Assignment 3

Due Tuesday, September 28, 2021 at 11:59 PM
in your GitHub repo.

Instructions:

Complete this assignment within the space on your GitHub repo in a folder called `assignment_03`. In this folder, save your answer to Question 1 in a file called `A3Q1_data.R`. In the same folder, save a copy of the sample file called `A3Q2_data.sh` that will contain your shell script for Question 2. Samples are available in the `assignment_03` folder within the code repository QMB6358F21.

When you are finished, submit your code by pushing your changes to your GitHub repo, following the instructions in Question 3. You are free to discuss your approach to each question with your classmates but you must `git push` your own work.

Question 1:

The repository QMB6358_car_auction_data contains 96 `.csv` files, which contain the records of monthly used car auctions over the years 2010 to 2017. The datasets list the sale prices of cars, the mileage of each car, whether the car has been in an accident, and whether the car has sustained major structural damage, which typically happens only as a result of an accident. Some auctions focus on the sale of damaged cars with "salvage" titles, aimed at the market for used car parts.

Clone the data repository to another location on your computer (outside of any other repo!). Use this location as your working directory but do not push the large dataset to the remote repository.

Your job is to collect these files to form one large dataset and print some summary statistics. Use the file `A3Q1_data.R` as a starting point. Complete it in stages by following these steps:

a) Write a loop that prints out the names of all the files. Use this to verify that the file names all match the names of the actual files.

b) Extend the loop to read in each file, assign it to a data frame called `A3Q1_data_sample` and print the function `table(A3Q1_data_sample[, 'accident'], A3Q1_data_sample[, 'damage'])` to show that the files are read in correctly. You should observe the sales of 100 cars in each auction, with some that have been in accidents and some with structural damage.

c) Extend the loop one more time by `bind`ing the files into a full dataset called `A3Q1_data`.

d) Verify that the statistics at the bottom of `A3Q1_data.R` indicate that the dataset has 9,600 rows, 6 columns, with cars 5,400 that have not been in accidents, 4,200 that have, and 600 of those with structural damage.

Question 2:

Now organize the data files into the dataset using UNIX commands. Complete the shell script `A3Q2_data.sh` to assemble the dataset in two different ways.

a) Use the `cat` command within a for loop to output the full dataset in the file `A3Q2a_full.csv`.

b) Use the `cat` command in a single line to output the full dataset in the file `A3Q2b_full.csv`.

c) Add some commands to the script `A3Q2_tests.R` to test the two datasets. Use commands similar to those at the bottom of `A3Q1_data.R` for testing the dataset in Question 1. Add a line at the bottom of `A3Q2_data.sh` to run this script and output the results to `A3Q2_results.out`.

Finally, running the script `A3Q2_data.sh` will assemble the datasets and calculate `summary` statistics and a `table` of observations for each method in both Questions 1 and 2. Use the output to verify that the datasets are correct.

Note: Do not save a large dataset in a folder within your GitHub repository for your assignments. The GitHub repository is designed to store text files with code but not data; there is a limit on the files size, as well. Please save any large files to a location on your computer outside of the repository.

Question 3:

Push your completed files to your GitHub repository following these steps. See the `README.md` in the folder `demo_04_version_control` in the QMB6358F20 course repository for more instructions.

1. Open GitBash and navigate to the folder inside your local copy of your git repo containing your assignments. Any easy way to do this is to right-click and open GitBash within the folder in Explorer. A better way is to navigate with UNIX commands.

2. Enter `git add .` to stage all of your files to commit to your repo. You can enter `git add my_filename.ext` to add files one at a time, such as `my_filename.ext`. in this example.

3. Enter `git commit -m "Describe your changes here"`, with an appropriate description, to commit the changes. This packages all the `add`ed changes into a single unit and stages them to `push` to your online repo.

4. Enter `git push origin main` to push the changes to the online repository. After this step, the changes should be visible on a browser, after refreshing the page.