US_Automobile_Sales_Recession_Choropleth

October 18, 2024

1 Create visualizations using Matplotib, Seaborn and Folium

Estimated time needed: **40** minutes

In this assignment, you will have the opportunity to demonstrate the skills you have acquired in creating visualizations using *Matplotlib, Seaborn, Folium*. After each task you will be required to save your plots as an image or screenshot using the filenames specified. You will be uploading these images during your final project submission so they can be evaluated by your peers.

2 Table of Contents

3 Objectives

After completing this lab you will be able to:

- Create informative and visually appealing plots with Matplotlib and Seaborn.
- Apply visualization to communicate insights from the data.
- Analyze data through using visualizations.
- Customize visualizations

4 Setup

For this lab, we will be using the following libraries:

- `pandas` for managing the data.
- `numpy` for mathematical operations.

- `matplotlib` for plotting.
- `seaborn` for plotting.
- `Folium` for plotting.

4.0.1 Installing Required Libraries

The following required libraries are pre-installed in the Skills Network Labs environment. However, if you run these notebook commands in a different Jupyter environment (e.g. Watson Studio or Ananconda), you will need to install these libraries by removing the `#` sign before `%pip` in the code cell below.

[ ]:

```
# All Libraries required for this lab are listed below. The libraries pre-installed on Skills Network Labs are commented.
# %pip install -qy pandas==1.3.4 numpy==1.21.4 matplotlib==3.5.0 seaborn folium
# Note: If your environment doesn't support "%pip install", use "!mamba install"
```

[3]:

```
%pip install seaborn
%pip install folium
```

4.0.2 Importing Required Libraries

We recommend you import all required libraries in one place (here):

[8]:

```
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import folium
```

Click here for a hint

You will require:- Numpy for many scientific computing in Python Pandas for creating and working on dataframe, also for plotting directly on dataframe/series The inline backend to generate the plots within the browser [%matplotlib inline] Matplotlib and its pyplot pacakge for plotting Seaborn for plotting

Click here for python solution

```
#Import Primary Modules:
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import folium
```

---

5 Scenario

In this assignment you will be tasked with creating plots which answer questions for analysing "historical_automobile_sales" data to understand the historical trends in automobile sales during recession periods. recession period 1 - year 1980 recession period 2 - year 1981 to 1982 recession period 3 - year 1991 recession period 4 - year 2000 to 2001 recession period 5 - year end 2007 to mid 2009 recession period 6 - year 2020 -Feb to April (Covid-19 Impact)

6 Data Description

The dataset used for this visualization assignment contains *historical_automobile_sales* data representing automobile sales and related variables during recession and non-recession period.

The dataset includes the following variables: 1. Date: The date of the observation. 2. Recession: A binary variable indicating recession perion; 1 means it was recession, 0 means it was normal. 3. Automobile_Sales: The number of vehicles sold during the period. 4. GDP: The per capita GDP value in USD. 5. Unemployment_Rate: The monthly unemployment rate. 6. Consumer_Confidence: A synthetic index representing consumer confidence, which can impact consumer spending and automobile purchases. 7. Seasonality_Weight: The weight representing the seasonality effect on automobile sales during the period. 8. Price: The average vehicle price during the period. 9. Advertising_Expenditure: The advertising expenditure of the company. 10.Vehicle_Type: The type of vehicles sold; Supperminicar, Smallfamiliycar, Mediumfamilycar, Executivecar, Sports. 11.Competition: The measure of competition in the market, such as the number of competitors or market share of major manufacturers. 12.Month: Month of the observation extracted from Date.. 13.Year: Year of the observation extracted from Date. By examining various factors mentioned above from the dataset, you aim to gain insights into how recessions impacted automobile sales for your company.

---

7 Importing Data

For your convenience, we have already written code to import the data below.

[4]:
```
from js import fetch
import io

URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
  ↪IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/
  ↪historical_automobile_sales.csv"
resp = await fetch(URL)
text = io.BytesIO((await resp.arrayBuffer()).to_py())
import pandas as pd
df = pd.read_csv(text)
print('Data downloaded and read into a dataframe!')
```

```
Data downloaded and read into a dataframe!
```

[5]:
```
df.describe()
```

[5]:
```
              Year   Recession  Consumer_Confidence  Seasonality_Weight  \
count   528.000000  528.000000           528.000000          528.000000
mean   2001.500000    0.214015           101.140170            0.575795
std      12.710467    0.410526            10.601154            0.454477
min    1980.000000    0.000000            73.900000            0.000000
25%    1990.750000    0.000000            94.035000            0.250000
50%    2001.500000    0.000000           100.740000            0.500000
75%    2012.250000    0.000000           108.240000            0.750000
max    2023.000000    1.000000           131.670000            1.500000

              Price  Advertising_Expenditure  Competition         GDP  \
count    528.000000               528.000000   528.000000  528.000000
mean   24964.991956              3067.456439     6.064394   40.073903
std     4888.073433              1139.564637     1.968350   16.249714
min     8793.663000              1009.000000     3.000000   12.508000
25%    21453.300500              2083.500000     4.000000   27.237500
50%    25038.691500              3072.000000     6.000000   39.214500
75%    28131.684750              4067.250000     8.000000   53.506500
max    44263.657000              4983.000000     9.000000   70.374000

       Growth_Rate  unemployment_rate  Automobile_Sales
count   528.000000         528.000000        528.000000
mean     -0.242001           2.453977       2352.718068
std       0.861268           1.119019       1645.321284
min      -4.227601           1.000000        102.000000
25%      -0.574049           1.600000        793.950000
50%      -0.013162           2.300000       2182.600000
75%       0.388932           2.900000       3614.800000
max       0.815074           6.000000      21147.000000
```

[6]:
```
df.columns
```

[6]:
```
Index(['Date', 'Year', 'Month', 'Recession', 'Consumer_Confidence',
       'Seasonality_Weight', 'Price', 'Advertising_Expenditure', 'Competition',
       'GDP', 'Growth_Rate', 'unemployment_rate', 'Automobile_Sales',
       'Vehicle_Type', 'City'],
      dtype='object')
```

---

8 Creating Visualizations for Data Analysis

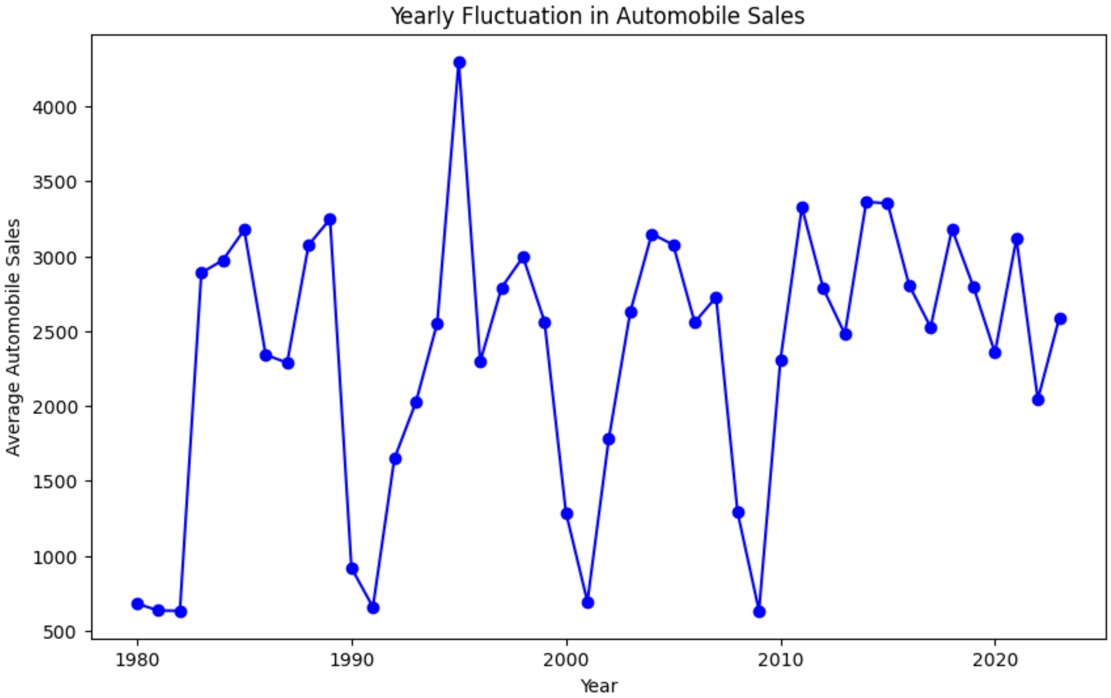
8.0.1 TASK 1.1: Develop a *Line chart* using the functionality of pandas to show how automobile sales fluctuate from year to year

Click here for a hint

You will require:- to group the year and calculate the average on the 'Automobile Sales', as the data has years and months column make use of .plot() with kind = 'line' donot forget to include labels and title

[10]:

```
df_line = df.groupby('Year')['Automobile_Sales'].mean()

# Create the figure and plot the line chart
plt.figure(figsize=(10, 6))
df_line.plot(kind='line', marker='o', color='b')

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Average Automobile Sales')
plt.title('Yearly Fluctuation in Automobile Sales')

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



Click here for a solution template

```
#create data for plotting
df_line = df.groupby(df['Year'])['Automobile_Sales'].mean()
#create figure
plt.figure(figsize=(10, 6))
df_line.plot(kind = 'line')
plt.xlabel('........')
plt.ylabel('.........')
```

```
    plt.title('......................')
    plt.show()
```

8.0.2 Include the following on the plot

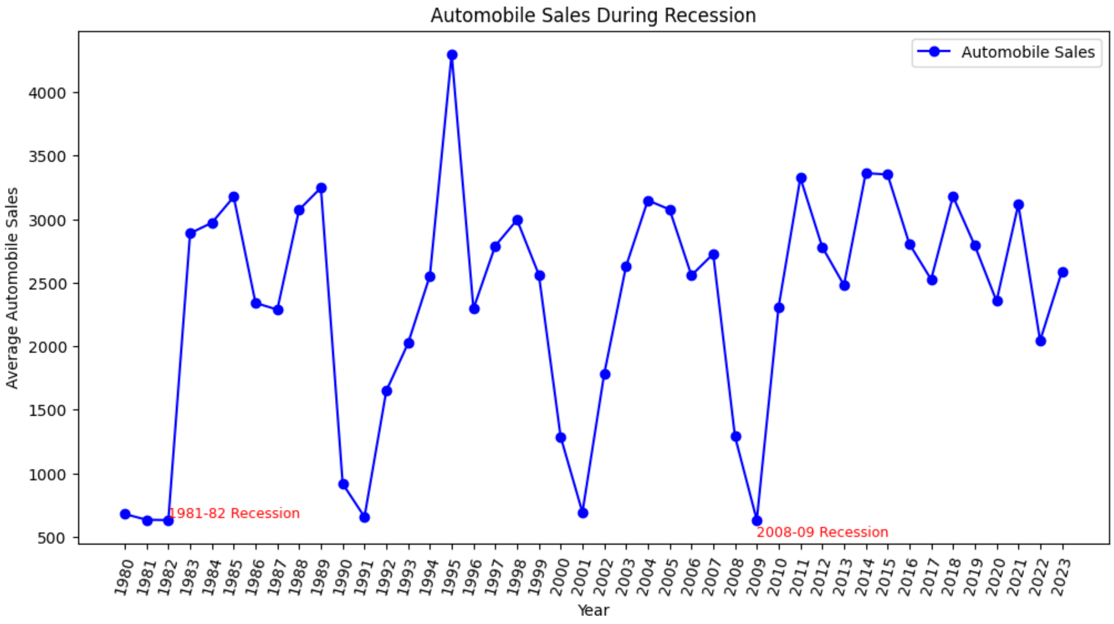
ticks on x- axis with all the years, to identify the years of recession annotation for at least two years of recession Title as Automobile Sales during Recession

Click here for a hint

```
<p>
You can create the list for the range 1980 till 2023 and pass that list to the plt.xticks func
You might need to rotate the ticks to an angle so that they fit in well on the axis
You can include annotation with plt.text(x, y, 'text to display')
</p>
```

[12]:

```
# Group data by 'Year' and calculate the mean of 'Automobile_Sales'
df_line = df.groupby('Year')['Automobile_Sales'].mean()

# Create the figure and plot
plt.figure(figsize=(12, 6))
df_line.plot(kind='line', marker='o', color='blue')

# Set x-axis ticks for every year
plt.xticks(ticks=list(range(1980, 2024)), rotation=75)

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Average Automobile Sales')
plt.title('Automobile Sales During Recession')

# Annotate two recession periods (modify the x and y coordinates as needed)
plt.text(1982, 650, '1981-82 Recession', fontsize=9, color='red')
plt.text(2009, 500, '2008-09 Recession', fontsize=9, color='red')

# Display the legend
plt.legend(['Automobile Sales'])

# Save the plot as an image (optional)
plt.savefig('automobile_sales_recession.png')

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



Click here for Solution template

```
plt.figure(figsize=(10, 6))
df_line = ...............
df_line.plot(kind = 'line')
plt.xticks(list(range(1980,2024)), rotation = 75)
plt.xlabel('..............')
plt.ylabel('...........')
plt.title('...................')
plt.text(1982, 650, '1981-82 Recession')
plt.text(......, ..., '..............')
plt.legend()
plt.show()
```

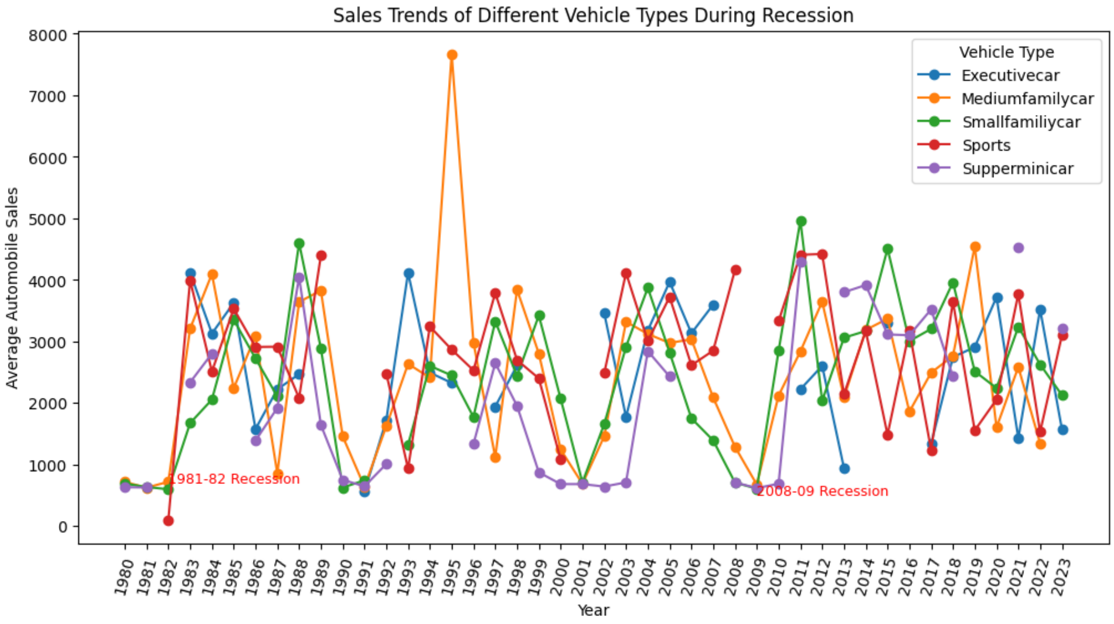
Save this plot as “Line_Plot_1.png” *Hint: You can right lick on the plot and then click on “Save image as” option to save it on your local machine*

TASK 1.2: Plot different lines for categories of vehicle type and analyse the trend to answer the question Is there a noticeable difference in sales trends between different vehicle types during recession periods? Click here for a hint

You will require:- create a separate dataframe where the column recession has a value of ‘1’ to group the year, vehicle_type and calculate the average on the ‘Automobile Sales’ one way is to - use as_index as false else you will endup with multiple-indexed datafame later set year as index and groupby vehicle over Sales and plot make use of .plot() with kind = ‘line’ do not forget to include labels and title

[13]:

```
# Group data by 'Year' and 'Vehicle_Type', and calculate the mean of␣
 ↪'Automobile_Sales'
df_vehicle = df.groupby(['Year', 'Vehicle_Type'])['Automobile_Sales'].mean().
 ↪unstack()

# Plot the trend for each vehicle type
plt.figure(figsize=(12, 6))
df_vehicle.plot(kind='line', marker='o', ax=plt.gca())

# Set x-axis ticks for every year
plt.xticks(ticks=list(range(1980, 2024)), rotation=75)

# Add labels and title
plt.xlabel('Year')
plt.ylabel('Average Automobile Sales')
plt.title('Sales Trends of Different Vehicle Types During Recession')

# Annotate two recession periods
plt.text(1982, 700, '1981-82 Recession', fontsize=9, color='red')
plt.text(2009, 500, '2008-09 Recession', fontsize=9, color='red')

# Display the legend
plt.legend(title='Vehicle Type')

# Save the plot as an image (optional)
plt.savefig('vehicle_sales_trends.png')

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



Click here for Solution template

```
df_rec = df[df['Recession']==1]
df_Mline = df_rec.groupby(['Year','Vehicle_Type'], as_index=False)['Automobile_Sales'].mean
df_Mline.set_index('Year', inplace=True)
df_Mline = df_Mline.groupby(['Vehicle_Type'])['Automobile_Sales']
df_Mline.plot(kind='line')
plt.xlabel('..............')
plt.ylabel('............')
plt.title('Sales Trend Vehicle-wise during Recession')
plt.legend()
plt.show()
```

From the above plot, what insights have you gained on the sales of various vehicle types? Type in your answer below: Inference

Inference: From this plot, we can understand that during recession period, the sales for 'Sports type vehicles' declined because of the high cost of the vehicle.while sales of the superminicar and smallfamilycar increased.

Save this plot as "Line_Plot_2.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

8.0.3 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.

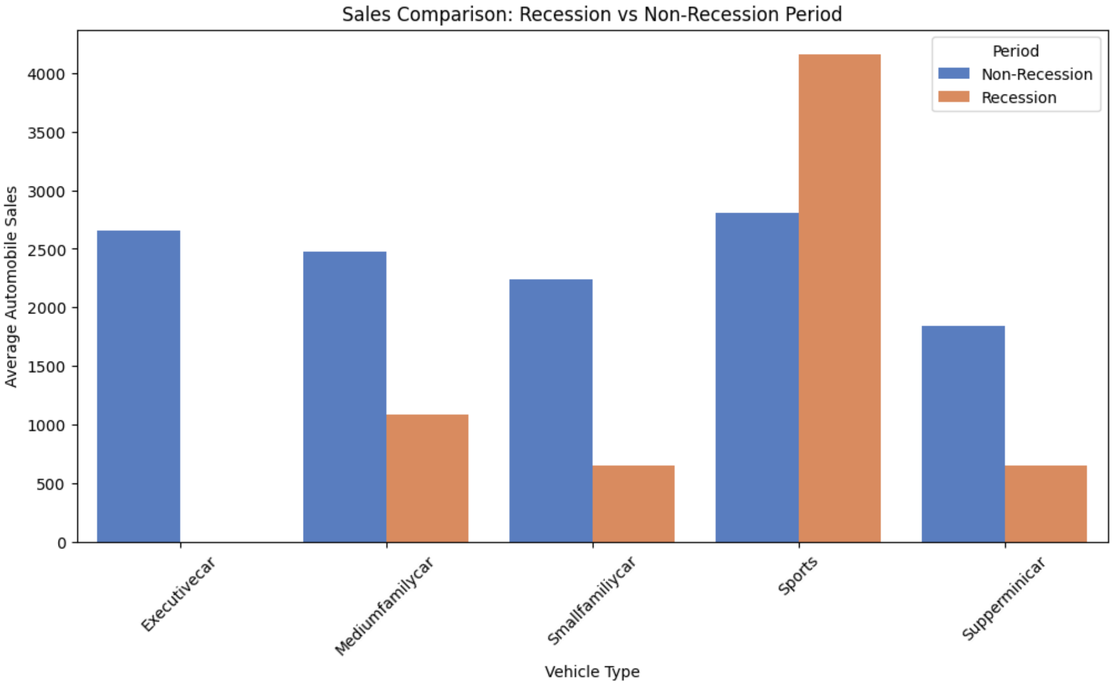
Click here for a hint

To visualize the average number of vehicles sold during recession and non-recession periods, you can use a bar chart You will need to group recession average Automobile_Sales and then plot it Make use of sns.barplot(x=x,y=y, data = df)

[14]:

```
# Define recession and non-recession periods (e.g., 2008-09 recession vs.␣
↪2010-12 non-recession)
df['Period'] = df['Year'].apply(lambda x: 'Recession' if x in [2008, 2009] else␣
↪'Non-Recession')

# Group data by 'Vehicle_Type' and 'Period', and calculate mean sales
df_period = df.groupby(['Vehicle_Type', 'Period'])['Automobile_Sales'].mean().
↪reset_index()

# Create the Seaborn bar plot
plt.figure(figsize=(12, 6))
```

```
sns.barplot(data=df_period, x='Vehicle_Type', y='Automobile_Sales', hue='Period', palette='muted')

# Add labels and title
plt.xlabel('Vehicle Type')
plt.ylabel('Average Automobile Sales')
plt.title('Sales Comparison: Recession vs Non-Recession Period')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Save the plot as an image (optional)
plt.savefig('seaborn_sales_comparison.png')

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



Click here for Solution template

```
new_df = df.groupby('Recession')['Automobile_Sales'].mean().reset_index()

# Create the bar chart using seaborn
plt.figure(figsize=(.........., ...........)
sns.barplot(x='Recession', y='Automobile_Sales', hue='Recession',  data=new_df)
plt.xlabel('...........')
```

```
    plt.ylabel('...............')
    plt.title('Average Automobile Sales during Recession and Non-Recession')
    plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
    plt.show()
```

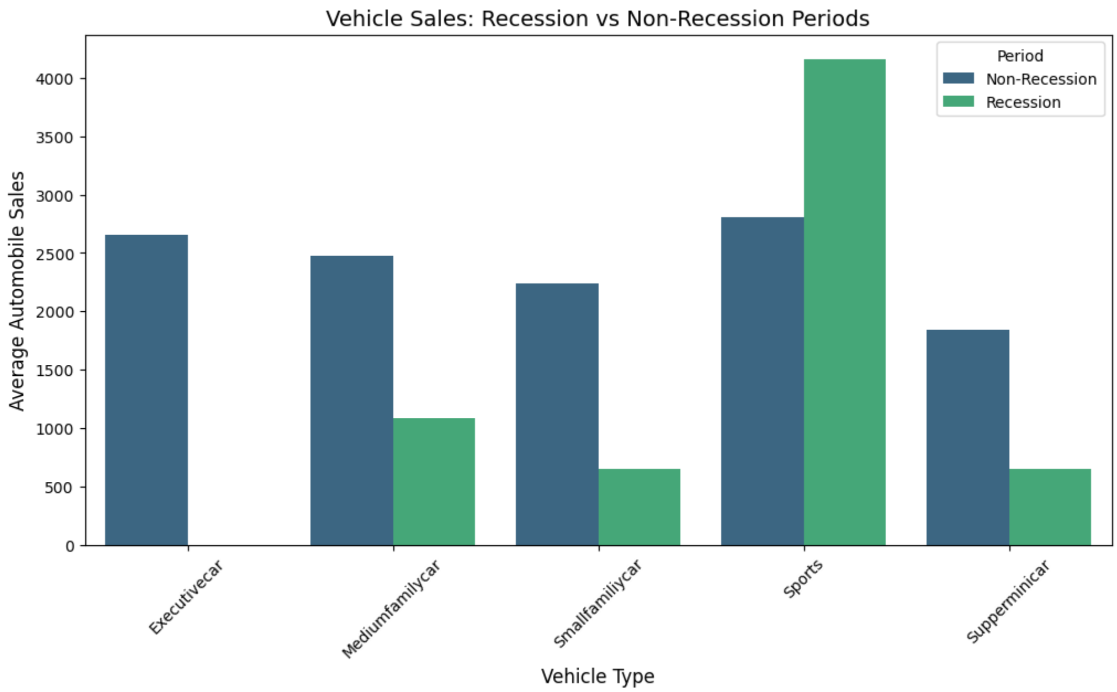
8.0.4 Now you want to compare the sales of different vehicle types during a recession and a non-recession period

We recommend that you use the functionality of **Seaborn Library** to create this visualization

Click here for a hint

To visualize sales of different vehicles during recession and non-recession periods, you can use a bar chart You will need to group Recession, Vehicle_Type for average Automobile_Sales and then plot it Make use of sns.barplot(x=x,y=y, data = df)

```
[15]: # Define Recession (e.g., 2008-09) vs Non-Recession periods (e.g., 2010-12)
df['Period'] = df['Year'].apply(lambda x: 'Recession' if x in [2008, 2009] else␣
 ↪'Non-Recession')

# Group data by 'Vehicle_Type' and 'Period', calculate average sales
df_period = df.groupby(['Vehicle_Type', 'Period'])['Automobile_Sales'].mean().
 ↪reset_index()

# Create the Seaborn bar plot
plt.figure(figsize=(12, 6))
sns.barplot(data=df_period, x='Vehicle_Type', y='Automobile_Sales',␣
 ↪hue='Period', palette='viridis')

# Add title and labels
plt.title('Vehicle Sales: Recession vs Non-Recession Periods', fontsize=14)
plt.xlabel('Vehicle Type', fontsize=12)
plt.ylabel('Average Automobile Sales', fontsize=12)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Save the plot as an image (optional)
plt.savefig('vehicle_sales_comparison.png')

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



Click here for Solution template

```
# Filter the data for recessionary periods
 recession_data = df[df['Recession'] == 1]

 dd=df.groupby(['Recession','Vehicle_Type'])['Automobile_Sales'].mean().reset_index()

 # Calculate the total sales volume by vehicle type during recessions
 #sales_by_vehicle_type = recession_data.groupby('Vehicle_Type')['Automobile_Sales'].sum().

 # Create the grouped bar chart using seaborn
 plt.figure(figsize=(10, 6))
 sns.barplot(x='......', y='........', hue='Vehicle_Type', data=dd)
 plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
 plt.xlabel('.............')
 plt.ylabel('..............')
 plt.title('Vehicle-Wise Sales during Recession and Non-Recession Period')

 plt.show()
```

8.0.5 From the above chart what insights have you gained on the overall sales of automobiles during recession? Type your answer below:-

Inference

From this plot, we can understand that there is a drastic decline in the overall sales of the automobiles during recession.However, the most affected type of vehicle is executivecar and sports

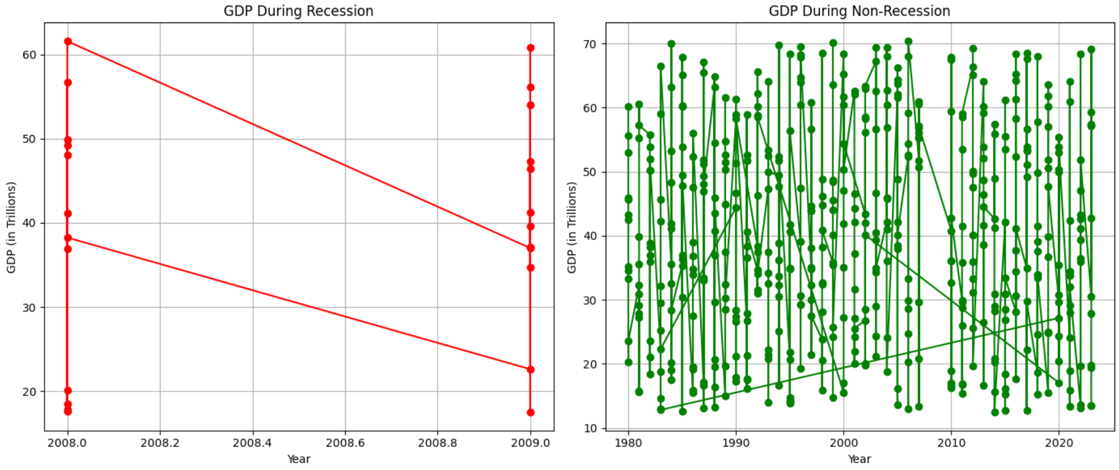
Save this plot as "Bar_Chart.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

8.0.6 TASK 1.4: Use sub plotting to compare the variations in GDP during recession and non-recession period by developing line plots for each period.

Now, you want to find more insights from the data to understand the reason. Plot a two line charts using subplotting to answer:- #### How did the GDP vary over time during recession and non-recession periods? Make use of add_subplot() from Matplotlib for this comparision.

[16]:

```
# Define recession and non-recession periods
recession_years = [2008, 2009]  # Adjust these based on your data
df_recession = df[df['Year'].isin(recession_years)]
df_non_recession = df[~df['Year'].isin(recession_years)]

# Create a figure with subplots
fig = plt.figure(figsize=(14, 6))

# Plot for Recession Period
ax1 = fig.add_subplot(1, 2, 1)  # 1 row, 2 columns, 1st subplot
ax1.plot(df_recession['Year'], df_recession['GDP'], marker='o', linestyle='-', color='r')
ax1.set_title('GDP During Recession')
ax1.set_xlabel('Year')
ax1.set_ylabel('GDP (in Trillions)')
ax1.grid(True)

# Plot for Non-Recession Period
ax2 = fig.add_subplot(1, 2, 2)  # 1 row, 2 columns, 2nd subplot
ax2.plot(df_non_recession['Year'], df_non_recession['GDP'], marker='o', linestyle='-', color='g')
ax2.set_title('GDP During Non-Recession')
ax2.set_xlabel('Year')
ax2.set_ylabel('GDP (in Trillions)')
ax2.grid(True)

# Adjust layout to prevent overlap
plt.tight_layout()

# Save the plot as an image (optional)
plt.savefig('gdp_comparison_subplots.png')

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



Click here for Solution template

```
 #Create dataframes for recession and non-recession period
 rec_data = df[df['Recession'] == 1]
 non_rec_data = df[df['Recession'] == 0]

 #Figure
 fig=plt.figure(figsize=(12, 6))

 #Create different axes for subploting
 ax0 = fig.add_subplot(1, 2, 1) # add subplot 1 (1 row, 2 columns, first plot)
 ax1 = fig.add_subplot(... ,... ,... ) # add subplot 2 (1 row, 2 columns, second plot).

 #plt.subplot(1, 2, 1)
 sns.lineplot(x='Year', y='GDP', data=rec_data, label='Recession', ax=ax0)
 ax0.set_xlabel('Year')
 ax0.set_ylabel('GDP')
 ax0.set_title('GDP Variation during Recession Period')

 #plt.subplot(1, 2, 2)
 sns.lineplot(x='......', y='......', data=........, label='.........',ax=...)
 ax1.set_xlabel('.....')
 ax1.set_ylabel('.......')
 ax1.set_title('..........')

 plt.tight_layout()
 plt.show()

#-----------------------------------------------Alternatively--------------
#Using subplot()
 plt.figure(figsize=(..........., .........))
```

```
#subplot 1
plt.subplot(1, 2, 1)
sns.lineplot(x='.........', y='......', data=.........., label='......')
plt.xlabel('.......')
plt.ylabel('..........')
plt.legend()
#subplot 1
plt.subplot(1, 2, 2)
sns.lineplot(x='.........', y='......', data=.........., label='......')
plt.xlabel('.......')
plt.ylabel('..........')
plt.legend()

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

8.0.7 Inference

From this plot, it is evident that during recession, the GDP of the country was in a low range, might have afected the overall sales of the company Save this plot as "Subplot.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

8.0.8 TASK 1.5: Develop a Bubble plot for displaying the impact of seasonality on Automobile Sales.

How has seasonality impacted the sales, in which months the sales were high or low? Check it for non-recession years to understand the trend

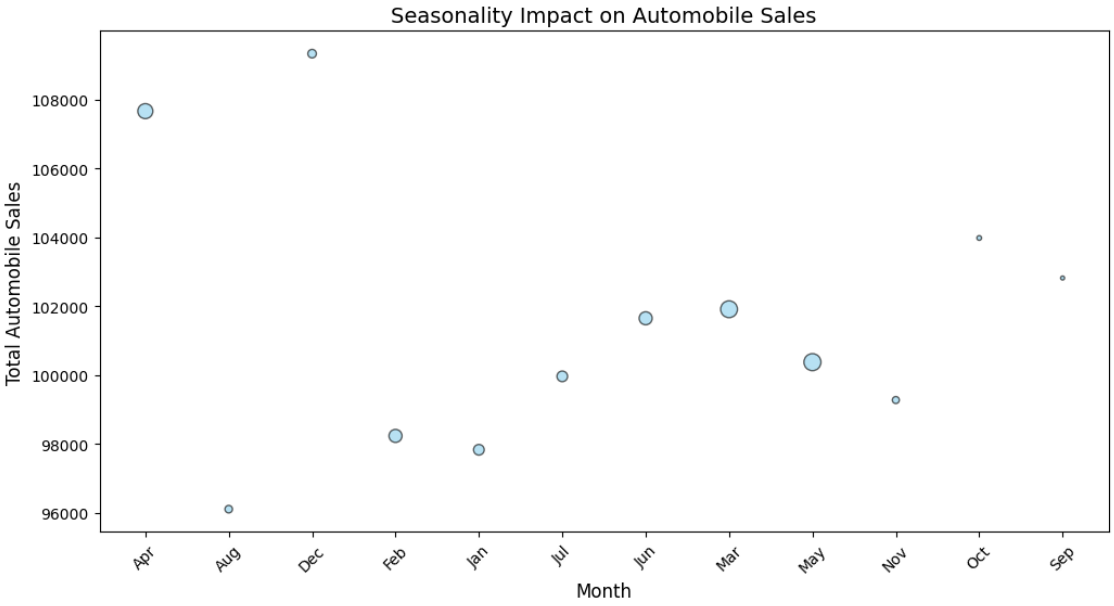
Develop a Bubble plot for displaying Automobile Sales for every month and use Seasonality Weight for representing the size of each bubble Title this plot as 'Seasonality impact on Automobile Sales'

Click here for a hint

You can create Bubble Chart by calling the scatter() Pass the 'Month' and 'Automobile_Sales' to the functions as x and y and then use Seasonality weight for size parameter

[17]:
```
# Filter data for non-recession years (e.g., excluding 2008-09)
recession_years = [2008, 2009]  # Modify based on your data
df_non_recession = df[~df['Year'].isin(recession_years)]

# Group data by 'Month' and calculate total sales and seasonality weight
df_seasonality = df_non_recession.groupby('Month').agg(
    {'Automobile_Sales': 'sum', 'Seasonality_Weight': 'mean'}
).reset_index()

# Create the bubble plot
plt.figure(figsize=(12, 6))
```

```
# Scatter plot with size representing seasonality weight
plt.scatter(
    df_seasonality['Month'],
    df_seasonality['Automobile_Sales'],
    s=df_seasonality['Seasonality_Weight'] * 100,  # Adjust bubble size
    alpha=0.6,
    color='skyblue',
    edgecolors='black'
)

# Add title and labels
plt.title('Seasonality Impact on Automobile Sales', fontsize=14)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Total Automobile Sales', fontsize=12)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Save the plot as an image (optional)
plt.savefig('seasonality_impact_bubble_plot.png')

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



Click here for Solution template

```
non_rec_data = df[df['Recession'] == 0]
```

```
size=non_rec_data['Seasonality_Weight'] #for bubble effect

sns.scatterplot(data=non_rec_data, x='........', y='........', size=size)

#you can further include hue='Seasonality_Weight', legend=False)

plt.xlabel('Month')
plt.ylabel('Automobile_Sales')
plt.title('Seasonality impact on Automobile Sales')

plt.show()
```

8.0.9 Inference

From this plot, it is evident that seasonality has not affected on the overall sales. However, there is a drastic raise in sales in the month of April Save this plot as "Bubble.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

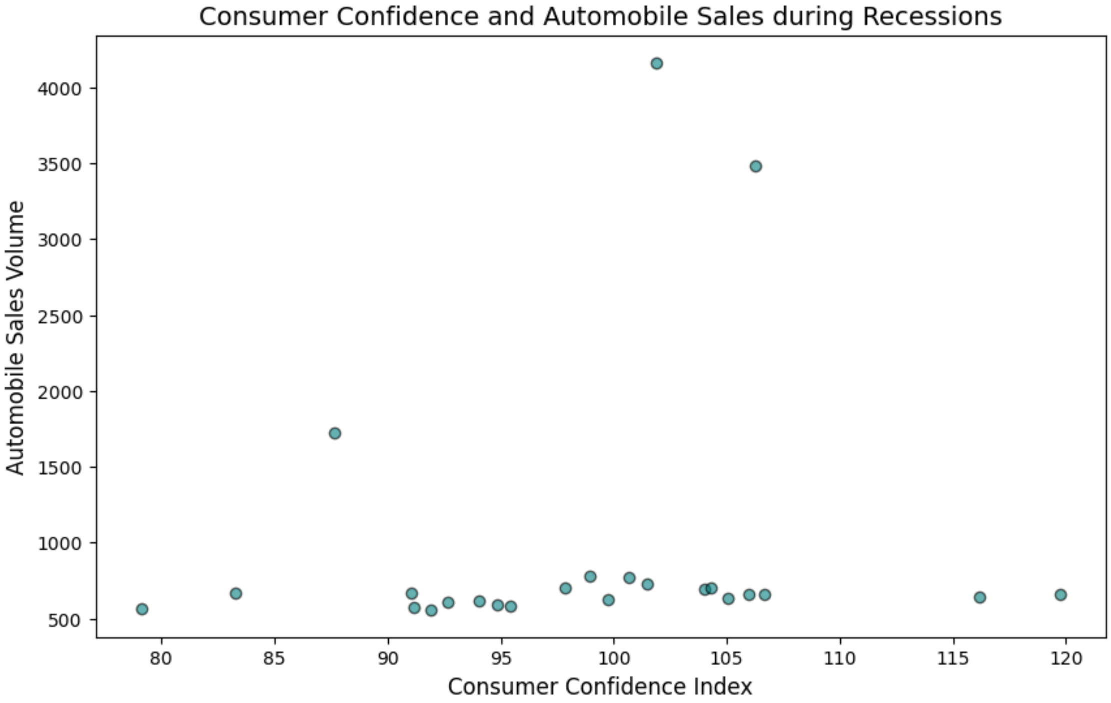
### TASK 1.6: Use the functionality of Matplotlib to develop a scatter plot to identify the correlation between average vehicle price relate to the sales volume during recessions. #### From the data, develop a scatter plot to identify if there a correlation between consumer confidence and automobile sales during recession period? Title this plot as 'Consumer Confidence and Automobile Sales during Recessions'

Click here for a hint

You can create dataframe where recession is '1'. Pass the 'Consumer_Confidence' and 'Automobile_Sales' to the plt.scatter()

[19]:
```
# Create dataframes for recession and non-recession periods
recession_years = [2008, 2009]  # Modify as needed
rec_data = df[df['Year'].isin(recession_years)]

# Create the scatter plot
plt.figure(figsize=(10, 6))

plt.scatter(
    rec_data['Consumer_Confidence'],  # Use the correct column name
    rec_data['Automobile_Sales'],     # Use the correct column name
    color='teal',
    alpha=0.6,
    edgecolors='black'
)

# Add title and axis labels
plt.title('Consumer Confidence and Automobile Sales during Recessions',␣
 ↪fontsize=14)
```

```
plt.xlabel('Consumer Confidence Index', fontsize=12)
plt.ylabel('Automobile Sales Volume', fontsize=12)

# Save the plot as an image (optional)
plt.savefig('consumer_confidence_vs_sales_scatter.png')

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



Click here for Solution template

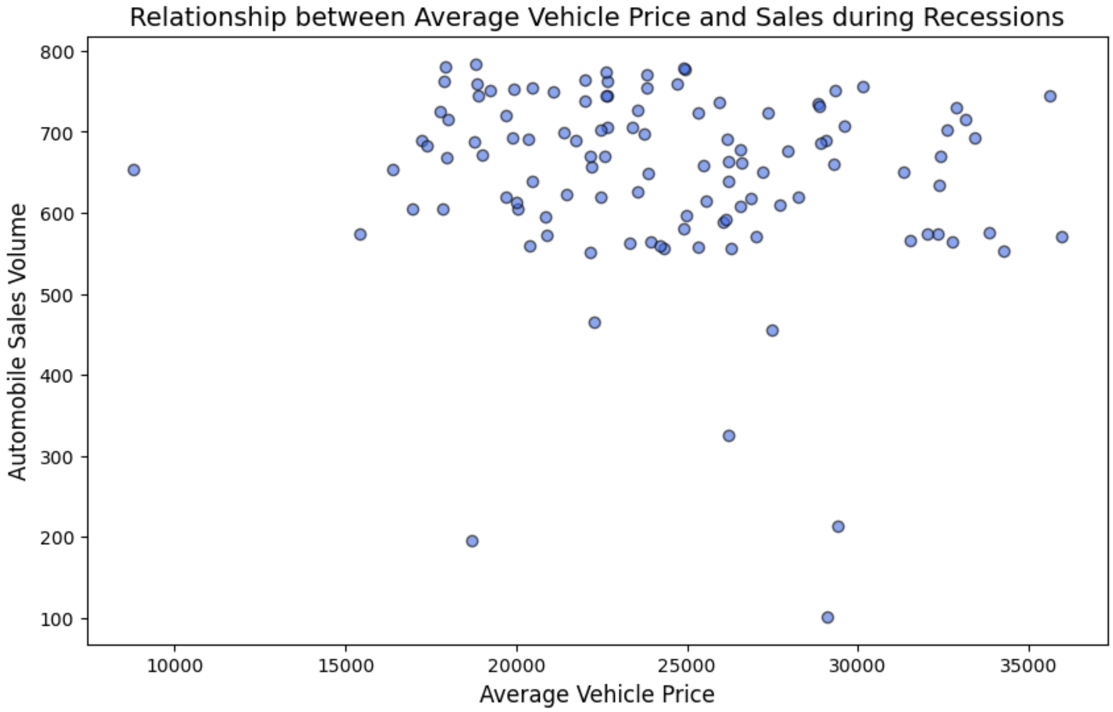
```
#Create dataframes for recession and non-recession period
rec_data = df[df['Recession'] == 1]
plt.scatter(rec_data['Consumer_Confidence'], rec_data['Automobile_Sales'])

plt.xlabel('.....')
plt.ylabel('.......')
plt.title('..........')
plt.show()
```

### How does the average vehicle price relate to the sales volume during recessions? Plot another scatter plot and title it as 'Relationship between Average Vehicle Price and Sales during Recessions'

[21]:

```
# Create a DataFrame for recession periods
rec_data = df[df['Recession'] == 1]
```

```
# Create the scatter plot
plt.figure(figsize=(10, 6))

# Use the correct column name for price
plt.scatter(
    rec_data['Price'],  # Update this with the correct column name
    rec_data['Automobile_Sales'],  # Update this with the correct column name
    color='royalblue',
    alpha=0.6,
    edgecolors='black'
)

# Add title and axis labels
plt.title('Relationship between Average Vehicle Price and Sales during␣
 ↪Recessions', fontsize=14)
plt.xlabel('Average Vehicle Price', fontsize=12)  # Update if necessary
plt.ylabel('Automobile Sales Volume', fontsize=12)  # Update if necessary

# Save the plot as an image (optional)
plt.savefig('average_price_vs_sales_scatter.png')

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



Click here for Solution template

```
#Create dataframes for recession and non-recession period
rec_data = df[df['Recession'] == 1]
plt.scatter(recession_data['Price'], rec_data['Automobile_Sales'])

plt.xlabel('.....')
plt.ylabel('.......')
plt.title('..........')
plt.show()
```

8.0.10 Inference

There is not much relation! Save this plot as "Scatter.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

### TASK 1.7: Create a pie chart to display the portion of advertising expenditure of XYZAutomotives during recession and non-recession periods. How did the advertising expenditure of XYZAutomotives change during recession and non-recession periods?

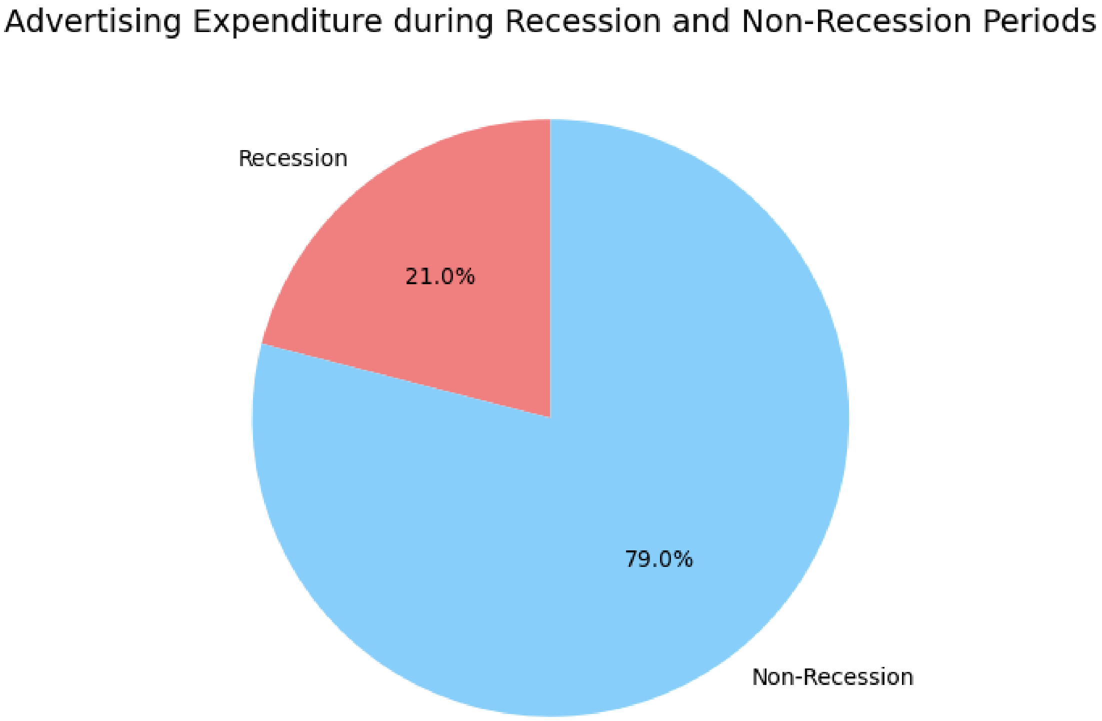
Click here for a hint

You can create two dataframe for recession and nonreccession period. Calculate the sum of Advertising_Expenditure for both dataframes Pass these total values to plt.pie(). May include labels as ['Recession', 'Non-Recession'] Feel Free to customie the pie further title this plot as - Advertising Expenditure during Recession and Non-Recession Periods

[22]:

```
# Filter the data for recession and non-recession periods
Rdata = df[df['Recession'] == 1]
NRdata = df[df['Recession'] == 0]

# Calculate the total advertising expenditure for both periods
RAtotal = Rdata['Advertising_Expenditure'].sum()  # Use the correct column name
NRAtotal = NRdata['Advertising_Expenditure'].sum()  # Use the correct column␣
  ↪name

# Create a pie chart for the advertising expenditure
plt.figure(figsize=(8, 6))

labels = ['Recession', 'Non-Recession']
sizes = [RAtotal, NRAtotal]
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90,␣
  ↪colors=['lightcoral', 'lightskyblue'])

plt.title('Advertising Expenditure during Recession and Non-Recession Periods',␣
  ↪fontsize=14)

# Display the pie chart
```

```
plt.show()
```



Click here for Solution template

```
# Filter the data
Rdata = df[df['Recession'] == 1]
NRdata = df[df['Recession'] == 0]

# Calculate the total advertising expenditure for both periods
RAtotal = Rdata['...........'].sum()
NRAtotal = NRdata['...........'].sum()

# Create a pie chart for the advertising expenditure
plt.figure(figsize=(8, 6))

labels = ['Recession', 'Non-Recession']
sizes = [RAtotal, NRtotal]
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)

plt.title('...........................')

plt.show()
```

From the above plot, what insights do you find on the advertisement expenditure during recession and non recession periods? Type your answer below:- Inference

It seems ABCAutomotives has been spending much more on the advertisements during non-recession periods as compared to during recession times. Fair enough!

Save this plot as "Pie_1.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

8.0.11 TASK 1.8: Develop a pie chart to display the total Advertisement expenditure for each vehicle type during recession period.

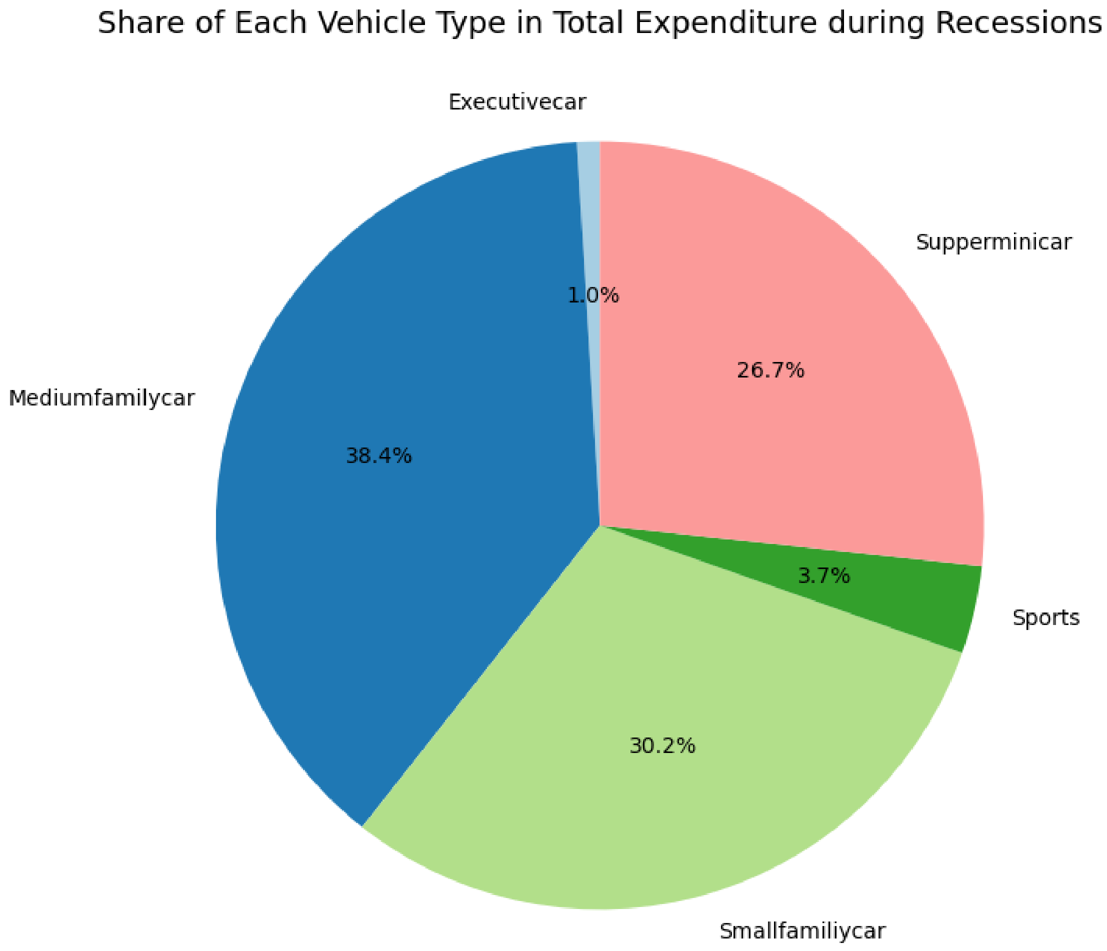
Can we observe the share of each vehicle type in total expenditure during recessions?

Click here for a hint

You will be required to group vehicle type for sum of advertisement expenditure. the plot a pie with the data, May include relevant labels title this plot as - Share of Each Vehicle Type in Total Expenditure during Recessions

[23]:

```
# Filter the data for recession periods
Rdata = df[df['Recession'] == 1]

# Calculate the advertisement expenditure by vehicle type during recessions
VTexpenditure = Rdata.groupby('Vehicle_Type')['Advertising_Expenditure'].sum() ␣
  ↪# Use the correct column names

# Create a pie chart for the share of each vehicle type in total expenditure␣
  ↪during recessions
plt.figure(figsize=(10, 8))

labels = VTexpenditure.index
sizes = VTexpenditure.values
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90, colors=plt.cm.
  ↪Paired.colors)

plt.title('Share of Each Vehicle Type in Total Expenditure during Recessions',␣
  ↪fontsize=14)

# Display the pie chart
plt.show()
```



Click here for Solution template

```
# Filter the data
Rdata = df[df['Recession'] == 1]

# Calculate the sales volume by vehicle type during recessions
VTexpenditure = Rdata.groupby('..........')['.............'].sum()

# Create a pie chart for the share of each vehicle type in total expenditure during recess
plt.figure(figsize=(..., ...))

labels = VTexpenditure.index
sizes = VTexpenditure.values
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)

plt.title('....................')
```

```
    plt.show()
```

8.0.12 Inference

During recession the advertisements were mostly focued on low price range vehicle. A wise decision! Save this plot as "Pie_2.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

### TASK 1.9: Develop a lineplot to analyse the effect of the unemployment rate on vehicle type and sales during the Recession Period. Analyze the effect of the unemployment rate on vehicle type and sales during the Recession Period #### You can create a lineplot and title the plot as 'Effect of Unemployment Rate on Vehicle Type and Sales'

Click here for a hint

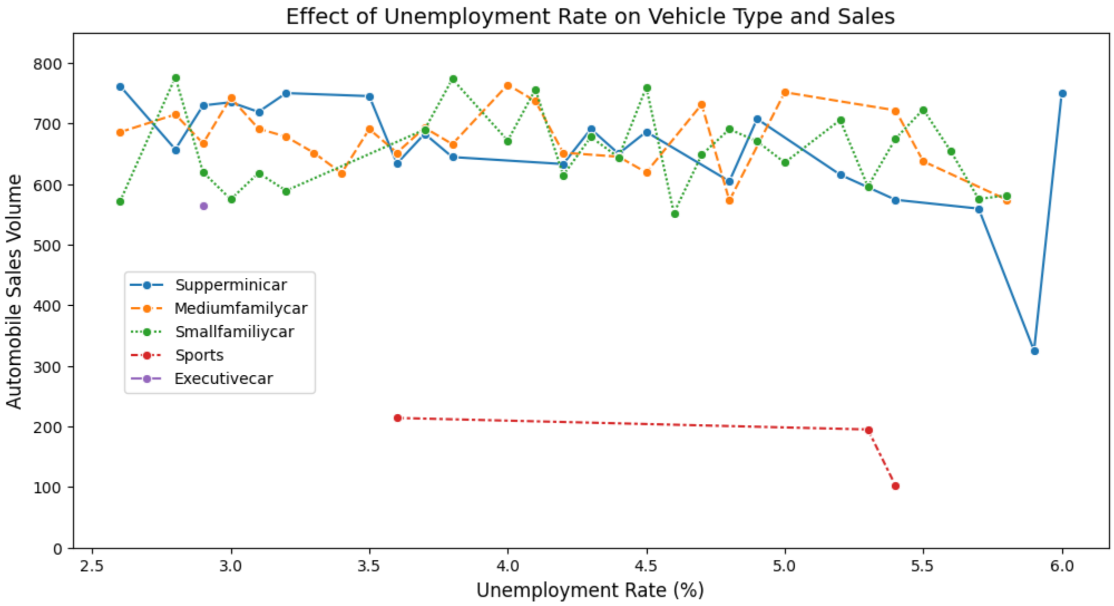
Filter out the data for recession period Make use of lineplot() from seaborn and pass the relavent data

[26]:

```
# Filter the data for recession periods
df_rec = df[df['Recession'] == 1]

# Create a line plot to analyze the effect of unemployment rate on vehicle type␣
  ↪and sales
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df_rec,
    x='unemployment_rate',     # Corrected column name
    y='Automobile_Sales',       # Ensure this matches the actual column name
    hue='Vehicle_Type',
    style='Vehicle_Type',
    markers='o',
    err_style=None
)

# Customize the plot
plt.title('Effect of Unemployment Rate on Vehicle Type and Sales', fontsize=14)
plt.ylim(0, 850)  # Adjust y-axis limits based on your data
plt.xlabel('Unemployment Rate (%)', fontsize=12)
plt.ylabel('Automobile Sales Volume', fontsize=12)
plt.legend(loc=(0.05, 0.3))

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

```
<ipython-input-26-b5179a8d0b88>:6: UserWarning:
The markers list has fewer values (1) than needed (5) and will cycle, which may
produce an uninterpretable plot.
```

```
sns.lineplot(
```



Click here for Solution template

```
df_rec = df[df['Recession']==1]
sns.lineplot(data=df_rec, x='..........', y='.........',
             hue='Vehicle_Type', style='Vehicle_Type', markers='o', err_style=None)
plt.ylim(0,850)
plt.legend(loc=(0.05,.3))
```

From the above plot, what insights have you gained on the sales of superminicar, smallfamilycar, mediumminicar? Type your answer below:- Inference

During recession, buying pattern changed, the sales of low range vehicle like superminicar,smallfamilycar and Mediumminicar

Save this plot as "line_plot_3.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

8.0.13 OPTIONAL : TASK 1.10 Create a map on the hightest sales region/offices of the company during recession period

[27]:

```
from pyodide.http import pyfetch

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())

path = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/us-states.json'
await download(path, "us-states.json")

filename = "us-states.json"
```

You found that the datset also contains the location/city for company offices. Now you want to show the recession impact on various offices/city sales by developing a choropleth

[34]:

```
# Import necessary libraries
from js import fetch
import io
import pandas as pd
import folium

# Fetch the dataset
URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/
 ↪historical_automobile_sales.csv"
resp = await fetch(URL)
text = io.BytesIO((await resp.arrayBuffer()).to_py())
df = pd.read_csv(text)

# Now, you can proceed with filtering and creating your choropleth map
# Filter the data for the recession period
recession_data = df[df['Recession'] == 1]

# Calculate the total sales by city
sales_by_city = recession_data.groupby('City')['Automobile_Sales'].sum().
 ↪reset_index()

# Create a base map centered on the United States
map1 = folium.Map(location=[37.0902, -95.7129], zoom_start=4)

# Create a choropleth layer using Folium
choropleth = folium.Choropleth(
    geo_data='us-states.json',  # GeoJSON file with state boundaries
    data=sales_by_city,
    columns=['City', 'Automobile_Sales'],
    key_on='feature.properties.name',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Automobile Sales during Recession'
).add_to(map1)

# Add tooltips to the choropleth layer
choropleth.geojson.add_child(
    folium.features.GeoJsonTooltip(fields=['name'], labels=True)
)

# Display the map
map1
```

```
[34]: <folium.folium.Map at 0x7f61498>
```

Click for Solution

```
# Filter the data for the recession period and specific cities
recession_data = data[data['Recession'] == 1]

# Calculate the total sales by city
sales_by_city = recession_data.groupby('City')['Automobile_Sales'].sum().reset_index()

# Create a base map centered on the United States
map1 = folium.Map(location=[37.0902, -95.7129], zoom_start=4)

# Create a choropleth layer using Folium
choropleth = folium.Choropleth(
    geo_data= 'us-states.json',  # GeoJSON file with state boundaries
    data=sales_by_city,
    columns=['City', 'Automobile_Sales'],
    key_on='feature.properties.name',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Automobile Sales during Recession'
).add_to(map1)


# Add tooltips to the choropleth layer
choropleth.geojson.add_child(
    folium.features.GeoJsonTooltip(['name'], labels=True)
)

# Display the map
map1
```

9 Congratulations! You have completed the lab

9.1 Authors

Dr. Pooja

`toggle ## Change Log`