Analytics Programming

Task 3

Nathan Hefner

A. Explain how the code works for the program you submitted in Task 2.

To create a code that completes all the requirements, I first had to import Panda, a Python library that allows the code to read and analyze data spreadsheets. I identified the file path to enable the program to open it and load it into a data frame.

I then check for duplicate values across all columns of the analyzed data, letting me filter out all duplicate rows into a new data frame.

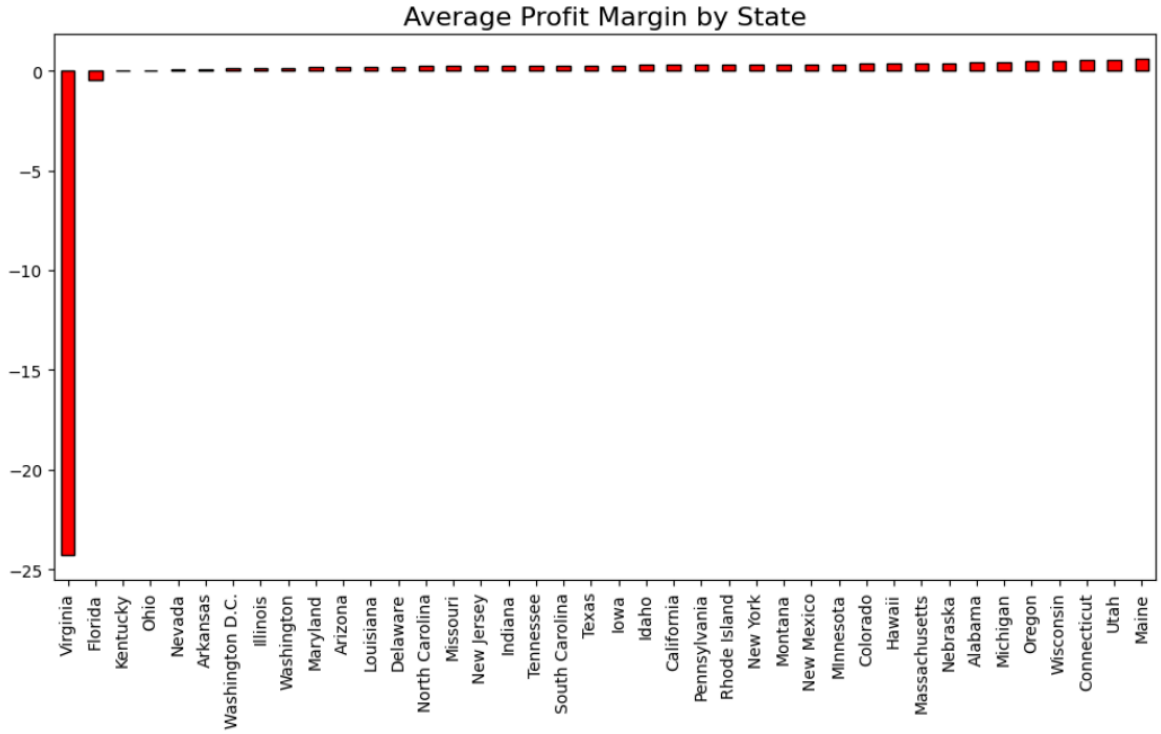
The next step is to compute each numeric column's descriptive statistics: minimum, maximum, median, and mean. The code achieves this by grouping the columns and calculating these values for each.

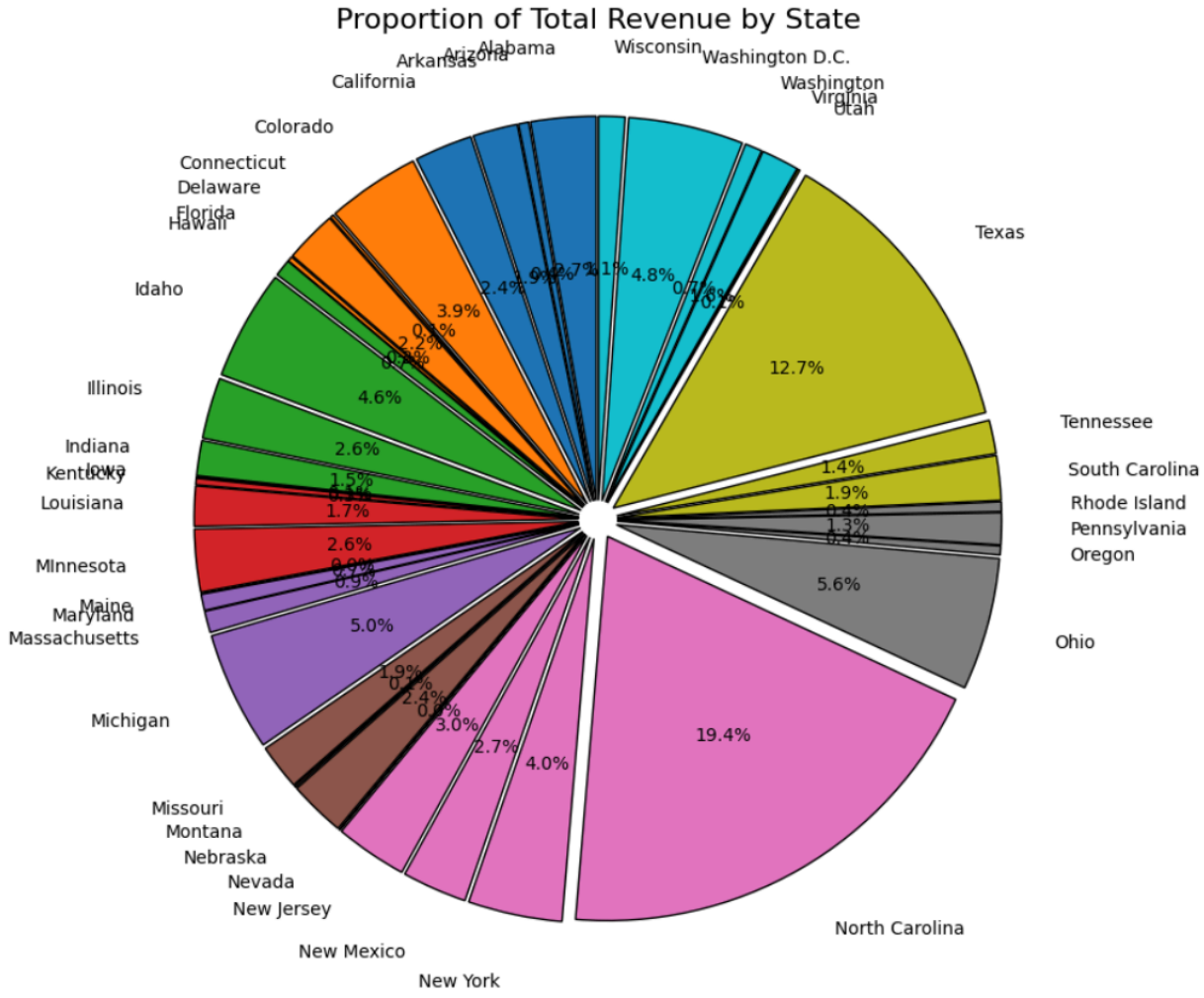
After this step, I filter the rows with negative Debt-to-Equity ratios by setting the value to less than zero and creating a separate data frame. The code then calculates the debt-to-income ratio by dividing the 'Total Long-Term Debt' by the 'Total Revenue" columns and creating a new data frame for these new values.

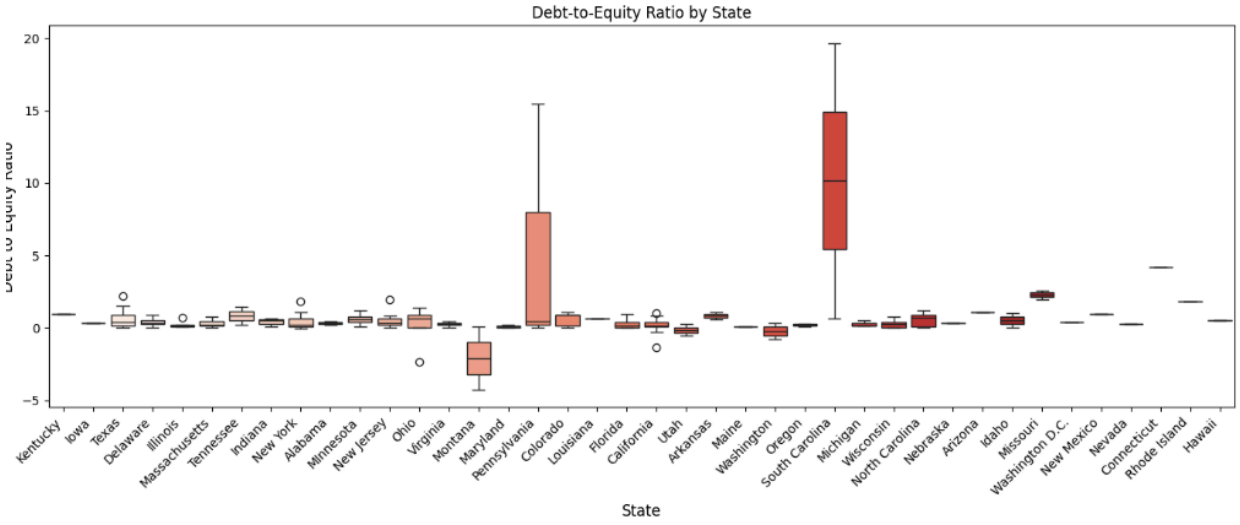
The code then concatenates the original DataFrame with the new Debt-to-Income DataFrame, combining it into one table for a more straightforward display.

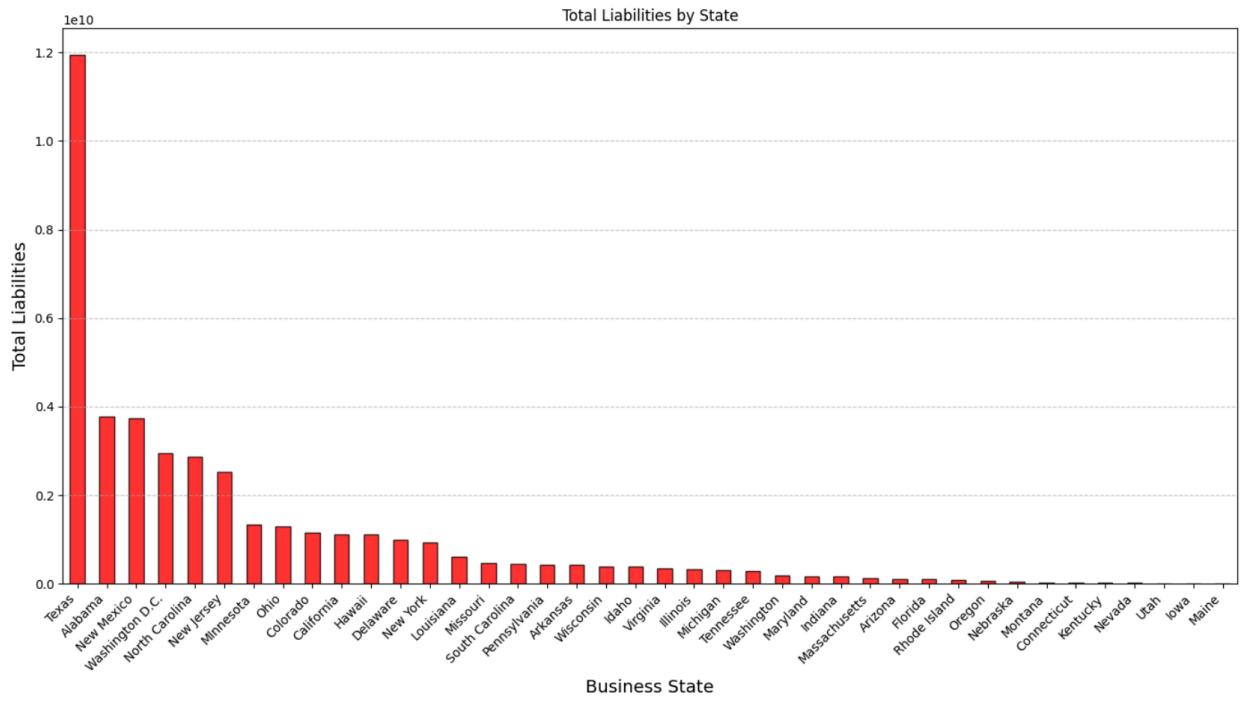
B. Provide 4 customized data visualizations.

Below are my four graphs that will help the business maximize profit by showing which states have the best profit margins, total revenue, debt to equity, and total liabilities of all states.




Proportion of Total Revenue by State
Alabama
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Hawaii
Idaho
Illinois
Indiana
Iowa
Kentucky
Louisiana
MInnesota
Maine
Maryland
Massachusetts
Michigan
Missouri
Montana
Nebraska
Nevada
New Jersey
New Mexico
New York
North Carolina
Ohio
Oregon
Pennsylvania
Rhode Island
South Carolina
Tennessee
Texas
Utah
Virginia
Washington
Washington D.C.
Wisconsin
4.8%
12.7%
1.4%
1.9%
1.3%
5.6%
19.4%
4.0%
2.7%
3.0%
2.4%
1.9%
5.0%
0.9%
2.6%
1.7%
1.5%
2.6%
4.6%
2.2%
3.9%
2.4%


Debt-to-Equity Ratio by State
Debt to Equity Ratio
State
Kentucky
Iowa
Texas
Delaware
Illinois
Massachusetts
Tennessee
Indiana
New York
Alabama
Minnesota
New Jersey
Ohio
Virginia
Montana
Maryland
Pennsylvania
Colorado
Louisiana
Florida
California
Utah
Arkansas
Maine
Washington
Oregon
South Carolina
Michigan
Wisconsin
North Carolina
Nebraska
Arizona
Idaho
Missouri
Washington D.C.
New Mexico
Nevada
Connecticut
Rhode Island
Hawaii



C. Explain how customized visualizations in part B were created.

I coded the data visuals in Jupyter Notebook by importing Matplotlib. This Python library allows the creation of different data graphs to help visualize the data imported (Hex use case: AD-hoc exploratory analysis).

For the first graph, I grouped the profit margin value by the mean of each business state, then sorted the values by least to most, revealing that Virginia has the lowest profit margin. With the values and data grouped and ready for display, I chose a bar graph to help visualize the contrast between each state.

For my second graph, I calculated the sum of each business state's total revenue, grouped these values according to their proper correspondent, and then displayed them as a pie chart. The states are ordered alphabetically, and the percentages of the state's total revenue against the country's total revenue are placed on their slices. This visualization will help the readers interpret which states contribute the most and least to total revenue.

For the boxplot graph, I configured the x-axis as the business state and the y-axis as debt to equity. Python can read the data from the Excel file and calculate the values for the boxplot using Seaborn. Seaborn calculates all states' minimum, maximum, median, and first and third quartiles. Which then displays the information on the graph. Also including outliers when needed. This graph type can help with data cleaning to see if there are unusual outliers and show which states have the most expansive and slimmest margins.

For my final graph, I made a bar graph in the code, grouped the values by state, summed up the total liabilities for each, and displayed the data in descending order. Allowing the data to be displayed makes it easier to see which states have the most liabilities.

Citations

"Hex Use Case: AD-Hoc Exploratory Analysis." *Hex*, hex.tech/use-case/exploratory-analysis/?utm_source=google&utm_medium=paid_search&utm_campaign=python&utm_content=matplotlib&utm_term=matplotlib+pyplot&utm_id=h_7015f000000VTMxAAO&utm_adgroup_id=170419746226&gad_source=1&gclid=Cj0KCQiAs5i8BhDmARIsAGE4xHygOFBew7q74zpGlRBYQpsokMB0iVsaDLMgY_WMB4DBC6jX6xWhNgYaAq6ZEALw_wcB. Accessed 14 Jan. 2025.