STAT 29000 Optional Projects

Topics: R, UNIX, & associated tools

Motivation: Practice, practice, practice. Continuing to learn about and use the tools we've addressed this semester will make you faster and more efficient. You will be able to break problems into logical parts, and will discover you get "stuck" less often.

Context: We've explored a wealth of tools in using R and UNIX. We are now going to use what you've learned to solve a variety of different problems.

Scope: The scope of this project encompasses all topics covered this semester, namely, R, UNIX, and associated tools.

Don't forget the very useful documentation shortcut `?`. To use, simply type `?` in the console, followed by the name of the function you are interested in.

You can also look for package documentation by using `help(package=PACKAGENAME)`.

——— Optional Project 1 ———

Question 1: closures at Disney

1a. Read in the dataset provided by `option1.RData`. What is the name of the dataset that was read in? How many rows and columns are in the dataset?

1b. In the `SPOSTMIN` column of our Disney dataset, the number -999 represents the ride being closed. Let's assume that the time a ride is closed is the difference between the datetime of the first occurence of a -999 value and the datetime of the first non -999 value after the first occurence of the previously mentioned -999 value. For example:

```
35 2012-01-08 16:19:00
-999 2012-01-08 16:36:00
-999 2012-01-08 16:41:00
40 2012-01-08 17:07:00
```

The duration of this closure would be 31 minutes. Create a `data.table` with the duration of ride closures in a column called `duration`, and the datetime of the beginning of the closure in a column called `datetime`. Lastly, put the ride name in a column called `ride_name`. So for the above example, the first row would read: `2012-01-08 16:36:00, 31, splash_mountain`.

1c. What is the date and time of the beginning of the first closure lasting 60 minutes or longer?

Question 2: create a question

2a. You were provided a dataset about Disney wait times. Create a question that can be answered using this data, and solve the question using any of the tools you've learned about this semester. Include both the question and solution.

2b. Create an interesting plot using the provided dataset. Make sure to take care making the graphic look nice (i.e. descriptive labels, themed colors, etc.). Include the code used to generate your graphic. Note that this plot must be completely different than any other graphic you've produced using this dataset in the past.

——— Optional Project 2 ———

Question 1: `tailnum.sh`

1a. Write a bash script called `tailnum.sh` that, given a tail number, summarizes information about the life of the aircraft. Specifically: the aircrafts avg trip distance, avg air time, most common destination (abbreviation), avg arrival delay, and number of flights.

The result needs to be formatted like the example outputs below:

```
./tailnum.sh N784AA
```

```
Num. of Flights: 3004
Avg. Distance: 2348.15
Avg. Air Time: 295.04
Avg. Arrival Delay: 3.22936
Main Destination: "BOS"
```

```
./tailnum.sh N783AA
```

```
Num. of Flights: 2897
Avg. Distance: 2310.03
Avg. Air Time: 288.602
Avg. Arrival Delay: 2.61685
Main Destination: "LAX"
```

Question 2: Indiana flights

2a. We've worked with the flight data in previous projects. Create a function that given the name or abbreviation of a state, and a list of years, returns a `data.table` where only data related to the state is returned, for only the years specified. Call this function `flights_for_state()`. The function signature should look like: `flights_for_state(state, years)`.

2b. Use the function you created in (1a) to get a `data.table` for flights into and out of Indiana for 2000-2002 and 2012-2014. How many rows and columns are in the resulting `data.table`?

2c. Out of the sampled years, which year had the most flights into Indiana? Which year had the most flights out of Indiana? Across all of the sampled years, what state was the number 1 destination for flights leaving Indiana? What state had the most flights where Indiana was the destination?

2d. Create a bar plot where the number of flights out of Indiana are plotted next to the flights into Indiana by year. Add a theme to your plot.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption2-2d.jpg

Hint: The function `position_dodge()` and the argument `position` in `geom_bar` are useful to unstack bar plots.

Question 3: create a question

3a. You have all of the flight data at your disposal. Create a question that can be answered using this data, and solve the question using any of the tools you've learned about this semester. Include both the question and solution.

3b. Create an interesting plot using the provided dataset. Make sure to take care making the graphic look nice (i.e. descriptive labels, themed colors, etc.). Include the code used to generate your graphic. Note that this plot must be completely different than any other graphic you've produced using this dataset in the past.

——— Optional Project 3 ———

Question 1: ufos

You've been provided with the dataset used to generate the following graphic called `UFO sightings between 1910 and 2013 per country`. Reverse engineer the graphic, and give it a theme of your choice. Include the code used to generate the graphic in the answer.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption3-1.jpg

Hint: Function `coord_polar()` from `ggplot2` package may be useful when creating pie charts.

Hint: `ggplot2` has some themes built in. One of these themes has been used in this example. Note that if you don't get the exact theme, this is ok.

Question 2: big macs

You've been provided with the dataset used to generate the following graphic called `Big Mac prices on July 9, 2019`. Reverse engineer the graphic, and give it a theme of your choice. Include the code used to generate the graphic in the answer.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption3-2.jpg

Hint: Look at the function `reorder()`. Note that it can be used within your graph aesthetics (example: `geom_line(aes(x=reorder(x_values), y=y_values))`)

Hint: `ggplot2` has some themes built in. One of these themes has been used in this example.

Hint: The package `scales` is useful for helping when creating plots with different types of scales (dollars, percentages, etc). Functions in the package can be combined with several functions in `ggplot2`, such as `scale_x_continuous`, `scale_y_continuous`, etc.

Question 3: phds

You've been provided with the dataset used to generate the following graphic called `PhDs awarded between 2008 and 2017 per field`. Reverse engineer the graphic, and give it a theme of your choice. Include the code used to generate the graphic in the answer.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption3-3.jpg

Hint: You will probably need to do some data transformation first. In addition to transforming the data to long format, you will need to get the mean number of PhDs awarded per broad field and year.

Hint: The argument `group` in `geom_line()` function is useful to help separate lines for different groupings. Be smart about your groupings. It can be a variable that combines many pieces of information.

Hint: The colors we used are Purdue Campus Gold (Hex: #C28E0E) and Purdue Slayter Sky Blue (Hex: #6E99B4)

Hint: `ggplot2` has some themes built in. One of these themes has been used in this example. Note that if you don't get the exact theme, this is ok.

——— Optional Project 4 ———

Question 1: getting the data

1a. Use the methods we used in project 11 to download nfl play-by-play data for 2013-2019 from http://nflsavant.com/pbp_data.php and combine the data into a single `data.table` (or `tibble` or `data.frame`) called `plays`. There are 4 empty columns, remove them.

Hint: http://nflsavant.com/pbp_data.php?year=2019 is a link to the 2019 play-by-play data.

1b. You can find 3 more datasets on nflsavant.com/about.php. Write code to download nfl weather data, nfl player data, and combine data into 3 separate `data.table`'s named `weather`, `players`, and `combine` respectively. Note remove any "V" prefixed columns, these are erroneous in this case.

Hint: You may need to use option `fill=T` to read in the combine data.

Question 2: is it true?

In project 12 we saw that anecdotal evidence doesn't always appear to be true. In this series of questions we will explore some often regurgitated information that may or may not have the evidence to back it up.

2a. Newscasters and fans alike have been known to make statements that indicate football games played on Thursday nights are typically lower scoring and/or less exciting affairs.

Use the `plays` dataset to calculate the number of yards gained on Thursday night games on average. How does this compare to the average number of yards for all other games?

2b. Use the `plays` dataset to calculate the number of points scored on Thursday night games on average. Let's assume all extra points are made and therefore each touchdown is worth 7 points. How does this compare to the average number of points scored for all other games? Ignore field goals for now.

2c. Another thing that is commonly talked about in football is that as the game goes on, runningback performance increases because they "wear down" the defense.

Create a plot that shows the average number of yards gained by runs (this can also be called the number of yards rushed), by quarter. Make sure and add a theme to the graphic.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption4-2c.jpg

Note: Quarter "5" is overtime. You can, but don't need to include this "quarter" into the graphic.

2d. Football has been earning a lot of complaints of becoming harder to watch. Arguments include more commercials, and too many penalties. Let's explore the latter.

Create a plot that shows the median number of penalties per game, by year. Does there appear to be a noticeable increase?

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption4-2d.jpg

2e. It is not uncommon to hear people predict that a cold day football game will lead to a run-heavy (or rush-heavy) game script, meaning there will be more than normal amounts of running.

Use the `weather` dataset in combination with the `plays` dataset to see if there is on average more rushing during cold day games. What is the average number of rushing plays during warmer games? What about during colder games? Consider cold if it is less than or equal to 32 degrees.

Hint: Note that the `GameId` from `plays` and from `weather` do not match.

Hint: You will need to use the columns `OffenseTeam` and `DefenseTeam` from the `plays` dataset in addition to the `away_team` and `home_team` columns from the weather dataset. I've provided you with a dataset to convert between formats. See `option4.RData`.

2f. A small part of football is (ironically) kicking the ball. As you could imagine, if it is windy out, you would expect it to be harder to kick the ball accurately.

Create a `data.table` with the following columns: `GameId` (from `plays` dataset), `date`, `wind_mph`, and `% missed fg`. In this instance, `% missed fg` is the percentage of field goals that were missed during the game.

Hint: Note that the `GameId` from `plays` and from `weather` do not match.

Hint: You will need to use the columns `OffenseTeam` and `DefenseTeam` from the `plays` dataset in addition to the `away_team` and `home_team` columns from the weather dataset. I've provided you with a dataset to convert between formats. See `option4.RData`.

Hint: A potentially useful term is 'FIELD GOAL IS NO GOOD'.

2g. Create a plot that shows `wind_mph` on the x-axis and `% missed fg` on the y axis. State any observations you may have.

https://datamine.purdue.edu/seminars/fall2019/stat29000finalprojectoption4-2g.jpg

Project Submission:

Submit your solutions for the project(s) at these URLs:

Optional Project 1: https://classroom.github.com/a/vJ_qL6oj

Optional Project 2: https://classroom.github.com/a/ylf254dW

Optional Project 3: https://classroom.github.com/a/KMU8M97_

Optional Project 4: https://classroom.github.com/a/RnKPnHTr

using the instructions found in the GitHub Classroom instructions folder on Blackboard.