QMB 3311: Python for Business Analytics
Department of Economics
College of Business
University of Central Florida
Spring 2022

Assignment 8

Due Sunday, April 24, 2021 at 11:59 PM
in your GitHub repository

Instructions:

Complete this assignment on your private GitHub repo in a folder called `assignment_08`. In this folder, save your answers to Questions 1 to 4 in a file called `my_A8_queries.py`, by completing the script in the file `my_A8_queries.py` in the course repository. When you are finished, submit it by uploading your files to your GitHub repo using any one of the approaches outlined in Question 5. You are free to discuss your approach to each question with your classmates but you must upload your own work.

Question 1:

The folder `assignment_08` contains three `.csv` files: `applications.csv`, `credit_bureau.csv`, and `demographic.csv`. The first dataset `applications.csv` contains the following variables.

`app_id`	=	a unique key for each customer who applied for credit
`ssn`	=	the social security number
`zip_code`	=	the the zip code in which the applicant resides
`income`	=	the applicant's reported income
`homeownership`	=	a categorical variable that indicates whether an applicant owns or rents a home
`purchases`	=	the monthly value of purchases on the account
`credit_limit`	=	the maximum amount that an applicant is approved to spend

You will use this dataset to estimate a regression model to predict the monthly amount of `purchases` for each customer.

a) Create a new database called `credit.db`.

b) Read in the `applications.csv` dataset and store the contents in a data frame called `applications` in your workspace.

c) Use the sample code in `my_A8_queries.py` to estimate a regression model to predict `purchases` as a function of the other variables in the dataset (ignoring the variables `app_id`, `ssn` and `zip_code`, which are keys for databases). Notice the value of the adjusted R-squared statistic.

d) `CREATE` a `TABLE` called `Applications` with a schema that is appropriate for the variables.

e) Populate the table `Applications` with the observations in the data frame `applications`.

Question 2:

Now use two files `applications.csv` and `credit_bureau.csv` in the folder `assignment_08`. The dataset `credit_bureau.csv` contains the following variables.

`ssn`	=	the consumers unique social security number
`zip_code`	=	the zip code in which the consumer resides
`fico`	=	the consumer's credit score
`num_late`	=	the number of number of times a consumer has made a payment after the due date
`past_def`	=	the number of number of times a consumer has defaulted on a line of credit
`num_bankruptcy`	=	the number of number of times a consumer has filed for bankruptcy

You will use the variables from both datasets to estimate a better regression model to predict monthly purchase volume.

a) Read the new dataset and store it in a data frame called `credit_bureau` in your workspace.

b) `CREATE` a `TABLE` called `CreditBureau` with a schema that is appropriate for the variables.

c) Populate the table `CreditBureau` with the observations in the data frame `credit_bureau`.

d) Join the two tables by `ssn` and `zip_code` and output the result as a `pandas` data frame called `app_bureau`.

e) Use the sample code in `my_A8_queries.py` to estimate a regression model to predict `purchases` as a function of the other variables in the dataset. (Again, ignore the variables `app_id`, `ssn` and `zip_code`, which are keys for databases.)

Question 3:

Now use all three files `applications.csv`, `credit_bureau.csv`, and `demographic.csv` in the folder `assignment_08`. The dataset `demographic.csv` contains the following variables.

`zip_code`	=	the zip code to indicate each geographic region
`avg_income`	=	the average income in each zip code
`density`	=	the population density in each zip code

You will use the variables from all three datasets to estimate an even better regression model to predict monthly purchase volume.

a) Read the new dataset and store it in a data frame called `demographic` in your workspace.

b) `CREATE` a `TABLE` called `Demographic` with a schema that is appropriate for the variables.

c) Populate the table `Demographic` with the observations in the data frame `demographic`.

d) Join the new table `Demographic` to the information from the other two tables by `zip_code`. You can use your query from Question 2 as a nested query. Output the result as a `pandas` data frame called `purchase_full`.

e) Use the sample code in `my_A8_queries.py` to estimate a regression model to predict `purchases` as a function of the other variables in the dataset. As above, ignore the variables `app_id`, `ssn` and `zip_code`, which are keys for databases.

Question 4:

Push your completed files to your GitHub repository following one of these three methods.

Method 1: In a Browser

Upload your code to your GitHub repo using the interface in a browser.

1. Browse to your `assignment_0X` folder in your repository (the "X" corresponds to Assignment X.).
2. Click on the "Add file" button and select "Upload files" from the drop-down menu.
3. Revise the generic message "Added files via upload" to leave a more specific message. You can also add a description of what you are uploading in the field marked "Add an optional extended description..."
4. Press the button "'Commit changes," leaving the buton set to "Commit directly to the `main` branch."

Method 2: With GitHub Desktop

Upload your code to your GitHub repo using the interface in GitHub Desktop.

1. Save your file within the folder in your repository within the folder referenced in GitHub Desktop.
2. When you see the changes in GitHub Desktop, add a description of the changes you are making in the bottom left panel.
3. Press the button "Commit to main" to commit those changes.
4. Press the button "Push origin" to push the changes to the online repository. After this step, the changes should be visible on a browser, after refreshing the page.

Method 3: At the Command Line

Push your code directly to the repository from the command line in a terminal window, such as GitBash on a Windows machine or Terminal on a Mac.

1. Open GitBash or Terminal 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, such as `cd`.
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_functions.py` in this Assignment.
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.