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:

- `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.

In [ ]:
```
# All Libraries required for this lab are listed below. The libraries pre-instal
# %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 [1]:
```
%pip install seaborn
%pip install folium
```

```
ib\site-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=0.25 in c:\users\91939\anaconda3\lib\site-
packages (from seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\91939\anaconda
3\lib\site-packages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\91939\anaconda3\lib\s
ite-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\91939\anaconda3\lib\site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\91939\anaconda3\lib
\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\91939\anaconda3\lib
\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\91939\anaconda3\lib\si
te-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\91939\anaconda3\lib\site
-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\91939\anaconda3\lib\s
ite-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\91939\anaconda3\l
ib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\91939\anaconda3\lib\site-
packages (from pandas>=0.25->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\91939\anaconda3\lib\sit
e-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\91939\anaconda3\lib\site-pack
ages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
Collecting foliumNote: you may need to restart the kernel to use updated package
s.

  Downloading folium-0.16.0-py2.py3-none-any.whl.metadata (3.6 kB)
Collecting branca>=0.6.0 (from folium)
  Downloading branca-0.7.2-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: jinja2>=2.9 in c:\users\91939\anaconda3\lib\site-p
ackages (from folium) (3.1.3)
Requirement already satisfied: numpy in c:\users\91939\anaconda3\lib\site-package
s (from folium) (1.26.4)
Requirement already satisfied: requests in c:\users\91939\anaconda3\lib\site-pack
ages (from folium) (2.31.0)
Requirement already satisfied: xyzservices in c:\users\91939\anaconda3\lib\site-p
ackages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\91939\anaconda3\lib\si
te-packages (from jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\91939\anacond
a3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\91939\anaconda3\lib\site-
packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\91939\anaconda3\lib
\site-packages (from requests->folium) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\91939\anaconda3\lib
\site-packages (from requests->folium) (2024.2.2)
Downloading folium-0.16.0-py2.py3-none-any.whl (100 kB)
   ---------------------------------------- 0.0/100.0 kB ? eta -:--:--
   ---- ----------------------------------- 10.2/100.0 kB ? eta -:--:--
   ----------- ---------------------------- 30.7/100.0 kB 325.1 kB/s eta 0:00:01
   -------------- ------------------------- 41.0/100.0 kB 487.6 kB/s eta 0:00:01
```

```
Installing collected packages: branca, folium
Successfully installed branca-0.7.2 folium-0.16.0
```

Importing Required Libraries

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

In [2]:
```
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 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.

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.

In [9]:
```
from pandas.io.parsers.readers import read_csv
import requests
from io import StringIO


URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDev
header = {
    'User-Agent' : 'Mozilla/5.0'
}

resp = requests.get(URL, headers= header)

df = read_csv(StringIO(resp.text))
df
```

1	2/29/1980	1980	Feb	1	98.75	0.75	2
2	3/31/1980	1980	Mar	1	107.48	0.20	2
3	4/30/1980	1980	Apr	1	115.01	1.00	3
4	5/31/1980	1980	May	1	98.72	0.20	2
...	...	...	...	...	...	...	
523	8/31/2023	2023	Aug	0	103.36	0.25	2
524	9/30/2023	2023	Sep	0	101.55	0.07	2
525	10/31/2023	2023	Oct	0	124.66	0.12	1
526	11/30/2023	2023	Nov	0	97.09	0.25	1
527	12/31/2023	2023	Dec	0	95.92	0.34	2

528 rows × 15 columns

In [10]:
```
df.describe()
```

Out[10]:

	Year	Recession	Consumer_Confidence	Seasonality_Weight	Price
count	528.000000	528.000000	528.000000	528.000000	528.000000
mean	2001.500000	0.214015	101.140170	0.575795	24964.991956
std	12.710467	0.410526	10.601154	0.454477	4888.073433
min	1980.000000	0.000000	73.900000	0.000000	8793.663000
25%	1990.750000	0.000000	94.035000	0.250000	21453.300500
50%	2001.500000	0.000000	100.740000	0.500000	25038.691500
75%	2012.250000	0.000000	108.240000	0.750000	28131.684750
max	2023.000000	1.000000	131.670000	1.500000	44263.657000

In [11]:
```
df.columns
```

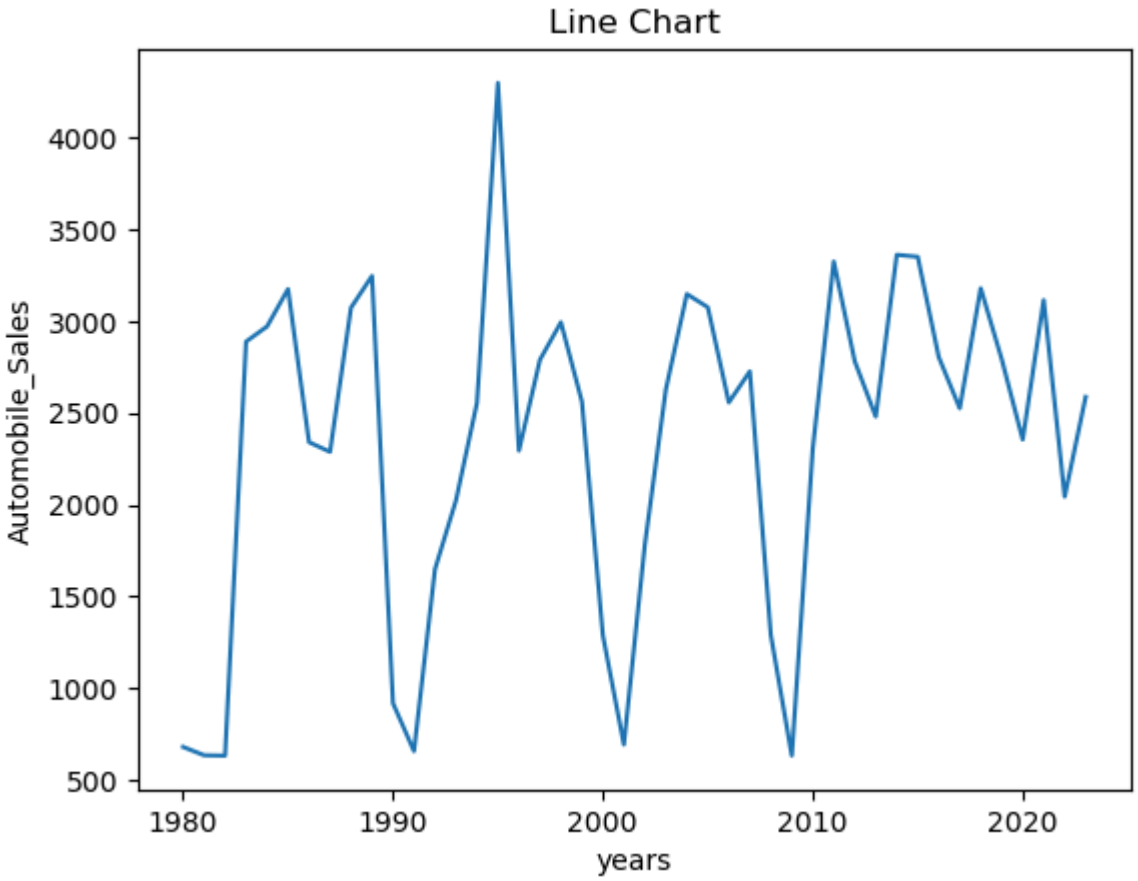
Out[11]:
```
Index(['Date', 'Year', 'Month', 'Recession', 'Consumer_Confidence',
       'Seasonality_Weight', 'Price', 'Advertising_Expenditure', 'Competition',
       'GDP', 'Growth_Rate', 'unemployment_rate', 'Automobile_Sales',
       'Vehicle_Type', 'City'],
      dtype='object')
```

Creating Visualizations for Data Analysis

pandas to show how automobile sales fluctuate from year to year

▶ Click here for a hint

In [12]:
```
grouped_df = df.groupby(by="Year")["Automobile_Sales"].mean().reset_index()

plt.plot(grouped_df["Year"], grouped_df["Automobile_Sales"])

plt.xlabel('years')
plt.ylabel('Automobile_Sales')
plt.title('Line Chart')

plt.show()
```



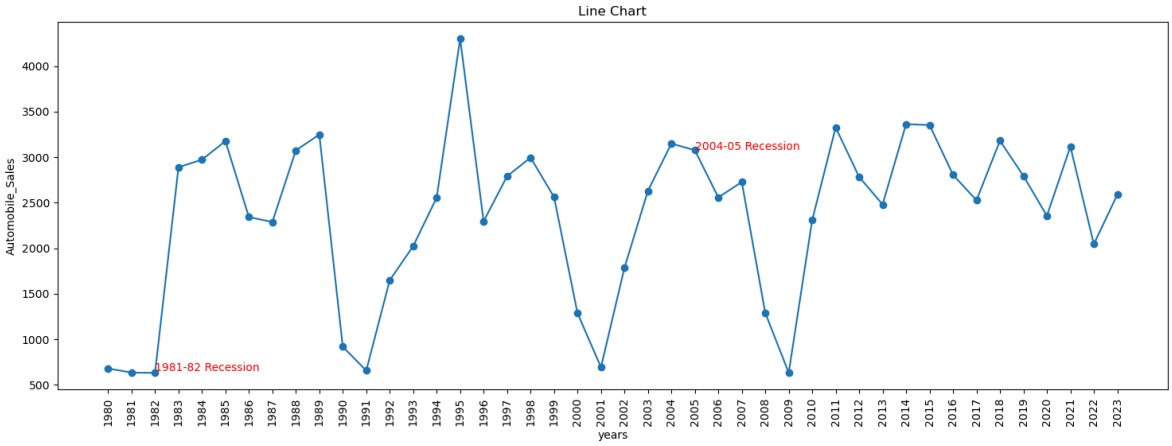
▶ Click here for a solution template

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

```
plt.figure(figsize=(18, 6))
plt.plot(years, grouped_df["Automobile_Sales"],marker='o')

plt.xticks(years,rotation=90)
plt.xlabel('years')
plt.text(1982, 650, '1981-82 Recession', color='r')
plt.text(2005, 3077, '2004-05 Recession', color='r')
plt.ylabel('Automobile_Sales')
plt.title('Line Chart')
plt.show()
```



▶ Click here for Solution template

Save this plot as "Line_Plot_1.png"

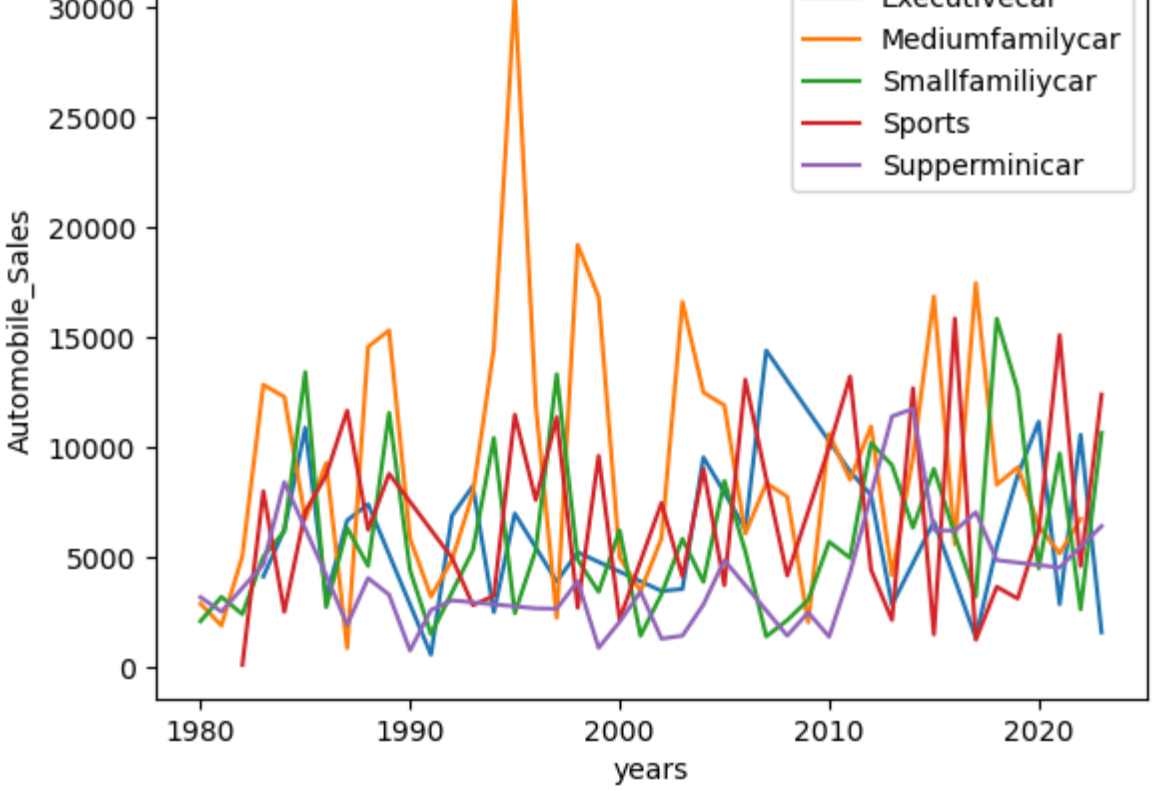
Hint: You can right lick on the plot and then click on "Save image as" option to save it on your local machine

---

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

In [14]:
```
grouped_df2 = df.groupby(by=["Year","Vehicle_Type"],as_index=False)["Automobile_
grouped_df2.set_index('Year',inplace=True)
grouped_df2 = grouped_df2.groupby(['Vehicle_Type'])['Automobile_Sales']

grouped_df2.plot(kind='line')
plt.xlabel('years')
plt.ylabel('Automobile_Sales')
plt.title('Line Chart')
plt.legend()
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:

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.

▶ Inference

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

---

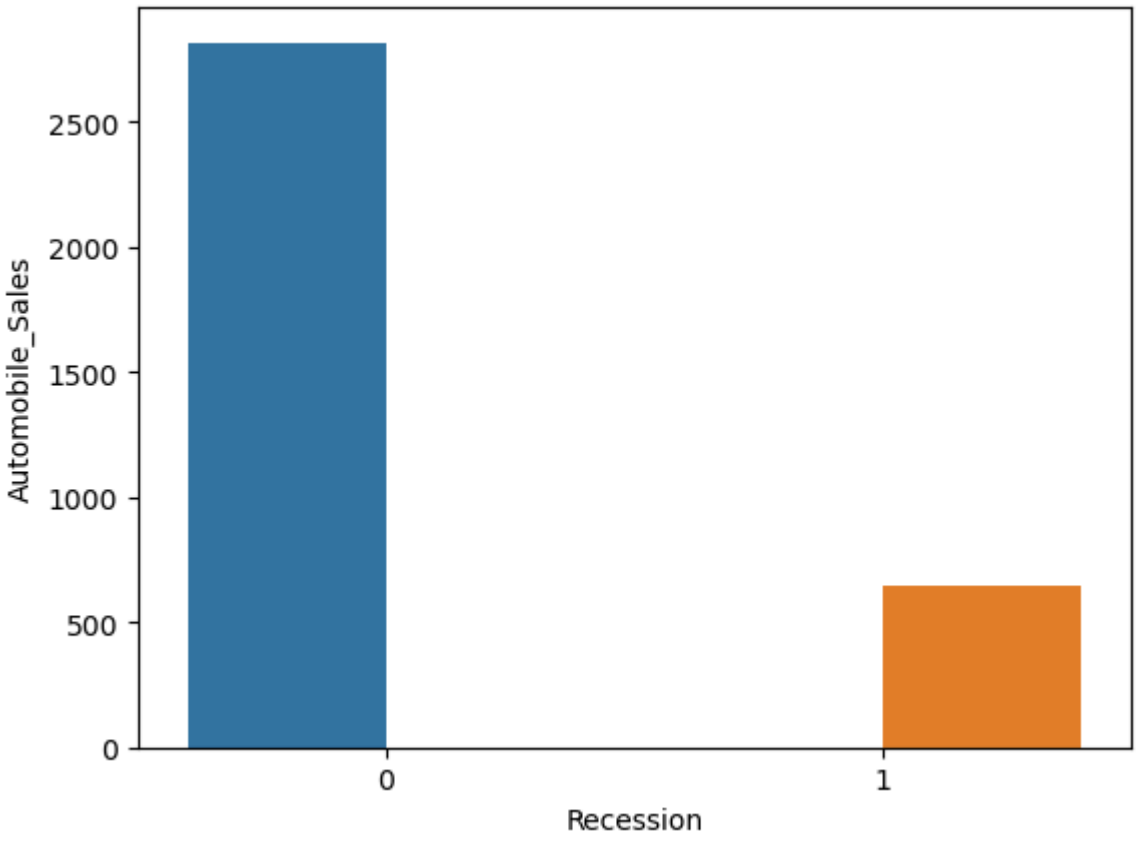
TASK 1.3: Use the functionality of **Seaborn Library** to create a visualization to compare the sales trend per vehicle type for a recession period with a non-recession period.

In [15]:
```
grouped_df3 = df.groupby(by=["Recession"],as_index=False)["Automobile_Sales"].me

sns.barplot(x='Recession', y='Automobile_Sales', hue='Recession',  data=grouped_
plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
plt.show()
```

```
C:\Users\91939\anaconda3\Lib\tokenize.py:529: RuntimeWarning: coroutine 'main' wa
s never awaited
  pseudomatch = _compile(PseudoToken).match(line, pos)
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
```

```
      1 grouped_df3 = df.groupby(by=["Recession"],as_index=False)["Automobile_Sales"].mean().reset_index()
----> 3 sns.barplot(x='Recession', y='Automobile_Sales', hue='Recession',  data=grouped_df3)
      4 plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
      5 plt.show()

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:2763, in barplot(data, x, y, hue, order, hue_order, estimator, errorbar, n_boot, units, seed, orient, color, palette, saturation, width, errcolor, errwidth, capsize, dodge, ci, ax, **kwargs)
   2760 if ax is None:
   2761     ax = plt.gca()
-> 2763 plotter.plot(ax, kwargs)
   2764 return ax

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:1587, in _BarPlotter.plot(self, ax, bar_kws)
   1585 """Make the plot."""
   1586 self.draw_bars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
   1588 if self.orient == "h":
   1589     ax.invert_yaxis()

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:767, in _CategoricalPlotter.annotate_axes(self, ax)
    764     ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
    766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:322, in Axes.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
   (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
    321     """
--> 322     handles, labels, kwargs = mlegend._parse_legend_args([self], *args, **kwargs)
    323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    324     self.legend_._remove_method = self._remove_legend

File ~\anaconda3\Lib\site-packages\matplotlib\legend.py:1361, in _parse_legend_args(axs, handles, labels, *args, **kwargs)
   1357     handles = [handle for handle, label
   1358                in zip(_get_legend_handles(axs, handlers), labels)]
   1360 elif len(args) == 0:  # 0 args: automatically detect labels and handles.
-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
   1362     if not handles:
   1363         log.warning(
   1364             "No artists with labels found to put in legend.  Note that "
   1365             "artists whose label start with an underscore are ignored "
   1366             "when legend() is called with no argument.")
```

```
   1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
   1292         handles.append(handle)
   1293         labels.append(label)

AttributeError: 'numpy.int64' object has no attribute 'startswith'
```



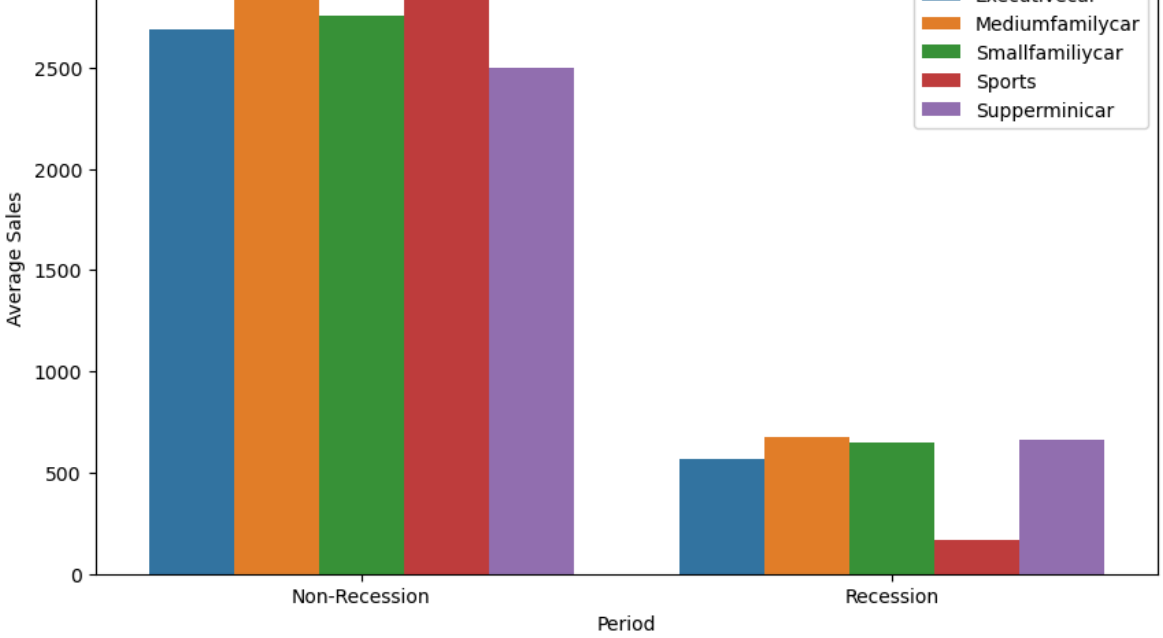
▶ Click here for Solution template

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 [16]:
```
grouped_df3=df.groupby(['Recession','Vehicle_Type'])['Automobile_Sales'].mean().

plt.figure(figsize=(10, 6))
sns.barplot(x='Recession', y='Automobile_Sales', hue='Vehicle_Type', data=groupe
plt.xticks(ticks=[0, 1], labels=['Non-Recession', 'Recession'])
plt.xlabel('Period')
plt.ylabel('Average Sales')
plt.title('Vehicle-Wise Sales during Recession and Non-Recession Period')

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:-

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

▶ Inference

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

---

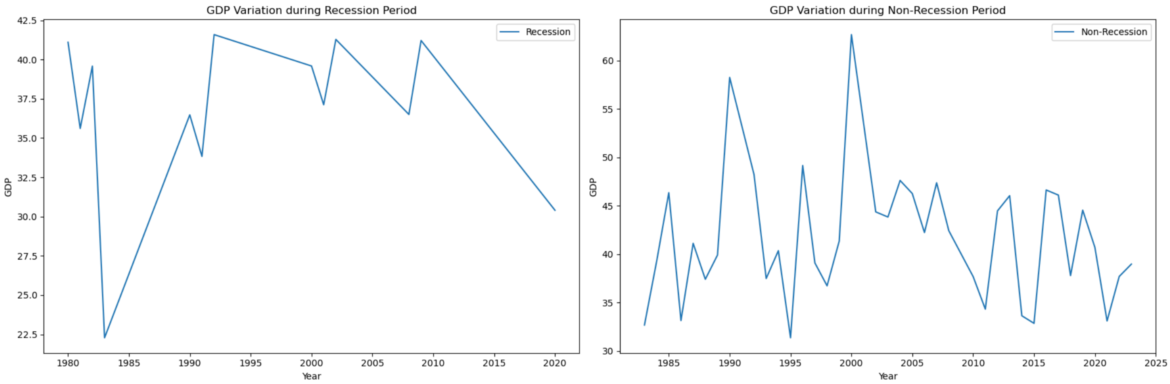
TASK 1.4: Use sub plotting to compare the variations in GDP during recession and non-recession period by developing line plots for each period.

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?

In [17]:
```
grouped_df4=df.groupby(['Recession','Year'])['GDP'].mean().reset_index()


#Create dataframes for recession and non-recession period
rec_data = grouped_df4[grouped_df4['Recession'] == 1]
non_rec_data = grouped_df4[grouped_df4['Recession'] == 0]

plt.figure(figsize=(18, 6))
plt.subplot(1, 2, 1)
sns.lineplot(x='Year', y='GDP', data=rec_data, label='Recession')
plt.title('GDP Variation during Recession Period')
plt.xlabel('Year')
plt.ylabel('GDP')
plt.legend()
#subplot 1
plt.subplot(1, 2, 2)
sns.lineplot(x='Year', y='GDP', data=non_rec_data, label='Non-Recession')
plt.title('GDP Variation during Non-Recession Period')
plt.xlabel('Year')
plt.ylabel('GDP')
plt.legend()

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

```
C:\Users\91939\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarnin
g: use_inf_as_na option is deprecated and will be removed in a future version. Co
nvert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\91939\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarnin
g: use_inf_as_na option is deprecated and will be removed in a future version. Co
nvert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\91939\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarnin
g: use_inf_as_na option is deprecated and will be removed in a future version. Co
nvert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\91939\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarnin
g: use_inf_as_na option is deprecated and will be removed in a future version. Co
nvert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



▶ Click here for Solution template

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.

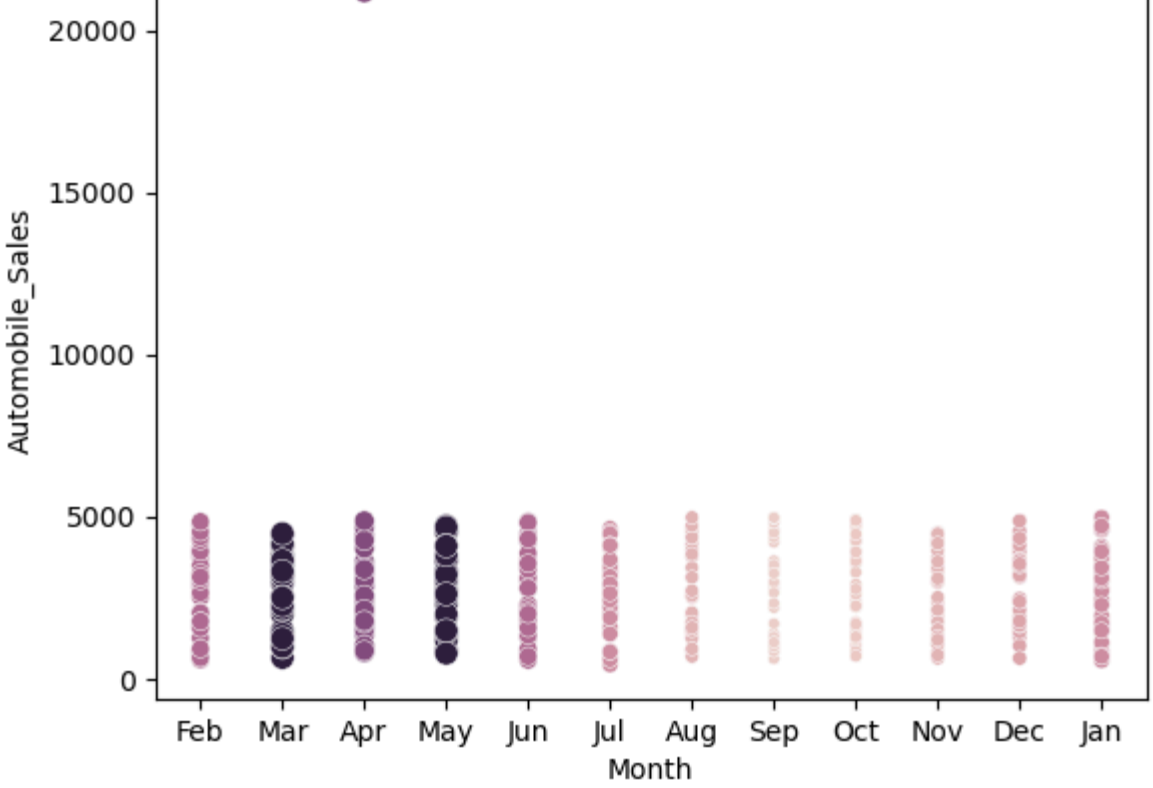
How has seasonality impacted the sales, in which months the sales were high or low? Check it for non-recession years to understand the trend

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'

▶ Click here for a hint

In [18]:
```
grouped_df5 = df[df['Recession']==0]
# grouped_df5=grouped_df5.groupby(['Seasonality_Weight','Month'])['Automobile_Sa

sns.scatterplot(data = grouped_df5 ,x='Month',y='Automobile_Sales',size='Seasona
plt.title(label="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 lick on the plot and then click on "Save image as" option to save it on your local machine

---

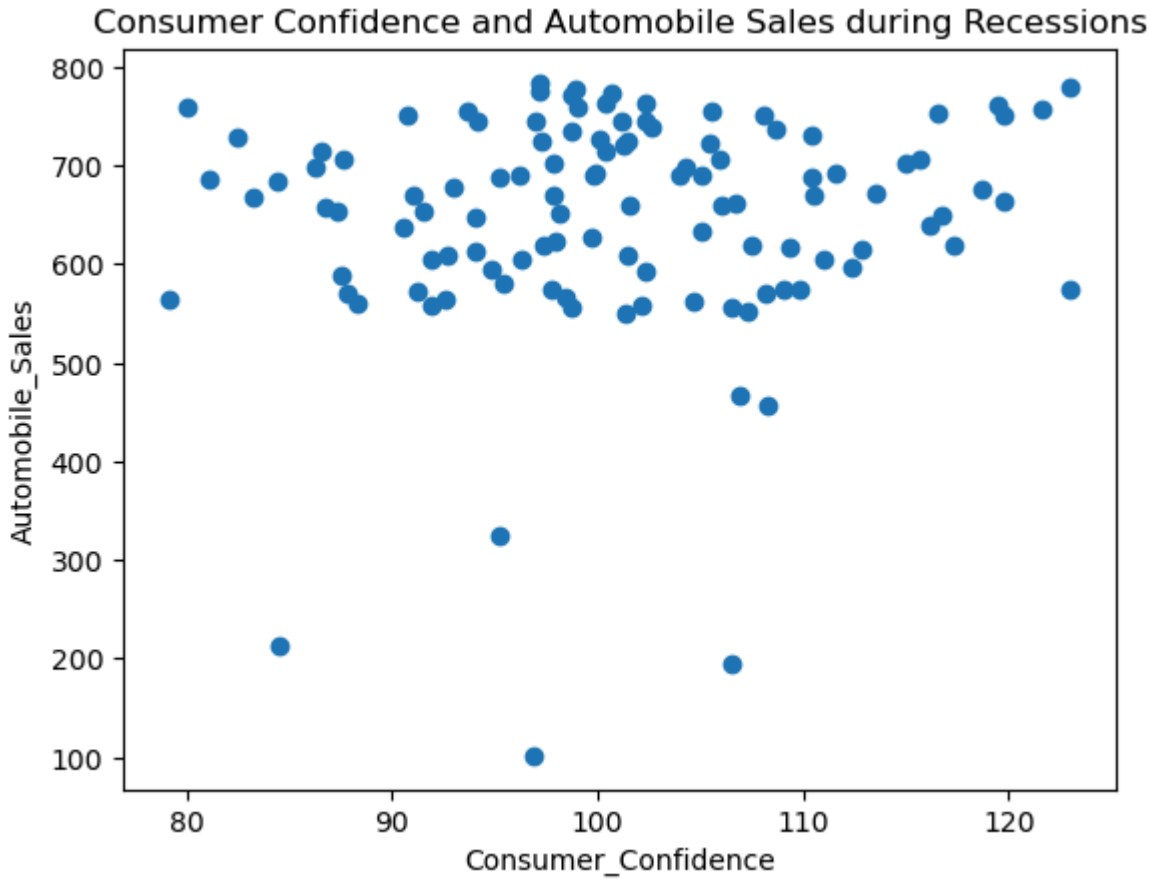
TASK 1.6: Use the functionality of Matplotlib to develop a scatter plot to identify the correlation between average vehicle price relate to the sales volume during recessions.

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

```
plt.title('Consumer Confidence and Automobile Sales during Recessions')
plt.xlabel('Consumer_Confidence')
plt.ylabel('Automobile_Sales')
plt.show()
```

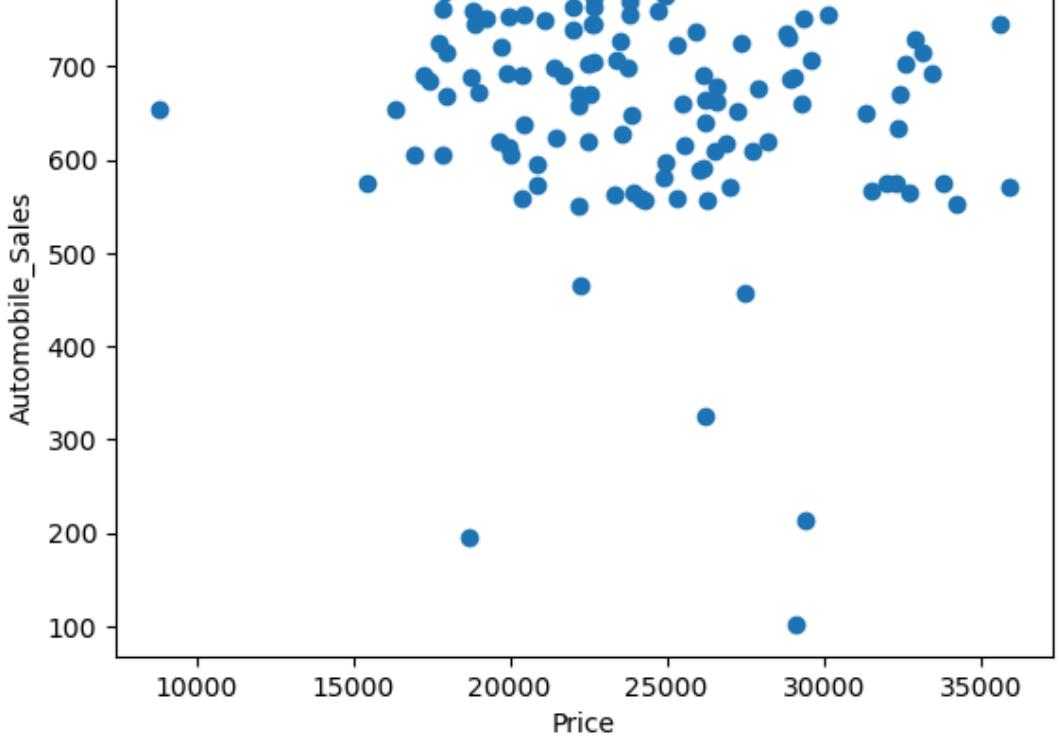


▶ Click here for Solution template

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 [20]:
```
grouped_df6 = df[df['Recession']==1]
plt.scatter('Price' , 'Automobile_Sales',data=grouped_df6)
plt.title('Rlationship between Average Vehicle Price and Slaes during Recessiong
plt.xlabel('Price')
plt.ylabel('Automobile_Sales')
plt.show()
```



▶ Click here for Solution template

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

---

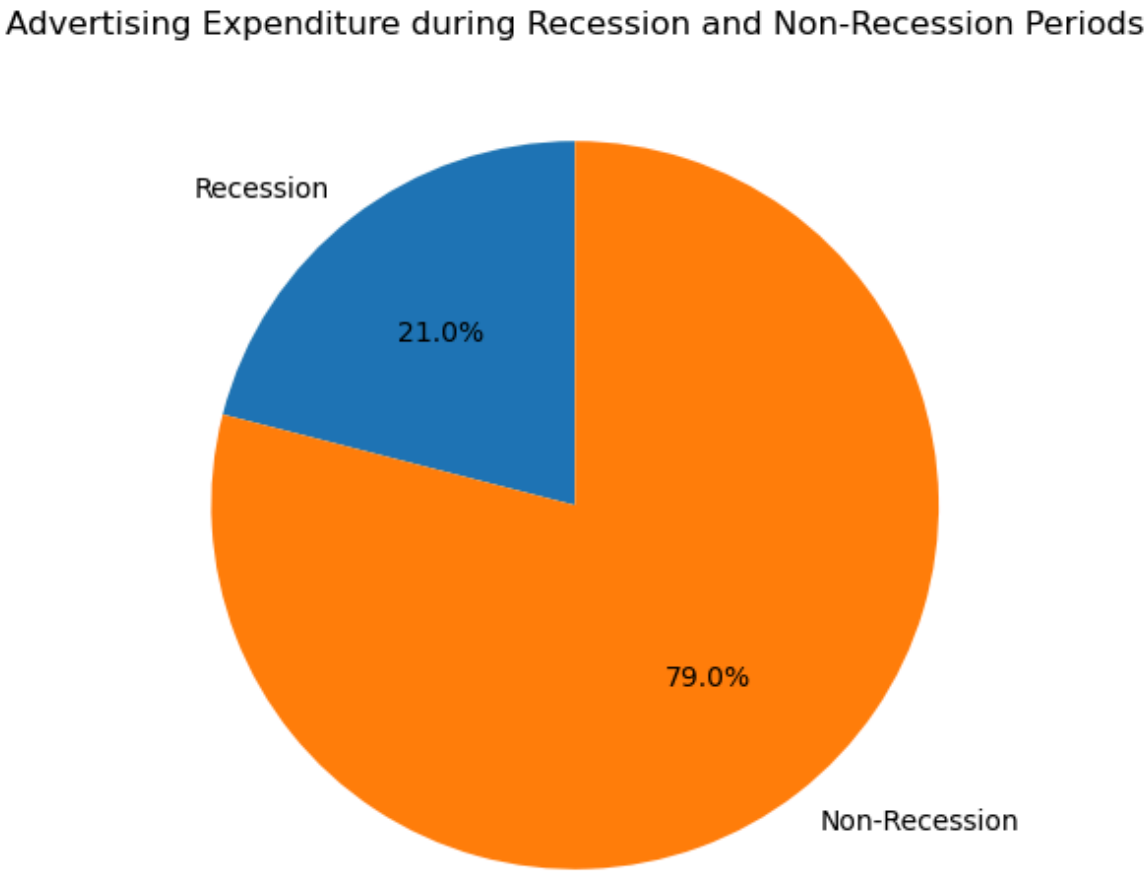
TASK 1.7: Create a pie chart to display the portion of advertising expenditure of XYZAutomotives during recession and non-recession periods.

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

▶ Click here for a hint

In [21]:
```
# Filter the data
Rdata = df[df['Recession'] == 1]
NRdata = df[df['Recession'] == 0]

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

```
plt.figure(figsize=(8, 6))

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

plt.title('Advertising Expenditure during Recession and Non-Recession Periods')

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

▶ Inference

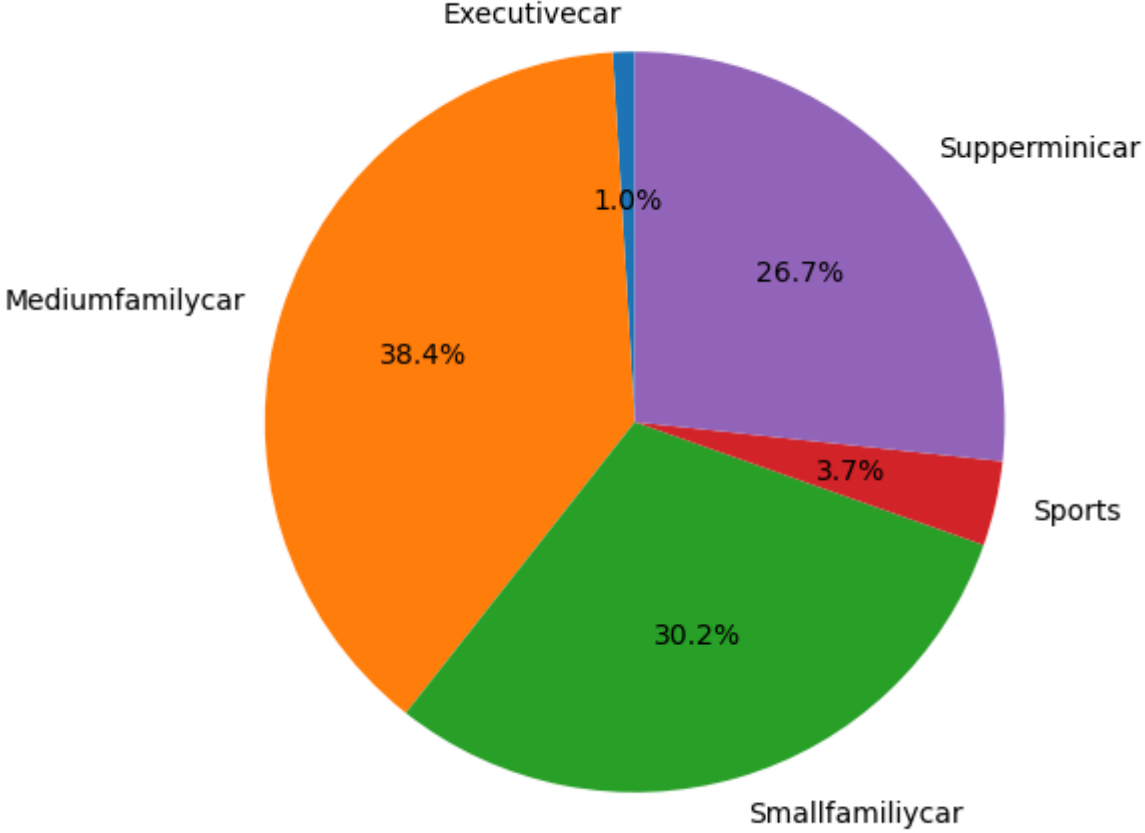
on your local machine*

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 [22]:
```
# Filter the data
Rdata = df[df['Recession'] == 1]

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

# Create a pie chart for the share of each vehicle type in total sales during re
plt.figure(figsize=(8, 6))

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

plt.title('Share of Each Vehicle Type in Total Sales during Recessions')

plt.show()
```



▶ Click here for Solution template

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

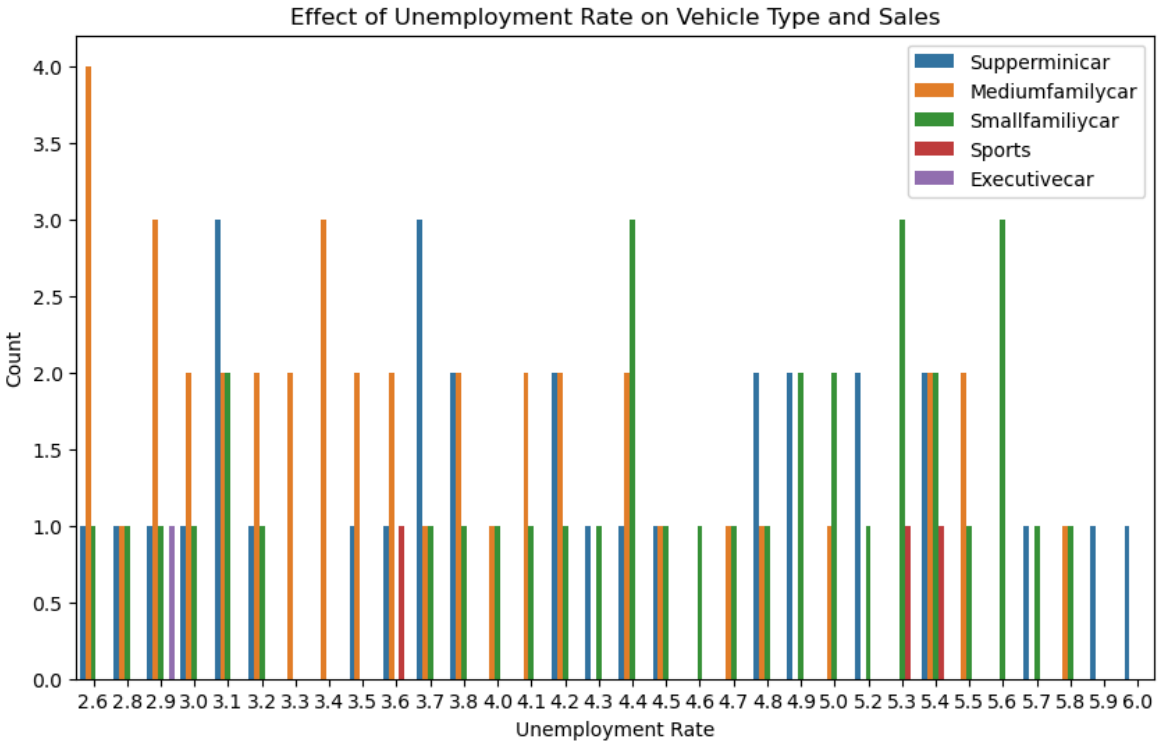
---

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'

In [23]:
```
data= df[df['Recession'] == 1]

plt.figure(figsize=(10, 6))

sns.countplot(data=data, x='unemployment_rate', hue='Vehicle_Type')

plt.xlabel('Unemployment Rate')
plt.ylabel('Count')
plt.title('Effect of Unemployment Rate on Vehicle Type and Sales')
plt.legend(loc='upper right')
plt.show()
```



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

▶ Inference

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

---

region/offices of the company during recession period

In [24]:
```
import requests

def download(url, filename):
  response = requests.get(url,headers)
  with open(filename, "w") as f:
      f.write(response.text)
path = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDe
filename = "us-states.json"
headers = { 'User-Agent' : 'Mozilla/5.0'}

download(path,filename)
```

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

In [25]:
```
# 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_i

# 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
```

▶ Click for Solution

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|