Foundation

Advanced Python

Weeks 2 & 3

Module 4 Exercises

Version	Date	Author	Comments
1.0	26 / 11 / 21	Nikola Ignjatovic	First draft

Module 4A: Matplotlib

1. Write a Python program to display the popularity of programming languages worldwide using

   a) column plot

   b) bar plot

   c) pie plot (adding 'other' to the sample data with the difference to make 100%). Make Python stand out by 'exploding' it slightly from the pie plot.

   Sample data:

Programming languages	Python	Java	JavaScript	C#	C++	PHP	R
Popularity (%)	29.93	17.78	8.79	6.73	6.45	5.76	3.92

   (Source: https://pypl.github.io/PYPL.html, August 2021)

2. The company "ABC Group" keeps daily records of its shares values (£). The opening, high, low and closing values for 2nd August 2021 are given in the sample below:

Open	**High**	**Low**	**Close**
558.750000	570.559998	554.500000	569.065002

   Write a Python program to draw line plot showing ABCs shares values during 2nd August 2021.

3. The company "ABC Group" also keeps weekly records of its shares values (£). The opening, high, low and closing values for each day between 2nd and 6th August 2021 are given in the sample below:

Date	Open	High	Low	Close
02-08-2021	558.750000	570.559998	554.500000	569.065002
03-08-2021	556.835021	558.710022	552.890015	556.429993
04-08-2021	557.658439	578.070007	555.650024	556.469971
05-08-2021	559.213903	580.479980	555.039978	556.859985
06-08-2021	558.307553	559.659973	556.250000	557.034628

Write a Python program to draw line plot of ABCs shares values during this period.
Tip: Create a grouped line plot where each day will be represented with a separate line. Include the legend to list the dates.

4. Change the group line plot produced in task 3 so that each line in the plot represents a point during the day (open, high, low, close). The 5 weekdays should be displayed on the x-axis, and share values (£) on the y-axis. Include the legend to list the points during the day (day-points) for which shares values are given: open, high, low, close.

5. Use the sample data from question 3 to write a Python program to draw a column plot of ABCs shares values during this period.

Tip: create a grouped column plot where each day-point will be represented with a separate column (grouping the 4 columns together for each of the 5 days). Include the legend to list the day-points.

6. Use the sample data from question 3 to write a Python program to draw a column plot of ABCs stock's return on each of the 5 days.
   The stock's return (or its performance in percentage terms) is calculated as the difference between the stock's open and close, divided by the open.

   Note: question 6 requires data for two day-points only: 'open' and 'close'.

7. Combine the two plots from questions 5 and 6 into one figure using

   a) subplots() function

   b) subplot() function

8. ABC Group is looking at their domestic and international opportunities, comparing the revenue that can be generated to the probability of obtaining the revenue. Write a Python program that loads data from the file abc_dom_int and draws a grouped scatter plot, where the first plot compares revenue vs probability for domestic opportunities, and the second plot compares revenue vs probability for international opportunities. Add a label to each scatter point. The program should be able to work with any number of opportunities (not just the 10 listed in the sample file abc_dom_int).

   Use the following link to find out how to add a label to each scatter point
   https://www.delftstack.com/howto/matplotlib/matplotlib-label-scatter-plot-points/

9. The sample file abc_training_costs provides data for ABC's trainings and related costs. Write a Python program that loads data from the file abc_training_costs and draws a figure combining four column subplots positioned in two rows:

   - The first subplot, positioned in 1st row and 1st column, to show the total training costs incurred by each office (Finland, France, Germany, Ireland, Spain)
   - The second subplot, positioned in 1st row and 2nd column, to show the total training costs by level (Management, Non management)
   - The third subplot, positioned in 2nd row and 1st column, to show the total training costs by Course-type (Technical, Soft skills)
   - The fourth subplot, positioned in 2nd row and 2nd column, to show the total training costs by Course-type (Technical, Soft skills).

   Combine the four subplots using:

   a) subplots() function

   b) subplot() function

   Tip: implement the SUMIF Excel function and use it to produce data for all four plots

10. The sample data from questions 2 & 3 are stored in the file abc_shares_data.

    a) Perform the task in the same scenario as in question 2 by loading the data needed for the plot from the file abc_shares_data into a NumPy array

    b) Perform the task in the same scenario as in question 3 by loading the data needed for the plot from the file abc_shares_data into a NumPy array

c) Perform the task in the same scenario as in question 4 by loading the data needed for the plot from the file abc_shares_data into a NumPy array

Note: the file abc_shares_data is just a sample. The solutions must work for any number of rows and columns.

Tip: Questions 10b and 10c can be done in two ways:

1) Using the list of arrays:

   In Question 10b extract data for each day, convert them to float data type and store them in a list of arrays 'days_data', so that:
   index 0 corresponds to data array for day1,
   index 1 corresponds to data array for day2,
   . . .
   index N-1 corresponds to data array for dayN (N=num_rows)

   In Question 10c extract data for each point during the day, convert them to float data type and store them in a list of arrays 'day_points_data', so that:
   index 0 corresponds to data array for open,
   index 1 corresponds to data array for high,
   index 2 corresponds to data array for low,
   index 3 corresponds to data array for close (in general N, where N=num_cols)

2) Find out how the `exec()` function works and use it in question 10b to execute the statement to populate arrays for day1 - day5, and in question 10c to execute the statement to populate arrays for day_point1 data (open) to day_point4 data (close)

Implement the solution for questions 10b and 10c in both ways.

11. Modify the Python program produced for question 5 to load data from the file abc_shares_data into a NumPy array and to generalise the solution to work for any number of rows and columns, using the list of arrays, as done in question 10c.

12. Modify the Python program produced for question 5 to load data from the file abc_shares_data into a NumPy array and to generalise the solution to work for any number of rows and columns, using the `exec()` function, as done in question 10c.

13. Modify the Python program produced for question 6 to load data from the file abc_shares_data into a NumPy array and to generalise the solution to work for any number of rows and columns. Assume that the file abc_shares_data lists dates in the 1st column, open values in the 2nd column, and close values in the last column.

    Note: Since question 13 (as question 6) requires data for two day-points only: 'open' and 'close', there is no need to involve the `exec()` function or use the list of arrays to store data for each day-point. Simply extract data from the 1st and the last column of the array.

14. Modify the Python programs produced for questions 7a and 7b to load data from the file abc_shares_data into a NumPy array and to generalise the solution to work for any number of rows and columns, using the list of arrays for the first plot, as done in question 11. For the second plot extract data from the 1st and the last column of the array as done in question 13.

15. Modify the Python programs produced for questions 7a and 7b to load data from the file abc_shares_data into a NumPy array and to generalise the solution to work for any number of rows and columns, using the `exec()` function for the first plot, as done in question 12. For the second plot extract data from the 1st and the last column of the array as done in question 13.

Module 4B: Seaborn

1. Using the 'titanic' built-in Seaborn data set, create a distribution plot to visualise the survival distribution

   a) by each of the three passenger classes (first, second, third)

   b) by each of the three passenger types (man, woman, child)

   c) by each gender (male, female)

2. Bundle the three plots created in question 1 together into one multi-plot (grid of subplots) with one row and tree columns.

3. Using the 'titanic' Seaborn built-in dataset, illustrate the number of male and female passengers on the Titanic split by class using:

   a) countplot() function, including the numbers on top of each bar

   b) count plot through catplot() function, including the numbers in the centre of each bar

   c) barplot() function, including the numbers on top of each bar

   d) bar plot through catplot() function, including the numbers in the centre of each bar

   e) multiple bar plot, where each subplot relates to a different gender value. Include the numbers in the centre of each bar. Use the 'bar' value for kwarg kind.

   f) multiple bar plot, where each subplot relates to a different gender value. Include the numbers in the centre of each bar. Use the 'count' value for kwarg kind.

4. Using the 'titanic' Seaborn built-in dataset, illustrate the average age per class of the titanic passengers:

   a) using the barplot() function

   b) using the catplot() function

   Include the numbers at the centre of bars.

5. Using the 'titanic' Seaborn built-in dataset, illustrate the average age per gender of the titanic passengers:

   a) using the barplot() function

   b) using the catplot() function

   Include the numbers at the centre of bars.

6. Using the 'titanic' Seaborn built-in dataset, illustrate the average age per gender split by class of the titanic passengers:

   a) using the barplot() function

   b) using the catplot() function

   Include the numbers at the centre of bars.

7. Using the 'titanic' Seaborn built-in dataset, illustrate

   a) the number of passengers per class split by the type of person (man, woman, child). Use the 'ocean' colour palette and display values on top of the bars in purple colour. using the catplot() function

b) the survival rate of passengers per class split by type of person. Use the 'inferno' colour palette and display values in the centre of the bars in white colour.

Include the numbers at the centre of bars.

8. Using the 'titanic' Seaborn built-in dataset, illustrate the number of passengers by class split by each age group, for each facet of survival, following the sinking of the titanic

   a) creating a new data frame using queries, consisting of the following columns: class, age_range, survived and a column listing the number of passengers for each (class, age_range, survived) combination

   b) creating and populating a new column within the existing data frame
   step 1: define a function set_age_range() that will populate the new column 'age_range' with the different age ranges, as defined below
   step 2: create a new column in the titanic data frame and populate it passing the custom-made set_age_range() function to the apply() method

   Categorise the age using the following categories:
   <18
   18-25
   26-34
   35-44
   45-54
   55-64
   65+

   Include numbers of top of the bars.

   Tip: to create the data frames for questions 8a) and 8b) refer to questions 13a) and 13b) from Pandas exercises (Module 3B).

9. Using the new data frame created in question 8a, illustrate the average number between survived and deaths per class across all age ranges, including the numbers on top of each bar.

10. Using the new data frame created in question 8a, illustrate the average number between survived and non-survived passengers per class, split by age range, including the numbers on top of each bar.
Reflect on the data in the data frame created in task 8a to figure out how to resolve the task.

11. Using the expanded titanic data frame created in question 8b, illustrate the passengers' survival rate per class, split by age range, including the numbers on top of each bar.
Reflect on the data in the expanded titanic data frame to figure out how to resolve the task.

12. Using the 'titanic' Seaborn built-in dataset, create a scatter plot to show the dependency:

a) between 'age' and 'fare' variables

b) between 'age' and 'fare' variables against the data variable 'sex'

c) between 'age' and 'fare' variables against the data variable 'who' (type of person) representing each type of person with a different size of data points.

d) same as in task 10c) but show each size with a different colour

e) same as in task 10c) but instead of the size, use a different shape for data points of each type of person.

f) same as in task 10c) but create a multiple plot consisting of two subplots, each showing values for each facet of the genders: male, female

13. Using the 'titanic' Seaborn built-in dataset, create a joint plot to combine a scatterplot(), showing the bivariate distribution of passengers' age and fare, and two histplot(), showing marginal distributions of ages on top and of fares on the right. Add a linear regression fit to scatter plot (using regplot()) and univariate KDE curves to hist plots. Drop missing values from the data before plotting.

14. Using the 'titanic' Seaborn built-in dataset, change the scatter plot created in task 10a by including the rugplot() to add ticks ('rugs') along the x and y axes. The rugs on the bottom should illustrate the distribution of passengers' age alone. The rugs on the left should illustrate the distribution of passengers' fare alone.