3. Infer correct values
         1. Be careful
            1. Malle is probably Male
            2. Be fmale coule be female by adding a letter, or male by removing a letter
   2. Ignore it (Denote it as missing)
      1. If you can’t be sure what the correct datum should be, then that means it is missing and your software should note it as such.
   3. Impute a sensible value
      1. Multiple imputation
      2. Hot-deck
      3. Model the values
3. Report
   1. Note all of the steps
   2. Generate a codebook
      1. Variable labels
      2. Cleaning script which takes raw data and give an end product using an interoperable file format and an rds (which retains the structure of the data)
   3. Check to ensure that the problems were addressed aka that the data are as clean as possible