Homework 2: Summarizing and Cleaning Data Pt. 2

And some GitHub

Econ 245

Overview

For this homework assignment, you will be working with Airbnb data from Seattle, Washington. Airbnb is an American online marketplace that offers arrangement for lodging, primarily homestays, or tourism experiences. They are a hotel alternative where you can rent directly from locals in your desired area. Some of the variables we are particularly interested in are:

- `id` - gives the unique identification number for the listing.
- `property_type` - describes whether the property is a house, apartment, townhouse etc.
- `price` - the price for a night at the property.
- `host_is_superhost` - states whether the host is a superhost. A superhost is someone that has many listings and has great experience and feedback on Airbnb.
- `room_type` - documents what type of room the listing is.
- `cleaning_fee` - states the cleaning fee (if any) that must be paid by the visitor.
- `neighbourhood` - gives the neighborhood of Seattle.
- `host_response_rate` - gives a percentage of the amount of inquiries the host responds to.
- `host_listings_count` - gives the number of listings the host has.
- `host_acceptance_rate` - gives a percentage of the amount of bookings the host accepts.
- `cancellation_policy` - the cancellation policy.
- `square_feet` - the square footage of the listing.

To start, make sure you have the `tidyverse` package installed and loaded in Rstudio. If not, use the `install.packages` function to install them before you begin.

To Receive Credit

- Save the file as assignment_2.R. Make sure your capitalization is correct as the autograder is case-sensitive.
- Make sure all changes to the original dataset are done within the R script.
- For this assignment, you will also be using GitHub. This assignment will be completed in pairs. You and your partner are responsible for submitting one GitHub repository with one assignment_2.R file. The grade received will be both of your grades for this assignment. Submit to gradescope using the *GitHub* option. Part 1 will go through setting up the *GitHub*. The pairing for this assignment is displayed in Table 1.[1]

[1] Names and emails were taken directly from the UCSB class roster.

Table 1: Partners for Assignment 2

Person 1		Person 2	
Name	Email	Name	Email
Camilo	ecabbategranada@umail.ucsb.edu	Roberto	roberto_amaralsantos@umail.ucsb.edu
Akanksha	akanksha_arora@umail.ucsb.edu	Luorao	lbian@ucsb.edu
Sebastian	sebastian_brown@umail.ucsb.edu	Kenneth	kenneth_chan@umail.ucsb.edu
Nir	nir@umail.ucsb.edu	Toshio	ferrazares@umail.ucsb.edu
Nicolas	nfuertessegura@umail.ucsb.edu	Ariel	yeqing@umail.ucsb.edu
Yang	yang_gao@umail.ucsb.edu	Vincent	mguan@umail.ucsb.edu
Xin	xin_jiang@umail.ucsb.edu	Emma	emmajulie@umail.ucsb.edu
Jack	jkeefer@umail.ucsb.edu	Jahyeon	jahyeon_koo@umail.ucsb.edu
Dingyue	dingyueliu@umail.ucsb.edu	Jeffrey	jrromine@umail.ucsb.edu
James	jstuart-turner@umail.ucsb.edu	Sandy	sandysum@umail.ucsb.edu
Dario	dariotrujanoochoa@umail.ucsb.edu	Shu Chen	shuchentsao@umail.ucsb.edu
Micah	micahvillarreal@umail.ucsb.edu	Ravi	vora@umail.ucsb.edu
Lei	leiyue@umail.ucsb.edu	David	zingher@umail.ucsb.edu

Grading on Coding Questions

Grading on the coding portion of the homework will come in two types of questions: *Public Questions* and *Private Questions*. Public Questions can be submitted as many times as you like to the autograder, and the autograder will give detailed feedback. On the other hand, Private Questions can be thought of as a mini quiz within the homework. While you still have as many times to upload your answer as you want, the autograder will not provide any feedback, Professor Startz will not provide any guidance or assistance (but getting advice from classmates on Nectir or elsewhere is completely okay). Private Questions will be marked on the homework assignment.

Part 1: GitHub

This part will be setup in for two people, person 1 and person 2. It doesn't matter who is person 1 and who is person 2. It just matters the Github is setup on both computers. There will be some missing steps in Part 1. For example, you may have to figure out on your own how to download `git` onto your computer. Happy Git with R will be a good resource when you get stuck.

1. Person 1 will create a new GitHub repository.[2] You can name it whatever you like.

2. Person 1 will invite Person 2 to the repository. Person 2 will accept.

3. Person 1 and 2 will sync the Github repository locally with their `R`. If you do not remember, follow the steps from lecture and the guided exercises.

4. Person 2 adds your datasets to the GitHub repository. Do not make any sub-folders. Commit and push your changes. After done, access your repository via the internet and make sure it updated. Person 1, pull from Github. Make sure you see the files on your local computer. Refer to the resource linked above if you have issues (like a `conlfict`). **HINT**: You may need to hop on zoom and discuss what's going on for a while. Remember, if you can figure out how to do this now you won't need to learn it later!

5. Person 1 creates assignment_2.R. Do not put it in a sub-folder. Commit your changes and push. Person 2 pull from Github. Make sure you have it locally.

[2] This means you will need a GitHub account. Create one if you have not done so already.

6. Person 1 and 2 make their own branches. Each person should individually solve Part 2. Feel free to look at the other person's work on their own branch, make comments, and compare. This is meant to be collaborative.

7. After completion of Part 2, choose which branch to merge into `master`.

8. Once merged into master, have one of the team members submit the assignment on Gradescope. Make sure to add your teammate to the group!

Part 2: Coding

1. Setting up the data.

   a) Import the data using the `read_csv` function, and save the data set as a tibble with the name airbnb.
   b) Using the `View` and `colnames` function, take a look around the data and familiarize yourself with the columns of interest. This question is not graded, but it is always helpful to look at your data before beginning any computation to minimize errors.
   c) (*Private Question*) If you used the `colnames` function, you likely noticed that the column `neighbourhood` is spelt with a "u". Change the column name `neighbourhood` to `neighborhood` using the `rename` function. Be sure to save the updated tibble as airbnb.

2. Piping practice and creating summary statistics.

   a) The goal of this question to answer the question: which neighborhoods are most popular for Airbnb and what characteristics do they have? First, using the `count` function, count the number of occurrences each neighborhood appears in the airbnb data. Save this tibble as neighborhoods. It should have 2 columns. See Table 2 for an example of the output.
   b) Update the neighborhoods tibble from part (a) to get rid of any `NA` rows in the `neighborhood` column using the `filter` function, and then use the `arrange` and `head` functions to get the top 20 most frequently listed neighborhoods in descending order. Hence, your tibble should have two columns: `neighborhood` and `n`, and should have no more than 20 rows.
   c) Create a new tibble named airbnb_top_neighborhoods which will be the airbnb tibble, but only including the neighborhoods in the neighborhoods tibble. You will want to utilize the `filter` function and the `%in%` operator.
   d) Now we will create summary statistics of these top 20 neighborhoods. In particular, we want to know the mean of the `square_feet` and `price` columns, along with the standard deviation, minimum, and maximum of the `price` column. This will help us answer the question of whether more square footage accounts for a higher listing price, or if there are other factors that determine listing price. Using the `group_by`, `summarize`, and `arrange` functions, create a new tibble named summary_stats_top_neighborhoods that has the column names `neighborhood`, `avg_square_feet`, `avg_price`, `sd_price`, `max_price`, `min_price`. Arrange the tibble so that `avg_square_feet` is in descending order. Is square footage the only input to price? See Table 3 for example.
   e) (*Private Question*) Using the method of element extracting from a matrix (see Guided Exercises), save the highest `avg_square_ft` as a variable named highest_avg_square_ft.
   f) (*Private Question*) Similarly to part (e), save the second highest average price as a variable named second_avg_price.

Table 2: Sample of select rows for Question 2a.

neighborhood	n
NA	416
Capitol Hill	351
Ballard	213
Belltown	204
Minor	192

* Note that your tibble may have different values.

Table 3: Sample of select rows for Question 2d.

neighborhood	avg_square_feet	avg_price	sd_price	max_price	min_price
Wallingford	1375.000	131.3357	93.23641	575	39
Minor	1358.333	130.2969	78.09861	450	22
Green Lake	1272.500	152.6538	105.57905	550	40
Queen Anne	1233.333	168.7647	134.70768	975	20
Lower Queen Anne	1000.000	142.5783	69.25829	498	38

* Note that your tibble may have different values.

Submitting to Gradescope

When you get to the submission screen on Gradescope, click GitHub instead of upload. Then you'll be asked to sync your GitHub repository (See Figure 1).

You'll be asked to sync your GitHub and which repository and branch to use. After you choose, click upload and your files should be graded.

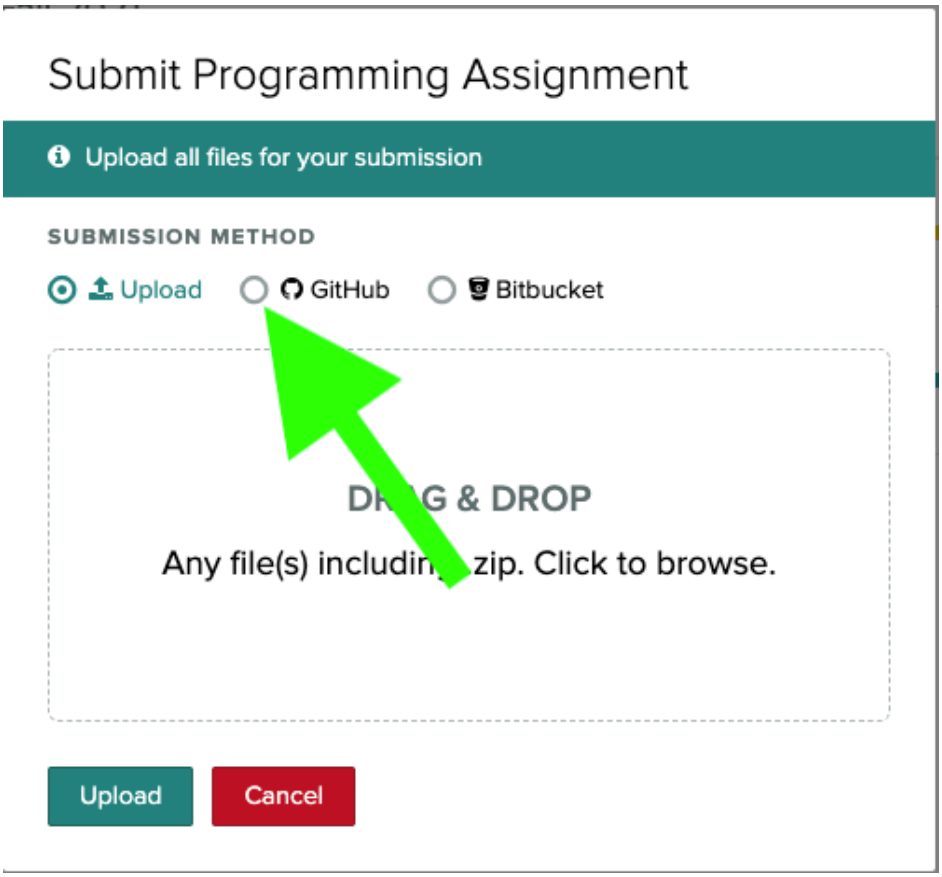

Figure 1: Where to Click to Submit