Final Assignment: Part 2 - Create Dashboard with Plotly and Dash

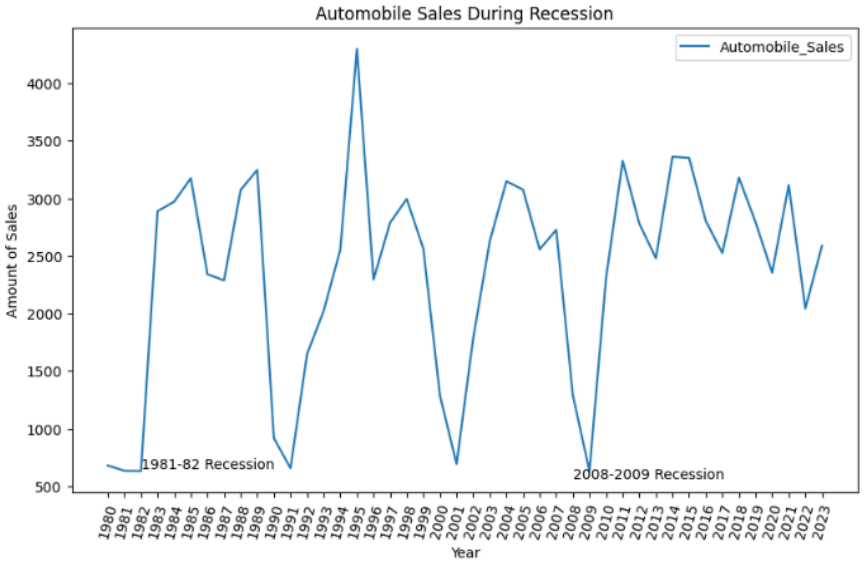
This file only contains the images from Dash uploaded for the final exam.

Submitted by Kailani Bailey on June 1, 2024

Task 1.1

Develop a Line Plot using the functionality of Pandas to show how automobile sales fluctuate from year to year.

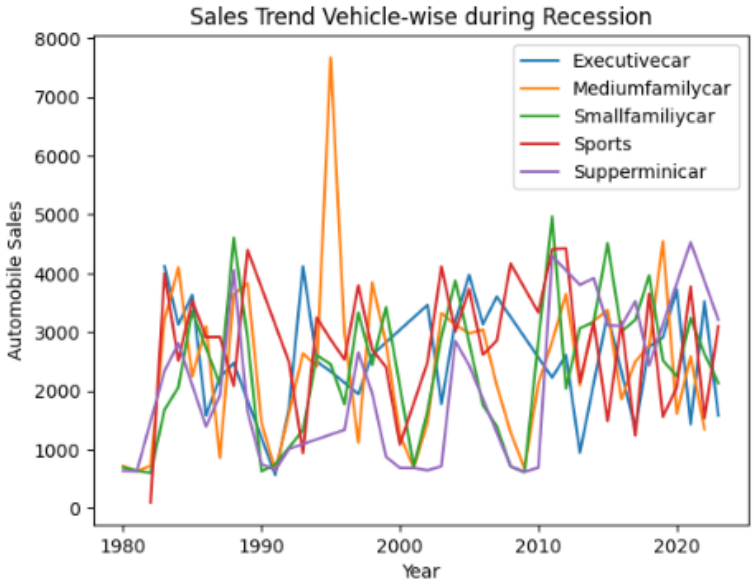
Submit the image *Line_Plot_1.png*



Task 1.2

Plot different lines for categories of vehicle type and analyze the trend to answer the question, "Is there noticeable difference in sales trends between different vehicle types during recession periods?"

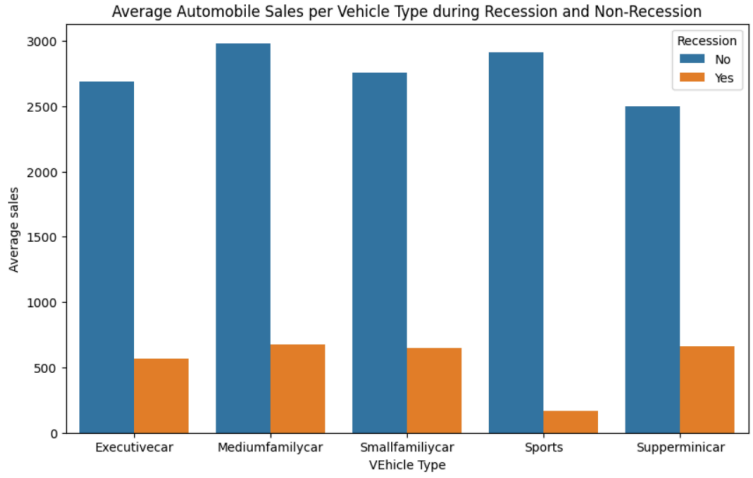
Submit the image *Line_Plot_2.png*



Task 1.3

Use the functionality of Seaborn Library to create a visualization to compare the sales trend per vehicle type for a recession period with a non-recession period.

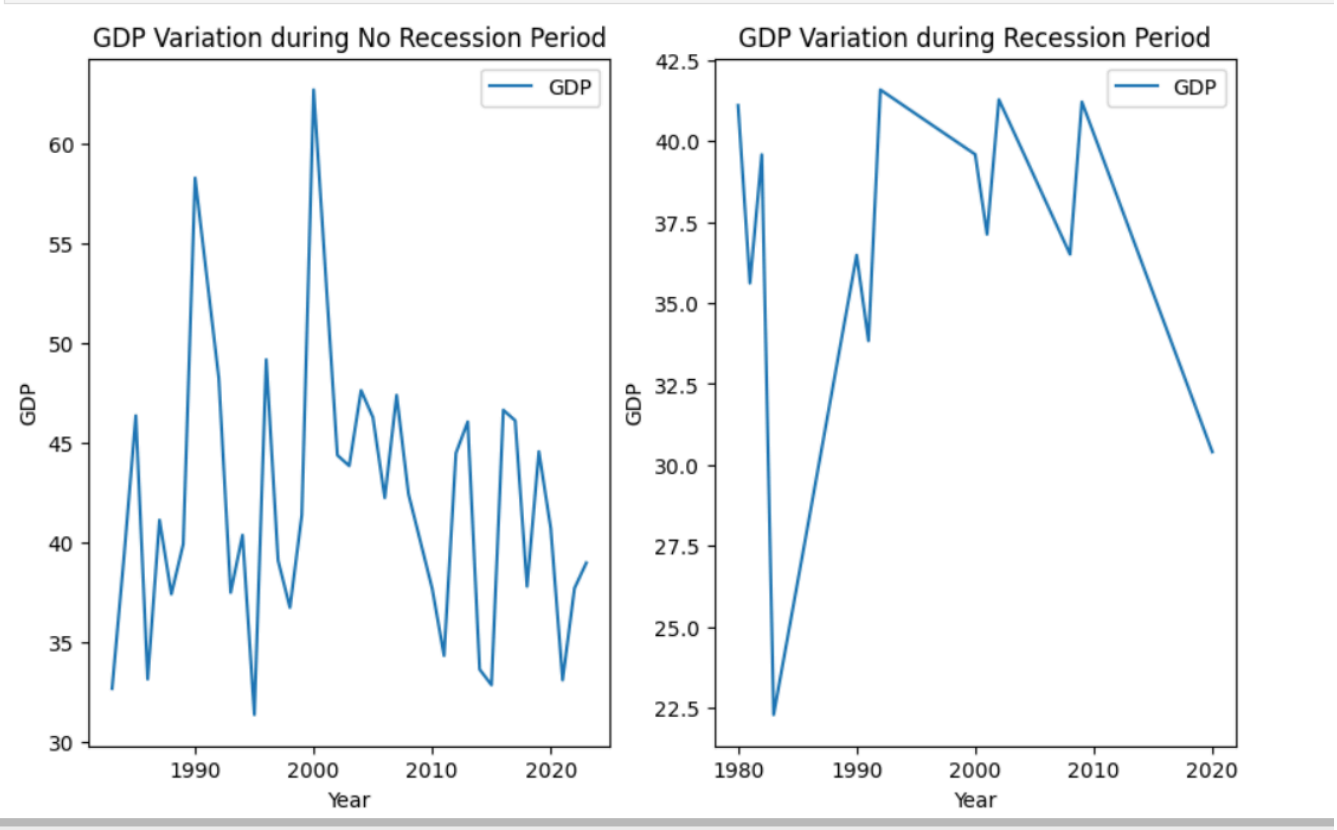
Submit the image *Bar_Chart.png*



Task 1.4

Use subplotting to compare the variations of GDP during recession and non-recession period by developing line plots for each period.

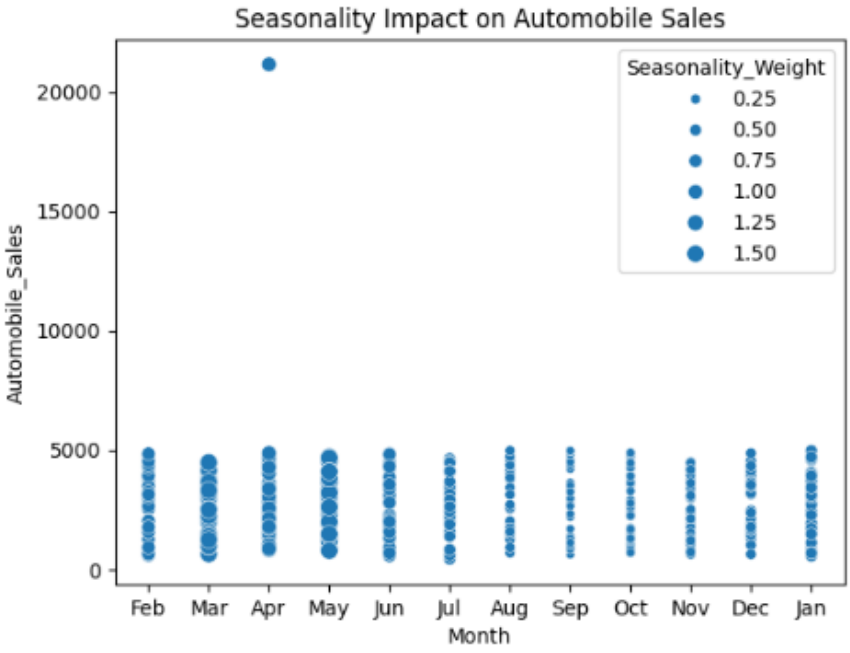
Submit the image *Subplot.png*



Task 1.5

Develop a Bubble Plot for displaying the impact of seasonality on Automobile Sales.

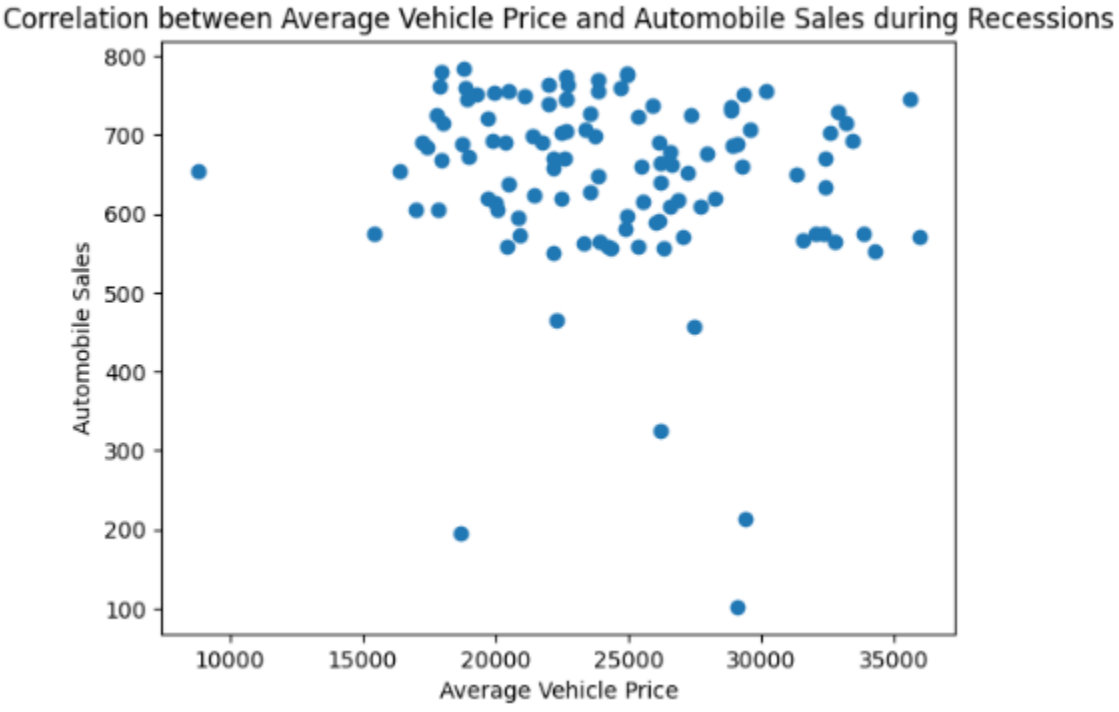
Submit the image *Bubble.png*



Task 1.6

Use the functionality of Matplotlib to develop a Scatter Plot to identify the correlation between average vehicle price related to the sales volume during recessions.

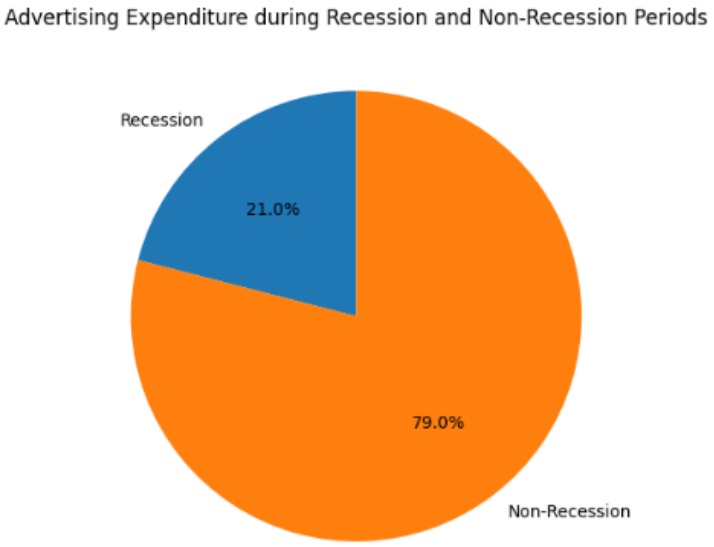
Submit the image *Scatter.png*



Task 1.7

Create a Pie Chart to display the portion of advertising expenditure of XYZAutomotives during recession and non-recession periods.

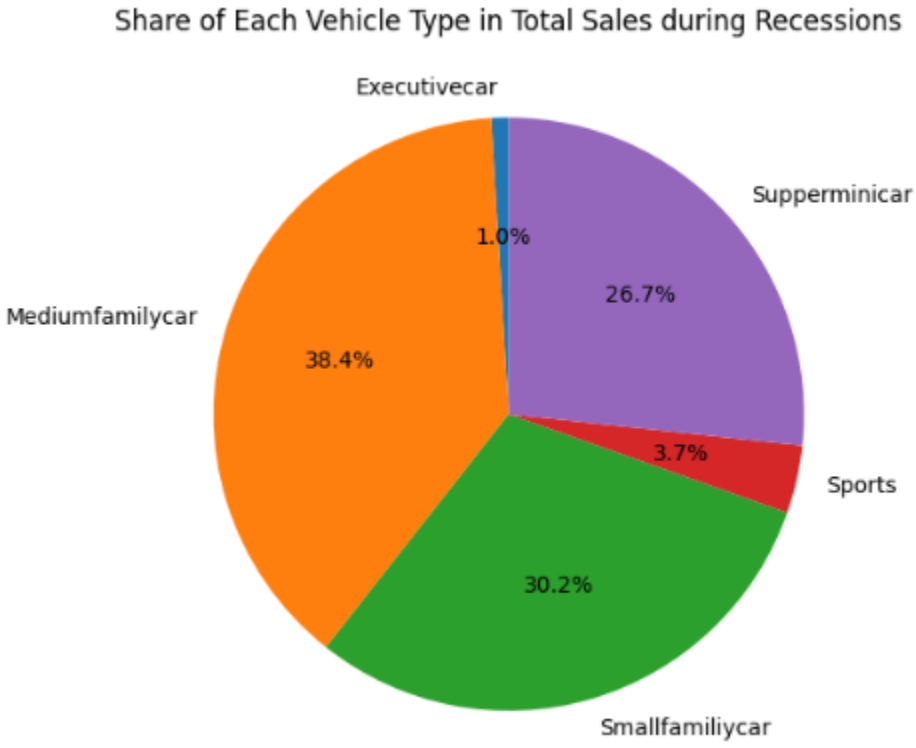
Submit the image *Pie_1.png*



Task 1.8

Develop a Pie Chart to display the total advertising expenditure for each vehicle type during recession period.

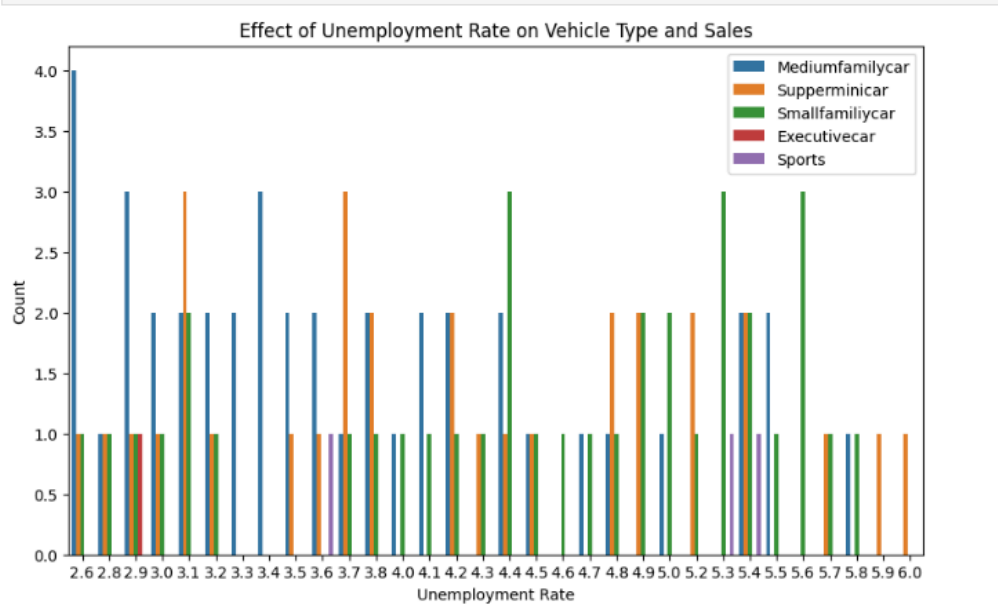
Submit the image *Pie_2.png*



Task 1.9

Develop a Line Plot to analyze the effect of the unemployment rate on vehicle type and sales during the recession period.

Submit image *Line_Plot_3.png*



Task 2.1

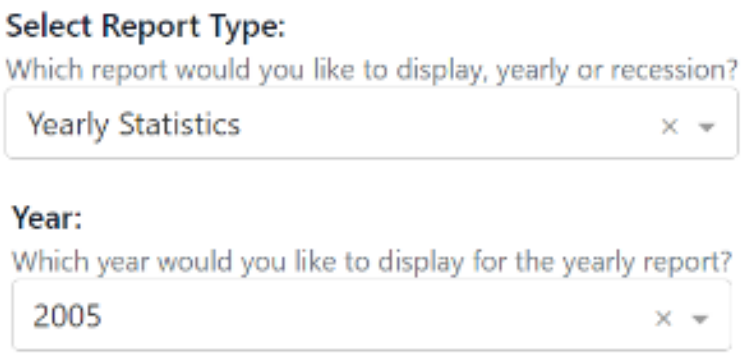
Create a Dash application and give it a meaningful title.

Submit the image *Title.png*

Task 2.2

Add drop-downs to your dashboard with appropriate titles and options.

Submit the image *Dropdown.png*



Task 2.3

Add a division for output display with appropriate ‘id’ and ‘classname’ properties.

Submit the image *Outputdiv.png*

```
        html.Div(
            [
                html.Label(
                    "Select Report Type:",
                    className="text-base font-semibold text-gray-900",
                    htmlFor="input-report",
                ),
                html.P(
                    "Which report would you like to display, yearly or recession?",
                    className="text-sm text-gray-500",
                ),
                dcc.Dropdown(
                    options=[
                        {"label": "Yearly Statistics", "value": "Yearly"},
                        {"label": "Recession Period Statistics", "value": "Recession"},
                    ],
                    value="Yearly",
                    id="input-report",
                ),
            ],
            className="mt-4",
        ),
        html.Div(
            [
                html.Label(
                    "Year:",
                    className="text-base font-semibold text-gray-900",
                    htmlFor="input-year",
                ),
                html.P(
                    "Which year would you like to display for the yearly report?",
                    className="text-sm text-gray-500",
                ),
                dcc.Dropdown(
                    sorted(df.Year.unique()), value=2005, id="input-year", disabled=True
                ),
            ],
            className="mt-4",
        ),
        html.Section(
            [
                dcc.Graph(id="plot-1"),
                dcc.Graph(id="plot-2"),
                dcc.Graph(id="plot-3"),
                dcc.Graph(id="plot-4"),
            ],
            className="flex flex-wrap items-center justify-center",
        ),
    ],
    className="flex flex-col items-center",
```

Task 2.4

Creating Callbacks: Define the callback function to update the input container based on the selected statistics and the output container.

Submit the *Callbacks.png*

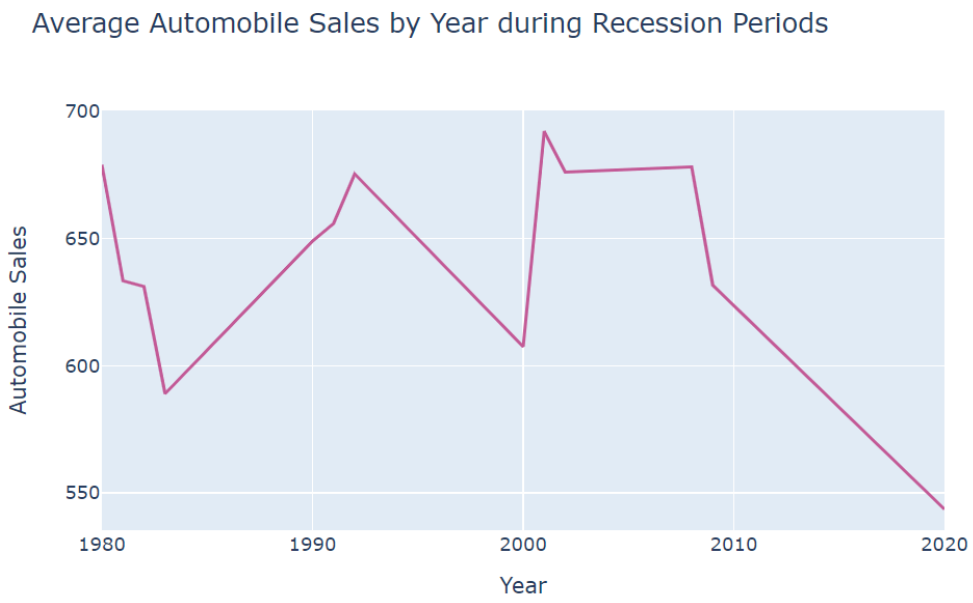
```
@callback(Output("input-year", "disabled"), Input("input-report", "value"))
def disable_year(report_value):
    if report_value == "Recession":
        return True
    else:
        return False


@callback(
    [
        Output(component_id="plot-1", component_property="figure"),
        Output(component_id="plot-2", component_property="figure"),
        Output(component_id="plot-3", component_property="figure"),
        Output(component_id="plot-4", component_property="figure"),
    ],
    [
        Input(component_id="input-report", component_property="value"),
        Input(component_id="input-year", component_property="value"),
    ],
)
```

Task 2.5

Create and display graphs for Recession Report Statistics.

Submit the image *RecessionReportGraphs.png*



Task 2.6

Create and display graphs for Yearly Report Statistics.

Submit the image *YearlyReportGraphs.png* for the report types.