

Robert Pavlik

D598 Analytics Programming

Task 3

A.

The program written for Task 2 performs data analysis on a dataset of 150 US companies and follows several steps:

1. Importing Libraries and Dataset:
 - a. The pandas library is imported to manage and manipulate data.
 - b. The dataset is read using `pd.read_excel()` for an Excel file named 'D598 Data Set.xlsx'.
 - c. The dataset is stored in a DataFrame called 'ds'.
2. Checking and Removing Duplicates:
 - a. The code checks for any duplicate rows in the dataset using `ds.duplicated()`
 - b. If duplicates exist, they are removed using `ds.drop_duplicates()`
3. Grouping Data by State:
 - a. The companies in the dataset are grouped by the 'Business State' column using the `groupby()` method. This groups the data based on each unique state
4. Calculating Descriptive Statistics:
 - a. For each numeric column related to financial data (like 'Total Long-term Debt', 'Total Equity', etc.) the program calculates descriptive statistics (mean, median, min, max) using the `agg()` function. These statistics help in understanding the distribution of financial metrics by state.
5. Filtering Negative Debt-to-Equity Ratios:
 - a. The program identifies companies with negative 'Debt to Equity' ratios by filtering the DataFrame: `ds[ds['Debt to Equity'] < 0]`.
6. Calculating Debt-to-Income Ratio:
 - a. A new column 'Debt-to-Income Ratio' is calculated for each business by dividing 'Total Long-term Debt' by 'Total Revenue'. This provides insights into how much debt a company has in relation to its income.
7. Concatenating DataFrames:
 - a. The newly calculated 'Debt-to-Income Ratio' is added back into the original DataFrame. This is done using `pd.concat()` to ensure that the DataFrame now includes this additional metric.
8. Displaying Results:
 - a. Finally the program prints the first five rows of the resulting DataFrame, including the newly added 'Debt-to-Income Ratio' column

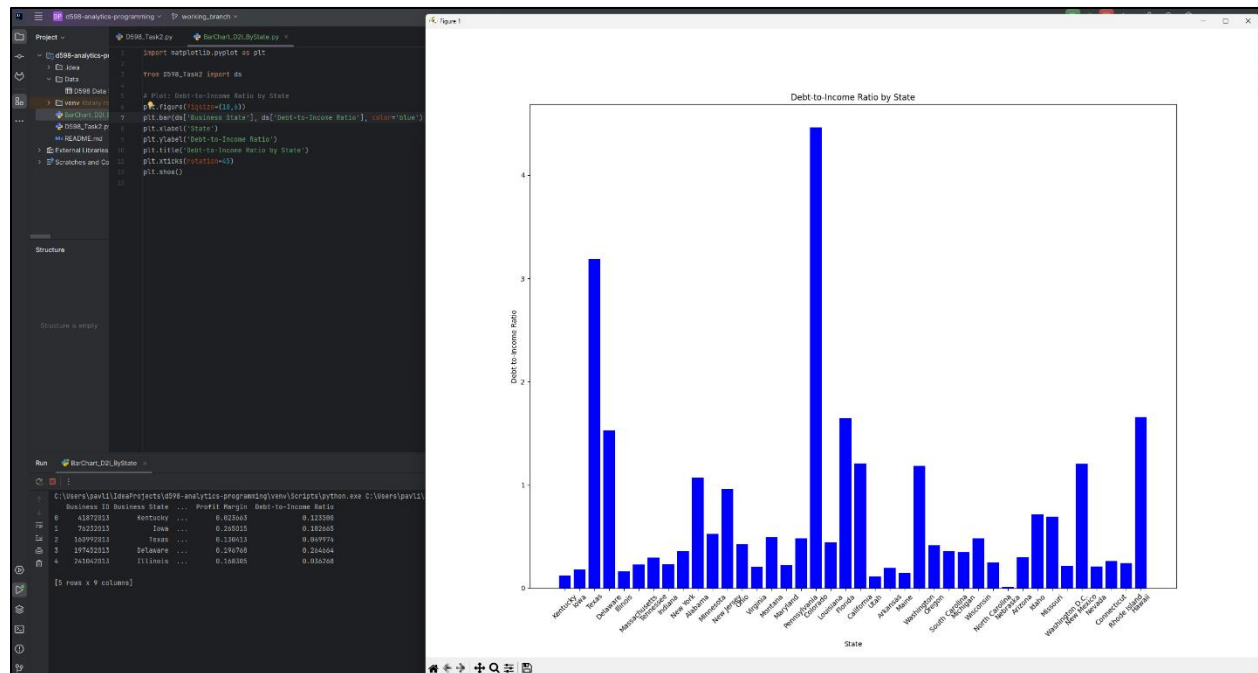
```

1  #Import Library
2  import pandas as pd
3
4  dataset_path = 'data/D598 Data Set.xlsx'
5
6  #Import the dataset
7  ds = pd.read_excel(dataset_path, engine='openpyxl')
8
9  #Check for duplicates and remove
10 if ds.duplicated().any():
11     #Remove duplicates
12     ds = ds.drop_duplicates()
13
14 #Group data by State
15 grouped_ds = ds.groupby('Business State')
16
17 #Calculate statistics (mean, median, min, max)
18 stats_ds = grouped_ds.agg({
19     'Total Long-term Debt': ['mean', 'median', 'min', 'max'],
20     'Total Equity': ['mean', 'median', 'min', 'max'],
21     'Debt to Equity': ['mean', 'median', 'min', 'max'],
22     'Total Liabilities': ['mean', 'median', 'min', 'max'],
23     'Total Revenue': ['mean', 'median', 'min', 'max'],
24     'Profit Margin': ['mean', 'median', 'min', 'max']
25 }).reset_index()
26
27 #Filter rows where 'Debt to Equity' is negative
28 negative_debt_equity_ds = ds[ds['Debt to Equity'] < 0]
29 💡
30 #Calculate Debt-to-Income Ratio for each business
31 ds['Debt-to-Income Ratio'] = ds['Total Long-term Debt'] / ds['Total Revenue']
32
33 #Concatenate original dataframe with the newly calculated Debt-to-Income Ratio
34 final_ds = ds
35
36 #Display the final result
37 print(final_ds.head())

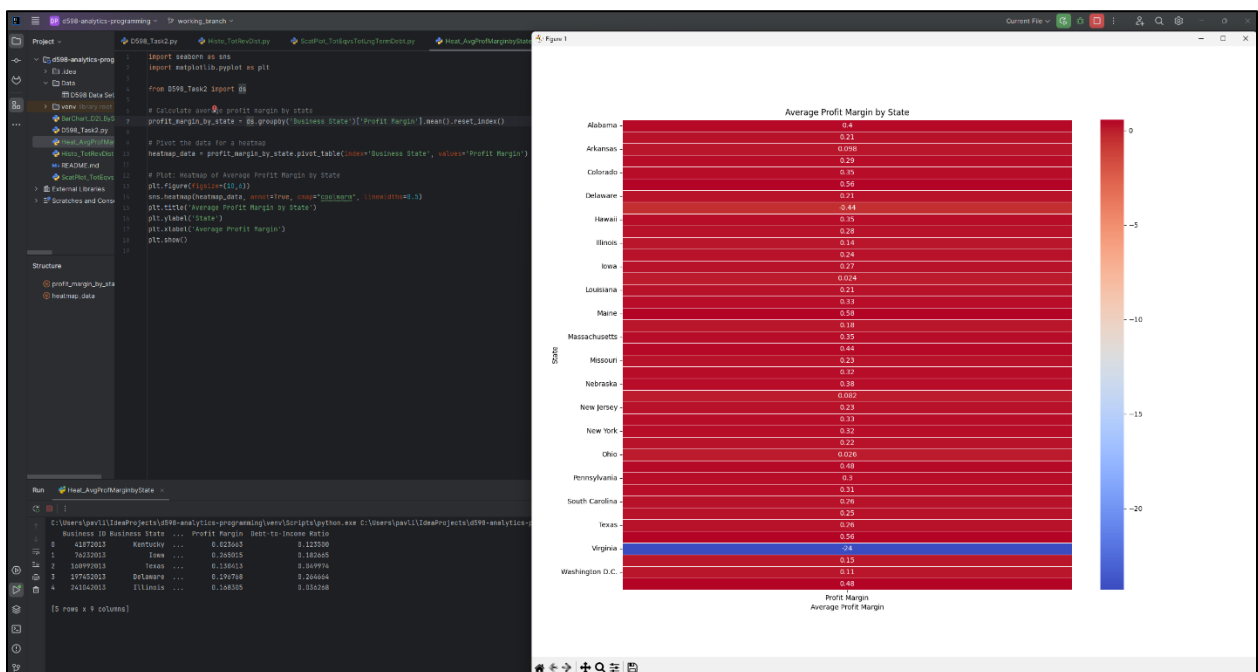
```

B.

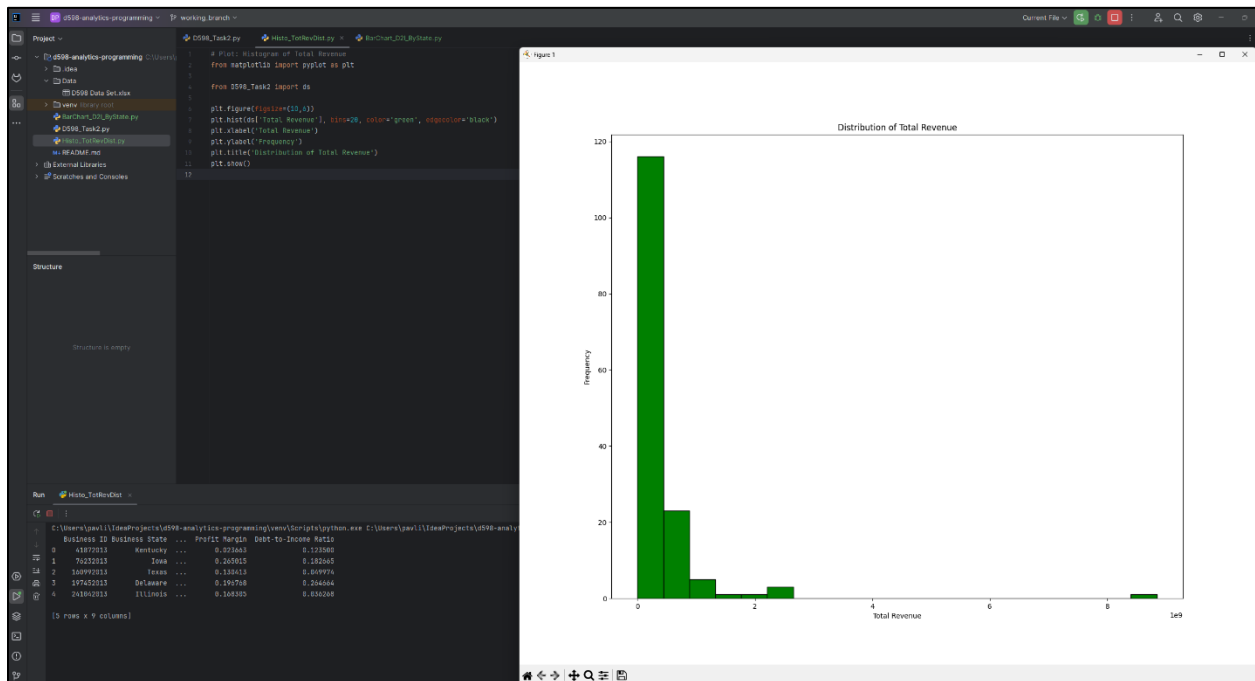
Bar Chart: Debt-to-Income Ratio by State



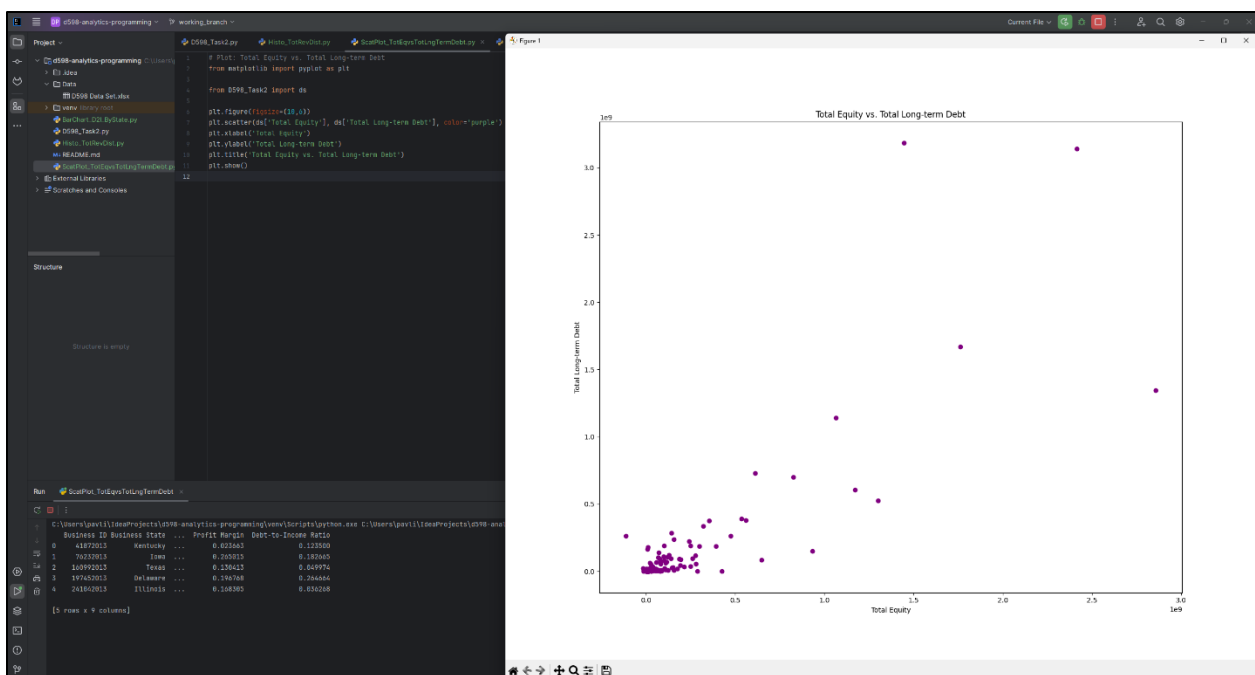
Heatmap: Average Profit Margin by State



Histogram: Total Revenue Distribution



Scatter Plot: Total Equity vs Total Long-term Debt



C.

1. Bar Chart: Debt-to-Income Ratio by State

The bar chart visualizes the debt-to-income ratio for companies in different states. The `plt.bar()` function from Matplotlib is used to create a bar chart, where each bar represents a state, and the height corresponds to the average debt-to-income ratio. The x-axis contains the state names, and the y-axis represents the ratio.

2. Heatmap: Average Profit Margin by State

The heatmap visualizes the average profit margin for each state in a color-coded format, allowing for quick identification of states with higher or lower profit margins. States are listed on the y-axis, and the corresponding average profit margin is represented by the color intensity on the x-axis.

3. Histogram: Total Revenue Distribution

The histogram visualizes the distribution of total revenue across the dataset. The `plt.hist()` function is used, where the data is divided into bins (ranges of values), and the height of each bar indicates the frequency (number of companies) in the range. This gives insights into the concentration of total revenue values.

4. Scatter Plot: Total Equity vs Total Long-term Debt

The scatter plot is used to visualize the relationship between total equity and total long-term debt for each company. The `plt.scatter()` function creates a plot where each point represents a company, with total equity on the x-axis and total long-term debt on the y-axis. This plot helps to observe any correlation between equity and debt.