04-638 A: Programming for Data Analytics
Practice Analytics Task 1 [110 Pts]
Release Date: 13-October-2023
Due Date: 3-November-2023

Instructions

1. Include internal documentation as appropriate.
2. Submit a Jupyter Notebook file on Canvas. The file name should be andrewid.ipynb.

Introduction

In this course, you will have seven practice tasks to help you improve your programming and analytics skills and prepare you for your assignments. These tasks are low-stakes formative assessments, but they contribute to 10% of your grade. Being low-stakes tasks, the main aim is to facilitate learning. This implies that if you determine that learning did not occur, you can redo the assignment and have it reassessed. When reassessed, you have the chance to make up to 90% of the points/marks lost during the first grading cycle.

In this task, you will get some practice with matplotlib, numpy and pandas. The practice exercises are assessed for correctness.

All work can be completed in a Jupyter Notebook.

Task 4 Objectives:

- Perform numerical data manipulation using numpy.
- Create data visualization using matplotlib.
- Use pandas for elementary data preparation, transformation, manipulation, and analysis tasks.

Section 1: Numpy

Question 1: Analyzing Monthly Expenses [15 Pts]

Scenario:You are tasked with analyzing monthly expenses for a small business. You have two lists: one containing monthly expenses for the current year, and another for the previous year.

current_year_expenses = [1200, 1350, 1100, 980, 1500]

previous_year_expenses = [1150, 1300, 1050, 950, 1400]

a) Create a Python script that imports NumPy, sets up two arrays based on the two lists, and calculates the following statistics for both current and previous year's expenses:

- Mean (average) expense.

- Median expense.
- Standard deviation of expenses.
- Total expenses.

b) b. Imagine you are a financial analyst presenting these expense statistics to the company's executives. In your code, calculate and print the following based on the calculated means of the current and previous year's expenses:

- The difference between the mean expenses of the current year and the previous year.
- The percentage change in mean expenses from the previous year to the current yea*r*.

c) c. You are responsible for financial risk assessment. In your code, calculate and print the following practical metrics based on the standard deviation of expenses for the current and previous year:

- The coefficient of variation (CV) for both years (CV = (Standard Deviation / Mean) * 100).
- A risk assessment message for each year based on the CV (e.g., "Low Risk" for CV < 20, "Moderate Risk" for 20 <= CV < 50, "High Risk" for CV >= 50).

Question 2: Sales Forecasting with NumPy [10 Pts]

Scenario: Suppose you want to forecast monthly sales for the upcoming year based on historical sales data. You have access to the monthly sales data for the last three years.

Historical sales data for the last three years (12 months each year):

year1_sales = [12000, 13500, 11000, 9800, 15000, 12300, 13700, 11150, 9750, 14850, 13000, 15500]
year2_sales = [12500, 13000, 11200, 9900, 15200, 12600, 14200, 11350, 9950, 15000, 13500, 16000]
year3_sales = [12300, 13700, 11150, 9750, 14850, 12800, 13900, 11000, 9600, 14500, 12750, 14200]

a) Create a Python script that imports NumPy (you need not re-import NumPy though if you completed Question 1), sets up a suitable array (or arrays), and calculates the monthly sales forecast for the upcoming year using the historical data.

   We will keep it simple. To calculate the forecast, determine each month's average (mean) sales by considering the sales data from the last three years. This will give you a reasonable estimate of the upcoming year's monthly sales.

b) Consider that external factors such as discounts and promotions can significantly affect sales. Write Python code to adjust the monthly sales forecast based on a list of percentage changes for each month due to expected promotions.

List of percentage changes for each month due to promotions (e.g., 10% increase in March, 7% decrease in January):
promotion_percentages = [-0.07, 0.0, 0.10, 0.0, 0.0, -0.05, 0.0, 0.0, 0.08, 0.0, 0.0, 0.15]

Section 2: Matplotlib

Question 3: Basic matplotlib plot [15 Pts]

Scenario: Imagine you are analyzing the monthly sales data of a small business. You have a list of months and corresponding sales values. Your task is to create a basic line plot using matplotlib.

months = [Jan, Feb, Mar, Apr, May]
sales = [10000, 12000, 9000, 11000, 13000]

a) Create a Python script that imports matplotlib and plots a line graph of the monthly sales data.

- Set appropriate labels for the x-axis and y-axis.
- Add a title to the plot.

b) Modify the script to change the line style to a dashed line.
c) Change the line color to blue.
d) Add a legend with the label "Sales" to the plot.
e) Add a markdown cell to explain the patterns or trends in the data, if any.

Question 4: Enhancing the Sales Plot [15 Pts]

a) Modify the script from Question 3 to include the following:

- Plot the sales data using a red solid line.
- Add markers to data points.
- Include grid lines in the plot.

b) Imagine you are presenting this sales data to your company's management. Write Python code to add a label to the highest data point on the plot, indicating the maximum sales value. Use plt.annotate() to label the highest data point with the maximum sales value on your sales data plot.)

c) Add markdown cells to briefly explain how adding markers to data points in the plot can help them understand and make decisions based on the sales trends.
d) You are preparing a report for a client, and you've used grid lines in your sales plot. Write Python code to make the grid lines dash instead of solid.
e) Add a markdown cell to describe how the inclusion of grid lines aids in presenting data accurately and facilitating a better understanding of the sales performance.

Question 5: Plotting Monthly Sales with Trends [15 Pts]

a) Modify the script from Question 4 to include the following:

- Plot the sales data using a solid green line.
- Add a trendline (a linear regression line) to represent the overall sales trend.
- Add labels for the trendline and data points.

b) Analyze Seasonal Sales by Creating a Python function that takes the monthly sales data as input and identifies the months with the highest and lowest sales.
c) Compare Monthly Sales Across Years:

- Imagine you have data for sales from the previous year and want to compare it with the current year. Modify the sales plot from Question 4 to include sales data from the previous year, and use different colors for the current and previous years' data.

d) Highlight Special Events:

- You want to highlight special events or promotions that occurred during specific months. Modify the sales plot to include markers or annotations for these events (e.g., using '^' markers for promotions).

e) Explaining Insights:

- Add markdown cells to provide brief explanations of the insights that can be drawn from the sales plots in Question 5(a-d) and how this information can be useful for decision-making in a business context.

Section 3: Pandas

Question 6: Load, Inspect, and Clean Data [20 Pts]

a) Load and Inspect Data

- Load the dataset using pandas and display the first 5 rows.
- Provide the basic information about the dataset (number of rows, columns, data types).

- Identify and print the summary statistics (mean, median, min, max, etc.) for numerical columns.

b) Data Cleaning

- Handle missing values in the dataset if any. Use markdown cells to explain your strategy for dealing with missing data.
- Check for and remove any duplicate rows if there are.
- Check for any outliers (How do you check for outliers?)

Question 7: Elementary Data Analysis [20 Pts]

a) Comparative Analysis:

- Compare the crime rates across the five most frequent provinces for 2011-2012.
- Identify the province with the highest and lowest crime rates in the “All theft not mentioned elsewhere” category over the entire period.

b) Ranking:

- Rank the top 5 stations based on the average crime rate over the entire period.
- Identify the top 5 stations with the highest and lowest crime rates over the entire period.

c) Change Over Time:

- Identify the top 5 stations where there has been a significant decrease in crime rates over consecutive years.
- Plot the percentage change in crime rates for Pretoria Central, Cape Town Central, Brooklyn, and Pretoria West stations over the years.

d) Aggregate Analysis:

- Aggregate the data at the province level (Use only the 5 most frequent provinces) and analyze the overall crime trends for each province.
- Identify provinces (Use only the 5 most frequent provinces) where certain categories of crime are more prevalent.