Graphs with Matplotlib

A picture is worth a thousand words, and with Python's matplotlib library, fortunately, takes far less than a thousand words of code to create a production-quality graphic. Matplotlib is the brainchild of John Hunter (1968-2012), who, along with its many contributors, have put an immeasurable amount of time and effort into producing a piece of software utilized by thousands of scientists worldwide. Learning matplotlib can be a frustrating process at times. The problem is not that matplotlib's documentation is lacking: the documentation is actually extensive.

Using the library Matplotlib and the provided data files create the following graphs. Your program should open and read the given data files and carefully collect the data and plot the graphs.

Please use Jupyter Notebook to complete this assignment.

a) **Pie chart**
Create a pie chart that shows the percentage of employees in each department within a company. The provided file: *employee_count_by_department.txt* contains the data required in order to generate this pie chart.

Below shows what the output of the program should look like:
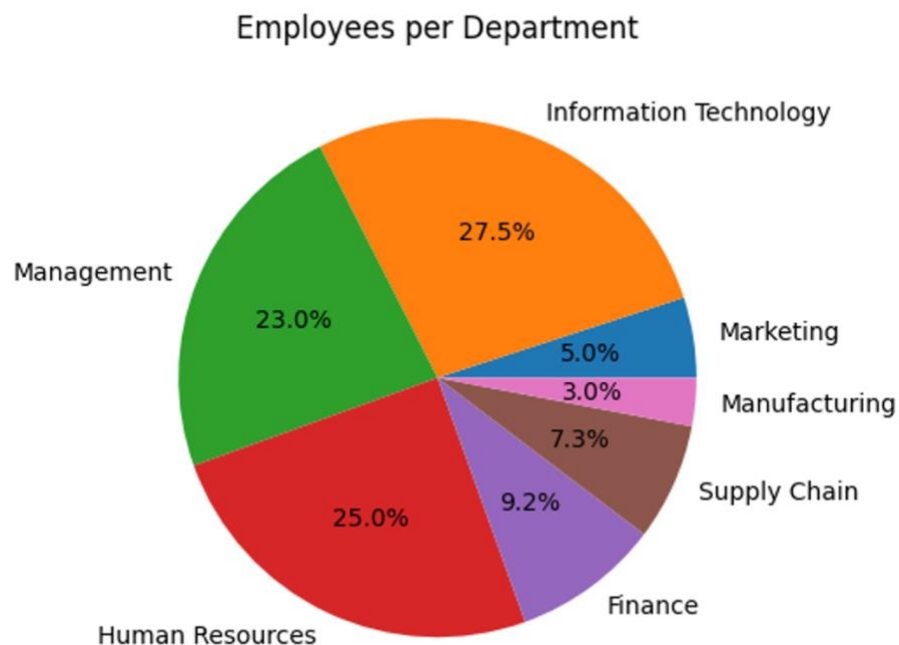
b) Line Graph

Create a line graph that shows a company's profit over the past ten years. The provided file: *last_ten_year_net_profit.txt* contains the data required in order to generate this line graph.
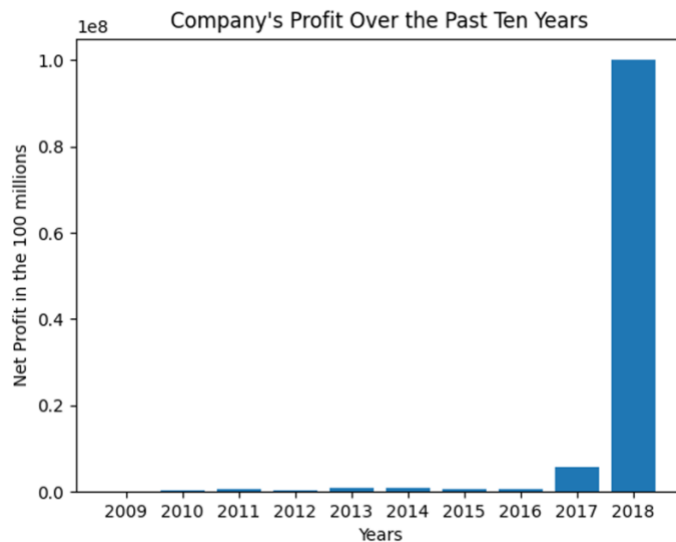
Below shows what the output of the program should look like:

c) Bar Graph

Create a bar graph that shows a company's profit over the past ten years. The provided file: *last_ten_year_net_profit.txt* contains the data required in order to generate this bar graph.

Below shows what the output of the program should look like:

Info: Be sure to label all the axes of each graph with meaningful labels and provide an appropriate title to each of the graphs (mentioned above) being generated. Hint, the names of the files and the headers provided in each of the data files should help.

Write a Report Summary
Using Microsoft Word or any text editor, answer the following questions. Please describe your answers, do not just say yes or no.

1. Did you complete your assignment and did your program run without errors?
2. Did your program produce the correct result?
3. Did you test your program thoroughly?
4. How much time did you spend completing your assignment?
5. Did you write the program yourself? Did you get any help from anyone?
6. How did you resolve the issues when you encountered obstacles to completing your program? Did you use Google or other resources to get help? Describe how Google or other resources was abled or not able to assist you.
7. What did you learn from doing this assignment?
8. Any other information you would like to share with your instructor?

What to submit

1. Submit your Python program files (.ipynb files)
2. Submit your program output
3. Submit your learning report summary