STAT 413/613 HW 1

Ifebunandu Jerome Okeke

2021-01-27

Instructions

Admin elements:
1. Upload a photo (headshot) of yourself into your canvas profile
2. Review the Syllabus and the academic integrity code.
3. Fill in your information in the Student Info spreadsheet under the Canvas Collaboration site.

Analysis Elements: Rename the starter file under the analysis directory as `hw_01_yourname.Rmd` and use it for your solutions.
1. Modify the "author" field in the YAML header.
2. Stage and Commit R Markdown and HTML files (no PDF files).
3. **Push both .Rmd and HTML files to GitHub**.
- Make sure you have knitted to HTML prior to staging, committing, and pushing your final submission.
4. **Commit each time you answer a part of question, e.g. 1.1**
5. **Push to GitHub after each major question**, e.g., College Scorecard and World Bank Data
- **Committing and Pushing are graded elements for this homework.**
6. When complete, submit a response in Canvas that you have completed the homework. No need to submit your files.

- Only include necessary code to answer the questions.
- Most of the functions you use should be from the tidyverse. Too much base R will result in point deductions.
- Use Pull requests and or email to ask me any questions. If you email, ensure your most recent code is pushed to GitHub whether it is working or not.

Learning Outcomes:

- Operate with Git and GitHub.
- Apply concepts and methods from STAT 412/612.

Canvas Picture, Syllabus, and Student Info

Review the Syllabus on Canvas and answer the following questions:

I, *Ifebunandu Okeke* have:

1. Added a photo of myself (headshot) to my Canvas profile

2. Reviewed the syllabus and the associated policies on the following date: 01/25/2021
3. Reviewed the American University policies on academic integrity, and understand how they apply to this course and agree to comply with them for this course
4. Filled in my information in the Student Info spreadsheet on Canvas collaborations

College Scorecard

The data folder contains "college_score_200601.csv", a subset of the data in the College Scorecard database as of June 1, 2020. These data contain information on colleges in the United States. The variables include:

- `UNITID` and `OPEID`: Identifiers for the colleges.
- `INSTNM`: Institution name
- `ADM_RATE`: The Admission Rate.
- `SAT_AVE`: Average SAT equivalent score of students admitted.
- `UGDS`: Enrollment of undergraduate certificate/degree-seeking students
- `COSTT4_A`: Average cost of attendance (academic year institutions)
- `AVGFACSAL`: Average faculty salary
- `GRAD_DEBT_MDN`: The median debt for students who have completed
- `AGE_ENTRY`: Average age of entry
- `ICLEVEL`: Level of institution (1 = 4-year, 2 = 2-year, 3 = less than 2-year).
- `MN_EARN_WNE_P10`: Mean earnings of students working and not enrolled 10 years after entry.
- `MD_EARN_WNE_P10`: Median earnings of students working and not enrolled 10 years after entry.
- `FEMALE`: Share of female students
- `PCT_WHITE`: Percent of the population from students' zip codes that is White, via Census data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
C_score <- read_csv(file = "./../data/college_score_200601.csv")
```

```
## Parsed with column specification:
## cols(
##   UNITID = col_double(),
##   OPEID = col_double(),
##   MN_EARN_WNE_P10 = col_character(),
##   MD_EARN_WNE_P10 = col_character(),
##   INSTNM = col_character(),
##   STABBR = col_character(),
##   SAT_AVG = col_character(),
```

```
##   ADM_RATE = col_character(),
##   UGDS = col_character(),
##   COSTT4_A = col_character(),
##   AVGFACSAL = col_character(),
##   GRAD_DEBT_MDN = col_character(),
##   AGE_ENTRY = col_character(),
##   FEMALE = col_character(),
##   PCT_WHITE = col_character(),
##   ICLEVEL = col_double()
## )
```

2.1. If you used the default settings for reading in the data, 11 variables are probably type character when they should be numeric.

Which ones?

Solution:

They 11 variables are :

MN_EARN_WNE_P10 MD_EARN_WNE_P10 SAT_AVG ADM_RATE UGDS COSTT4_A AVGFACSAL GRAD_DEBT_MDN AGE_ENTRY FEMALE PCT_WHITE

2.2. Why were they read in as type character?

They are are incorrectly read as character because some of the missing data are labeled as " Null and privacy suppressed"

3. Use a readr function to fix these variables to be numeric and save the updated tibble.

Solution:

```
C_score %>%
  mutate(MN_EARN_WNE_P10 = parse_number(MN_EARN_WNE_P10),
         MD_EARN_WNE_P10 = parse_number(MD_EARN_WNE_P10),
         SAT_AVG  = parse_number(SAT_AVG ),
         ADM_RATE = parse_number(ADM_RATE),
         UGDS = parse_number(UGDS),
         COSTT4_A = parse_number(COSTT4_A),
         AVGFACSAL= parse_number(AVGFACSAL),
         GRAD_DEBT_MDN= parse_number(GRAD_DEBT_MDN),
         AGE_ENTRY = parse_number(AGE_ENTRY),
         FEMALE = parse_number(FEMALE),
         PCT_WHITE = parse_number(PCT_WHITE)) ->

  C_score
```

```
## Warning: Problem with `mutate()` input `MN_EARN_WNE_P10`.
## i 1989 parsing failures.
## row col expected            actual
##  18  -- a number NULL
##  24  -- a number PrivacySuppressed
##  26  -- a number NULL
##  45  -- a number PrivacySuppressed
##  55  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
##
## i Input `MN_EARN_WNE_P10` is `parse_number(MN_EARN_WNE_P10)`.

## Warning: 1989 parsing failures.
## row col expected            actual
##  18  -- a number NULL
##  24  -- a number PrivacySuppressed
##  26  -- a number NULL
##  45  -- a number PrivacySuppressed
##  55  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `MD_EARN_WNE_P10`.
## i 1989 parsing failures.
## row col expected            actual
##  18  -- a number NULL
##  24  -- a number PrivacySuppressed
##  26  -- a number NULL
##  45  -- a number PrivacySuppressed
##  55  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
##
## i Input `MD_EARN_WNE_P10` is `parse_number(MD_EARN_WNE_P10)`.

## Warning: 1989 parsing failures.
## row col expected            actual
##  18  -- a number NULL
##  24  -- a number PrivacySuppressed
##  26  -- a number NULL
##  45  -- a number PrivacySuppressed
##  55  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `SAT_AVG`.
## i 5508 parsing failures.
## row col expected actual
##   3  -- a number   NULL
##   7  -- a number   NULL
##   8  -- a number   NULL
##  12  -- a number   NULL
```

```
##  13  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input ‘SAT_AVG‘ is ‘parse_number(SAT_AVG)‘.

## Warning: 5508 parsing failures.
## row col expected actual
##   3  -- a number   NULL
##   7  -- a number   NULL
##   8  -- a number   NULL
##  12  -- a number   NULL
##  13  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.

## Warning: Problem with ‘mutate()‘ input ‘ADM_RATE‘.
## i 4800 parsing failures.
## row col expected actual
##   3  -- a number   NULL
##   7  -- a number   NULL
##   8  -- a number   NULL
##  12  -- a number   NULL
##  13  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input ‘ADM_RATE‘ is ‘parse_number(ADM_RATE)‘.

## Warning: 4800 parsing failures.
## row col expected actual
##   3  -- a number   NULL
##   7  -- a number   NULL
##   8  -- a number   NULL
##  12  -- a number   NULL
##  13  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.

## Warning: Problem with ‘mutate()‘ input ‘UGDS‘.
## i 765 parsing failures.
## row col expected actual
##  86  -- a number   NULL
## 174  -- a number   NULL
## 180  -- a number   NULL
## 181  -- a number   NULL
## 203  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input ‘UGDS‘ is ‘parse_number(UGDS)‘.

## Warning: 765 parsing failures.
```

```
## row col expected actual
##  86  -- a number   NULL
## 174  -- a number   NULL
## 180  -- a number   NULL
## 181  -- a number   NULL
## 203  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `COSTT4_A`.
## i 3375 parsing failures.
## row col expected actual
##   8  -- a number   NULL
##  18  -- a number   NULL
##  24  -- a number   NULL
##  55  -- a number   NULL
##  62  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input `COSTT4_A` is `parse_number(COSTT4_A)`.

## Warning: 3375 parsing failures.
## row col expected actual
##   8  -- a number   NULL
##  18  -- a number   NULL
##  24  -- a number   NULL
##  55  -- a number   NULL
##  62  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `AVGFACSAL`.
## i 2794 parsing failures.
## row col expected actual
##  18  -- a number   NULL
##  62  -- a number   NULL
##  65  -- a number   NULL
##  70  -- a number   NULL
##  71  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input `AVGFACSAL` is `parse_number(AVGFACSAL)`.

## Warning: 2794 parsing failures.
## row col expected actual
##  18  -- a number   NULL
##  62  -- a number   NULL
##  65  -- a number   NULL
##  70  -- a number   NULL
##  71  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
```

```
## Warning: Problem with `mutate()` input `GRAD_DEBT_MDN`.
## i 1530 parsing failures.
## row col expected            actual
##  17  -- a number PrivacySuppressed
##  19  -- a number PrivacySuppressed
##  21  -- a number PrivacySuppressed
##  24  -- a number PrivacySuppressed
##  25  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
##
## i Input `GRAD_DEBT_MDN` is `parse_number(GRAD_DEBT_MDN)`.

## Warning: 1530 parsing failures.
## row col expected            actual
##  17  -- a number PrivacySuppressed
##  19  -- a number PrivacySuppressed
##  21  -- a number PrivacySuppressed
##  24  -- a number PrivacySuppressed
##  25  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `AGE_ENTRY`.
## i 626 parsing failures.
## row col expected            actual
##  86  -- a number NULL
## 174  -- a number NULL
## 180  -- a number PrivacySuppressed
## 181  -- a number PrivacySuppressed
## 184  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
##
## i Input `AGE_ENTRY` is `parse_number(AGE_ENTRY)`.

## Warning: 626 parsing failures.
## row col expected            actual
##  86  -- a number NULL
## 174  -- a number NULL
## 180  -- a number PrivacySuppressed
## 181  -- a number PrivacySuppressed
## 184  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.

## Warning: Problem with `mutate()` input `FEMALE`.
## i 1492 parsing failures.
## row col expected            actual
##  18  -- a number PrivacySuppressed
##  24  -- a number PrivacySuppressed
##  30  -- a number PrivacySuppressed
##  58  -- a number PrivacySuppressed
```

```
##  70  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
##
## i Input `FEMALE` is `parse_number(FEMALE)`.
```

```
## Warning: 1492 parsing failures.
## row col expected            actual
##  18  -- a number PrivacySuppressed
##  24  -- a number PrivacySuppressed
##  30  -- a number PrivacySuppressed
##  58  -- a number PrivacySuppressed
##  70  -- a number PrivacySuppressed
## ... ... ........ ..................
## See problems(...) for more details.
```

```
## Warning: Problem with `mutate()` input `PCT_WHITE`.
## i 2136 parsing failures.
## row col expected actual
##  18  -- a number   NULL
##  26  -- a number   NULL
##  45  -- a number   NULL
##  58  -- a number   NULL
##  67  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
##
## i Input `PCT_WHITE` is `parse_number(PCT_WHITE)`.
```

```
## Warning: 2136 parsing failures.
## row col expected actual
##  18  -- a number   NULL
##  26  -- a number   NULL
##  45  -- a number   NULL
##  58  -- a number   NULL
##  67  -- a number   NULL
## ... ... ........ ......
## See problems(...) for more details.
```

4. How is average faculty salary associated the mean earnings of students ten years after initial enrollment? Create an appropriate plot and interpret the plot to justify your answer.

5. Does the level of the institution seem to be associated with the mean earnings of students ten years after enrollment? Reproduce this plot in R to explore this relationship and interpret the plot:

6. Plot the mean earnings 10 years after enrollment for level 1 institutions as the Y axis against PCT_WHITE and, in a second plot, against FEMALE.

- Use a log scale as appropriate.
- Add a loess smoother.
- Describe and interpret the relationship, if any, in each of the plots.

7. Create a scatter plot of the mean earnings 10 years after enrollment (Y) axis) compared to the median earnings 10 years after enrollment (X axis) using log scales for both.

- Add an abline.
- Interpret the plot and the relationship between the two variables.

8. Compute a ranking of level 1 universities based on the ratio of mean earnings 10 years after enrollment compared to median graduation debt.

- Identify the top 5 best (highest ROI should be #1) and the bottom 5 worst?
- What is American University's rank and ROI?
- Extra Credit:
  - Reproduce the following plot so the *AU line automatically adjusts as the new data is entered*:
  - What is AU's new ranking and ROI if the median earnings are used?

World Bank Data

The World Bank provides loans to countries with the goal of reducing poverty. The dataframes in the data folder were taken from the public data repositories of the World Bank.

- country.csv: Contains information on the countries in the data set.
  - The variables are:
  - `Country_Code`: A three-letter code for the country. Note not all rows are countries; some are regions.
  - `Region`: The region of the country.
  - `IncomeGroup`: Either `"High income"`, `"Upper middle income"`, `"Lower middle income"`, or `"Low income"`.
  - `TableName`: The full name of the country.
- fertility.csv: Contains the fertility rate information for each country for each year.
  - For the variables `1960` to `2017`, the values in the cells represent the fertility rate in total births per woman for that year.
  - Total fertility rate represents the number of children that would be born to a woman if she were to live to the end of her childbearing years and bear children in accordance with age-specific fertility rates of the specified year.
- life_exp.csv: Contains the life expectancy information for each country for each year.
  - For the variables `1960` to `2017`, the values in the cells represent life expectancy at birth in years for the given year.
  - Life expectancy at birth indicates the number of years a newborn infant would live if prevailing patterns of mortality at the time of its birth were to stay the same throughout its life.
- population.csv: Contains the population information for each country.
  - For the variables `1960` to `2017`, the values in the cells represent the total population in number of people for the given year.
  - Total population is based on the de facto definition of population, which counts all residents regardless of legal status or citizenship. The values shown are midyear estimates.

1. Use relative paths and a readr function to load these files into four tibbles.

2. These data are messy. The observational units in `fert`, `life`, and `pop` are locations in space-time (e.g. Aruba in 2017). Recall tidy data should have one observational unit per row.

- Use dplyr 1.0 functions to tidy these three tibbles and save the updated data frames.
- Use an approach where the tidying function also ensures the variable for `year` is a numeric.

3. Use dplyr functions to combine the three tibbles into one and save to a new tibble which includes the fertility rate, population, and life expectancy in each year as well as the region for each country.

4. Make a scatterplot of fertility rate vs life expectancy, color-coding by region and annotating size by the population.

- Include only the years 1960, 1970, 1980, 1990, 2000, and 2010.
- Facet by these years.
- Your final plot should look like this (Each element of the formatting is graded):
- **Interpret the plot in one sentence**.

5. Calculate the total population for each region for each year.

- Exclude 2018.

- Make a line plot of year versus total population, color-coding by region and using a log scale.
- Your final plot should look like this:
- **Interpret the plot in one sentence to identify the fastest growing regions**.

6. Make a bar plot of population vs region for the year 2010.

- Order the bars on the y-axis in **decreasing** order of population.
- Your final plot should look like this: