Classroom

Content for running instances of STAT 545/547M at UBC

View the Project on GitHub STAT545-UBC/Classroom

Homework 06: Data wrangling wrap up

Overview

Due November 09, 2018 at 23:59.

This is the *first* assignment of STAT 547M (despite it being named Homework 06).

Your task is to complete two of the six (numbered) topics below.

Evaluation

Note that we're being less strict on grading now. Due to student feedback, for STAT 547, we've raised the percentage value of an assignment that's complete and well done from 80% to 90%. To get higher than 90%, your submission will need to contain a "wow" factor. This does not mean doing more work (such as doing more than two tasks) – it means upping the quality of the work that you do have.

The rubrics listed on the homework page apply. More generally, you can think of the different levels as follows:

0: Did not attempt.

1: One or more elements are missing or sketchy. Missed opportunities to complement code and numbers with a figure and interpretation. Technical problem that is relatively easy to fix. It’s hard to find the report in this crazy repo.

2: Hits all the elements. No obvious mistakes. Pleasant to read. No heroic detective work required. Well done! This is worth 90%.

3: Exceeded the requirements in number of dimensions. Developed novel tasks that were indeed interesting and “worked”. Impressive use of R – maybe involving functions, packages or workflows that weren’t given in class materials. Impeccable organization of repo and report. You learned something new from reviewing their work and you’re eager to incorporate it into your work.

Submitting

Once you’re done the assignment, go back to UBC canvas, and find the “Homework 06” page. Here, do the following:

1. Provide a link to your homework repository.
2. Write a brief reflection about your experience with this assignment: what was hard/easy, problems you solved, helpful tutorials you read, etc. No need to write lots here.

Although you don’t have to do this until the very end, we highly recommend you commit and push to your homework repo regularly!

1. Character data

Read and work the exercises in the Strings chapter or R for Data Science.

2. Writing functions

Pick one:

- Write one (or more) functions that do something useful to pieces of the Gapminder or Singer data. It is logical to think about computing on the mini-data frames corresponding to the data for each specific country, location, year, band, album, … This would pair well with the prompt below about working with a nested data frame, as you could apply your function there.
    - Make it something you can't easily do with built-in functions. Make it something that's not trivial to do with the simple `dplyr` verbs. The linear regression function presented here is a good starting point. You could generalize that to do quadratic regression (include a squared term) or use robust regression, using `MASS::rlm()` or `robustbase::lmrob()`.

- If you plan to complete the homework where we build an R package, write a couple of experimental functions exploring some functionality that is useful to you in real life and that might form the basis of your personal package.

3. Work with the candy data

In 2015, we explored a dataset based on a Halloween candy survey (but it included many other odd and interesting questions). Work on something from this homework from 2015. It is good practice on basic data ingest, exploration, character data cleanup, and wrangling.

4. Work with the `singer` data

The `singer_location` dataframe in the `singer` package contains geographical information stored in two different formats: 1. as a (dirty!) variable named `city`; 2. as a latitude / longitude pair (stored in `latitude`, `longitude` respectively). The function `revgeocode` from the *ggmap* library allows you to retrieve some information for a pair (vector) of longitude, latitude (warning: notice the order in which you need to pass lat and long). Read its manual page.

1. Use `purrr` to map latitude and longitude into human readable information on the band's origin places. Notice that `revgeocode(... , output = "more")` outputs a dataframe, while `revgeocode(... , output = "address")` returns a string: you have the option of dealing with nested

dataframes.
You will need to pay attention to two things:

- Not all of the track have a latitude and longitude: what can we do with the missing information? (*filtering*, …)
- Not all of the time we make a research through `revgeocode()` we get a result. What can we do to avoid those errors to bite us? (look at *possibly()* in `purrr`…)

2. Try to check wether the place in `city` corresponds to the information you retrieved.

3. If you still have time, you can go visual: give a look to the library `leaflet` and plot some information about the bands. A snippet of code is provided below.

```
singer_locations %>%
  leaflet()  %>%
  addTiles() %>%
  addCircles(popup = ~artist_name)
```

5. Work with a list

Work through and write up a lesson from the purrr tutorial:

- Trump Android Tweets
- Simplifying data from a list of GitHub users

6. Work with a nested data frame

Create a nested data frame and map a function over the list column holding the nested data. Use list extraction or other functions to pull interesting information out of these results and work your way back to a simple data frame you can visualize and explore.

Here’s a fully developed prompt for Gapminder:

- See the split-apply-combine lesson from Jenny Bryan

- Nest the data by country (and continent).
- Fit a model of life expectancy against year. Possibly quadratic, possibly robust (see above prompt re: function writing).
- Use functions for working with fitted models or the broom package to get information out of your linear models.
- Use the usual dplyr, tidyr, and ggplot2 workflows to explore, e.g., the estimated cofficients.

Inspiration for the modelling and downstream inspiration

- Find countries with interesting stories. - Sudden, substantial departures from the temporal trend is interesting. How could you operationalize this notion of “interesting”?
- Use the residuals to detect countries where your model is a terrible fit. Examples: Are there are 1 or more freakishly large residuals, in an absolute sense or relative to some estimate of background variability? Are there strong patterns in the sign of the residuals? E.g., all pos, then all neg, then pos again.
- Fit a regression using ordinary least squares and a robust technique. Determine the difference in estimated parameters under the two approaches. If it is large, consider that country “interesting”.
- Compare a linear and quadratic fit

This project is maintained by STAT545-UBC

Hosted on GitHub Pages — Theme by orderedlist