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.

Table of Contents

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

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.

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"

%pip install seaborn
%pip install folium
print ('Done')
```

```
Done
```

Importing Required Libraries

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

```
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
print ('Done')
```

```
Done
```

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)

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.

Importing Data

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

```
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!
```

```
df.describe()
```

```
              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
```

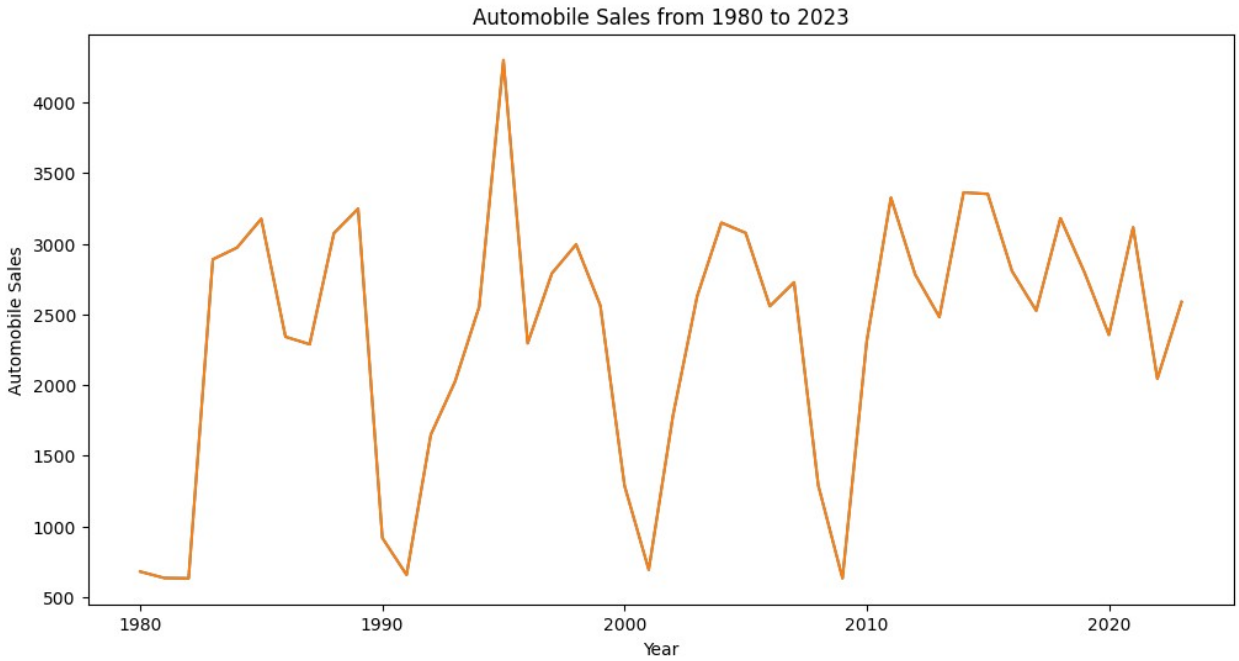
```
1.500000

              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

df.columns

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')

df.head()

        Date  Year Month  Recession  Consumer_Confidence
Seasonality_Weight  \
0  1/31/1980  1980   Jan          1               108.24
0.50
1  2/29/1980  1980   Feb          1                98.75
0.75
2  3/31/1980  1980   Mar          1               107.48
0.20
3  4/30/1980  1980   Apr          1               115.01
```

```
1.00
4  5/31/1980  1980   May           1                 98.72
0.20

       Price  Advertising_Expenditure  Competition     GDP
Growth_Rate  \
0  27483.571                     1558            7  60.223
0.010000
1  24308.678                     3048            4  45.986    -
0.309594
2  28238.443                     3137            3  35.141    -
0.308614
3  32615.149                     1653            7  45.673
0.230596
4  23829.233                     1319            4  52.997
0.138197

   unemployment_rate  Automobile_Sales     Vehicle_Type        City
0                5.4             456.0    Supperminicar     Georgia
1                4.8             555.9    Supperminicar    New York
2                3.4             620.0  Mediumfamilycar    New York
3                4.2             702.8    Supperminicar    Illinois
4                5.3             770.4  Smallfamiliycar  California
```

Creating Visualizations for Data Analysis

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

```
plt.figure(figsize=(12, 6))
df_line=df.groupby('Year')['Automobile_Sales'].mean()
df_line.plot(kind = 'line')
df_line.plot(x=df_line.index, y=df_line.values)
plt.xlabel('Year')
plt.ylabel('Automobile Sales')
plt.title('Automobile Sales from 1980 to 2023')
plt.show()
```

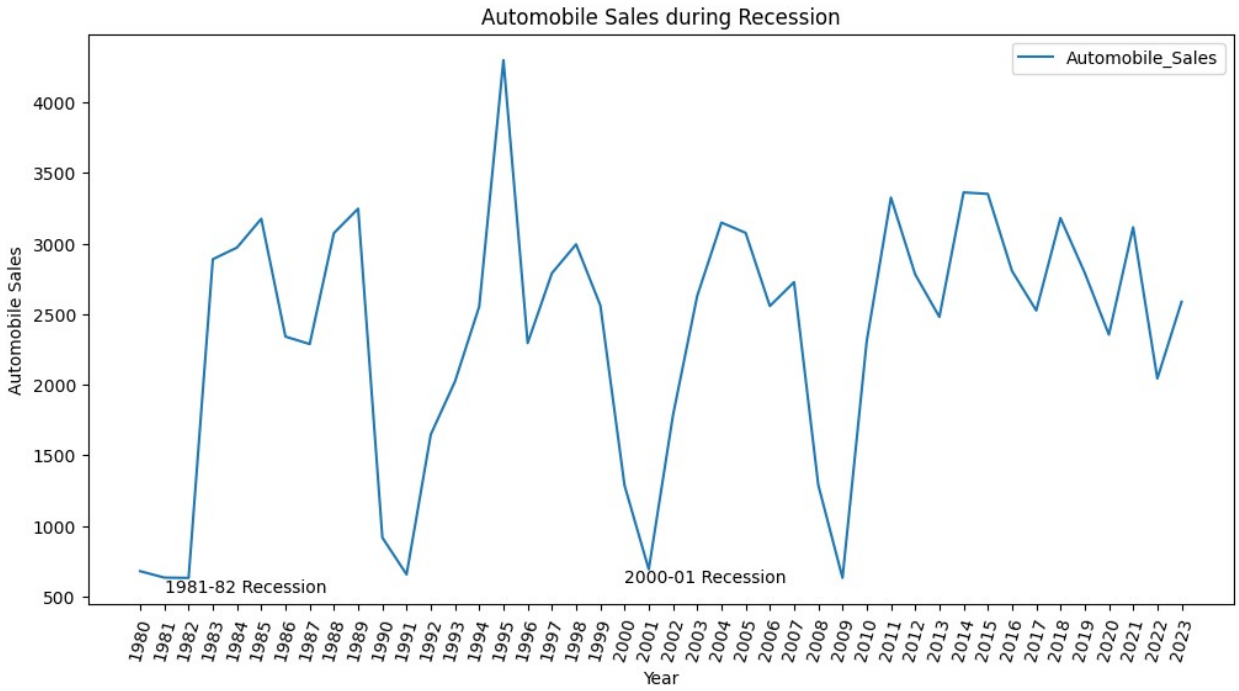


Include the following on the plot

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

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*

```
plt.figure(figsize=(12, 6))
df_line = df.groupby('Year')['Automobile_Sales'].mean()
df_line.plot(kind = 'line')
plt.xticks(list(range(1980,2024)), rotation = 75)
plt.xlabel('Year')
plt.ylabel('Automobile Sales')
plt.title('Automobile Sales during Recession')
plt.text(1981, 530, '1981-82 Recession')
plt.text(2000, 600, '2000-01 Recession')
plt.legend()
plt.show()
```



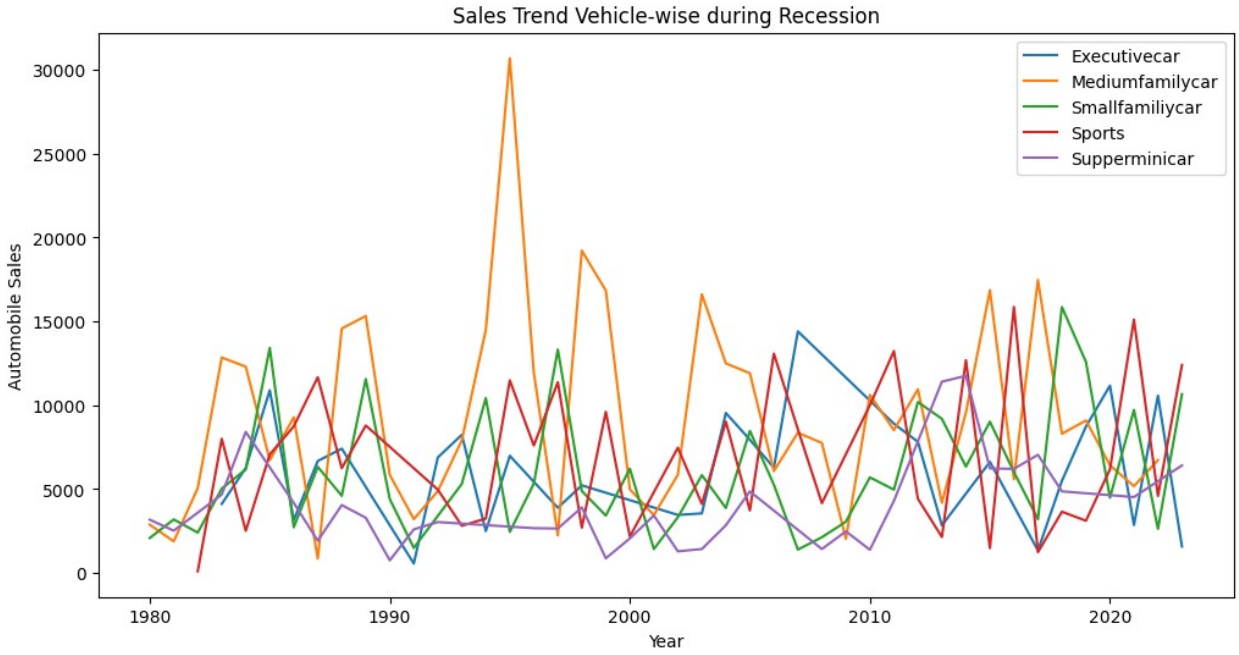
TASK 1.2: Plot different lines for categories of vehicle type and analyse the trend to answer the question Is there a noticeable difference in sales trends between different vehicle types during recession periods?

```
df_Mline = df.groupby(['Year','Vehicle_Type'], as_index=False)
['Automobile_Sales'].sum()
df_Mline.set_index('Year', inplace=True)
df_Mline = df_Mline.groupby(['Vehicle_Type'])['Automobile_Sales']
df_Mline.head()
```

```
Year
1980     2882.5
1980     2085.7
1980     3179.2
1981     1885.2
1981     3192.1
1981     2523.6
1982     5062.9
1982     2408.9
1982      102.0
1983     4120.2
1983    12845.6
1983     5030.6
1983     7998.7
1983     4677.1
1984     6261.6
1984    12299.7
```

```
1984     6175.8
1984     2514.0
1984     8415.9
1985    10892.0
1985     7068.4
1986     3163.0
1986     8740.0
1986     4178.5
1987     6676.0
Name: Automobile_Sales, dtype: float64
```

```
plt.figure(figsize=(12, 6))
df_Mline.plot(kind='line')
plt.xlabel('Year')
plt.ylabel('Automobile Sales')
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:

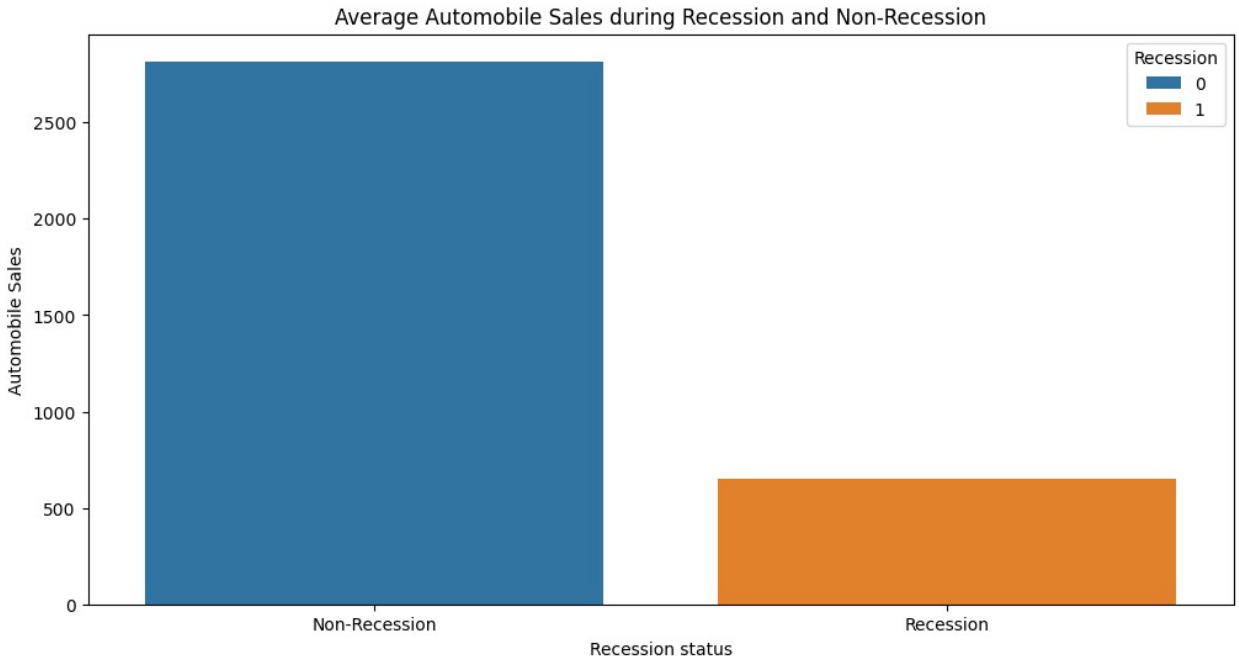
Save this plot as "Line_Plot_2.png" *Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine*

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.

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

```
   Recession  Automobile_Sales
0          0       2816.753590
1          1        648.516814
```

```
plt.figure(figsize=(12, 6))
sns.barplot(x='Recession', y='Automobile_Sales', hue='Recession',
data=new_df)
plt.xlabel('Recession status')
plt.ylabel('Automobile Sales')
plt.title('Average Automobile Sales during Recession and Non-
Recession')
plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
plt.show()
```



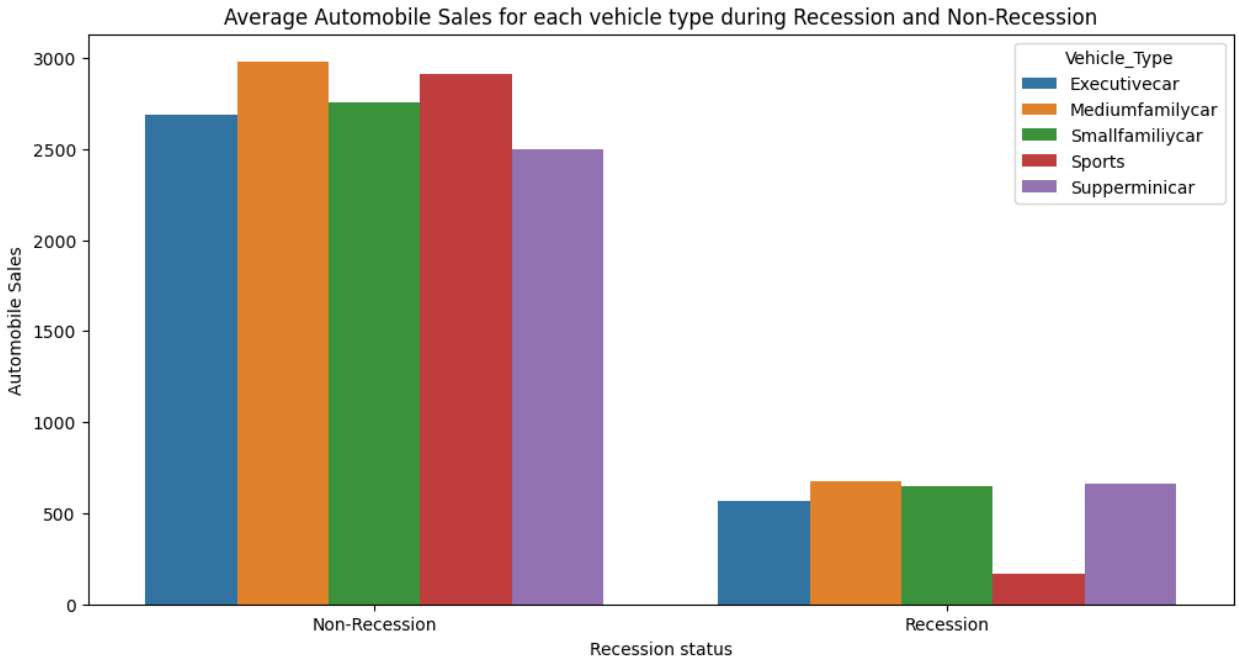
Now you want to compare the sales of different vehicle types during a recession and a non-recession period

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

```
new_df1 = df.groupby(['Recession','Vehicle_Type'])
['Automobile_Sales'].mean().reset_index()
new_df1
```

	Recession	Vehicle_Type	Automobile_Sales
0	0	Executivecar	2686.436232
1	0	Mediumfamilycar	2981.501935
2	0	Smallfamiliycar	2752.658140
3	0	Sports	2910.636264
4	0	Supperminicar	2495.242222
5	1	Executivecar	564.000000
6	1	Mediumfamilycar	674.847619
7	1	Smallfamiliycar	650.733333
8	1	Sports	170.333333
9	1	Supperminicar	659.270968

```
plt.figure(figsize=(12, 6))
sns.barplot(x='Recession', y='Automobile_Sales', hue='Vehicle_Type',
data=new_df1)
plt.xlabel('Recession status')
plt.ylabel('Automobile Sales')
plt.title('Average Automobile Sales for each vehicle type during
Recession and Non-Recession')
plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
plt.show()
```



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

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*

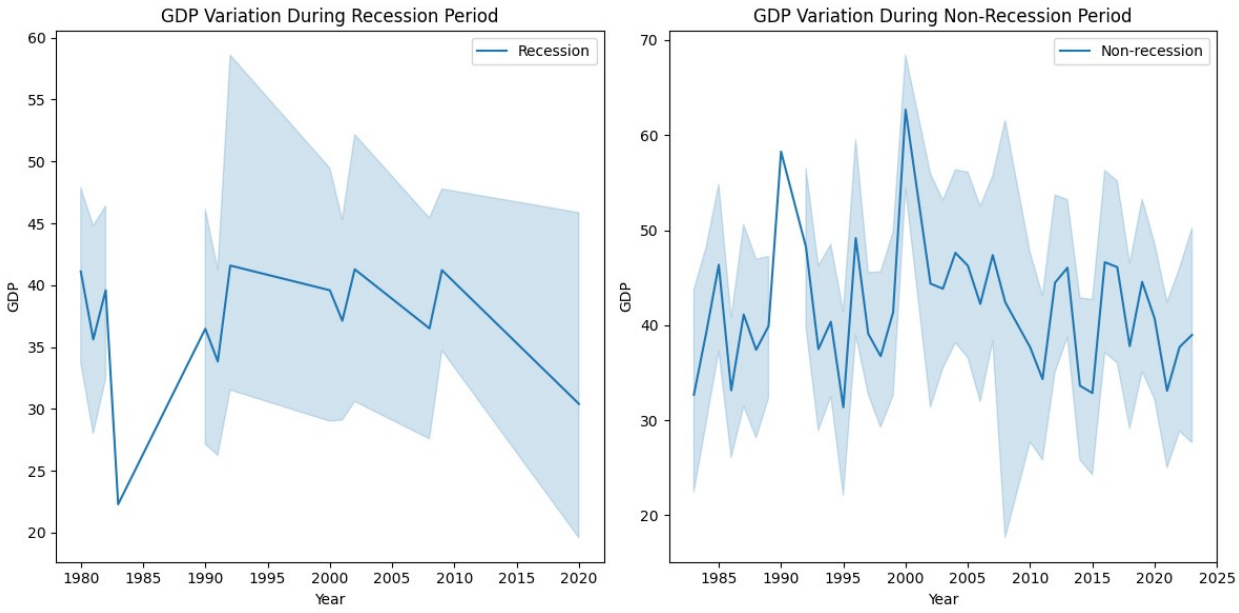
TASK 1.4: Use sub plotting to compare the variations in GDP during recession and non-recession period by developing line plots for each period.

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.

```
#Create dataframes for recession and non-recession period
recession_data = df[df['Recession'] == 1]
non_recession_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(1, 2, 2) # add subplot 2 (1 row, 2 columns,
second plot).

#plt.subplot(1, 2, 1)
sns.lineplot(x='Year', y='GDP', data=recession_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='Year', y='GDP', data=non_recession_data, label='Non-
recession',ax=ax1)
ax1.set_xlabel('Year')
ax1.set_ylabel('GDP')
ax1.set_title('GDP Variation During Non-Recession Period')

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



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*

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

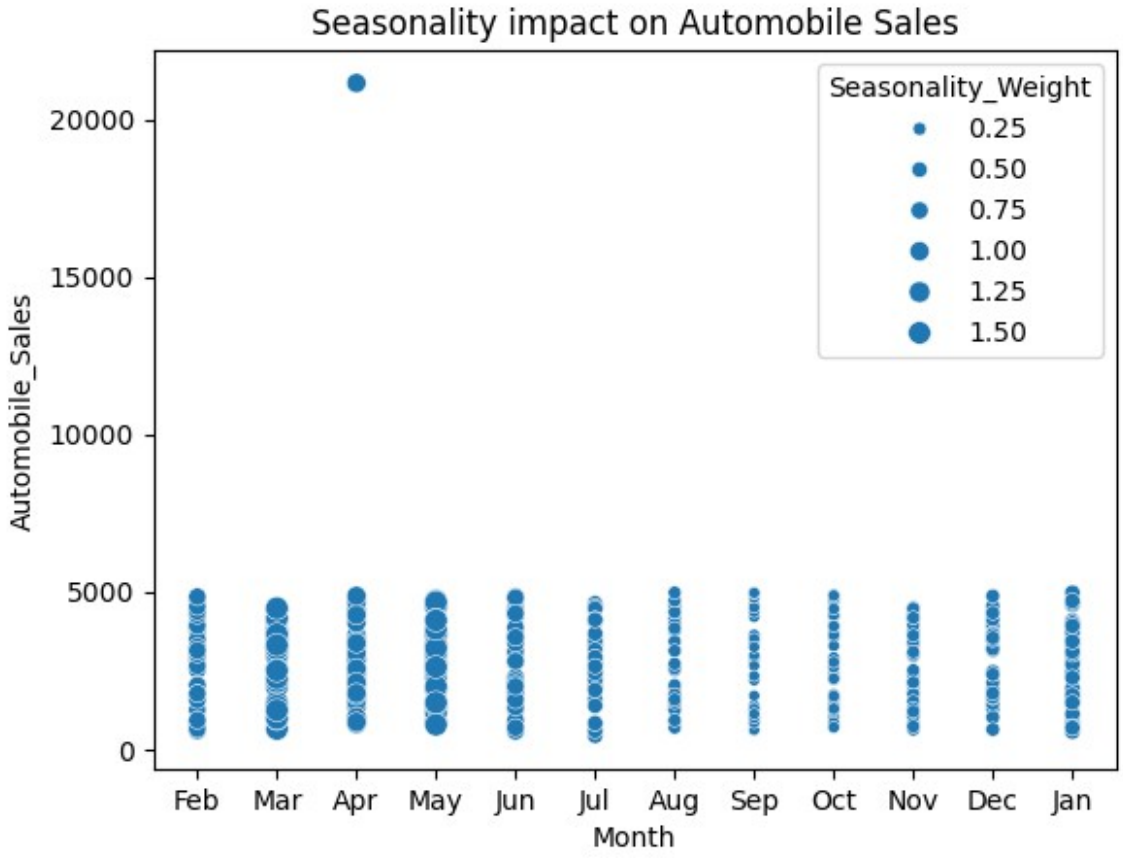
How has seasonality impacted the sales, in which months the sales were high or low? Check it for non-recession years to understand the trend

Develop a Bubble plot for displaying Automobile Sales for every month and use Seasonality Weight for representing the size of each bubble

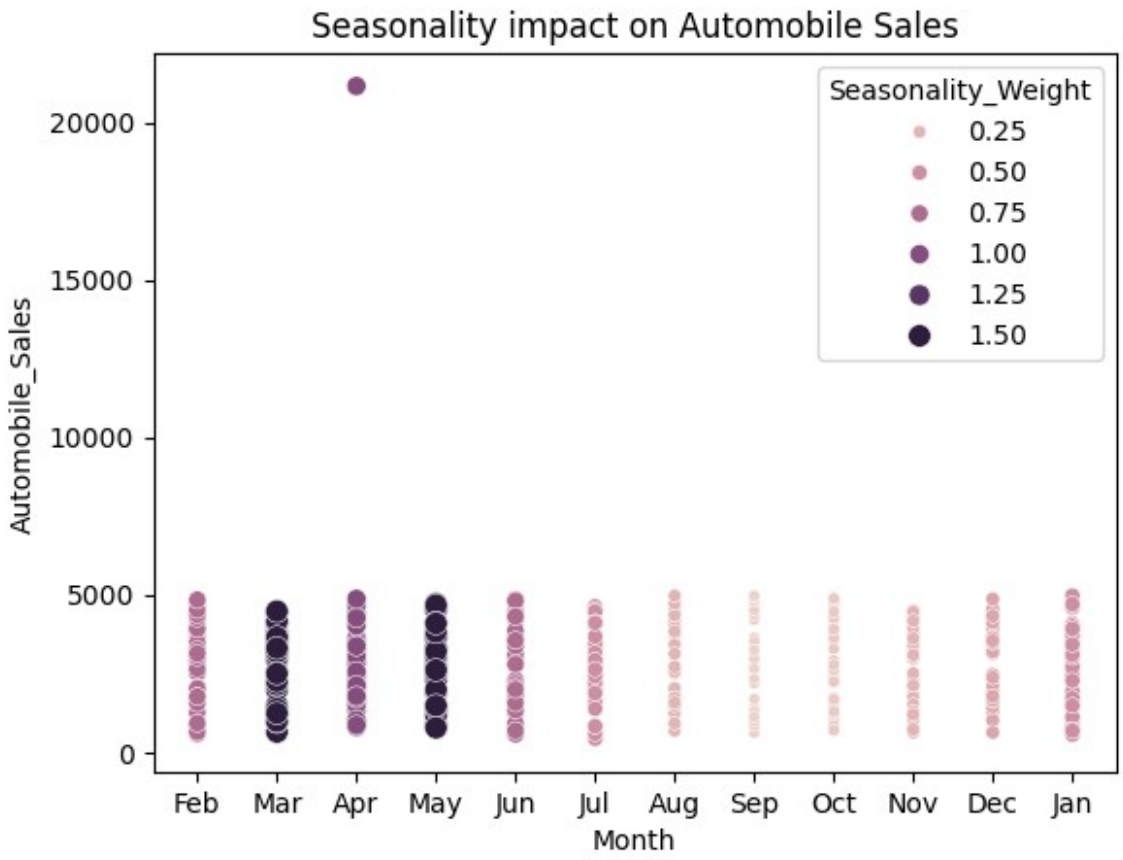
Title this plot as 'Seasonality impact on Automobile Sales'

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

size=nonrecession_data['Seasonality_Weight']

sns.scatterplot(x='Month', y='Automobile_Sales',
data=nonrecession_data, size=size)

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

```
plt.show()
```




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

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

sns.scatterplot(data=non_rec_data, x='Month', y='Automobile_Sales',
size=size, hue='Seasonality_Weight')

#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()
```



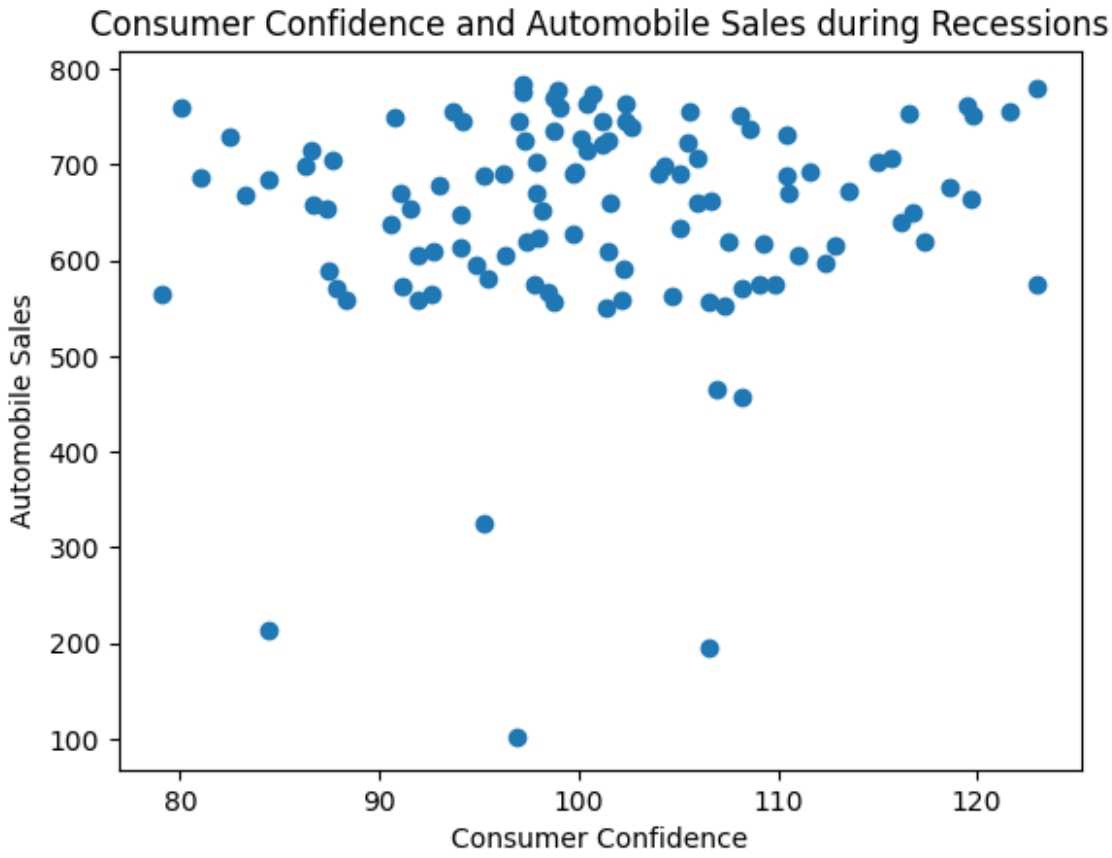
Inference

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'

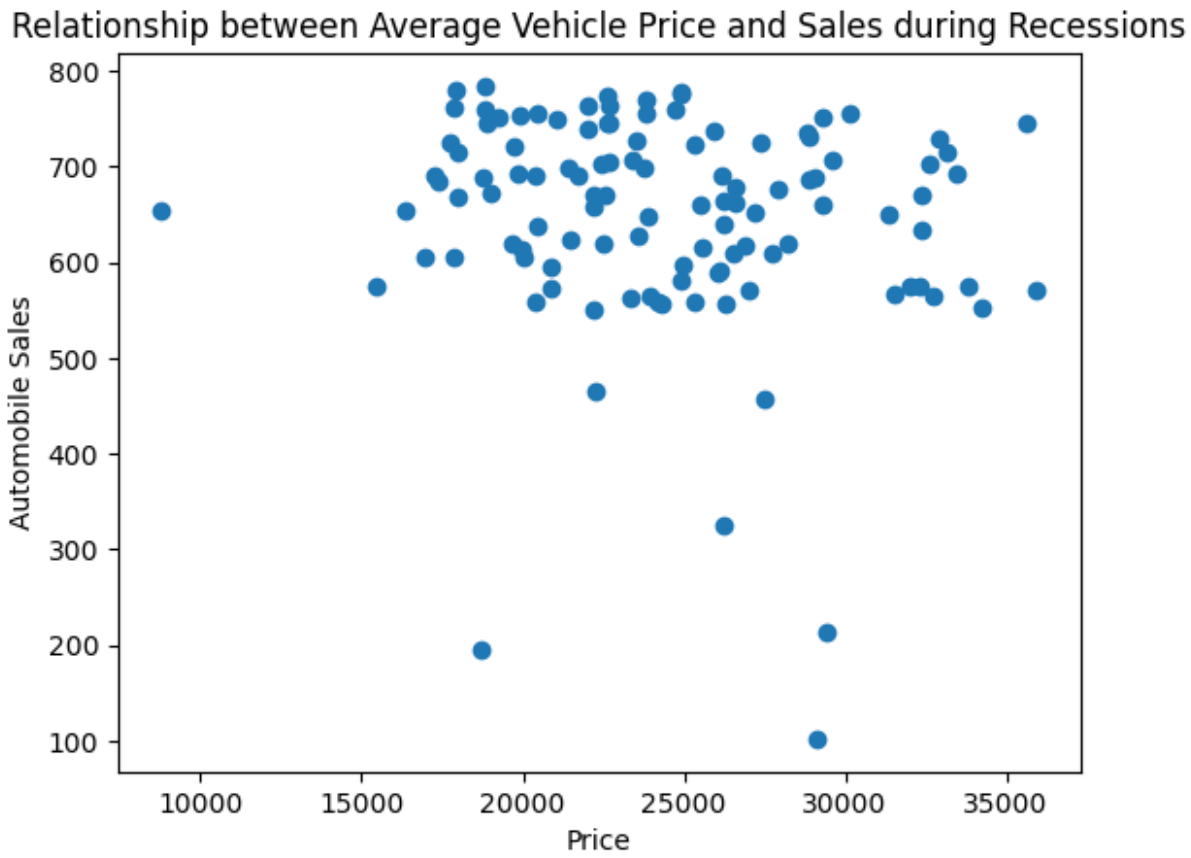
```
#Create dataframes for recession and non-recession period
rec_data = df[df['Recession'] == 1]
plt.scatter(rec_data['Consumer_Confidence'],
rec_data['Automobile_Sales'])

plt.xlabel('Consumer Confidence')
plt.ylabel('Automobile Sales')
plt.title('Consumer Confidence and Automobile Sales during
```

```
Recessions')
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'

```
#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('Price')
plt.ylabel('Automobile Sales')
plt.title('Relationship between Average Vehicle Price and Sales during
Recessions')
plt.show()
```



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*

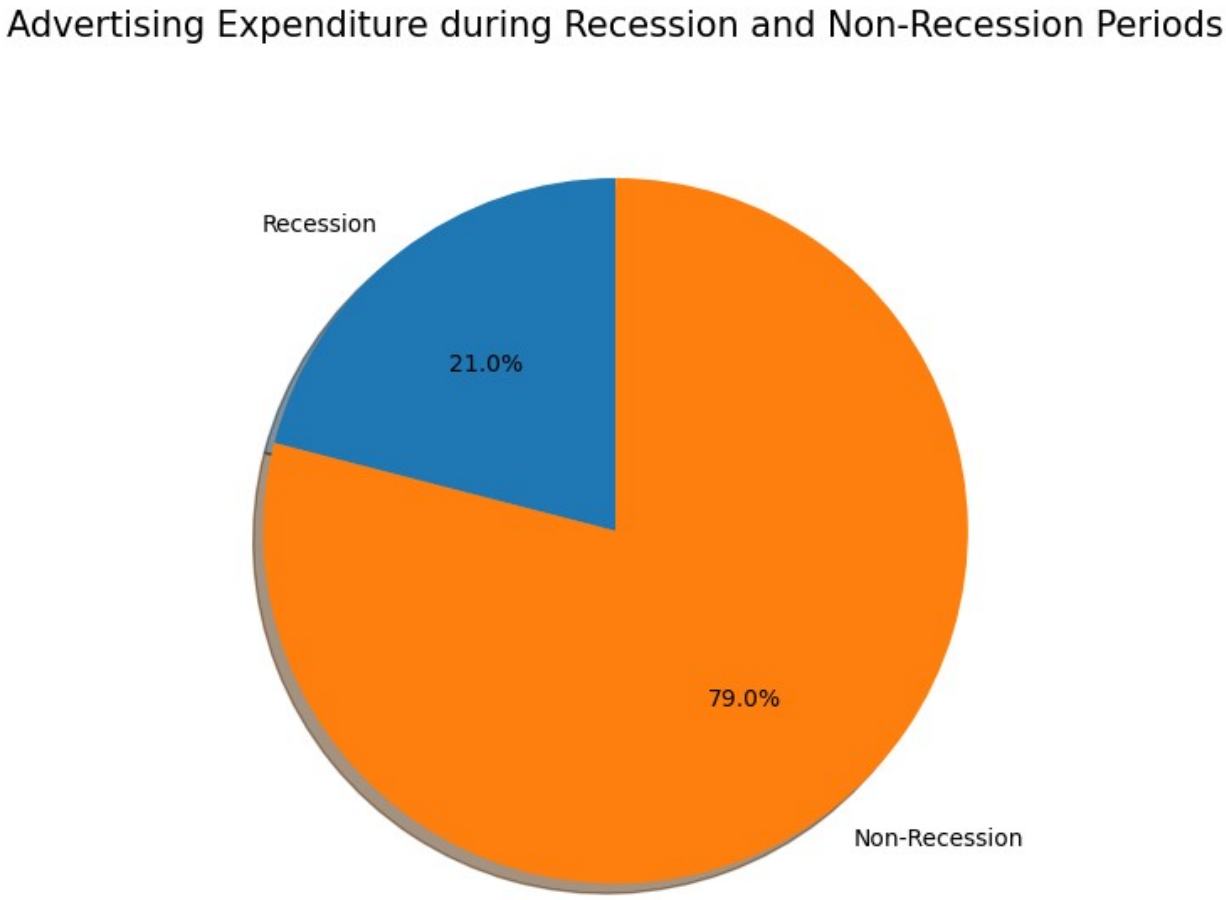
TASK 1.7: Create a pie chart to display the portion of advertising expenditure of XYZAutomotives during recession and non-recession periods. How did the advertising expenditure of XYZAutomotives change during recession and non-recession periods?

```
#Create dataframes
recession_data = df[df['Recession'] == 1]
nonrecession_data = df[df['Recession'] == 0]
nonrecession_data.head()
```

```
          Date  Year Month  Recession  Consumer_Confidence  \
113  2/28/1983  1983   Feb          0               126.06
114  3/31/1983  1983   Mar          0                92.67
115  4/30/1983  1983   Apr          0                96.39
116  5/31/1983  1983   May          0                91.81
117  6/30/1983  1983   Jun          0               106.16
```

```
     Seasonality_Weight      Price  Advertising_Expenditure
Competition  \
113                0.75  15201.649                     4621
4
114                1.50  18359.070                     3038
8
115                1.00  25984.306                     4490
8
116                1.50  28692.333                     4677
8
117                0.75  25856.841                     2118
5

        GDP  Growth_Rate  unemployment_rate  Automobile_Sales  \
113  12.819    -1.119666                2.4            3799.1
114  12.945     0.009733                1.5            3533.5
115  25.206     0.486432                2.0            3240.4
116  14.630    -0.722898                2.3             908.1
117  18.765     0.220357                2.3            4769.8

        Vehicle_Type        City
113           Sports     Georgia
114  Smallfamiliycar  California
115    Supperminicar    Illinois
116  Smallfamiliycar  California
117  Mediumfamilycar    Illinois
```

```
#Calculate advertising expenditure for both dataframes:
recession_data_advertising =
recession_data['Advertising_Expenditure'].sum()
nonrecession_data_advertising =
nonrecession_data['Advertising_Expenditure'].sum()

#Construct pie chart:
plt.figure(figsize=(8, 6))

labels = ['Recession', 'Non-Recession']
sizes = [recession_data_advertising, nonrecession_data_advertising]
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90,
shadow=True)

plt.title('Advertising Expenditure during Recession and Non-Recession
Periods', y=1.12, fontsize = 15)

plt.axis('equal')

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:-

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*

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?

```
#Create dataframe with recession periods:
recession_data = df[df['Recession'] == 1]
recession_data.head()
```

```
        Date  Year Month  Recession  Consumer_Confidence
Seasonality_Weight  \
0  1/31/1980  1980   Jan          1               108.24
0.50
```

```
1  2/29/1980  1980   Feb          1                98.75
0.75
2  3/31/1980  1980   Mar          1               107.48
0.20
3  4/30/1980  1980   Apr          1               115.01
1.00
4  5/31/1980  1980   May          1                98.72
0.20

       Price  Advertising_Expenditure  Competition     GDP
Growth_Rate  \
0  27483.571                     1558            7  60.223
0.010000
1  24308.678                     3048            4  45.986    -
0.309594
2  28238.443                     3137            3  35.141    -
0.308614
3  32615.149                     1653            7  45.673
0.230596
4  23829.233                     1319            4  52.997
0.138197

   unemployment_rate  Automobile_Sales     Vehicle_Type        City
0                5.4             456.0    Supperminicar     Georgia
1                4.8             555.9    Supperminicar    New York
2                3.4             620.0  Mediumfamilycar    New York
3                4.2             702.8    Supperminicar    Illinois
4                5.3             770.4  Smallfamiliycar  California
```

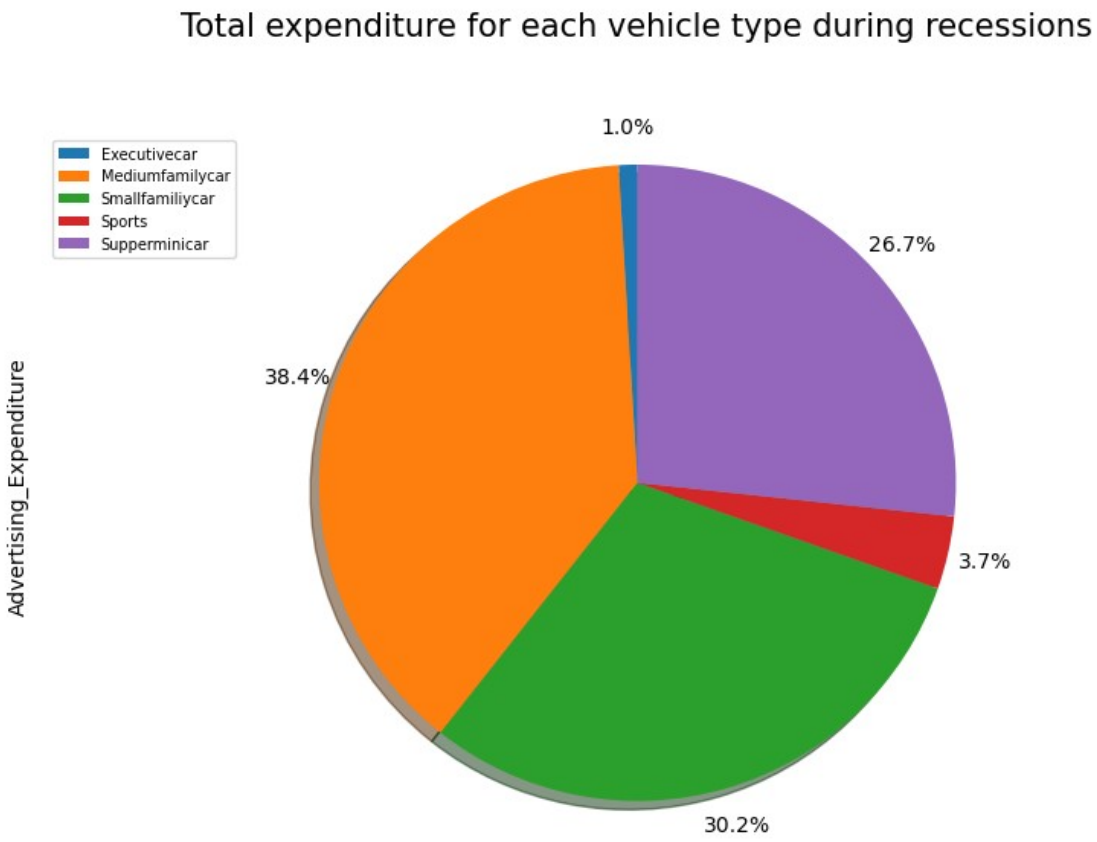
```
TAE_vehicle_type = recession_data.groupby('Vehicle_Type',
axis=0).sum()

print(type(recession_data.groupby('Vehicle_Type', axis=0)))

TAE_vehicle_type.head()
```

```
<ipython-input-50-e85340b7c368>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.sum is deprecated. In a future
version, numeric_only will default to False. Either specify
numeric_only or select only columns which should be valid for the
function.
  TAE_vehicle_type = recession_data.groupby('Vehicle_Type',
axis=0).sum()

<class 'pandas.core.groupby.generic.DataFrameGroupBy'>

                  Year  Recession  Consumer_Confidence
Seasonality_Weight  \
Vehicle_Type

Executivecar      1991          1                92.57
```

```
1.00
Mediumfamilycar  83752          42               4304.72
15.96
Smallfamiliycar  71761          36               3637.77
10.14
Sports            6002           3                287.88
0.14
Supperminicar    61831          31               3086.84
15.01

                        Price  Advertising_Expenditure  Competition
GDP  \
Vehicle_Type

Executivecar      32749.672                     3243             6
36.537
Mediumfamilycar  980455.366                   130612           261
1456.428
Smallfamiliycar  911602.196                   102759           209
1324.108
Sports            77216.504                    12568            17
130.522
Supperminicar    757359.957                    90728           187
1313.825

                  Growth_Rate  unemployment_rate  Automobile_Sales
Vehicle_Type
Executivecar        -0.415305                2.9             564.0
Mediumfamilycar    -13.891808              158.8           28343.6
Smallfamiliycar    -12.331361              162.1           23426.4
Sports               0.678674               14.3             511.0
Supperminicar        2.692273              129.4           20437.4
```

```
#Creating pie chart:
colors_list = ['red', 'yellow', 'green', 'blue', 'orange', 'purple']
explode_list = [0.1, 0, 0, 0, 0.1, 0.1] # ratio for each continent
with which to offset each wedge.

TAE_vehicle_type['Advertising_Expenditure'].plot(kind='pie',
                            figsize=(10, 6),
                            autopct='%1.1f%%',
                            startangle=90,
                            shadow=True,
                            labels=None,         # turn off labels on
pie chart
                            pctdistance=1.12,    # the ratio between
the center of each pie slice and the start of the text generated by
autopct
                            #colors=colors_list,  # add custom colors
                            #explode=explode_list # 'explode' lowest 3
```

```
continents
                                )

# scale the title up by 12% to match pctdistance
plt.title('Total expenditure for each vehicle type during recessions',
y=1.12, fontsize = 15)

plt.axis('equal')

# add legend
plt.legend(labels=TAE_vehicle_type.index, loc='upper left',
fontsize=7)

plt.show()
```



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*

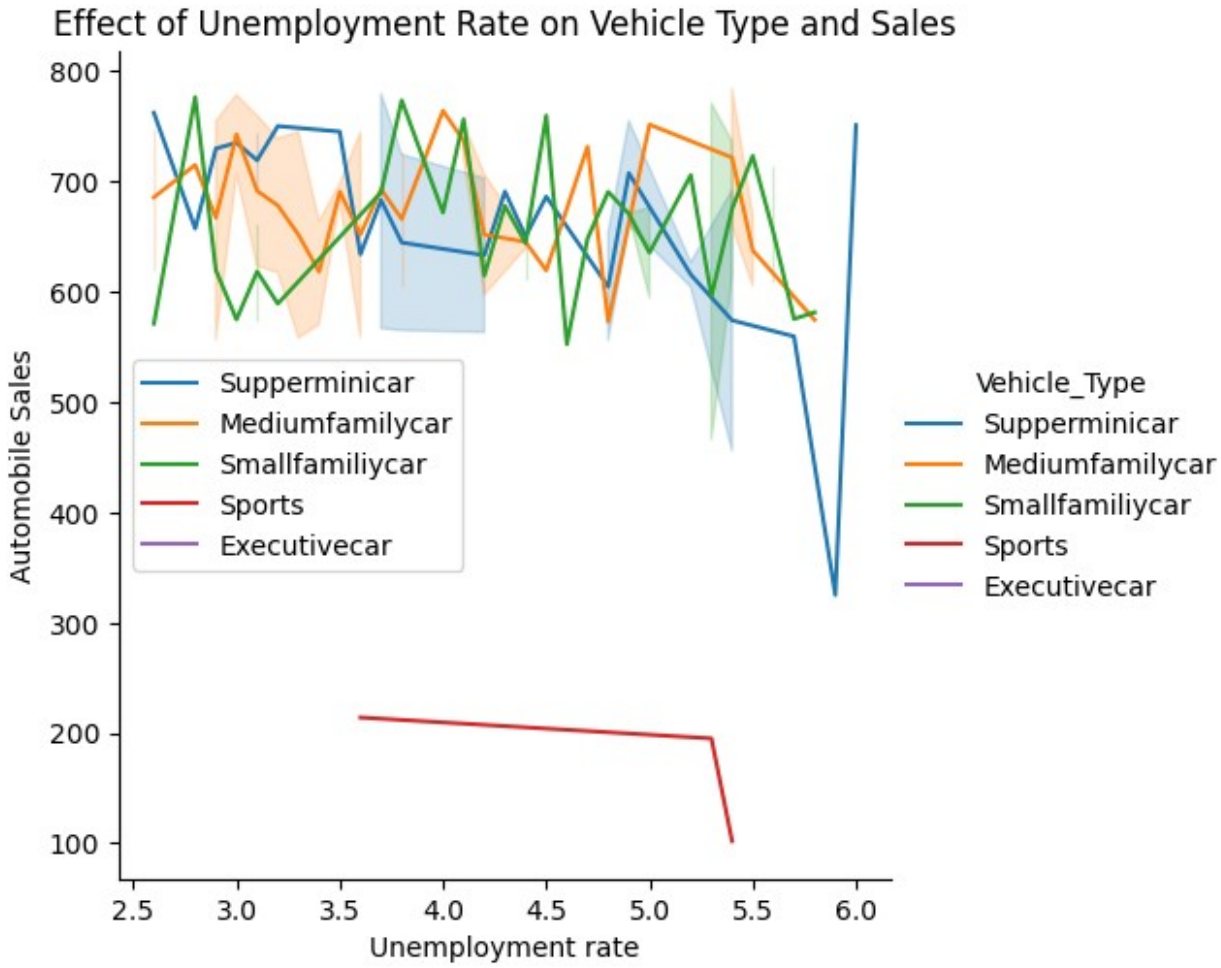
TASK 1.9: Develop a lineplot to analyse the effect of the unemployment rate on vehicle type and sales during the Recession Period. Analyze the effect of the unemployment rate on vehicle type and sales during the Recession Period #### You can create a lineplot and title the plot as 'Effect of Unemployment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

```
#Create dataframe with recession period:
recession_data = df[df['Recession'] == 1]
recession_data.head()
```

```
        Date  Year Month  Recession  Consumer_Confidence
Seasonality_Weight  \
0  1/31/1980  1980   Jan          1               108.24
0.50
1  2/29/1980  1980   Feb          1                98.75
0.75
2  3/31/1980  1980   Mar          1               107.48
0.20
3  4/30/1980  1980   Apr          1               115.01
1.00
4  5/31/1980  1980   May          1                98.72
0.20

       Price  Advertising_Expenditure  Competition     GDP
Growth_Rate  \
0  27483.571                     1558            7  60.223
0.010000
1  24308.678                     3048            4  45.986    -
0.309594
2  28238.443                     3137            3  35.141    -
0.308614
3  32615.149                     1653            7  45.673
0.230596
4  23829.233                     1319            4  52.997
0.138197

   unemployment_rate  Automobile_Sales     Vehicle_Type        City
0                5.4             456.0    Supperminicar     Georgia
1                4.8             555.9    Supperminicar    New York
2                3.4             620.0  Mediumfamilycar    New York
3                4.2             702.8    Supperminicar    Illinois
4                5.3             770.4  Smallfamiliycar  California
```

```
plt.figure(figsize=(12, 6))
sns.relplot(x='unemployment_rate', y='Automobile_Sales',
data=recession_data, kind='line', hue='Vehicle_Type')
plt.xlabel('Unemployment rate')
plt.ylabel('Automobile Sales')
plt.title('Effect of Unemployment Rate on Vehicle Type and Sales')
plt.legend()
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



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

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*

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

```
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"
print('GeoJSON file downloaded!')
```

```
GeoJSON file downloaded!
```

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

```
world_geo = r'https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-
SkillsNetwork/Data%20Files/us-states.json'

# create a plain world map
world_map = folium.Map(location=[0, 0], zoom_start=2)
world_map
```

```
<folium.folium.Map at 0x6d83580>
```

```
#Create dataframe with recession period:
recession_data = df[df['Recession'] == 1]
recession_data.head()
```

```
        Date  Year Month  Recession  Consumer_Confidence
Seasonality_Weight  \
0  1/31/1980  1980   Jan          1               108.24
0.50
1  2/29/1980  1980   Feb          1                98.75
0.75
2  3/31/1980  1980   Mar          1               107.48
0.20
3  4/30/1980  1980   Apr          1               115.01
1.00
4  5/31/1980  1980   May          1                98.72
0.20

       Price  Advertising_Expenditure  Competition     GDP
Growth_Rate  \
0  27483.571                     1558            7  60.223
0.010000
1  24308.678                     3048            4  45.986    -
0.309594
2  28238.443                     3137            3  35.141    -
```

```
0.308614
3  32615.149                      1653             7  45.673
0.230596
4  23829.233                      1319             4  52.997
0.138197
```

```
   unemployment_rate  Automobile_Sales     Vehicle_Type        City
0                5.4             456.0    Supperminicar     Georgia
1                4.8             555.9    Supperminicar    New York
2                3.4             620.0  Mediumfamilycar    New York
3                4.2             702.8    Supperminicar    Illinois
4                5.3             770.4  Smallfamiliycar  California
```

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

```
         City  Automobile_Sales
0  California           18982.6
1     Georgia           17470.5
2    Illinois           16874.7
3    New York           19954.6
```

```
# 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
```

```
<folium.folium.Map at 0x6e2cb90>
```

Congratulations! You have completed the lab

Authors

Dr. Pooja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2024-01-05	0.2.1	Sowmyaa Gurusamy	Updated the lab instructions
2023-06-17	0.2	Pooja	Initial Lab Creation
2023-05-01	0.1	Shengkai	Create Lab Template