

## D598 Task 2

## A. Python Program

This project required writing a program in Python to analyze the dataset of 150 U.S. companies managed by the investment firm. This was done in VS code/Jupyter notebook to write out the code.

Below is the Python script that fulfills the task requirements:

```
import pandas as pd

# Step 1: Load the data
file_path = r"C:\Users\StaphonSmith\Desktop\D598 Data Set (1).xlsx" # Replace this with the correct path to your dataset
df = pd.read_excel(file_path, sheet_name="1-150 V2") # Load the sheet named "1-150 V2"

# Step 2: Check for duplicate rows
duplicates = df[df.duplicated()]
print("Duplicate Rows:\n", duplicates)
```

[2] ✓ 0.0s Python

... Duplicate Rows:  
Empty DataFrame  
Columns: [Business ID, Business State, Total Long-term Debt, Total Equity, Debt to Equity, Total Liabilities, Total Revenue, Profit Margin]  
Index: []

```
# Step 3: Group by 'Business State' and calculate descriptive statistics
grouped_stats = df.groupby('Business State').agg(
    {
        'Total Long-term Debt': ['mean', 'median', 'min', 'max'],
        'Total Equity': ['mean', 'median', 'min', 'max'],
        'Debt to Equity': ['mean', 'median', 'min', 'max'],
        'Total Liabilities': ['mean', 'median', 'min', 'max'],
        'Total Revenue': ['mean', 'median', 'min', 'max'],
        'Profit Margin': ['mean', 'median', 'min', 'max'],
    }
)

grouped_stats.columns = ['_'.join(col).strip() for col in grouped_stats.columns.values]
grouped_stats.reset_index(inplace=True)
print("Grouped Descriptive Statistics by State:\n", grouped_stats)
```

[3] ✓ 0.0s Python

```
[3] ✓ 0.0s Python
Grouped Descriptive Statistics by State:
  Business State  Total Long-term Debt_mean  Total Long-term Debt_median \
0      Alabama      6.743890e+08      6.743890e+08
1      Arizona      6.508800e+07      6.508800e+07
2      Arkansas      8.960450e+07      8.960450e+07
3      California    3.273964e+07      4.721500e+06
4      Colorado      8.470025e+07      2.932700e+07
5      Connecticut    1.851200e+07      1.851200e+07
6      Delaware      6.156325e+07      6.016500e+07
7      Florida      1.230350e+07      1.809000e+06
8      Hawaii      6.055000e+08      6.055000e+08
9      Idaho      2.463500e+06      2.463500e+06
10     Illinois      1.132533e+07      9.029000e+06
11     Indiana      2.168600e+07      5.112000e+06
12     Iowa      6.252000e+06      6.252000e+06
13     Kentucky      1.688900e+07      1.688900e+07
14     Louisiana      3.790000e+08      3.790000e+08
15     Minnesota      7.493429e+07      1.016900e+07
16     Maine      8.960000e+05      8.960000e+05
17     Maryland      1.448550e+07      9.585000e+05
18     Massachusetts    7.717600e+06      4.911000e+06
19     Michigan      3.611400e+07      2.160900e+07
20     Missouri      1.730845e+08      1.730845e+08
21     Montana      8.730500e+06      8.730500e+06
22     Nebraska      1.891900e+07      1.891900e+07
...
38      0.108641      0.108641
39      0.122358      0.837386

[40 rows x 25 columns]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

```
# Step 4: Filter businesses with negative debt-to-equity ratios
negative_de_ratio = df[df['Debt to Equity'] < 0]
print("Businesses with Negative Debt-to-Equity Ratios:\n", negative_de_ratio)
```

[4] ✓ 0.0s

Python

```
... Businesses with Negative Debt-to-Equity Ratios:
```

	Business ID	Business State	Total Long-term Debt	Total Equity \
18	934562013	Ohio	263880000.0	-111297000.0
57	8343652013	Washington	10603000.0	-13271658.0
87	9323722013	California	21560000.0	-15691000.0
109	10919832013	Utah	2010000.0	-3602481.0
117	11245242013	California	556000.0	-2063203.0
142	14535932013	Montana	16459000.0	-3842372.0
143	14639722013	New York	187000.0	-13037879.0

	Debt to Equity	Total Liabilities	Total Revenue	Profit Margin
18	-2.370953	592174000.0	719783000	0.320697
57	-0.798921	16625000.0	8949401	0.448119
87	-1.374036	30048000.0	37782000	0.505955
109	-0.557949	6302000.0	17757388	0.732562
117	-0.269484	3819000.0	1100539	-0.084923
142	-4.283552	32720000.0	33073414	0.582444
143	-0.014343	15900000.0	2389053	0.300978

```
# Step 5: Calculate debt-to-income ratio (long-term debt divided by revenue)
df['Debt to Income Ratio'] = df['Total Long-term Debt'] / df['Total Revenue']
```

[5] ✓ 0.0s

Python

```
# Step 6: Concatenate the new column to the original DataFrame
print("DataFrame with Debt-to-Income Ratio:\n", df.head())
```

[6] ✓ 0.0s

Python

```
... DataFrame with Debt-to-Income Ratio:
```

	Business ID	Business State	Total Long-term Debt	Total Equity \
0	41872013	Kentucky	16889000.0	18046000.0
1	76232013	Iowa	6252000.0	18293621.0
2	160992013	Texas	19200000.0	177858000.0
3	197452013	Delaware	117592000.0	278773000.0
4	241042013	Illinois	4408000.0	52064000.0

	Debt to Equity	Total Liabilities	Total Revenue	Profit Margin \
0	0.935886	25986000.0	136753000	0.023663
1	0.341758	14474000.0	34226553	0.265015
2	0.107951	72787000.0	384196000	0.130413
3	0.421820	558749000.0	444306000	0.196768
4	0.084665	19898000.0	121541000	0.168305

	Debt to Income Ratio
0	0.123500
1	0.182665
2	0.049974
3	0.264664
4	0.036268

B. No sources/citations were used in creating the code except for general course materials.