Assignment [100 marks, weight: 30%]

BEE1038: Introduction to Data Science in Economics

In this assignment, you will demonstrate your understanding and mastery of programming in Python using data science tools.

What you will have learnt by the end of Week 8 should cover almost everything you will need, and what you learnt is already enough to start working on some problems. If you are stuck then read through the notebooks again. If you are still unsure, then have a look online. *Google* and *Stack OverFlow* are your friends!

The grade of this assignment contributes 30% towards your overall grade in the course. The following aspects need to be shown:

- Basic Python code and functions
- Manipulation and calculations on NumPy arrays and Pandas data frame
- Preparing and preprocessing data.
- Doing a basic plot, and changing plot markers, colors, etc.
- Improving and extending analysis.

Your submission will be a compressed file (.zip) containing:

1. A copy of your Python script named **solution_code.ipynb** (done in Jupyter Notebook).
2. Same copy printed as a PDF, **solution_code.pdf** [1]
3. Three .png images of your final plots: one that replicates the plot in Problem 4 (**P4.png**), one that replicates the plots in Problem 5 (H) (**P5H.png**), and those that show any additional analysis in Problem 6 (**P6A.png**, etc.).

Your scripts must be sufficient to reproduce your answers to all questions and plots. You are responsible for making sure that your Jupyter Notebook file will open without errors. Submissions that do not open may receive a zero.

Deadline is **Monday 21nd March at 15:00 (GMT)**

Collaboration & Misconduct: You are encouraged to think about this assignment in groups or ask each other for help. If you do, you should do the following: 1) write your own code (no code

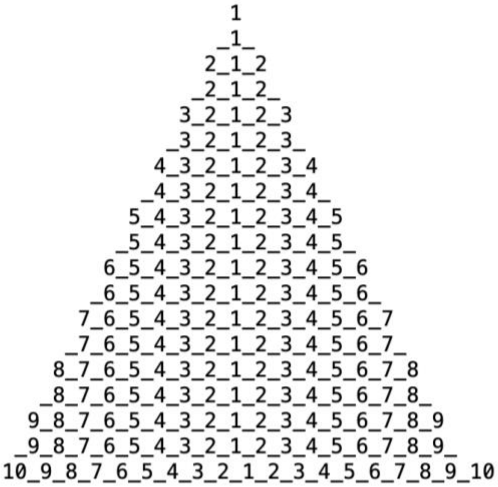
[1] See here how to create a PDF from a .ipynb file: https://vle.exeter.ac.uk/mod/forum/discuss.php?d=194496

copying from others), 2) Report the names of all people that you worked with in your submission, 3) if you received help from someone, write that explicitly, 4) plagiarism of code or writeup will not be tolerated; do not copy blocks of code in your answers, and 5) do not post your solutions online (even after the release of your marks). For those who want to evidence your experience to recruiters, make sure you share a private link to your project/work (or undiscoverable link). **If we can find your answers online anytime until September this year, you will be reported for misconduct.**

The University takes poor academic practice and academic misconduct very seriously and expects all students to behave in a manner which upholds the principles of academic honesty. Please make sure you familiarise yourself with the general guidelines and rules from this link[2] and this link[3].

Problem 1 [15 marks]

Write a function that accepts a number *n* as an input, and it returns n rows that look like the following pattern. Run your function for n = 19 (the output below is for n=19).



```
                   1
                  _1_
                 2_1_2
                _2_1_2_
               3_2_1_2_3
              _3_2_1_2_3_
             4_3_2_1_2_3_4
            _4_3_2_1_2_3_4_
           5_4_3_2_1_2_3_4_5
          _5_4_3_2_1_2_3_4_5_
         6_5_4_3_2_1_2_3_4_5_6
        _6_5_4_3_2_1_2_3_4_5_6_
       7_6_5_4_3_2_1_2_3_4_5_6_7
      _7_6_5_4_3_2_1_2_3_4_5_6_7_
     8_7_6_5_4_3_2_1_2_3_4_5_6_7_8
    _8_7_6_5_4_3_2_1_2_3_4_5_6_7_8_
   9_8_7_6_5_4_3_2_1_2_3_4_5_6_7_8_9
  _9_8_7_6_5_4_3_2_1_2_3_4_5_6_7_8_9_
10_9_8_7_6_5_4_3_2_1_2_3_4_5_6_7_8_9_10
```

[2] http://as.exeter.ac.uk/academic-policy-standards/tqa-manual/aph/managingacademicmisconduct/

[3] https://vle.exeter.ac.uk/pluginfile.php/1794/course/section/27399/A%20Guide%20to%20Citing%2C%20Referencing%20and%20Avoiding%20Plagiarism%20V.2.0%202014.pdf

Problem 2 [15 marks]

*For this problem, you are **not allowed** to use the sort() function provided by Python or any of its libraries.*

A. Write a function that you will call *TripleSort()* that takes as input 3 integers and returns a list containing those numbers in ascending order.

For example TripleSort(5,9,2) should return [2,5,9]

B. Using the *TripleSort()* function you have created, create another function *ListSort()* that takes a list of integers of any length as input, and sorts them ascendingly using the *TripleSort()* function.

C. Demonstrate that your ListSort algorithm can sort the list [5,2,1,6,8,0,4,9,18,-8,-100,100,26,9,18,28,30]

D. Set the NumPy random seed to 90. Generate a 1-dimensional NumPy array of size 3000 consisting of random integers between -5000 and 5000. Use the %timeit function to calculate the time for your *ListSort()* algorithm to run.

E. Comment on how you could improve your *ListSort()* and *TripleSort()* algorithms to increase the efficiency

Problem 3 [20 marks]

A. Set the NumPy random seed to 120

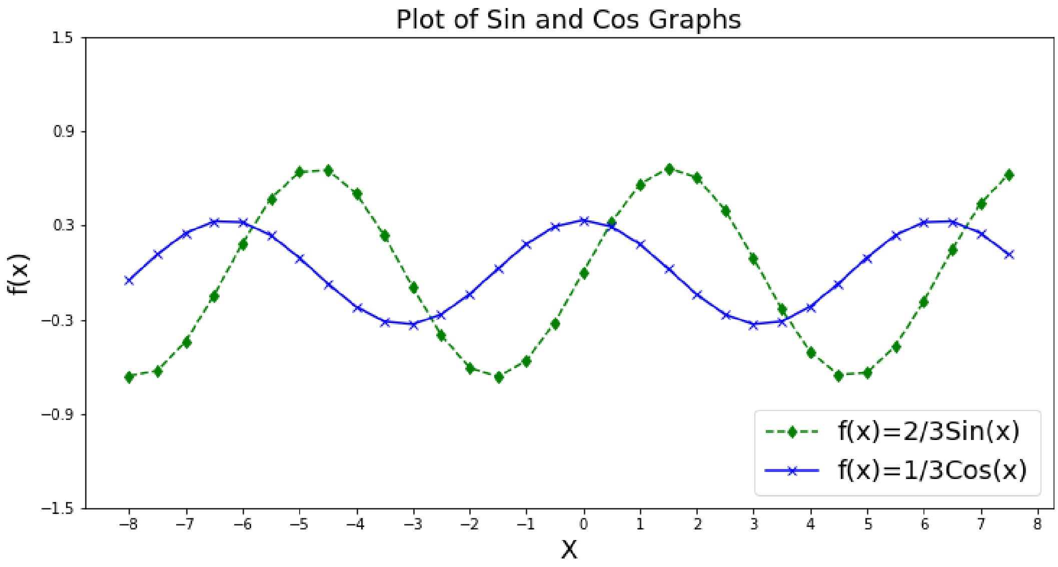
B. Create a 50x6 array of random integers between -10 and 200 (both included), and print it!

C. Print the number of elements that are strictly more than 50 in each column

D. Print the number of rows that contain negative values

Problem 4 [20 marks]

In this problem, you need to reproduce the plot shown below, as accurately as possible, from scratch. First, you will need to generate your x-axis data, and calculate the two series of your y-axis data using the simple functions shown in the legend



Problem 5 [20 marks]

In this problem, you will use a dataset called *TikTok Top Followers*. Please follow the instructions below for your data analysis.

A. Load the *TikTok_TopFollowed.csv* file in your notebook, and print the data set. Print the number of rows.

B. Print a list of the unique *Countries* (no repetition) represented in this data set.

C. You will notice that country names have an unnecessary leading space (e.g., “ United States”. Write a code to remove the leading space from every row in the ‘Country’ column. Save changes to your data frame. Re-run code in B to make sure it is solved now.

D. Create a new column: *‘American_Account’*: binary (*1* : *Country* is “United States”, *0* : otherwise)

E. Create a new column: *‘American_Person’*: boolean (*True* : *Country* is “United States” AND *Brand_Account* is NULL (not “Yes”), *False* : otherwise)

F. Calculate the following (you do not need to print the percentage sign ‘%’ in the answers):

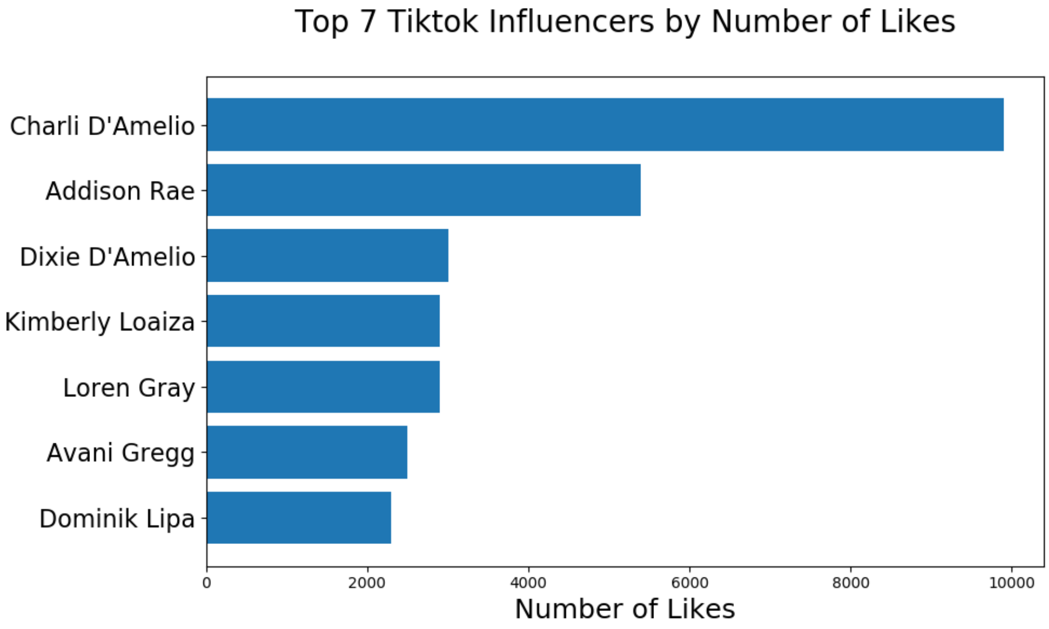
   a. Percentage of American accounts in the list = %100 * number of American Accounts / total number of accounts

   b. Percentage of American persons among all persons = %100 * ???/ ???

   c. Of all the American persons in the list, calculate the percentage of those who have 1 billion likes or more = %100 * ???/ ???

G. Create a new data frame, *df_persons*, which contains data only for persons i.e., *Brand_Account* is NULL (or alternatively *Brand_Account* is not “Yes”).

H. Reproduce the following plot: you will get marks for reproducing the plot as accurately as possible, taking into consideration the steps undertaken to reach the final figure.



Problem 6 [10 marks]

For this problem, use the same data from Problem 5 to perform compelling extra analysis. There are other columns in the *TikTok Top Followers* data set, you will get marks if you find a compelling and interesting visualisation of these too (one plot is enough, but you may produce as many as you want as long as they are all tied into one main idea). Make sure you provide textual description or analysis of the plot.