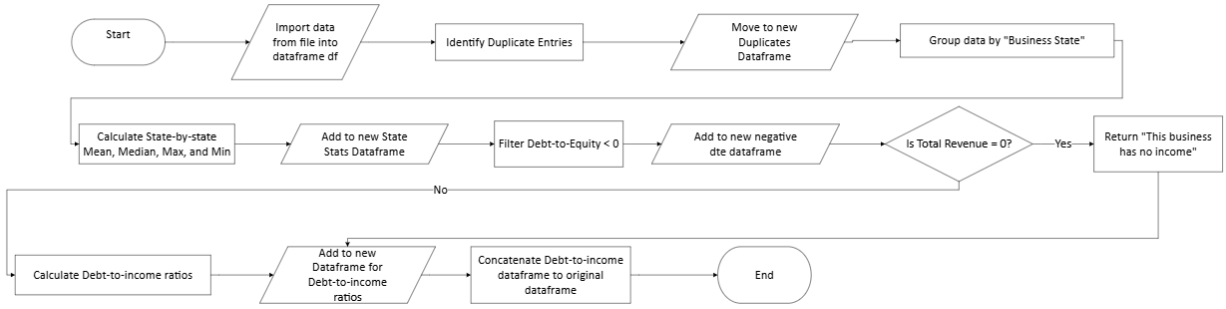
D598 - Task 1 - Aaron Pacheco

A.



B.

BEGIN

```
# Step 1: Import the data file into a data frame
IMPORT pandas library
READ the data file into a DataFrame called df

# Step 2: Identify any duplicate rows in the data set
FIND duplicate rows in df
STORE duplicates in a separate DataFrame called duplicate_data

# Step 3: Group all IDs by state, then run descriptive statistics
GROUP df by 'Business State'
CALCULATE mean, median, min, max for all numeric columns in each state group
STORE results in a new DataFrame called state_stats

# Step 4: Filter the data frame for businesses with negative debt-to-equity ratios
SELECT rows from df where debt_to_equity < 0
STORE results in a new DataFrame called negative_debt_equity

# Step 5: Create a new data frame for debt-to-income ratio
IF Total Revenue = 0
RETURN “This business has no income.”
ELSE
CALCULATE debt_to_income = Total Long Term Debt / Total Revenue
STORE results in a new DataFrame called dti_df with 'ID' and 'debt_to_income'

# Step 6: Concatenate the debt-to-income ratio DataFrame with the original data
CONCATENATE df with dti_df along columns
STORE as final_df
```

END

C. The first step requires us to “Import the date file into a data frame.” This step is covered in the flow chart in Cell 2, and in the Pseudocode it is covered under #Step One. Importing the pandas library is required to import the file and utilize dataframes and many of the data analysis tools I use.

The second step is “Identify any duplicate rows in the data set.” This step is covered in the flowchart in Cells 3-4 and in the pseudocode under #Step 2. The duplicate rows must first be identified by the program, then all rows that are found to be duplicates need to be removed from the original dataset in order to ensure our future calculations are accurate and not being skewed by messy data.

The third step is “Group all IDs by state, then run descriptive statistics (mean, median, min, & max) for all numeric variables by state and store this result as a new data frame.” This step is covered by cells 5-7 in the flowchart and in the pseudocode under #Step 3. We have to first take all of the data and group it based on the “Business State” column. Once we have all of the data group, we will calculate the mean, median, minimum, and maximum of each separate state. Then, we will store the results of all of these calculations in a new dataframe titled “state_stats.”

The fourth step is “Filter the data frame to identify all businesses with debt-to-equity ratios that are negative.” This step is covered in the flowchart in cells 8-9 and in the pseudocode under #Step 4. We will first have to use a command to find all rows that have a value less than 0 in the Debt to Equity column. Then, each row that meets this criteria will need to be added to a new dataframe titled “negative_debt_equity.”

The fifth step is “ Create a new data frame that provides the debt-to-income ratio for every business in the data set. Debt-to-income ratio is defined as long-term debt divided by revenue.” This is covered in the flow chart starting with the cell “Is Total Revenue = 0” and ending with “Add to new dataframe for debt-to-income ratios” and in the pseudocode under #Step 5. First, we will have to check if a business’s total revenue equals zero, in order to avoid any divide-by-zero errors. Then, if total revenue for a business equals zero, we will return “This business has no income.” Otherwise, if total revenue does not equal zero, we will have to calculate the Debt-to-income ratios by defining it as “Total Long Term Debt / Total Revenue.” Once we calculate each companies’ ratios, we will input the business IDs and their ratios into a new dataframe titled “dti_df.” The “This business has no revenue” return will be stored under the business’s debt-to-income ratio in the new dataframe.

The final step is “Concatenate the debt-to-income ratio data frame you created with the original data frame.” This is covered in the flowchart in the last cell before “END”, and in the pseudocode under #Step 6. We will first concatenate the two dataframes into one, then we will store that new dataframe as “final_df.”

D. There are no sources to cite for this task.