MATH 530/630

Integrative Lab 1 - Exploratory Data Analysis

Contents

Overview

The goal of this lab is to carefully, thoroughly, and thoughtfully conduct an exploratory data analysis (EDA). You are also asked to communicate clearly about the steps in your EDA process with others, by sharing your R code, output, and narrative. As such, your code cannot "stand alone"- it is meant to complement / enhance / support your narrative.

Tools

You will use R Markdown to construct your EDA (For additional guidance on EDAs, see: R4DS: Exploratory Data Analysis). You'll submit your work as an html file knit from your `.Rmd` file (please leave the default code chunk options for `eval = TRUE` and `echo = TRUE`). In that document, you'll use `dplyr`, `tidyr`, and `ggplot2` to do description and visualization. You may also wish to use the `janitor` package to make `tabyl`s, and some of the accompanying `adorn` functions.

Your lab should serve as your own personal cheatsheet in the future for things to do with a new dataset. Give yourself the cheatsheet you deserve! Remember:

- `rmarkdown` should be your EDA *documentation* tool
- your own words with `markdown` formatting are your ONLY *narrative* tool
- `dplyr` should be your *data manipulation* tool
- `tidyr` should be your *data reshaping* tool
- `janitor::tabyl` should be your *data table-making* tool
  - you may wish to combine with `knitr::kable()` for formatting tables
- `ggplot2` should be your *data visualization* tool

For all things, graphical and tabular, if you're dissatisfied with a result, discuss the problem, what you've tried and move on (remember my 30-minute rule). You'll need this loaded at the top:

```
library(tidyverse) # all the good stuff
library(readxl) # for reading in xlsx files
library(here) # for here!
library(janitor) # for clean_names & tabyl
library(knitr) # for kable
```

The `tidyverse` meta-package includes `dplyr`, `ggplot2`, and `tidyr`.

Data

You will start with the `reprohealth` dataset, which is distributed as an R package from Alison Presmanes Hill's github.

Install it, and remember to use this code only in your R console, not in a script or .Rmd file:

```
install.packages("remotes") # install the remotes package
library(remotes) # load remotes package so you can install from github
install_github("apreshill/reprohealth") # install the package
```

Then at the top of your lab, copy and paste this code:

```
library(reprohealth) # load the package
library(readr) # to get parse_number function
wb_reprohealth # call the data
wb_stats <- wb_reprohealth %>% # save the data to your local environment
  mutate(year = parse_number(year)) # fix year
```

You can name `wb_stats` anything else you want, this is just an example.

Explore the `wb_reprohealth` data

- How many variables/columns?
- How many rows/observations?
- Which variables are numbers?
- Which are categorical variables (numeric or character variables with variables that have a fixed and known set of possible values; aka factor variables)?
- Complete this sentence: "There is one row per…"

Explore a new dataset

Go to the gapminder site and find a new indicator that interests you.

- Download the data file to a folder in your `.Rproj` directory, and open the `.xlsx` file to see where the data is (i.e., which sheet).
- Read the data file into your EDA .Rmd document using the `readxl` package (remember, it must be installed first). Use the `here` package, with the `here` function, to build up the relative file path (example code-through). Here is an example code chunk:

```
internet <- read_xlsx(here::here("data", "Internet_user_per_100.xlsx"),
                      sheet = 1) %>%
  clean_names() # highly recommended

internet
```

```
# A tibble: 275 x 23
   internet_users_~ x1990 x1991 x1992 x1993    x1994    x1995    x1996
   <chr>            <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>
 1 Abkhazia            NA    NA    NA    NA NA       NA       NA
 2 Afghanistan          0    NA    NA    NA NA       NA       NA
 3 Akrotiri and Dh~    NA    NA    NA    NA NA       NA       NA
 4 Albania              0    NA    NA    NA NA        0.0112   3.22e-2
 5 Algeria              0    NA    NA    NA  3.61e-4  0.00177  1.74e-3
 6 American Samoa       0    NA    NA    NA NA       NA       NA
```

```
 7 Andorra              0    NA    NA    NA NA       NA        1.53e+0
 8 Angola               0    NA    NA    NA NA       NA        7.76e-4
 9 Anguilla            NA    NA    NA    NA NA       NA       NA
10 Antigua and Bar~     0    NA    NA    NA NA        2.20     2.86e+0
# ... with 265 more rows, and 15 more variables: x1997 <dbl>, x1998 <dbl>,
#   x1999 <dbl>, x2000 <dbl>, x2001 <dbl>, x2002 <dbl>, x2003 <dbl>,
#   x2004 <dbl>, x2005 <dbl>, x2006 <dbl>, x2007 <dbl>, x2008 <dbl>,
#   x2009 <dbl>, x2010 <dbl>, x2011 <dbl>
```

- Answer the same 5 questions as you did above for the `reprohealth` data.
- Gather the data (most data from gapminder has columns in years).

```
internet_tidy <- internet %>%
  rename(country = 1) %>%
  gather(year, users_per_100, -country) %>%
  mutate(year = readr::parse_number(year))
```

```
# another example dataset
age_marriage <- read_xlsx(here::here("data", "indicator_age_of_marriage.xlsx"),
                          sheet = 1)

age_marriage
```

```
# A tibble: 185 x 117
   ...1  `1616` `1666` `1685` `1710` `1716` `1735` `1760` `1766` `1775`
   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 Afgh~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 2 Alba~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 3 Alge~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 4 Ango~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 5 Arge~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 6 Arme~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 7 Aust~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 8 Aust~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 9 Azer~     NA     NA     NA     NA     NA     NA     NA     NA     NA
10 Baha~     NA     NA     NA     NA     NA     NA     NA     NA     NA
# ... with 175 more rows, and 107 more variables: `1780` <dbl>,
#   `1785` <dbl>, `1791` <dbl>, `1800` <dbl>, `1810` <dbl>, `1815` <dbl>,
#   `1825` <dbl>, `1835` <dbl>, `1840` <dbl>, `1845` <dbl>, `1855` <dbl>,
#   `1860` <dbl>, `1865` <dbl>, `1866` <dbl>, `1870` <dbl>, `1875` <dbl>,
#   `1879` <dbl>, `1880` <dbl>, `1885` <dbl>, `1887` <dbl>, `1890` <dbl>,
#   `1895` <dbl>, `1897` <dbl>, `1900` <dbl>, `1901` <dbl>, `1903` <dbl>,
#   `1905` <dbl>, `1906` <dbl>, `1907` <dbl>, `1910` <dbl>, `1911` <dbl>,
#   `1915` <dbl>, `1920` <dbl>, `1921` <dbl>, `1925` <dbl>, `1928` <dbl>,
#   `1930` <dbl>, `1931` <dbl>, `1935` <dbl>, `1937` <dbl>, `1939` <dbl>,
#   `1940` <dbl>, `1941` <dbl>, `1942` <dbl>, `1943` <dbl>, `1944` <dbl>,
#   `1945` <dbl>, `1946` <dbl>, `1947` <dbl>, `1948` <dbl>, `1949` <dbl>,
#   `1950` <dbl>, `1951` <dbl>, `1952` <dbl>, `1953` <dbl>, `1954` <dbl>,
#   `1955` <dbl>, `1956` <dbl>, `1957` <dbl>, `1958` <dbl>, `1959` <dbl>,
#   `1960` <dbl>, `1961` <dbl>, `1962` <dbl>, `1963` <dbl>, `1964` <dbl>,
#   `1965` <dbl>, `1966` <dbl>, `1967` <dbl>, `1968` <dbl>, `1969` <dbl>,
#   `1970` <dbl>, `1971` <dbl>, `1972` <dbl>, `1973` <dbl>, `1974` <dbl>,
#   `1975` <dbl>, `1976` <dbl>, `1977` <dbl>, `1978` <dbl>, `1979` <dbl>,
#   `1980` <dbl>, `1981` <dbl>, `1982` <dbl>, `1983` <dbl>, `1984` <dbl>,
#   `1985` <dbl>, `1986` <dbl>, `1987` <dbl>, `1988` <dbl>, `1989` <dbl>,
```

```
#   `1990` <dbl>, `1991` <dbl>, `1992` <dbl>, `1993` <dbl>, `1994` <dbl>,
#   `1995` <dbl>, `1996` <dbl>, `1997` <dbl>, `1998` <dbl>, ...
```

```
marriage_tidy <- age_marriage %>%
  rename(country = 1) %>%
  gather(year, age_first_marriage, -country) %>%
  mutate(year = readr::parse_number(year))
marriage_tidy
```

```
# A tibble: 21,460 x 3
   country      year age_first_marriage
   <chr>       <dbl>              <dbl>
 1 Afghanistan  1616                 NA
 2 Albania      1616                 NA
 3 Algeria      1616                 NA
 4 Angola       1616                 NA
 5 Argentina    1616                 NA
 6 Armenia      1616                 NA
 7 Australia    1616                 NA
 8 Austria      1616                 NA
 9 Azerbaijan   1616                 NA
10 Bahamas      1616                 NA
# ... with 21,450 more rows
```

Pathological in the same way as the previous...

```
# yet another example dataset
alcohol <- read_xlsx(here::here("data", "indicator_alcohol_consumption _20100830.xlsx"), sheet = 1)

alcohol
```

```
# A tibble: 189 x 25
   ...1  `1985` `1986` `1987` `1988` `1989` `1990` `1991` `1992` `1993`
   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 Afgh~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 2 Alba~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 3 Alge~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 4 Ando~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 5 Ango~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 6 Anti~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 7 Arge~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 8 Arme~     NA     NA     NA     NA     NA     NA     NA     NA     NA
 9 Aust~     NA     NA     NA     NA     NA     NA     NA     NA     NA
10 Aust~     NA     NA     NA     NA     NA     NA     NA     NA     NA
# ... with 179 more rows, and 15 more variables: `1994` <dbl>,
#   `1995` <dbl>, `1996` <dbl>, `1997` <lgl>, `1998` <dbl>, `1999` <lgl>,
#   `2000` <dbl>, `2001` <dbl>, `2002` <dbl>, `2003` <dbl>, `2004` <dbl>,
#   `2005` <dbl>, `2006` <dbl>, `2007` <dbl>, `2008` <dbl>
```

```
alc_tidy <- alcohol %>%
  rename(country = 1) %>%
  gather(year, alc_per_adult, -country) %>%
  mutate(year = readr::parse_number(year))
alc_tidy
```

```
# A tibble: 4,536 x 3
   country              year alc_per_adult
```

```
   <chr>                <dbl>          <dbl>
 1 Afghanistan           1985             NA
 2 Albania               1985             NA
 3 Algeria               1985             NA
 4 Andorra               1985             NA
 5 Angola                1985             NA
 6 Antigua and Barbuda   1985             NA
 7 Argentina             1985             NA
 8 Armenia               1985             NA
 9 Australia             1985             NA
10 Austria               1985             NA
# ... with 4,526 more rows
```

- Join this new data using `dplyr` with the `reprohealth` data you have been working with, and make some observations about the process and result. You may need:
  - This ModernDive section on joins
  - This tutorial by Jenny Bryan
  - This chapter in R4DS
  - **Hint!:** you'll probably need to use `tidyr::gather` on your new dataset first, before the join. Tread carefully here!
  - Remember, if you're dissatisfied with a result, discuss the problem, what you've tried and move on (remember my 30-minute rule).

```
# let's do this
int_repro <- wb_stats %>%
  left_join(internet_tidy)
agem_repro <- wb_stats %>%
  left_join(marriage_tidy)
alc_repro <- wb_stats %>%
  left_join(alc_tidy)
```

Explore questions

In the R4DS Exploratory Data Analysis chapter, the authors say:

> "Your goal during EDA is to develop an understanding of your data. The easiest way to do this is to use questions as tools to guide your investigation... EDA is fundamentally a creative process. And like most creative processes, the key to asking quality questions is to generate a large quantity of questions."

Your mission in this last section is to pick at least three of the tasks below and attack each with a **table and figure**. In your narrative, reword each "task" into a logical research question(s). Make observations about what your tables/figures show and about the process. If you want to do something comparable but different, i.e. swap one quantitative variable for another, be our guest! If you are feeling inspired and curious, then we're doing this right. Go for it.

Task menu

- Get the maximum and minimum of children per woman (`tot_fertility`) for all continents.
- Look at the spread of children per woman (`tot_fertility`) across countries within the continents.
- Compute a trimmed mean of maternal mortality (`mat_mortality`) for different years. Or a weighted mean, weighting by population. Just try something other than the plain vanilla mean.
- How does maternal mortality (`mat_mortality`) vary across different continents?

- Report the absolute and/or relative abundance of countries with low maternal mortality (`mat_mortality`) over time by continent: Compute some measure of worldwide maternal mortality - you decide - a mean or median or some other quantile or perhaps your current age. Then determine how many countries on each continent have a maternal mortality rate less than this benchmark, for each year.
- Find countries with interesting stories. Open-ended and, therefore, hard. Promising but unsuccessful attempts are encouraged. This will generate interesting questions to follow up on in class.
- Make up your own! Between the coverage in class, readings, and DataCamp plus the list above, you get the idea.

Companion graphs

For each table, make sure to include a relevant figure. One tip for starting is to draw out on paper what you want your x- and y-axis to be first and what your `geom` is; that is, start by drawing the plot you want `ggplot` to give you.

Your figure does not have to depict every last number from the data aggregation result. Use your judgement. It just needs to complement the table, add context, and allow for some sanity checking both ways.

Notice which figures are easy/hard to make, which data formats make better inputs for plotting functions vs. for human-friendly tables.

Report your process

You're encouraged to reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc. Give credit to your sources, whether it's a blog post, a fellow student, an online tutorial, etc.

Grading

This lab is worth 20 points total, scored as follows:

8 points for your initial EDA:

- 8 (Strong attempt): EDA reflects strong independent problem solving, with clearly thought out attempts to approach questions and problems, and a diligent and honest effort to answer questions and find the solutions.
- 4 (Adequate attempt): EDA reflects some attempt to approach the posed tasks, but approach appears to be superficial and lacks depth of analysis. No obvious mistakes. Pleasant to read. No head scratchers. Solid and complete.
- 0 (No attempt or incomplete): Didn't tackle all sections. Or didn't make companion graphs. Or didn't interpret anything but left it all to the "reader". Or more than one technical problem that is relatively easy to fix.

12 points for the quality of the final EDA:

- 12 (Exceptional): EDA is thorough, concise, and clearly demonstrates ability to competently and thoughtfully work with data as well as how to report on that process as a complement to code. Impeccable organization and presentation in the report.
- 8 (Adequate): Hits all the elements in all sections. No obvious mistakes. Pleasant to read. No head scratchers. Solid and complete.

- 4 (Inadequate): EDA attempts to address question with substantial inaccuracies in analysis and/or interpretation. Didn't tackle all sections. Or didn't make companion graphs. Or didn't interpret anything but left it all to the "reader". Or more than one technical problem that is relatively easy to fix.
- 0 (Insufficient): Nothing to grade, assignment was late.