Problem Set #04 Statistics 506

Due: Oct 24, 10am on Canvas

Instructions

- **Review the attribution of sources discussions.**
- Submit the output of your Quarto document on Canvas. The output should include a link to your GitHub repository for this assignment.
- Unless otherwise explicitly stated, all problems should be solved in **R**.
- Your output file should include all code required to solve the problem. Code folding may be useful to make the output more readable.
- Use a consistent and readable code style for your code. Lack of a consistent and readable code style will negatively affect your grade.
- Some of these exercises may require you to use commands or techniques that were not covered in class or in the course notes. You can use the web as needed to identify appropriate approaches. Part of the purpose of these exercises is for you to learn to be resourceful and self sufficient. Questions are welcome at all times, but please make an attempt to locate relevant information yourself first.
- Be sure to properly document any functions you write using roxygen, and add comments as appropriate to make it clear what you are doing.
- If submitting an HTML file, please make sure to make it self-contained.

Submitting Stata and SAS

As discussed in the previous problem sets, getting Stata (and SAS) to behave with Quarto is challenging. For any problems that involve these, you may do the following:

- Stata: Write a .Do file to solve the problem, then include its input and output directly into your submission, by copying the Stata output window and including a non-evaluated chunk.

- SAS: Include only the code (and your commentary if needed) in the Quarto document, and include a link to the SAS results a separate file in your GitHub repository

  When looking at the SAS results, there's a button to "Download results as an HTML file"; you can download and push that html file directly to your repository.

Problem 1 - Tidyverse

Use the **tidyverse** for this problem. In particular, use piping and **dplyr** as much as you are able. **Note**: Use of any deprecated functions will result in a point loss.

Install and load the package **nycflights13**.

a. Generate a table (which can just be a nicely printed tibble) reporting the mean and median departure delay per airport. Generate a second table (which again can be a nicely printed tibble) reporting the mean and median arrival delay per airport. Exclude any destination with under 10 flights. Do this exclusion through code, not manually.

   Additionally,

   - Order both tables in descending mean delay.
   - Both tables should use the airport *names* not the airport *codes*.
   - Both tables should print all rows.

b. How many flights did the aircraft model with the fastest average speed take? Produce a tibble with 1 row, and entires for the model, average speed (in MPH) and number of flights.

Problem 2 - `get_temp()`

Use the **tidyverse** for this problem. In particular, use piping and **dplyr** as much as you are able. **Note**: Use of any deprecated functions will result in a point loss.

Load the Chicago NNMAPS data we used in the visualization lectures. Write a function `get_temp()` that allows a user to request the average temperature for a given month. The arguments should be:

- `month`: Month, either a numeric 1-12 or a string.
- `year`: A numeric year.
- `data`: The data set to obtain data from.
- `celsius`: Logically indicating whther the results should be in celsius. Default `FALSE`.
- `average_fn`: A function with which to compute the mean. Default is `mean`.

The output should be a numeric vector of length 1. The code inside the function should, as with the rest of this problem, use the **tidyverse**. Be sure to sanitize the input.

Prove your code works by evaluating the following. Your code should produce the result, or a reasonable error message.

```
get_temp("Apr", 1999, data = nnmaps)
get_temp("Apr", 1999, data = nnmaps, celsius = TRUE)
get_temp(10, 1998, data = nnmaps, average_fn = median)
get_temp(13, 1998, data = nnmaps)
get_temp(2, 2005, data = nnmaps)
get_temp("November", 1999, data =nnmaps, celsius = TRUE,
         average_fn = function(x) {
           x %>% sort -> x
           x[2:(length(x) - 1)] %>% mean %>% return
         })
```

Problem 3 - SAS

This problem should be done entirely within SAS.

Access the RECS 2020 data and download a copy of the data. You may import the CSV or load in the sas7bdat file directly. (This is **not** the 2009 version we used in lecture.) You'll probably also need the "Variable and response cookbook" to identify the proper variables. Load or import the data into SAS.

a. What state has the highest percentage of records? What percentage of all records correspond to Michigan? (Don't forget to account for the sampling weights!)

b. Generate a histogram of the total electricity cost in dollars, amongst those with a strictly positive cost.

c. Generate a histogram of the log of the total electricity cost.

d. Fit a linear regression model predicting the log of the total electricity cost based upon the number of rooms in the house and whether or not the house has a garage. (Don't forget weights.)

e. Use that model to generate predicted values and create a scatterplot of predicted total electricity cost vs actual total electricity cost (**not** on the log scale).

Problem 4 - Multiple tools

It is not uncommon during an analysis to use multiple statistical tools as each has their own pros and cons. The problem is based on an actual analysis I've done, with a different data set. The data was originally stored in a large SAS database, but the researcher was most familiar with Stata so I carried out the analysis there. During the course of the project, there was a particular analysis that Stata could not do, so I switched over to R. We're going to mimic this workflow here.

We'll use the Survey of Household Economics and Decisionmaking from the Federal Resereve. The data and Codebook documentation can be found at https://www.federalreserve.gov/consumerscommunities/shed_data.htm. Use the 2022 version.

The researcher's interest is in whether long-term concerns about climate change impact current day concerns about financial stability. To address this, the particular research question of interest is whether **the respondent's family is better off, the same, or worse off finanicially compared to 12 month's ago** can be predicted by **thinking that the chance of experiencing a natural disaster or severe weather event will be higher, lower or about the same in 5 years**. We also want to control for

- How they rate the economic conditions today in the country.
- Whether they own (with or without a mortgage) or rent or neither their home.
- Education (use the 4-category version)
- Race (use the 5-category version)

We're going to pretend the raw data is extremely large and that we need to extract the subset of the data we're going to use before we can open it in Stata or R.

Additionally, the data comes from a complex survey design, so we need to account for that in the analysis.

a. Take a look at the Codebook. For very minor extra credit, how was the Codebook generated? (No loss of points if you skip this.)

SAS

b. Import the data into SAS (you can load the SAS data directly or import the CSV) and use `proc sql` to select only the variables you'll need for your analysis, as well as subsetting the data if needed. You can carry out variable transformations now, or save it for Stata.

c. Get the data out of SAS and into Stata. (Note that this could mean saving the data in SAS format, then importing into Stata; or exporting from SAS into Stata format then loading it in Stata; or exporting from SAS into a generic format and importing into Stata - whichever works for you.)

(Note: Include you SAS code in the Quarto as specified; you **do not** need to include an HTML copy of the SAS results for this question.)

Stata

d. Demonstrate that you've successfully extracted the appropriate data by showing the number of observations and variables. (Report these values via Stata code don't just say "As we see in the Properties window". The Codebook should give you a way to ensure the number of rows is as expected.)

e. The response variable is a Likert scale; convert it to a binary of worse off versus same/better.

f. Use the following code to tell Stata that the data is from a complex sample:

   ```
   svyset CaseID [pw=weight_pop]
   ```

   (Modify `CaseID` and `weight_pop` as appropriate if you have different variable names; those names are taken from the Codebook.)

   Carry out a logisitic regression model accounting for the complex survey design. Be sure to treat variables you think should be categorical appropriately. From these results, provide an answer to the researchers question of interest.

   Notice that the model does not provide a pseudo-R^2. R has the functionality to do this.

g. Get the data out of Stata and into R.

R

h. Use the survey package to obtain the pseudo R^2. Use the following code to set up the complex survey design:

```
svydesign(id = ~ caseid, weight = ~ weight_pop, data = dat)
```

Obtain the pseudo-R^2 value for the logistic model fit above and report it.

(**Note**: If you decide to re-fit the model in R, read the first paragraph of the "Details" in the model-fitting function help to choose the appropriate family.)