ECP 3004: Python for Business Analytics
Department of Economics
College of Business
University of Central Florida
Spring 2021

Assignment 2

Due Sunday, January 31, 2021 at 11:59 PM
in your GitHub repository

Instructions:

Complete this assignment within the space on your GitHub repo in a folder called `assignment_02`. In this folder, save your answers to Questions 1 and 2 in a file called `my_functions.py`, following the sample script in the folder `assignment_02` 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 3. You are free to discuss your approach to each question with your classmates but you must upload your own work.

Question 1:

Follow the function design recipe to define functions for all of the following Exercises. For each function, create three examples to test your functions. Record the definitions in the sample script `my_functions.py`

Example 1 Write a python function `average()` that will calculate the average of two numbers. It should have two arguments, the two numbers in the average. A junior developer at your firm made an initial attempt in the sample script `my_functions.py` but left to join a start-up in California. You might have to make some adjustments to the function.

Example 2 Write a python function `area_of_circle()` that will calculate the area of a circle. The only argument should be the radius of the circle.

Example 3 Write a python function `volume_of_cylinder()` that will calculate the volume of a cylinder. The first argument should be the radius of the circle at the base of the cylinder. The second argument should be the height of the cylinder. You can call your function `area_of_circle()` within this function.

Example 4 Write a python function `utility()` that will calculate the value of the Cobb-Douglass utility function $u(x, y; \alpha) = x^{\alpha} y^{1-\alpha}$. The first two arguments are x and y, respectively, and the third is α.

Example 5 Write a python function `logit()` that will calculate the logit link function

$$\ell(x; \beta_0, \beta_1) = Prob(y = 1|x) = \frac{e^{x\prime\beta}}{1 + e^{x\prime\beta}} = \frac{e^{\beta_0 + x\beta_1}}{1 + e^{\beta_0 + x\beta_1}}.$$

The first argument is x and the last two are β_0 and β_1.

Question 2:

For all of the Exercises in Question 1, use your examples to test the functions you defined. Complete the code at the bottom of your `my_functions.py` script so that it will make the comparisons between your expected answers and the output from your functions. When you run the whole block of code at the bottom, it should show the results of all your comparisons.

Question 3:

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_01` folder in your repository.
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.