Staphon Smith

D598 Analytical Programming - Task 3

A. Explanation of Code

In this section, we will discuss the core functions and logic behind the Python code used to analyze the provided dataset, including the key data wrangling steps.

1. **Importing the Data:** The dataset is imported using the pandas library in Python, which allows the data to be loaded into a DataFrame for easy manipulation.

2. **Identifying Duplicate Rows:** We check for and remove any duplicate rows in the dataset, ensuring the integrity of the data before analysis.

3. **Descriptive Statistics by State:** The data is grouped by state, and descriptive statistics (mean, median, min, max) are computed for each numeric variable within each state.

4. **Filtering for Negative Debt-to-Equity Ratios:** We filter the dataset to identify companies with negative debt-to-equity ratios, which may indicate financial distress.

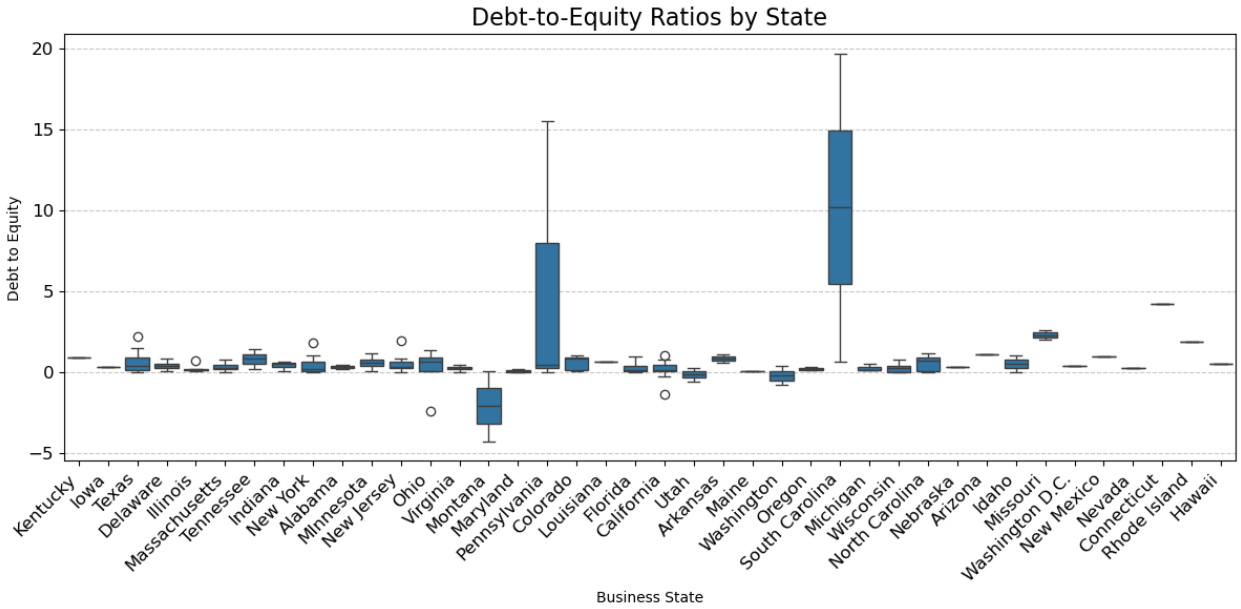
5. **Debt-to-Income Ratio:** A new data frame is created to calculate the debt-to-income ratio, which is defined as long-term debt divided by revenue.

6. **Concatenating DataFrames:** The newly created debt-to-income ratio data is concatenated with the original data for further analysis.

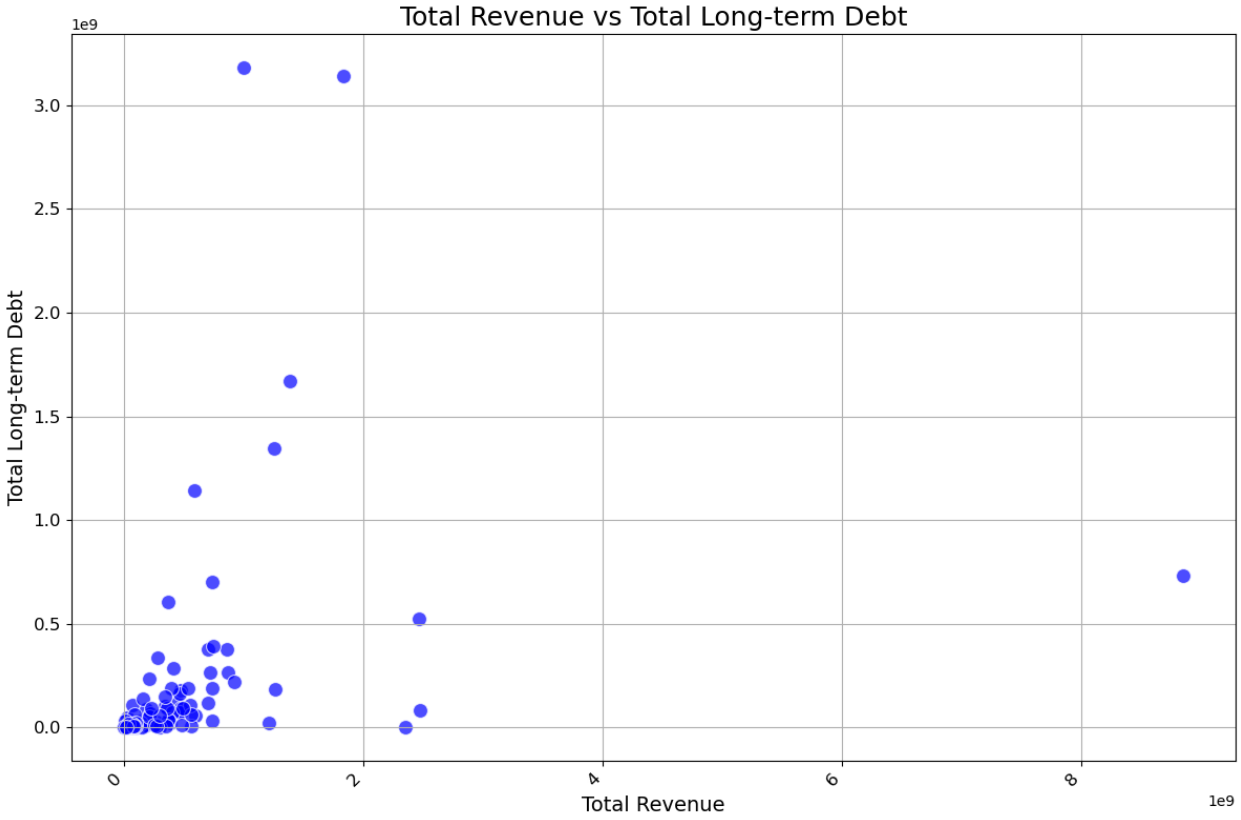
Staphon Smith

B: Customized Data Visualizations

Visualization 1:

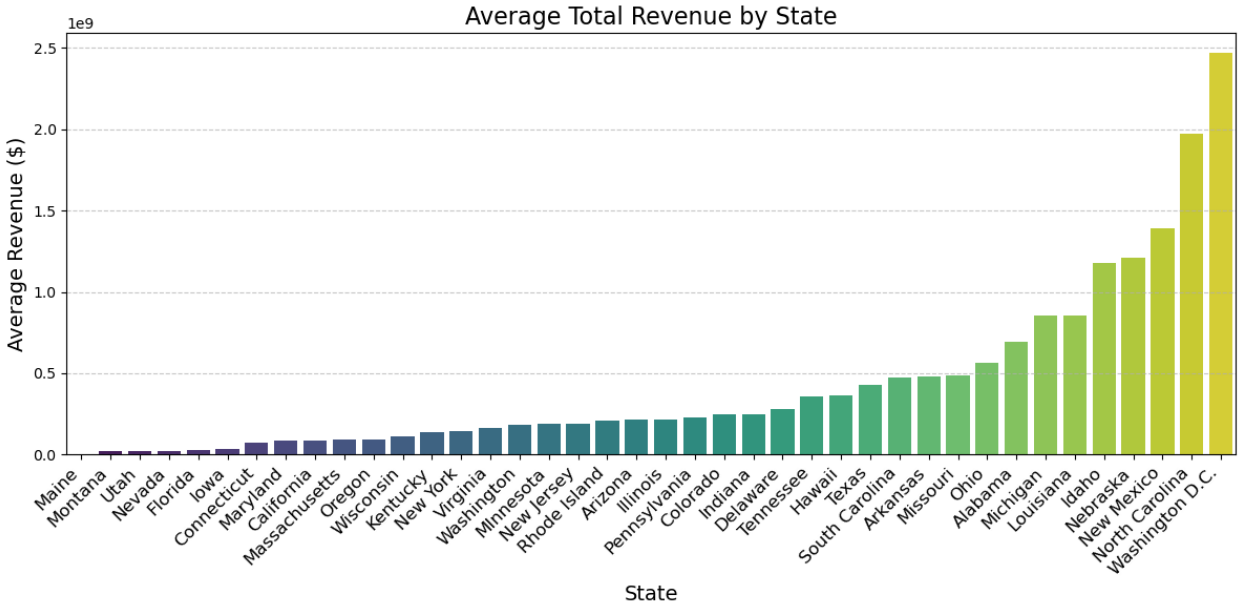


Visualization 2:

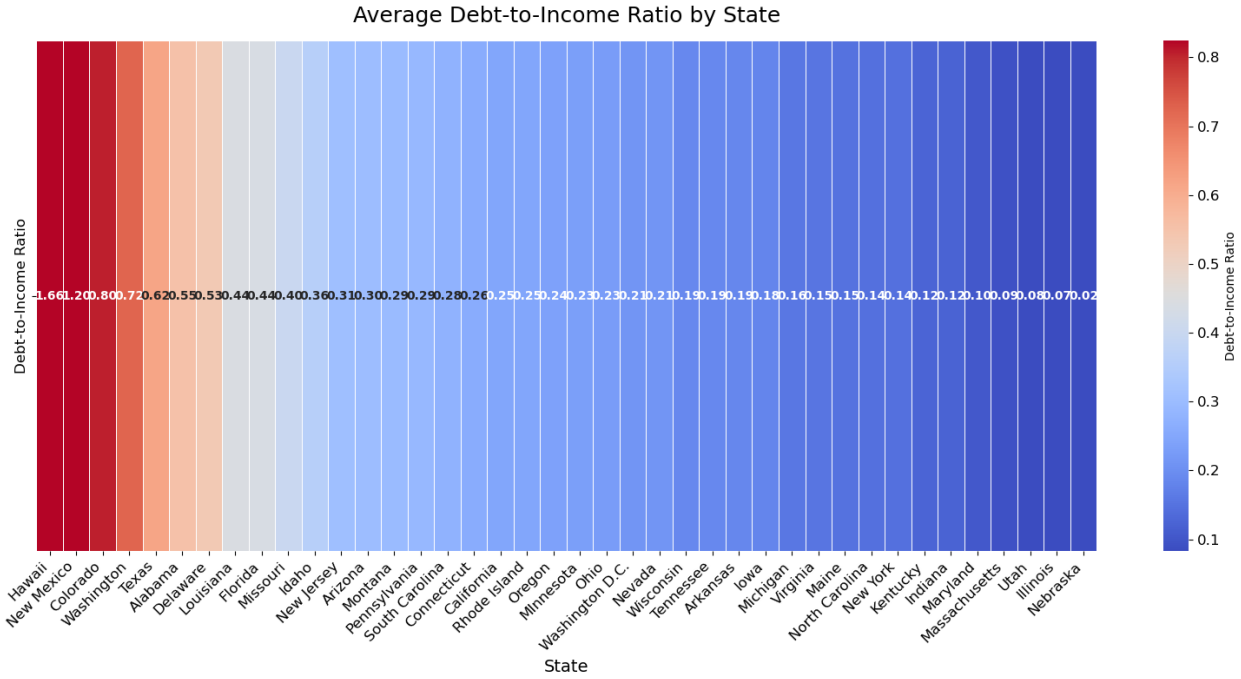


Visualization 3:

Staphon Smith



Visualization 4:



C. Explanation of Visualizations

Histogram of Debt-to-Income Ratios: A histogram was chosen for this visualization because it is ideal for displaying the frequency distribution of numerical data. The x-axis represents the debt-to-income ratio, and the y-axis shows the number of companies falling into each range.

```
plt.figure(figsize=(12, 6))

# Compute average revenue by state
avg_revenue = df.groupby('Business State')['Total
Revenue'].mean().sort_values().reset_index()

# Create barplot with hue assigned correctly
sns.barplot(data=avg_revenue, x='Business State', y='Total Revenue',
hue='Business State', palette='viridis', legend=False)

plt.title('Average Total Revenue by State', fontsize=16)
plt.xlabel('State', fontsize=14)
plt.ylabel('Average Revenue ($)', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```

Bar Chart of Debt-to-Equity by State: A bar chart was used to compare the average debt-to-equity ratio by state. This chart helps the stakeholders quickly identify regions with higher leverage, which could warrant closer scrutiny.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Adjust figure size and create boxplot
plt.figure(figsize=(12, 6))  # Set the figure size
sns.boxplot(data=df, x='Business State', y='Debt to Equity')

# Add title and label formatting
plt.title('Debt-to-Equity Ratios by State', fontsize=16)
plt.xticks(rotation=45, ha='right', fontsize=12)  # Rotate x-axis labels
and adjust font size
```

```
plt.yticks(fontsize=12)  # Increase font size of y-axis labels

# Add gridlines for better visual clarity
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show plot
plt.tight_layout()  # Ensure all labels and title fit
plt.show()
```

Scatter Plot of Revenue vs. Long-Term Debt: The scatter plot helps visualize the relationship between revenue and long-term debt, allowing stakeholders to spot companies that may have higher debt burdens relative to their revenue.

```
# Scatter plot of Total Revenue vs Total Long-term Debt with improvements
plt.figure(figsize=(12, 8))  # Set figure size
sns.scatterplot(data=df, x='Total Revenue', y='Total Long-term Debt',
alpha=0.7, s=100, color='blue')  # Adjust point size and color

# Add title and axis labels with larger fonts
plt.title('Total Revenue vs Total Long-term Debt', fontsize=18)
plt.xlabel('Total Revenue', fontsize=14)
plt.ylabel('Total Long-term Debt', fontsize=14)

# Adjust tick labels if needed
plt.xticks(rotation=45, ha='right', fontsize=12)  # Rotate x-axis labels
for readability
plt.yticks(fontsize=12)  # Increase font size of y-axis labels

# Add gridlines for better visual clarity
plt.grid(True)

# Ensure labels fit within the plot area
plt.tight_layout()

# Show the plot
plt.show()
```

Heatmap of Debt-to-Income Ratios by State: A heatmap is created to show the debt-to-income ratios across different states, providing an easy-to-understand color-coded representation of

financial health in each region. A heatmap is used to visually represent the debt-to-income ratio across states, providing a color-coded display that makes it easy to identify regions with higher or lower ratios. This is helpful for visualizing patterns and anomalies across states.

```
plt.figure(figsize=(16, 8))  # Increase figure size for better spacing

# Create heatmap with improved readability of annotations
ax = sns.heatmap(debt_income_avg.set_index('Business State').T,
                 cmap="coolwarm", annot=True, fmt=".2f",
                 linewidths=0.5, cbar_kws={'label': 'Debt-to-Income
Ratio'},
                 annot_kws={"size": 10, "weight": "bold"},  # Smaller font
size for annotations
                 vmax=np.percentile(debt_income_avg["Debt-to-Income
Ratio"], 95),
                 vmin=np.percentile(debt_income_avg["Debt-to-Income
Ratio"], 5),
                 cbar=True)

# Improve title and labels
plt.title('Average Debt-to-Income Ratio by State', fontsize=18, pad=15)
plt.xlabel('State', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)  # Rotate x-axis labels
and set horizontal alignment
plt.yticks(fontsize=12)

# Adjust colorbar font size
cbar = ax.collections[0].colorbar
cbar.ax.tick_params(labelsize=12)

# Use tight_layout() to ensure all elements fit and avoid overlapping
plt.tight_layout()

plt.show()
```

D. Sources

This report and the code used for data analysis are based on the provided "D598 Data Set" and supporting materials. No external sources were quoted or paraphrased.