Koç University COMP125
Programming with Python: Midterm 4
Instructor: Beren Semiz
Date: May 23rd, 2021

12:00 PM – 1:40 PM
Deadline to submit on Blackboard: 1:45 PM

Instructions

Make sure you read and understand every part of this document

Download & Extract the zip to your Desktop:

Blackboard -> Assessments -> MT4-> MT4.zip

Zip file includes:

miniFunc.ipynb   courseGrades.ipynb   grades.txt   Midterm4_Spring2021.pdf

This midterm contains 2 programming questions. Each question may contain multiple parts.

Download MT4.zip from Blackboard and unzip the contents to a convenient location on your computer. The .ipynb files contain starter codes for the programming questions. **You should open them with Jupyter Notebook (not Spyder).** Remaining files are sample text/data files. Solve each question in its own file. You cannot use any library/module other than **random, math, numpy and matplotlib.**

This time, you won't be working with functions. **Jupyter Notebook files will be leading you about the steps you should implement. If you want, you can add new cells while writing your codes.**

When you are finished, compress your MT4 folder containing all of your **.ipynb files** (do not try to convert them to .py). The result should be a SINGLE compressed file (file extension must be .zip or .rar). Upload this compressed file to Blackboard.

Multiple attempts are allowed. We will grade only the LAST version submitted to Blackboard. **After 1: 45 PM, the system will be closed.**

By submitting this file, you automatically agree to the honor pledge: "You certify that you have completed this exam on your own without any help from anyone else. You understand that the only sources of authorized information in this exam are (i) the course textbook, (ii) the material that is posted at Blackboard for this class, and (iii) any study notes handwritten by yourself. You have not used, accessed or received any information from any other unauthorized source in taking this exam. The effort in the exam belongs completely to you."

Communication before & during exam:

Only clarification questions regarding the midterm are allowed.
Installation related or my_code_is_not_working questions will not be answered.
Join Zoom Meeting: https://kocun.zoom.us/j/95050481467
Meeting ID: 950 5048 1467
Passcode: comp125

Q1: Mini Functions – 50 pts (10 + 15 + 10 + 15)

The file associated with this question is miniFunc.ipynb. Solve the following parts – note that each part is independent.

Part A (10 pts): You are given the following set of equations:

$$\begin{gathered} y + z = 4 \\ x + 2y + 4z = 12 \\ 2x - 3y - z = 4 \end{gathered}$$

Solve the given linear system of equations using Numpy. Your result should be an array including the values for x, y and z.

For the given system the result should be as follows (both will be accepted):

If you don't use print: `array([12.,  8., -4.])`

If you use print: `[12.  8. -4.]`

Part B (15 pts): You are given two 1D Numpy arrays in the first cell. You will be calculating the covariance of these two arrays using the following formula (you are not allowed to use np.cov). Your final result should be a **float**:

$$cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X[i]-\bar{X})(Y[i]-\bar{Y})$$

where X is the first array, $\bar{X}$ is the mean of the first array, Y is the second array and $\bar{Y}$ is the mean for the second array.

For the given arrays, the result should be -0.012393406593406589

Part C (10 pts): Let's say we have a square matrix called **A**. If the inverse ($\mathbf{A}^{-1}$) and transpose ($\mathbf{A}^{T}$) of this matrix are equal, then A is an orthogonal matrix.

In the first cell, you are given a list of lists (you can assume that it is square and valid, you don't need to check for errors). First, convert it to a Numpy array.

- Then, write a code to check whether the given matrix is an orthogonal matrix.
- If it is orthogonal, you should print 'A is orthogonal', if not you should print 'A is not orthogonal'.

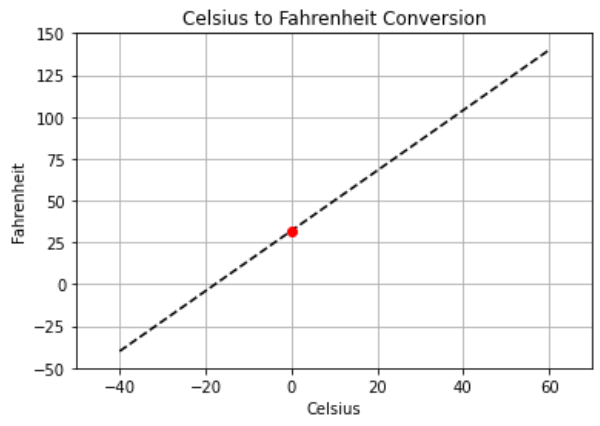
For the given list of lists, your result should be `A is orthogonal`

Part D (15 pts): Conversion between Fahrenheit (F) and Celsius (C) is achieved through the following formula.

$$F = \frac{9}{5}C + 32$$

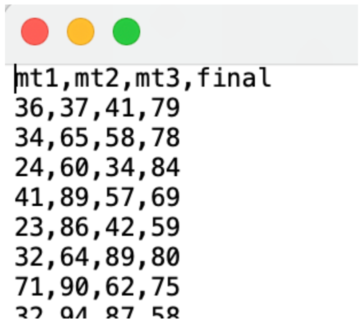
- First, create 100 equally spaced C values between [-40, +60] and compute $F = \frac{9}{5}C + 32$ at each of these C values.
- Then plot a line graph of the C values and F values found in the previous step
  - Your line should be a black dashed line.
  - Title of plot should be: "Celsius to Fahrenheit Conversion"
  - X axis label should be: "Celsius"
  - Y axis label should be: "Fahrenheit"
  - Limits should be as follows: xmin = -50, xmax = 70, ymin = -50, ymax = 150
  - You should have grids on your plot.
- Finally, add a red circle marker at (0,32). ***Hint:*** *You can again use plt.plot() function.*

Your plot should look like this:



Q2: Course Grades - 50 pts (15 + 10 + 10 + 15)

The file associated with this question is courseGrades.ipynb and it uses the grades.txt as input. The first few rows of the input file are as follows (note that the full file is longer!). The first line contains the headers of the table: mt1 is Midterm 1, mt2 is Midterm 2, mt3 is Midterm 3, final is the final exam grade.



```
mt1,mt2,mt3,final
36,37,41,79
34,65,58,78
24,60,34,84
41,89,57,69
23,86,42,59
32,64,89,80
71,90,62,75
32,94,87,58
```

Part A (15 pts):

- First, open "grades.txt" and read its contents.
- Then, calculate the average course grade for each student. While calculating each student's course grade, Mt1 and Mt2 weigh 20%, Mt3 weighs 25% and final weighs 35%.
- Your result should be a 1D Numpy array (array[0] is the 1st student in the file, array[1] is the 2nd student in the file, etc.)

For the given file the result should be as follows (both will be accepted):

If you don't use print:

```
array([ 52.5 ,  61.6 ,  54.7 ,  64.4 ,  52.95,  69.45,  73.95,  67.25,
        46.3 ,  39.  ,  50.75,  57.6 ,  47.7 ,  43.95,  40.85,  62.25,
        63.65,  39.3 ,  47.85,  39.25,  37.45,  30.7 ,  50.1 ,  49.15,
        42.8 ,  87.9 ,  88.3 ,  21.85,  62.9 ,  44.6 ,  51.  ,  85.75,
        56.9 ,  36.65,  39.  ,  39.55,  34.2 ,  43.9 ,  58.95,  36.7 ,
        23.15,  45.9 ,  84.35,  49.2 ,  54.9 ,  19.5 ,  23.75,   8.5 ,
        24.05,  37.65,  55.55,  63.85,  44.8 ,  85.7 ,  76.75,  47.05,
        58.  ,  65.  ,  36.45,   0.  ,  64.5 ,  63.1 ,  52.3 ,  91.5 ,
        44.65,  36.85,  44.65,  41.3 ,  27.4 ,  35.2 , 100.  ,  73.15,
        34.8 ,  52.65,  27.75,  37.3 ,  49.05,  47.8 ,  82.05,  48.55,
        59.4 ,  49.  ,  21.75,  49.95,  38.15,  66.65,  62.35,  62.95,
        42.95,  74.25,  28.3 ,  31.2 ,  37.05,  39.1 ,  23.1 ,  69.5 ,
        18.3 ,  64.1 ,  47.4 ,  46.75,  62.2 ])
```

If you use print:

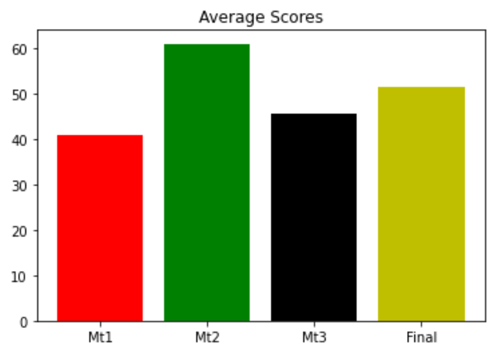
```
[  52.5    61.6    54.7    64.4    52.95   69.45   73.95   67.25   46.3    39.
   50.75   57.6    47.7    43.95   40.85   62.25   63.65   39.3    47.85   39.25
   37.45   30.7    50.1    49.15   42.8    87.9    88.3    21.85   62.9    44.6
   51.     85.75   56.9    36.65   39.     39.55   34.2    43.9    58.95   36.7
   23.15   45.9    84.35   49.2    54.9    19.5    23.75    8.5    24.05   37.65
   55.55   63.85   44.8    85.7    76.75   47.05   58.     65.     36.45    0.
   64.5    63.1    52.3    91.5    44.65   36.85   44.65   41.3    27.4    35.2
  100.     73.15   34.8    52.65   27.75   37.3    49.05   47.8    82.05   48.55
   59.4    49.     21.75   49.95   38.15   66.65   62.35   62.95   42.95   74.25
   28.3    31.2    37.05   39.1    23.1    69.5    18.3    64.1    47.4    46.75
   62.2 ]
```

Part B (10 pts):

- Plot a **bar chart** where each bar corresponds to the average class score obtained from a specific exam (i.e.: First bar will represent the average class score for Mt1, second bar will represent the average class score for Mt2, etc.).
  - Bar colors should be: **red, green, black and yellow** for Mt1, Mt2, Mt3 and Final, respectively.
  - You need to update your xticks as follows: **Mt1, Mt2, Mt3, Final**

  - Title of the plot should be: ‘Average Scores’
  - You do not need to change any other argument.

Your plot should look like this:

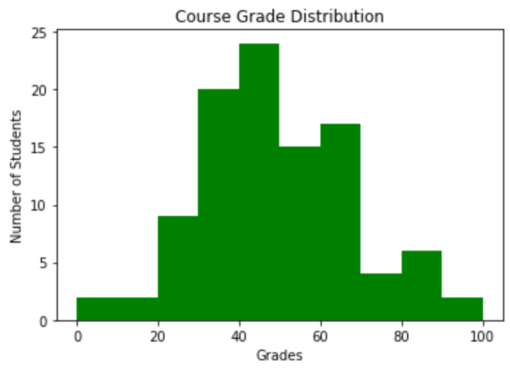


Part C (10 pts):

Plot the average course grade distribution using a **histogram**.

- You will be using the array you constructed in Part A.
- There should be 10 bins and the bins should be green.
  - *These 10 bins will be representing the grade ranges (e.g.: [70-80), [80-90), etc.)*
- X label should be ‘Grades’
- Y label should be ‘Number of Students’
- Title of the plot should be ‘Course Grade Distribution’
- You do not need to change any other argument.

Your plot should look like this:



Part D (15 pts):

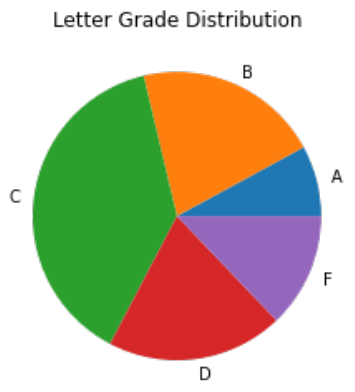
Finally, you would like to determine the letter grades for this course. Here is the letter grade rule:

[0 – 30) : F
[30-40) : D
[40-60) : C
[60-80) : B
[80-100]: A

As we covered in the lectures and Lab 12, histogram function has some return values. Using these values, plot a **pie chart** to visualize the letter grade distribution.

- Slice labels should be the letter grades: A, B, C, D, F
- Title should be 'Letter Grade Distribution'
- You do not need to change any other argument.

Your plot should look like this:



Submission + Honor Pledge

Solve each question in its own file. **DO NOT CHANGE THE NAMES OF THE FILES. DO NOT USE TURKISH CHARACTERS.**

When you are finished, compress your MT4 containing all of your answers:

miniFunc.ipynb courseGrades.ipynb

The result should be a SINGLE compressed file (file extension: .zip or .rar). Upload this compressed file to Blackboard.