Cleaning Messy Data in R

R-Ladies St. Louis

Crystal Lewis

11/12/2019

Agenda

- About me
- What is messy data
- What is clean data
- Common data cleaning steps
- Other data cleaning tips
- Let’s clean some data

About me

- Missouri Prevention Science Institute, University of Missouri, Columbia, in the College of Education
    - Director, Data Strategy
- I’ve also done some teaching at LaunchCode, CoderGirl Data Analysis track
- I’m on the R-Ladies Committee for the UseR! 2020 Conference here in St. Louis
    - Need for abstract mentors https://forms.gle/v7X8agaa6EaHEDDz7
- rstudio::conf 2020 diversity scholarship recipient

Messy data

- Most people have heard of the 80/20 dilemma
- Forbes article
    - 76% of data scientists view data preparation as the least enjoyable part of their work
- I actually enjoy it…..sometimes!

Why should you clean your data?

- Towards Data Science article
- Garbage in, Garbage out
- Quality data beats fancy algorithms
- You want data that is:
  - Valid - Does the data conform to constraints
  - Accurate - Is the data true
  - Complete - Missingness will happen but there are ways to mitigate this
  - Consistent - Is data consistent across variables
  - Uniform - Are all units within a column and across datasets measured the same way

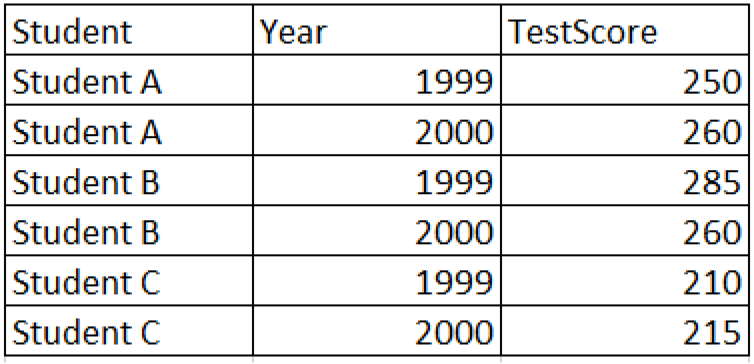
Other good data cleaning rules

- Descriptive variable names with no spaces
- Only one piece of information per column
- No characters (ex: $) in columns unless it is a string variable
- No unclear values in cells
- No duplicate entries
- De-identify data if necessary (drop columns)

How should you structure your data?

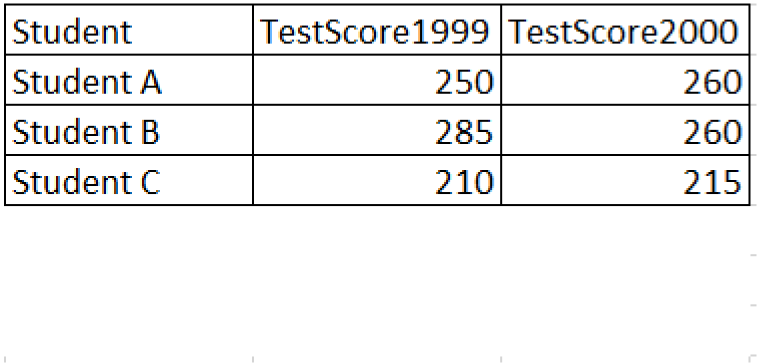
- Tidy Data (Table 1) vs. (Table 2)

Table 1



Student	Year	TestScore
Student A	1999	250
Student A	2000	260
Student B	1999	285
Student B	2000	260
Student C	1999	210
Student C	2000	215

Table 2



Student	TestScore1999	TestScore2000
Student A	250	260
Student B	285	260
Student C	210	215

Examples of messy data

- There are endless types of data you may encounter
  - Spreadsheets, text files, PDFs, word documents, APIs, webscraping, databases, googlesheets, etc.
- Let's look at some examples of messy data
  - Example 1: Long variable names with spaces, unnecessary rows and cols, unclear values, col with more than one piece of info
  - Example 2: Poor structure, values as variable names, missing data (need to fill)
  - Example 3: Unstructured data
  - Example 4: Characters in numeric cols, some values not valid, unclear variables, non-uniform cols

Why use R and RStudio

- Allows us to perform cleaning tasks without ever touching the raw data
- It also allows:
  - Our cleaning to be reproducible and reusable
  - Us to export clean data or do analyses on clean data within R
  - Us to document our steps
  - Create codebooks
  - To show data descriptives in reports

My typical data cleaning steps

1. Read in file/s and explore data
2. Drop columns (Ex: De-identify data - remove name)
3. Rename columns (Ex: Descriptive names, remove spaces)
4. Filter data (Ex: remove those with missing IDs)
5. Remove duplicates
6. Transform/create cols (Ex: string->numeric, or remove $ or %)
7. Recode variables (Ex: NA->0, reverse code likert scale)
8. Transform data (Ex: from wide to long or long to wide)
9. Merge data and/or append data
10. Add variable labels
11. Make codebook
12. Export data

Cleaning steps and associated packages/functions

Step	Package::Function
Read file	readxl::read_excel; readr::read_csv
Read files from folder	list.files, lapply (base)
Explore data	dplyr::glimpse; names, str, summary, table (base)
Explore cont.	skimr::skim; janitor::tabyl
Select cols	dplyr:: select; starts_with, contains, ends_with
Rename cols	purr::set_names; setNames (base R); dplyr::rename
Filter rows	dplyr::filter

Common data cleaning (cont.)

Step	Package::Function
Remove duplicate rows	dplyr::distinct
Create/Transform cols	dplyr::mutate, stringr::str_remove, str_extract; tidyr::extract
Split column	tidyr::separate; stringr::str_split
Concatenate 2 cols	paste0 (base); tidyverse::glue
Change col class	lubridate::mdy; as.numeric, as.string, as.factor (base)
Recode cols	dplyr::recode, na_if; tidyr::replace_na; ifelse (base)
Add value labels	labelled::labelled

Common data cleaning (cont.)

Step	Package::Function
Fill missing values	dplyr::coalesce
Long to Wide data	tidyr::spread, pivot_wider
Wide to Long data	tidyr::gather, pivot_longer
Merge data	dplyr::left_join, right_join, full_join
Append data	dplyr::bind_rows
Add variable labels	labelled::var_label
Make codebook	dataMaid::makeCodebook
Export data	readr::write_csv, openxlsx::write.xlsx; writexl::write_xlsx

Tidyverse

- Most of these packages can be loaded through the Tidyverse
  - "Opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures."
  - Included packages: magrittr, ggplot2, dplyr, tidyr, readr, purr, tibble, stringr, forcats
  - Many others not loaded automatically: ex: readxl, lubridate

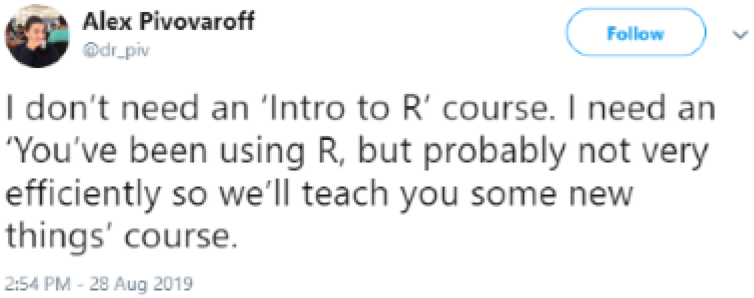
Piping

- Magrittr allows the use of piping (%>%)
    - Piping allows us to chain several chunks of code together
- I could write these 3 lines of code separately
    - data<-read_excel(“file.xlsx”)
    - data<-select(data,column1:column3)
    - data<-setNames(data, c(“ID”,“Gender”,“Test_Score”))
- OR I could use piping
    - data<-read_excel(“file.xlsx”)%>% select(column1:column3)%>% setNames(c(“ID”,“Gender”,“Test_Score”))

Other tips

- Use RProjects
  - Organize your directory within RProjects
  - Name files specifically and with no spaces
- Use RMarkdown as part of your reproducible research to showcase script, data documentation, and output all in one document
- Comment every step
- Use keyboard shortcuts
  - Alt key to highlight a column of data or make a multi-line cursor
  - Ctrl+Shift+M inserts pipe operator
  - Shift+Ctrl+R adds a header
  - Ctrl+Enter to run a line of R code

Endless ways to solve the same problem



Alex Pivovaroff
@dr_piv

I don't need an 'Intro to R' course. I need an 'You've been using R, but probably not very efficiently so we'll teach you some new things' course.

2:54 PM - 28 Aug 2019

Let’s clean some data

- Scenario: We are running a research study in a school district
- We currently have study data which includes StudyID and Treatment Status
- We want to know if treatment impacts student outcomes so we request data from the district
- We have a district Student ID in our data that we can use to merge files.
- Here is what we receive:
    - One file that includes demographics and attendance
    - One file that includes discipline referrals
- What might we need to clean in these files?
- Which file do you want to tackle first?

Contact Info

- Github: https://github.com/Cghlewis
- LinkedIn: https://www.linkedin.com/in/crystal-lewis-922b4193
- Twitter: @Cghlewis
- email: hamptoncg@missouri.edu

Other resources

- Julia Silge
- RPub, Alex Kaechele
- Tidy Data
- University of Chicago
- What they forgot to teach you about R
- Data Wrangling with Tidyverse