STAT 29000 Project 5

Topics: Assembling one data frame from one hundred data frames

Motivation: Combining separate data frames into one data frame is a useful but sometimes tedious skill. Practice builds experience. We will go through the full analysis of building one data frame, starting with more than 100 individual data frames.

Context: We will build such a data frame in the context of the AirBnB data, in which we have datasets about AirBnB properties from all over the world. These datasets were assembled at distinct points in time. For this reason, the datasets are not all stored in exactly the same way. Therefore, some data wrangling needs to be performed.

Scope: We walk through all of the nuances of data wrangling in this case study. Importantly, we document every step along the way, so that the data wrangling that we perform is totally reproducible to other human beings who might want to replicate our manipulations of the dataset.

Question 1: Examining the column names of data frames and making adjustments

Before you start question 1, please consider the examples given here:

`/class/datamine/data/examples/stat29000project5examples.R`

These will help you to better understand the techniques that we will use in this project.

We can also use sapply to see that they do not all have the same column names (if you check carefully, you will see that two of them do not have the same column names of columns)

```
sapply(myresults, names)
```

Find a way to discover which two data frames are the different ones, using R, rather than looking manually. More specifically:

1a. There is one data frame that does not have the same number of columns because it is missing the column `neighbourhood_group`. Find a way to discover which data frame it is, using R, rather than looking manually. Create a new column in this data frame, with all entries equal to NA. Make sure that this new column becomes the 5th column of this data frame.

Hint: You can rearrange columns of a data frame like this, for example, see

https://stackoverflow.com/questions/5620885/how-does-one-reorder-columns-in-a-data-frame

or many other similar help pages.

1b. Now there is one data frame remaining, which has different column names than the others. Find a way to discover which column it is, using R, rather than looking manually. Remove this element from the `myresults` list. See, for instance, these examples

https://stackoverflow.com/questions/652136/how-can-i-remove-an-element-from-a-list

or many other similar help pages. This leaves us with `myresults` as a list of 100 (rather than 101) data frames.

Question 2: Adjusting the types of columns in (some) of the data frames.

In the remaining 100 data frames, column 6 is usually a factor, but in two of the data frames, column 6 is an integer.

2a. Use `as.factor` to change the 6th column of these two data frames from integers into factors.

Question 3: Assembling one large data frame, based on the list of 100 data frames.

3a. Bind all of these 100 data frames into one big data frame, which should now have 1340094 rows and 16 columns. There are many ways to do this. For instance, see this page:

https://www.r-bloggers.com/concatenating-a-list-of-data-frames/

or many other similar help pages.

Project Submission:

Submit your solutions for the project at this URL: https://classroom.github.com/a/ubFQzpcK using the instructions found in the GitHub Classroom instructions folder on Blackboard.