ABACUS 9/16/21 – Tidy Data Tutorial

Adapted from *“Data Science with R”* by Garrett Grolemund

Welcome! Let’s start from basics and use an example dataset to demonstrate the advantage of “tidy” data.

1. Download R. Current version is R 4.1.1 (https://cran.r-project.org/bin/windows/base/) or (<https://cran.r-project.org/bin/macosx/>)
2. Download an interactive environment called RStudio. (https://www.rstudio.com/products/rstudio/) This is also packaged in Anaconda Navigator.
3. Open RStudio. Find the four interface windows: console, environment, files/plots/packages, and scripting.
4. The environment is a workspace that shows the active objects in your project. View the data by pressing the grid icon button. The scripting window accepts code. To run a line, click “Run” or hit command-enter.
5. By default files will be written to “Users” folder. To keep your project organized in a single folder set a “working directory”.
   * # getwd()
   * # setwd(“~/pathtodir/directory”)
6. Create a new R script using the toolbar in the upper left of the RStudio window.
7. The “tidyverse” is a collection of tools for data handling that needs to be brought into the RStudio interface. Use the UI and navigate to “packages” and install “tidyr” and “devtools” or use the following commands:
   * # install.packages(“tidyr”)
   * # install.packages(“devtools”)
8. The data we will use is a packaged dataset “DSR” on github, a hosting platform for data and software.
   * # install\_github("kylersc/DSR", force = TRUE)
   * # library(DSR)
   * # library(tidyr)
9. The DSR library contains datasets in different formats. Call the tables up using table1 through table6.
10. These are dataframes, 2D arrays where columns contain variables and each row is a corresponding value. Dataframes can be created, expanded/contracted, and data can be extracted.
11. Try a built-in dataframe “mtcars”
    * # library(dplyr)
    * # data() # These are all included datasets that are great practice for manipulating dataframes.
12. Diagram

    Description automatically generatedFor a dataset to be “tidy” it should follow 3 rules. “Tidy datasets are all alike but every messy dataset is messy in its own way.” – Hadley Wickham
    * Each column is a different variable.
    * Each row is a different observation.
    * Each cell is a different value.
13. Table 1 is tidy. Let’s see how this makes it easy to use simple but powerful operators. Calculate the rate of TB cases per country per year (#people per 10k diagnoses) :
    * # table1$cases
    * # mean(table1$cases)
    * # table1$cases / table1$population \* 10000
14. Try replicating this calculation with tables 2, 3, and 4.
15. In the tidyr package, the “gather” and “spread” commands are useful to alter the layout of the data table without disrupting the variable:value relationships. These are referred to as a key:value pair.
16. “Spread” turns the key:value columns into a tidy column by adding columns named with each value.
    * # table2
    * # spread(table2, type, count)
17. “Gather” does the opposite, making a “key” column with a new name from a set of column names and allows you to specify the new key:value pair.
    * # table4
    * # gather(table4, “year”, “cases”, 2:3)
18. “Spread” and “Gather” reshapes the data layout to address columns=variable and rows=observations. To address cell=single value we need to be able to split a single character column into multiple columns and vice versa for each corresponding variable. This is done with “Separate” and “Unite”.
19. In table3, the cases and population values are in the same column. These are discrete values, and we use “separate” to give each its own column.
    * # separate(table3, rate, into = c(“cases”, “population”))
20. In table6, the year is split into two columns. Use “unite” to create a new column containing single values.
    * # unite(table6, “year”, century, year, sep = “”)

These are basic functions for tidying data. They can be customized by including more arguments in the commands, the best way to understand the different features and options is to explore the documentation for each command.

For more practice with a larger dataset, try an example from “*Data Science in R”* using the “who” dataset.Included in the DSR package is a raw data set from the WHO Global Tuberculosis Report. It is labeled “who” and can be called with:

* + # View(who)

For more tutorials into tidying data, try the following webinars:

https://www.datacamp.com/community/tutorials/tidyverse-tutorial-r

https://www.rstudio.com/resources/webinars/a-gentle-introduction-to-tidy-statistics-in-r/