

Create visualizations using Matplotlib, 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

1. Objectives
2. Setup
 - A. Installing Required Libraries
 - B. Importing Required Libraries
3. Scenario
 - A. Data Description
4. Importing data
5. Creating Visualizations for Data Analysis

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 Anaconda), you will need to install these libraries by removing the `#` sign before `%pip` in the code cell below.

```
In [ ]: # 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"
```

```
In [3]: %pip install seaborn
        %pip install folium
```

Importing Required Libraries

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

```
In [3]: 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](#)

► [Click here for python solution](#)

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 period; 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, Smallfamilycar, 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.

```
In [ ]: from js import fetch
import io
import pandas as pd

URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDDeveloperSkillsNetwork-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)
print('Data downloaded and read into a dataframe!')
```

```
In [5]: df = pd.read_csv('historical_automobile_sales.csv')
df.head()
```

```
Out[5]:
```

	Date	Year	Month	Recession	Consumer_Confidence	Seasonality_Weight	Price	Advertising_Expenditure	Competition	GDP	Growth_Rate	unemployment
0	1/31/1980	1980	Jan	1	108.24	0.50	27483.571	1558	7	60.223	0.010000	
1	2/29/1980	1980	Feb	1	98.75	0.75	24308.678	3048	4	45.986	-0.309594	
2	3/31/1980	1980	Mar	1	107.48	0.20	28238.443	3137	3	35.141	-0.308614	
3	4/30/1980	1980	Apr	1	115.01	1.00	32615.149	1653	7	45.673	0.230596	
4	5/31/1980	1980	May	1	98.72	0.20	23829.233	1319	4	52.997	0.138197	

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

```
Out[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')
```

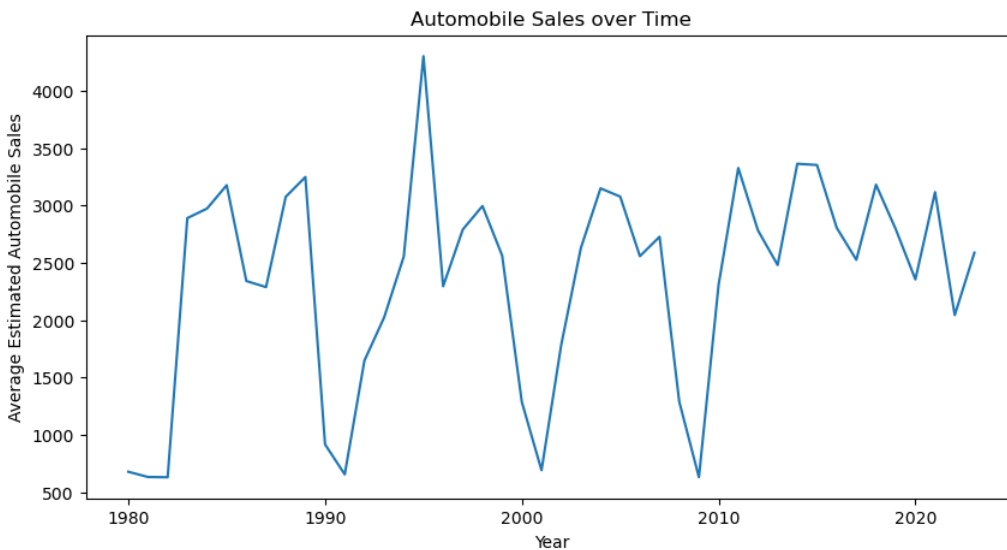
Creating Visualizations for Data Analysis

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

► [Click here for a hint](#)

```
In [7]: #perform group by on year for average sales
df_new = df.groupby('Year')['Automobile_Sales'].mean().to_frame()
df_new.reset_index(inplace=True)

plt.figure(figsize=(10, 5))
sns.lineplot(data=df_new, x="Year", y="Automobile_Sales")
plt.xlabel('Year')
plt.ylabel('Average Estimated Automobile Sales')
plt.title('Automobile Sales over Time')
plt.show()
```



► [Click here for a solution template](#)

Include the following on the plot

ticks on x- axis with all the years, to identify the years of recession

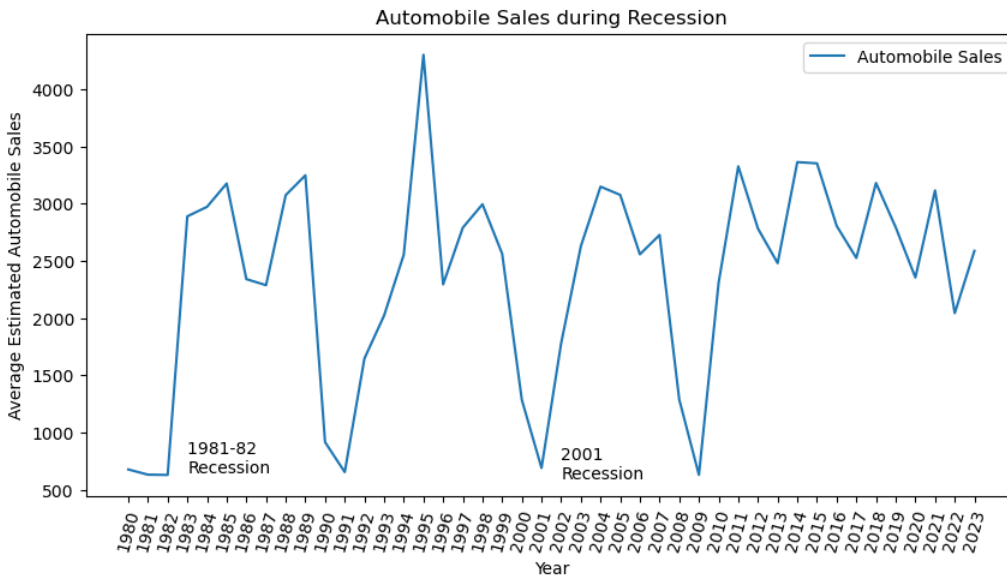
annotation for at least two years of recession

Title as Automobile Sales during Recession

► [Click here for a hint](#)

```
In [8]: #perform group by on year for average sales
df_new = df.groupby('Year')['Automobile_Sales'].mean().to_frame()
df_new.reset_index(inplace=True)

plt.figure(figsize=(10, 5))
sns.lineplot(data=df_new, x="Year", y="Automobile_Sales")
plt.xticks(list(range(1980, 2024)), rotation = 75)
plt.xlabel('Year')
plt.ylabel('Average Estimated Automobile Sales')
plt.title('Automobile Sales during Recession')
plt.text(1983, 650, '1981-82 \nRecession')
plt.text(2002, 600, '2001 \nRecession')
plt.legend(['Automobile Sales'])
plt.show()
```



► [Click here for Solution template](#)

Save this plot as "Line_Plot_1.png"

Hint: You can right click 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](#)

```
In [9]: # filter recession records
df_rec = df[df['Recession'] == 1]

#perform group by on year and vehicle type for average sales
df_cat = df_rec.groupby(["Year", "Vehicle_Type"], as_index=False)["Automobile_Sales"].mean()

#perform group by on vehicle type for normalized sales
df_cat['Normalized_Sales'] = df_cat.groupby('Vehicle_Type')['Automobile_Sales'].transform(lambda x: (x - x.min()) / (x.max() - x.min()))

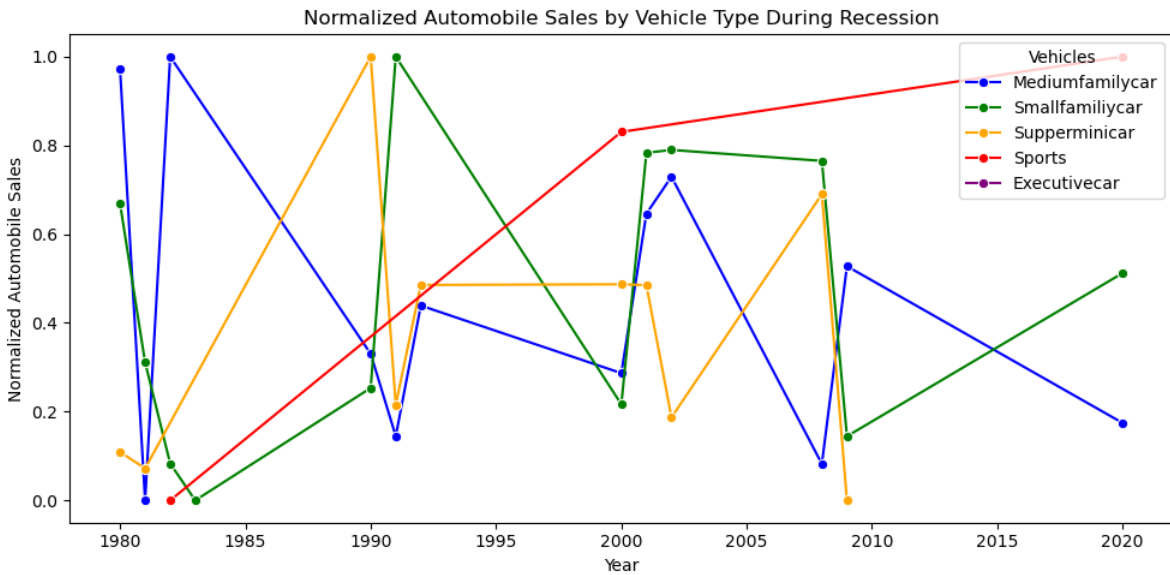
# set year as index
df_cat.set_index("Year", inplace=True)

# filter year, vehicle types and set color
years = df_cat.index.unique()
vehicles = df_cat["Vehicle_Type"].unique()
colors = ['blue', 'green', 'orange', 'red', 'purple']

# set fig size
plt.figure(figsize=(10, 5))

# Loop over vehicles and perform Line plot
for i, vehicle in enumerate(vehicles):
    df_v = df_cat[df_cat["Vehicle_Type"] == vehicle]
    sns.lineplot(data=df_v, x=df_v.index, y=df_v["Normalized_Sales"], marker='o', color=colors[i], label=vehicle)

plt.ylabel("Normalized Automobile Sales")
plt.xlabel("Year")
plt.title("Normalized Automobile Sales by Vehicle Type During Recession")
plt.legend(title='Vehicles', loc='upper right')
plt.tight_layout()
plt.show()
```



► [Click here for Solution template](#)

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

► Inference

[Save this plot as "Line_Plot_2.png"](#)

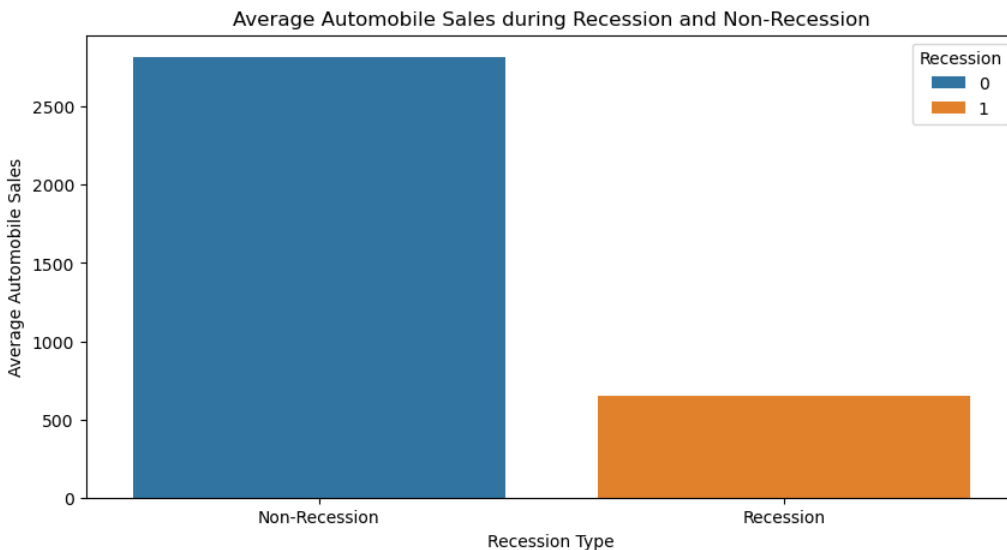
Hint: You can right click on the plot and then click on "Save image as" option to save it on your local machine

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.**

► [Click here for a hint](#)

```
In [10]: # perform group by on recession for average automobile sales
new_df = df.groupby('Recession')['Automobile_Sales'].mean().reset_index()

plt.figure(figsize=(10,5))
sns.barplot(x='Recession', y='Automobile_Sales', hue='Recession', data=new_df)
plt.xlabel('Recession Type')
plt.ylabel('Average Automobile Sales')
plt.title('Average Automobile Sales during Recession and Non-Recession')
plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
plt.show()
```



► [Click here for Solution template](#)

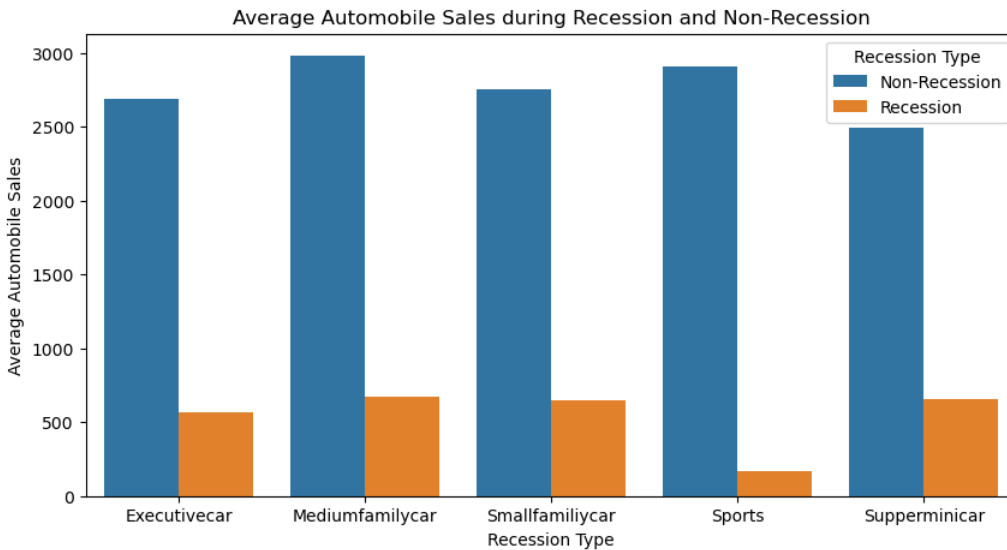
Now you want to compare the sales of different vehicle types during a recession and a non-recession period

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

► [Click here for a hint](#)

```
In [11]: # perform group by on recession and vehicle type for average automobile sales
new_df = df.groupby(['Recession', 'Vehicle_Type'])['Automobile_Sales'].mean().reset_index()
new_df["Recession"] = new_df["Recession"].replace([0,1],['Non-Recession', 'Recession'])

plt.figure(figsize=(10,5))
sns.barplot(x='Vehicle_Type', y='Automobile_Sales', hue='Recession', data=new_df)
plt.xlabel('Recession Type')
plt.ylabel('Average Automobile Sales')
plt.title('Average Automobile Sales during Recession and Non-Recession')
plt.legend(title='Recession Type')
plt.show()
```



► [Click here for Solution template](#)

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

► Inference

Save this plot as "Bar_Chart.png"

Hint: You can right click 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 comparison.

```
In [33]: #filter recession and non-recession data
rec_data = df[df['Recession'] == 1]
non_rec_data = df[df['Recession'] == 0]

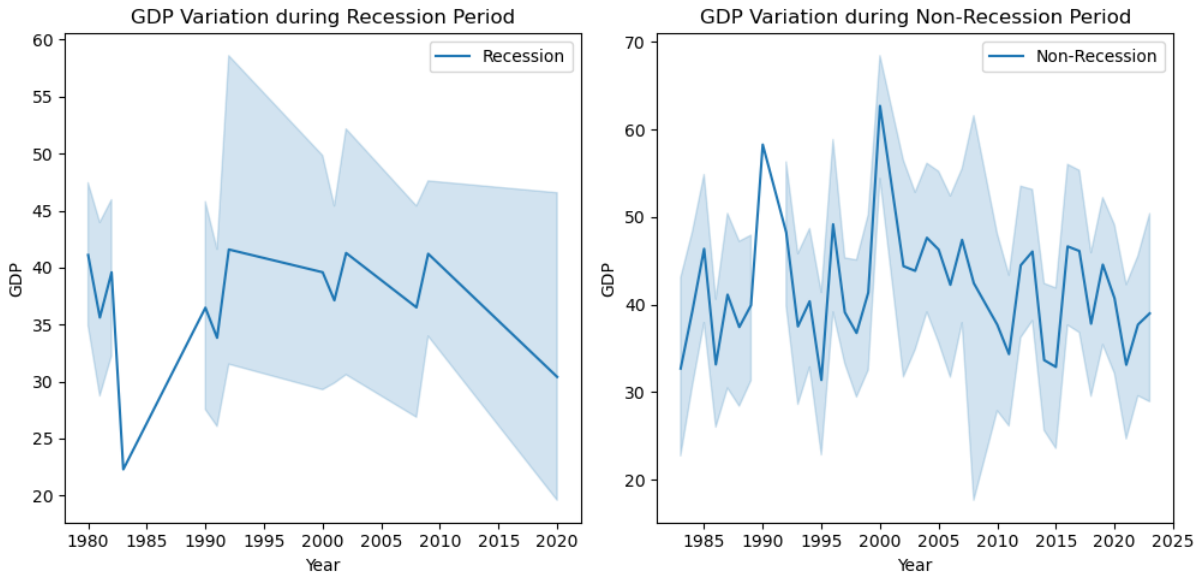
# create fig with size
fig=plt.figure(figsize=(10, 5))

# create subplot axes
ax0 = fig.add_subplot(1, 2, 1)
ax1 = fig.add_subplot(1, 2, 2)

#subplot 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')
```

```
#subplot 1
sns.lineplot(x='Year', y='GDP', data=non_rec_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()
```



► [Click here for Solution template](#)

Inference

From this plot, it is evident that during recession, the GDP of the country was in a low range, might have affected the overall sales of the company

Save this plot as "Subplot.png"

Hint: You can right click 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

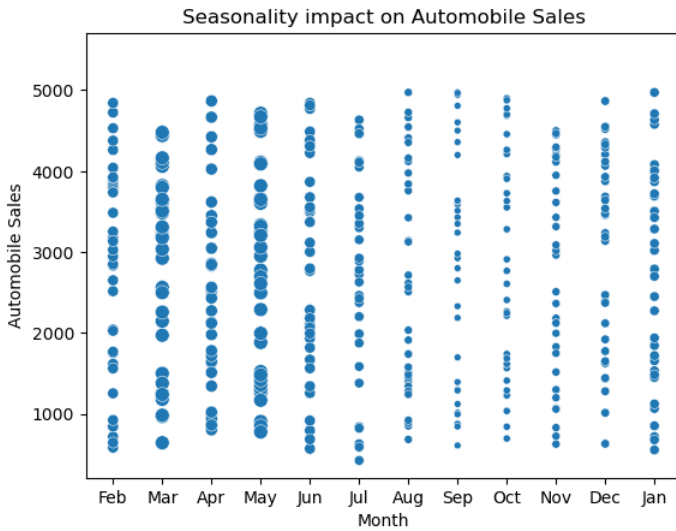
Title this plot as 'Seasonality impact on Automobile Sales'

► [Click here for a hint](#)

```
In [32]: non_rec_data = df[df['Recession'] == 0]

sns.scatterplot(data=non_rec_data, x='Month', y='Automobile_Sales', size=non_rec_data['Seasonality_Weight'], legend=False)

plt.xlabel('Month')
plt.ylabel('Automobile Sales')
plt.ylim(non_rec_data["Automobile_Sales"].min()*0.5, non_rec_data["Automobile_Sales"].median() * 2)
plt.title('Seasonality impact on Automobile Sales')
plt.show()
```



► [Click here for Solution template](#)

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 click 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.

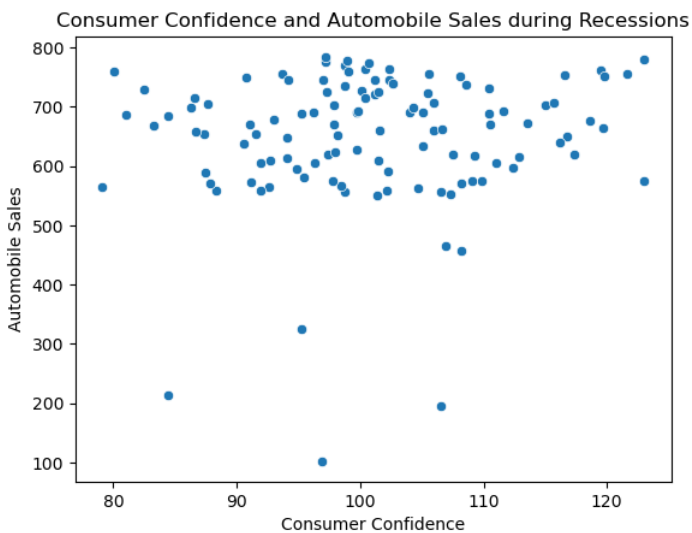
From the data, develop a scatter plot to identify if there a correlation between consumer confidence and automobile sales during recession period?

Title this plot as 'Consumer Confidence and Automobile Sales during Recessions'

► [Click here for a hint](#)

```
In [37]: df_rec = df[df["Recession"]==1]

sns.scatterplot(data=df_rec,x="Consumer_Confidence",y="Automobile_Sales")
plt.xlabel('Consumer Confidence')
plt.ylabel('Automobile Sales')
plt.title('Consumer Confidence and Automobile Sales during Recessions')
plt.show()
```



► [Click here for Solution template](#)

How does the average vehicle price relate to the sales volume during recessions?

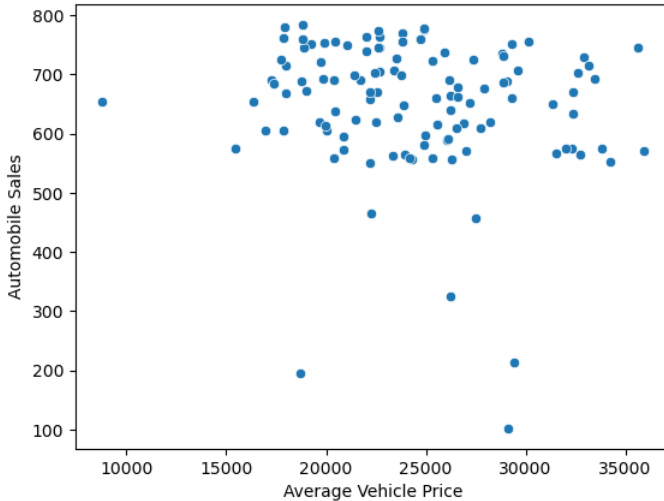
Plot another scatter plot and title it as 'Relationship between Average Vehicle Price and Sales during Recessions'

```
In [38]: rec_data = df[df['Recession'] == 1]

sns.scatterplot(data=rec_data, x='Price', y='Automobile_Sales')

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

Relationship between Average Vehicle Price and Sales during Recessions



► [Click here for Solution template](#)

Inference

There is not much relation!

Save this plot as "Scatter.png"

Hint: You can right click 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.

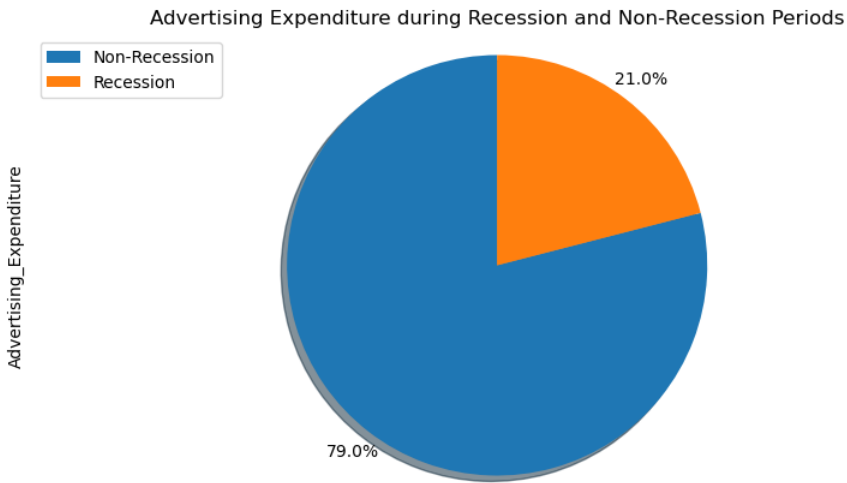
How did the advertising expenditure of XYZAutomotives change during recession and non-recession periods?

► [Click here for a hint](#)

```
In [47]: df_exp = df.groupby("Recession")["Advertising_Expenditure"].sum().reset_index()

df_exp['Advertising_Expenditure'].plot(kind='pie',
                                       figsize=(10, 5),
                                       autopct='%1.1f%%',
                                       startangle=90,
                                       shadow=True,
                                       labels=None,
                                       pctdistance=1.12
                                       )

plt.title('Advertising Expenditure during Recession and Non-Recession Periods')
plt.axis('equal')
plt.legend(labels=['Non-Recession', 'Recession'], loc='upper left')
plt.show()
```



► [Click here for Solution template](#)

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

► Inference

Save this plot as "Pie_1.png"

Hint: You can right click 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?

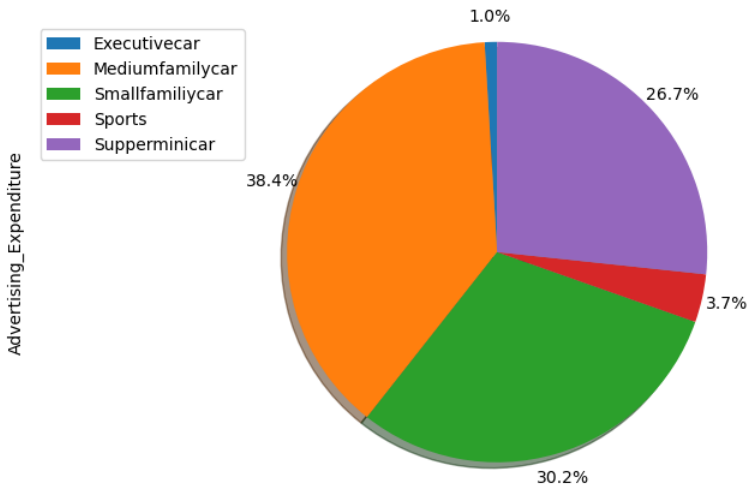
► [Click here for a hint](#)

```
In [52]: df_rec = df[df["Recession"]==1]
df_exp = df_rec.groupby("Vehicle_Type")["Advertising_Expenditure"].sum().reset_index()

df_exp['Advertising_Expenditure'].plot(kind='pie',
                                       figsize=(10, 5),
                                       autopct='%1.1f%%',
                                       startangle=90,
                                       shadow=True,
                                       labels=None,
                                       pctdistance=1.12
                                       )

plt.title('Share of Each Vehicle Type in Total Expenditure during Recessions',y=1.2)
plt.axis('equal')
plt.legend(labels=df_exp["Vehicle_Type"], loc='upper left')
plt.show()
```

Share of Each Vehicle Type in Total Expenditure during Recessions



► [Click here for Solution template](#)

Inference

During recession the advertisements were mostly focused on low price range vehicle. A wise decision!

[Save this plot as "Pie_2.png"](#)

Hint: You can right click 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

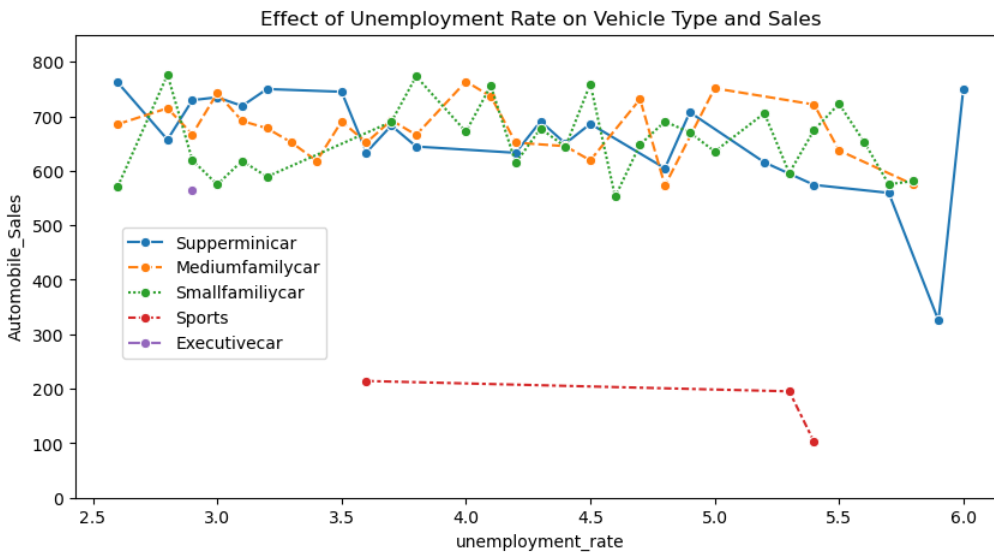
You can create a lineplot and title the plot as 'Effect of Unemployment Rate on Vehicle Type and Sales'

► [Click here for a hint](#)

```
In [62]: df_rec = df[df["Recession"]==1]

plt.figure(figsize=(10, 5))
sns.lineplot(data=df_rec, x='unemployment_rate', y='Automobile_Sales', hue='Vehicle_Type', style='Vehicle_Type', markers='o', err_style=None)
plt.ylim(0,850)
plt.legend(loc=(0.05,.3))
plt.title('Effect of Unemployment Rate on Vehicle Type and Sales')
plt.show()
```

C:\Users\aalib\AppData\Local\Temp\ipykernel_176736\2679092748.py:4: UserWarning:
The markers list has fewer values (1) than needed (5) and will cycle, which may produce an uninterpretable plot.
sns.lineplot(data=df_rec, x='unemployment_rate', y='Automobile_Sales', hue='Vehicle_Type', style='Vehicle_Type', markers='o', err_style=None)



► [Click here for Solution template](#)

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

► Inference

Save this plot as "line_plot_3.png"

Hint: You can right click 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 highest sales region/offices of the company during recession period

```
In [65]: import requests

async def download(url, filename):
    response = requests.get(url)
    if response.status_code == 200:
        with open(filename, "wb") as f:
            f.write(response.content)

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 dataset 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

```
In [66]: recession_data = df[df['Recession'] == 1]

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

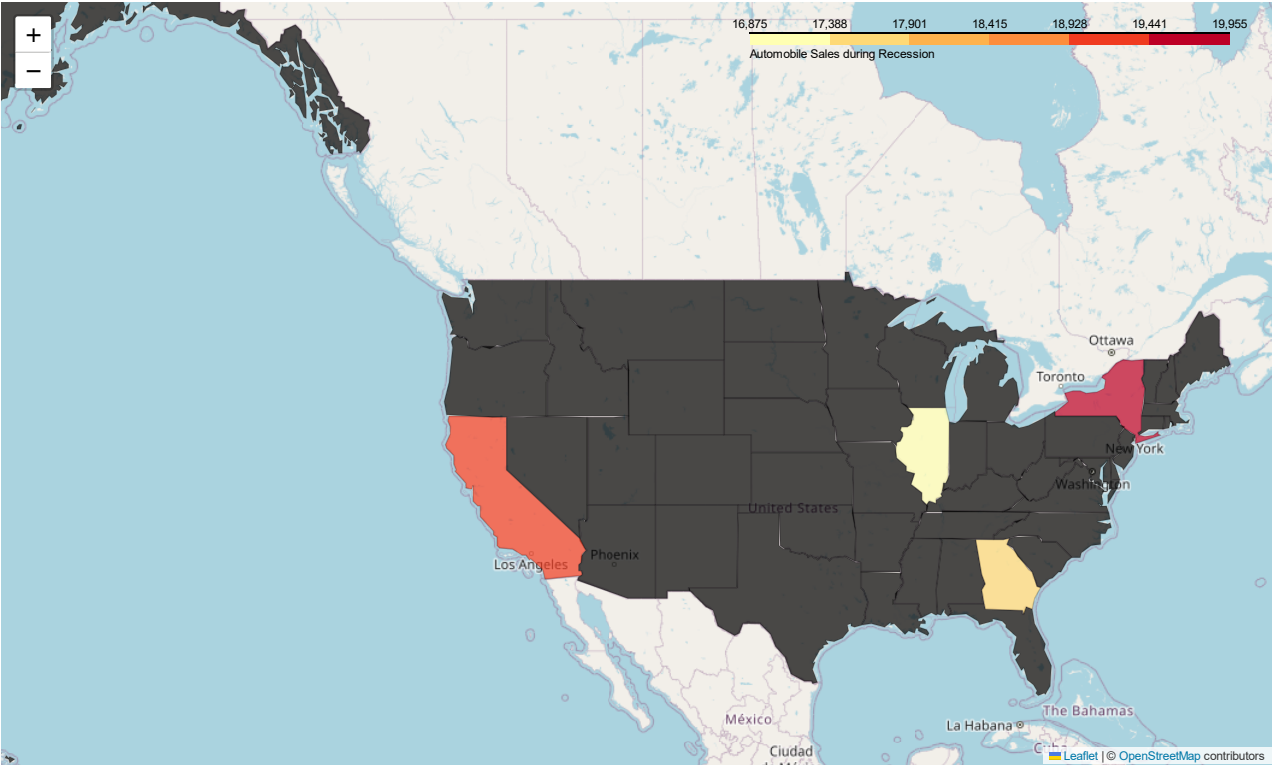
# Create a base map centered on the United States
us_map = folium.Map(location=[37.0902, -95.7129], zoom_start=4)

# Create a choropleth layer using Folium
choropleth = folium.Choropleth(
    geo_data='us-states.json',
    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(us_map)

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

# Display the map
us_map
```

Out[66]:



► Click for Solution

Congratulations! You have completed the lab

Authors

[Dr. Pooja](#)

Copyright © 2023 IBM Corporation. All rights reserved.