```
1.How does Seaborn relate to Matplotlib? (strengths and weaknesses of
each).
Seaborn and Matplotlib are both powerful Python libraries for data
visualization, but they serve different purposes and complement each
other well.
Here's how they relate, along with their strengths and weaknesses:
Relationship
•    Built on Matplotlib: Seaborn is essentially a high-level
interface built on top of Matplotlib. It simplifies the process of
creating complex visualizations and offers additional functionalities.
•    Enhanced Features: Seaborn automates many tasks, like choosing
aesthetic styles, handling color palettes, and working with
statistical data, that would require significant manual effort in
Matplotlib.


Seaborn
Strengths
1.   Simplified Syntax: Seaborn abstracts much of the complexity of
Matplotlib, making it easier to create advanced plots with minimal
code.
     Example: Creating a boxplot or violin plot with Seaborn is
straightforward, while Matplotlib requires more steps.
2.   Built-in Themes: It provides aesthetic default styles (whitegrid,
darkgrid, etc.) for better-looking plots out of the box.
3.   Statistical Integration: Seaborn integrates seamlessly with
pandas and has built-in support for statistical visualizations (e.g.,
KDE plots, categorical plots, and regression plots).
4.   Color Palettes: Includes diverse and visually appealing color
palettes (hls, husl, coolwarm, etc.) that are easy to apply.
5.   Faceted Plots: Facilitates the creation of grid-based plots for
analyzing subsets of data.
6.   Automatic Aggregation: Handles grouping and aggregation for
visualizations, saving significant manual effort.

Weaknesses
1.   Limited Flexibility: While it simplifies plotting, customizing
plots can be restrictive compared to Matplotlib.
2.   Dependency on Matplotlib: For very specific customizations, you
still need to use Matplotlib functions.
3.   Fewer Plot Types: Seaborn lacks support for some specialized
plots available in Matplotlib (e.g., 3D plots).

_
Matplotlib
Strengths
1.   Highly Customizable: Offers complete control over every aspect of
the plot, from axes scaling to marker styles.
2.   Wide Range of Plot Types: Supports specialized visualizations
like 3D plotting, polar charts, and quiver plots.
```

3.   Low-Level Control: Allows precise adjustments to elements like ticks, legends, and subplots.
4.   Stable and Mature: Matplotlib is one of the oldest Python visualization libraries and has a vast user base and community support.

Weaknesses
1.   Verbose Syntax: Simple tasks often require more code compared to Seaborn.
2.   Aesthetics: Default styles in Matplotlib are less visually appealing, though customization can improve this.
3.   Steeper Learning Curve: The need to understand its object-oriented interface makes Matplotlib more challenging for beginners.

```
#importing libraries

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv("historical_automobile_sales.csv")

df
```

```
           Date  Year Month  Recession  Consumer_Confidence  \
0     1/31/1980  1980   Jan          1               108.24   
1     2/29/1980  1980   Feb          1                98.75   
2     3/31/1980  1980   Mar          1               107.48   
3     4/30/1980  1980   Apr          1               115.01   
4     5/31/1980  1980   May          1                98.72   
..          ...   ...   ...        ...                  ...   
523   8/31/2023  2023   Aug          0               103.36   
524   9/30/2023  2023   Sep          0               101.55   
525  10/31/2023  2023   Oct          0               124.66   
526  11/30/2023  2023   Nov          0                97.09   
527  12/31/2023  2023   Dec          0                95.92   

     Seasonality_Weight      Price  Advertising_Expenditure  
Competition  \
0                  0.50  27483.571                     1558  
7   
1                  0.75  24308.678                     3048  
4   
2                  0.20  28238.443                     3137  
3   
3                  1.00  32615.149                     1653  
7   
4                  0.20  23829.233                     1319  
```

```
4
..                  ...        ...                        ...          .
..
523                0.25  27852.993                       1793
6
524                0.07  21183.704                       1028
5
525                0.12  15975.589                       1148
9
526                0.25  16862.288                       4850
5
527                0.34  25240.425                       2319
3

        GDP  Growth_Rate  unemployment_rate  Automobile_Sales  \
0    60.223     0.010000                5.4             456.0
1    45.986    -0.309594                4.8             555.9
2    35.141    -0.308614                3.4             620.0
3    45.673     0.230596                4.2             702.8
4    52.997     0.138197                5.3             770.4
..      ...          ...                ...               ...
523  57.169     0.764155                2.6            1579.6
524  59.315     0.036180                2.5            1123.4
525  19.472    -2.046169                2.5            1685.9
526  27.904     0.302179                2.9            2124.6
527  13.518    -1.064211                2.1            3538.5

        Vehicle_Type        City
0      Supperminicar     Georgia
1      Supperminicar    New York
2    Mediumfamilycar    New York
3      Supperminicar    Illinois
4    Smallfamiliycar  California
..               ...         ...
523     Executivecar    New York
524  Smallfamiliycar  California
525           Sports  California
526  Smallfamiliycar     Georgia
527  Smallfamiliycar     Georgia

[528 rows x 15 columns]
```

```
df.shape
```

```
(528, 15)
```

```
df.info
```

```
<bound method DataFrame.info of             Date  Year Month  Recession
Consumer_Confidence  \
0     1/31/1980  1980   Jan          1                108.24
1     2/29/1980  1980   Feb          1                 98.75
2     3/31/1980  1980   Mar          1                107.48
3     4/30/1980  1980   Apr          1                115.01
4     5/31/1980  1980   May          1                 98.72
..          ...   ...   ...        ...                   ...
523   8/31/2023  2023   Aug          0                103.36
524   9/30/2023  2023   Sep          0                101.55
525  10/31/2023  2023   Oct          0                124.66
526  11/30/2023  2023   Nov          0                 97.09
527  12/31/2023  2023   Dec          0                 95.92

     Seasonality_Weight      Price  Advertising_Expenditure
Competition  \
0                  0.50  27483.571                     1558
7
1                  0.75  24308.678                     3048
4
2                  0.20  28238.443                     3137
3
3                  1.00  32615.149                     1653
7
4                  0.20  23829.233                     1319
4
..                  ...        ...                      ...          .
..
523                0.25  27852.993                     1793
6
524                0.07  21183.704                     1028
5
525                0.12  15975.589                     1148
9
526                0.25  16862.288                     4850
5
527                0.34  25240.425                     2319
3

        GDP  Growth_Rate  unemployment_rate  Automobile_Sales  \
0    60.223     0.010000                5.4             456.0
1    45.986    -0.309594                4.8             555.9
2    35.141    -0.308614                3.4             620.0
3    45.673     0.230596                4.2             702.8
4    52.997     0.138197                5.3             770.4
..      ...          ...                ...               ...
523  57.169     0.764155                2.6            1579.6
524  59.315     0.036180                2.5            1123.4
525  19.472    -2.046169                2.5            1685.9
526  27.904     0.302179                2.9            2124.6
527  13.518    -1.064211                2.1            3538.5
```

```
        Vehicle_Type        City
0      Supperminicar     Georgia
1      Supperminicar    New York
2    Mediumfamilycar    New York
3      Supperminicar    Illinois
4    Smallfamiliycar  California
..               ...         ...
523     Executivecar    New York
524  Smallfamiliycar  California
525           Sports  California
526  Smallfamiliycar     Georgia
527  Smallfamiliycar     Georgia

[528 rows x 15 columns]>
```

```
df.describe
```

```
<bound method NDFrame.describe of             Date  Year Month
Recession  Consumer_Confidence  \
0     1/31/1980  1980   Jan          1                   108.24
1     2/29/1980  1980   Feb          1                    98.75
2     3/31/1980  1980   Mar          1                   107.48
3     4/30/1980  1980   Apr          1                   115.01
4     5/31/1980  1980   May          1                    98.72
..          ...   ...   ...        ...                      ...
523   8/31/2023  2023   Aug          0                   103.36
524   9/30/2023  2023   Sep          0                   101.55
525  10/31/2023  2023   Oct          0                   124.66
526  11/30/2023  2023   Nov          0                    97.09
527  12/31/2023  2023   Dec          0                    95.92

     Seasonality_Weight      Price  Advertising_Expenditure
Competition  \
0                   0.50  27483.571                     1558
7
1                   0.75  24308.678                     3048
4
2                   0.20  28238.443                     3137
3
3                   1.00  32615.149                     1653
7
4                   0.20  23829.233                     1319
4
..                   ...        ...                      ...          .
..
523                 0.25  27852.993                     1793
6
524                 0.07  21183.704                     1028
5
525                 0.12  15975.589                     1148
```

```
9
526                0.25  16862.288                     4850
5
527                0.34  25240.425                     2319
3

        GDP  Growth_Rate  unemployment_rate  Automobile_Sales  \
0    60.223     0.010000                5.4             456.0   
1    45.986    -0.309594                4.8             555.9   
2    35.141    -0.308614                3.4             620.0   
3    45.673     0.230596                4.2             702.8   
4    52.997     0.138197                5.3             770.4   
..      ...          ...                ...               ...   
523  57.169     0.764155                2.6            1579.6   
524  59.315     0.036180                2.5            1123.4   
525  19.472    -2.046169                2.5            1685.9   
526  27.904     0.302179                2.9            2124.6   
527  13.518    -1.064211                2.1            3538.5   

        Vehicle_Type        City  
0      Supperminicar     Georgia  
1      Supperminicar    New York  
2    Mediumfamilycar    New York  
3      Supperminicar    Illinois  
4    Smallfamiliycar  California  
..               ...         ...  
523     Executivecar    New York  
524  Smallfamiliycar  California  
525           Sports  California  
526  Smallfamiliycar     Georgia  
527  Smallfamiliycar     Georgia  

[528 rows x 15 columns]>
```

```
df.dtypes
```

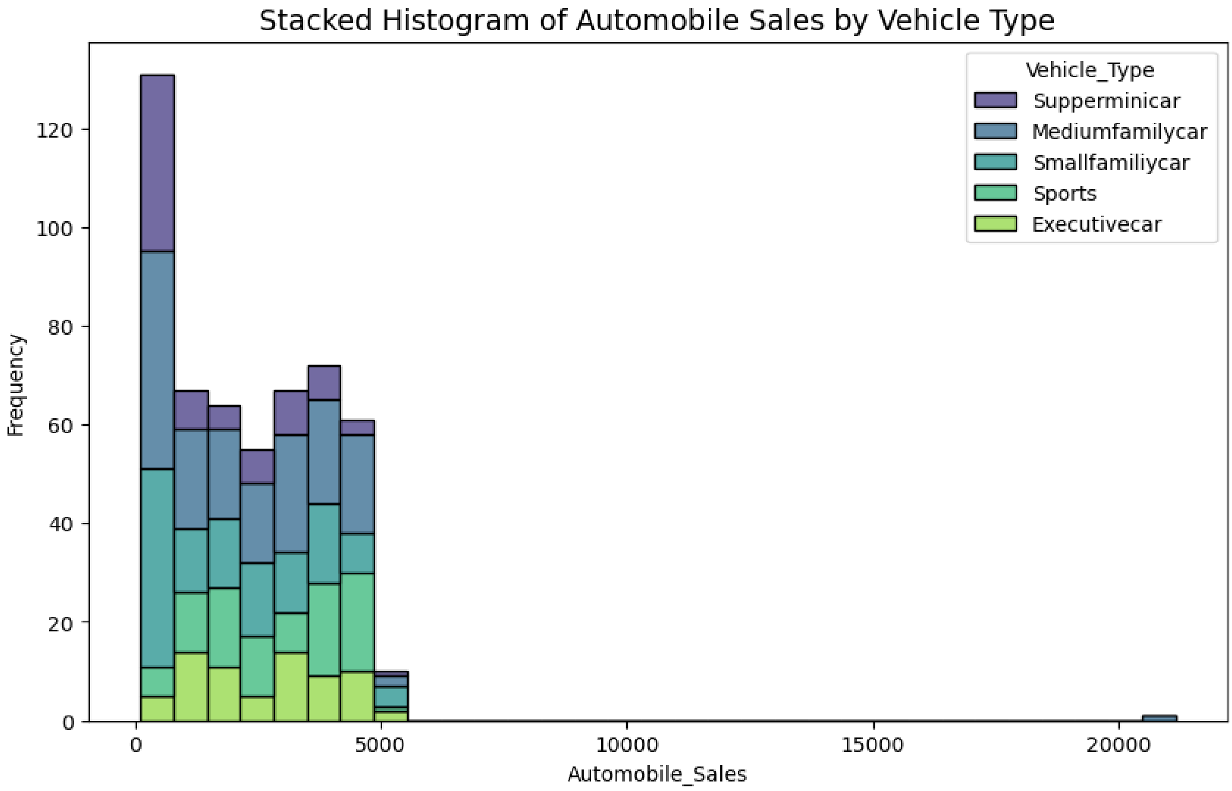
```
Date                          object
Year                           int64
Month                         object
Recession                      int64
Consumer_Confidence          float64
Seasonality_Weight           float64
Price                        float64
Advertising_Expenditure        int64
Competition                    int64
GDP                          float64
Growth_Rate                  float64
unemployment_rate            float64
Automobile_Sales             float64
Vehicle_Type                  object
```

```
City                        object
dtype: object
```

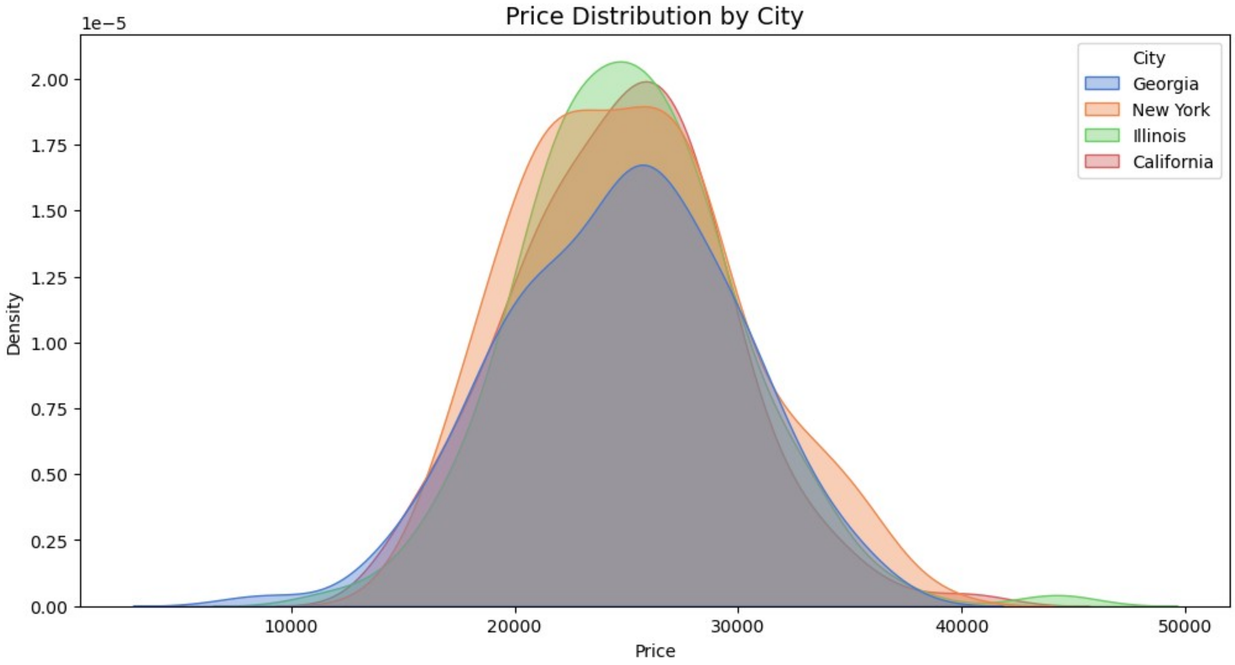
```
df.columns
```

```
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')
```

```
#Using Seaborn library create a stacked histogram which will capture
Automobile Sales for Vehicle Types.
plt.figure(figsize=(10, 6))
sns.histplot(
    data=df,
    x="Automobile_Sales",
    hue="Vehicle_Type",
    multiple="stack",
    palette="viridis",
    kde=False
)
plt.title("Stacked Histogram of Automobile Sales by Vehicle Type",
fontsize=14)
plt.xlabel("Automobile_Sales")
plt.ylabel("Frequency")
plt.show()
```

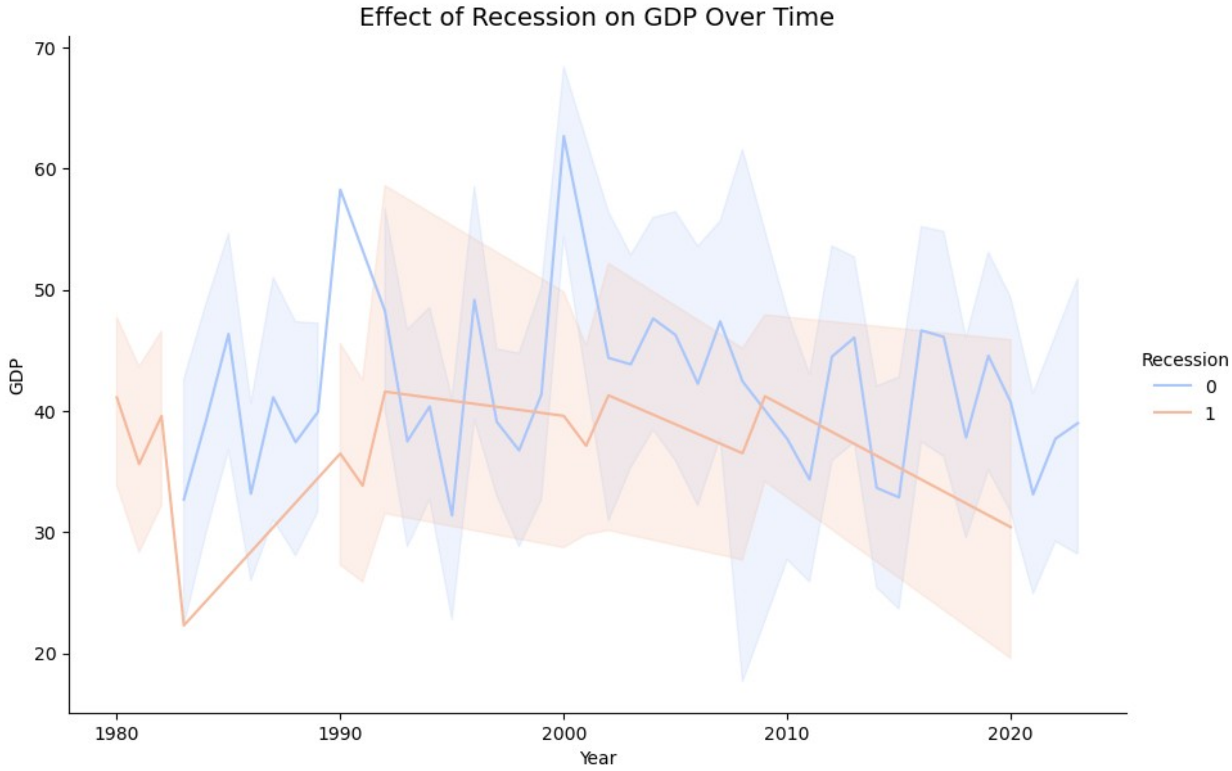


```
#Create a distribution plot (kernel density estimation plot) to
understand the distribution of Price for different Cities.
plt.figure(figsize=(12, 6))
sns.kdeplot(data=df, x="Price", hue="City", fill=True, alpha=0.4,
palette="muted")
plt.title("Price Distribution by City", fontsize=14)
plt.xlabel("Price")
plt.ylabel("Density")
plt.show()
```

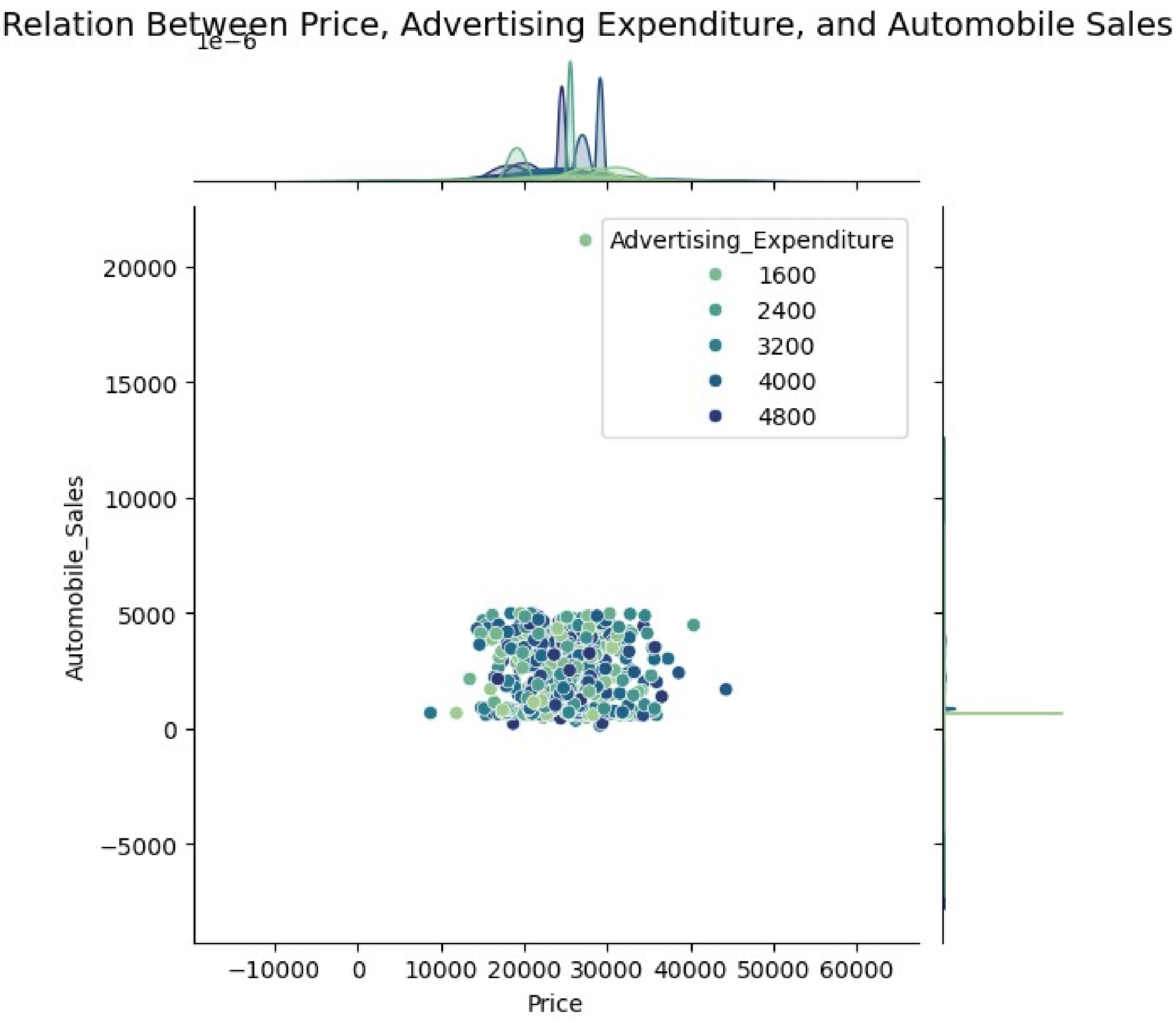


```
#Create a relation plot to visualise how along with the effect of
Recession, GDP has changed with time(Year)
plt.figure(figsize=(14, 8))
sns.relplot(
    data=df,
    x="Year",
    y="GDP",
    hue="Recession",
    kind="line",
    palette="coolwarm",
    height=6,
    aspect=1.5
)
plt.title("Effect of Recession on GDP Over Time", fontsize=14)
plt.xlabel("Year")
plt.ylabel("GDP")
plt.show()
```

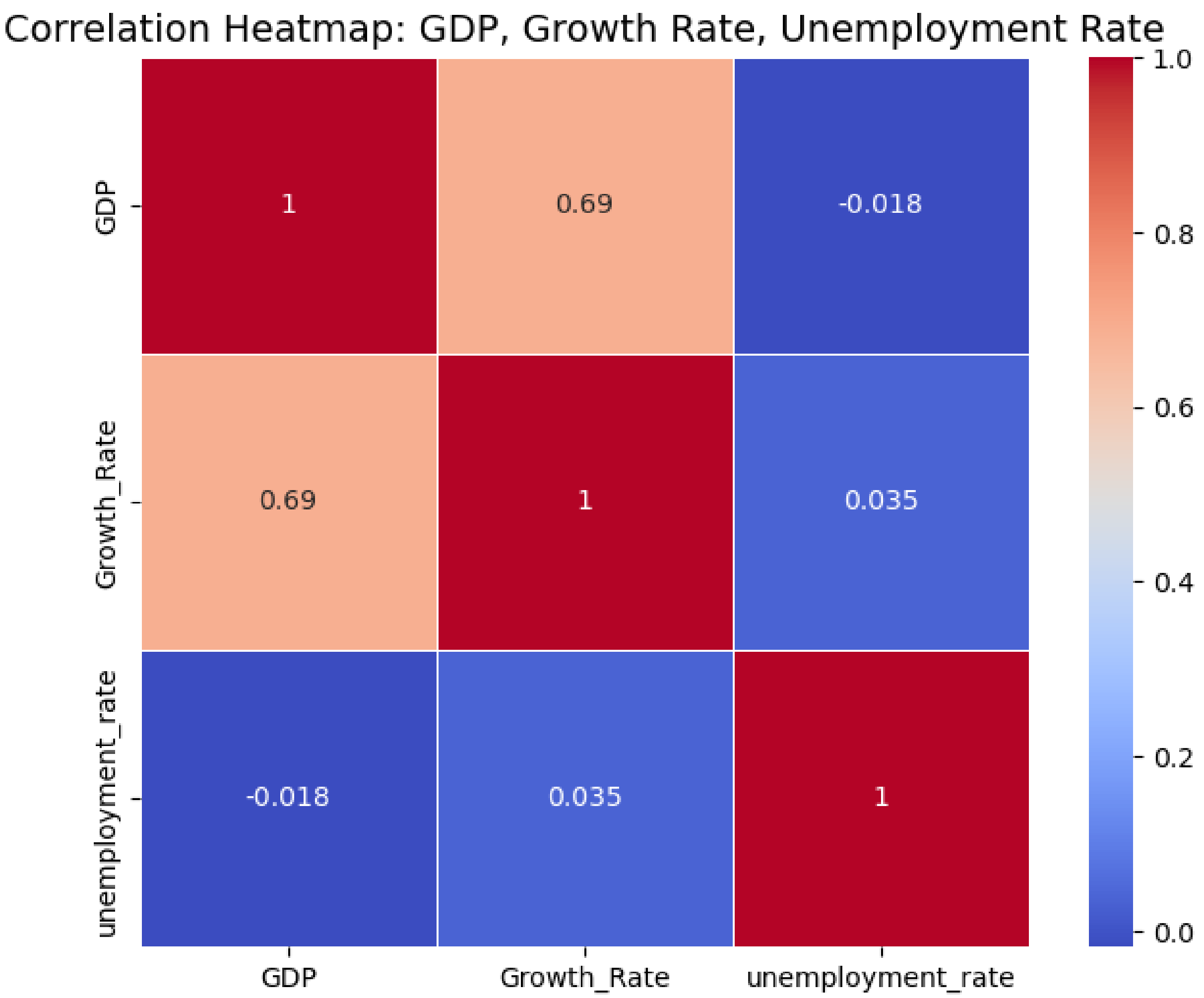
```
<Figure size 1400x800 with 0 Axes>
```



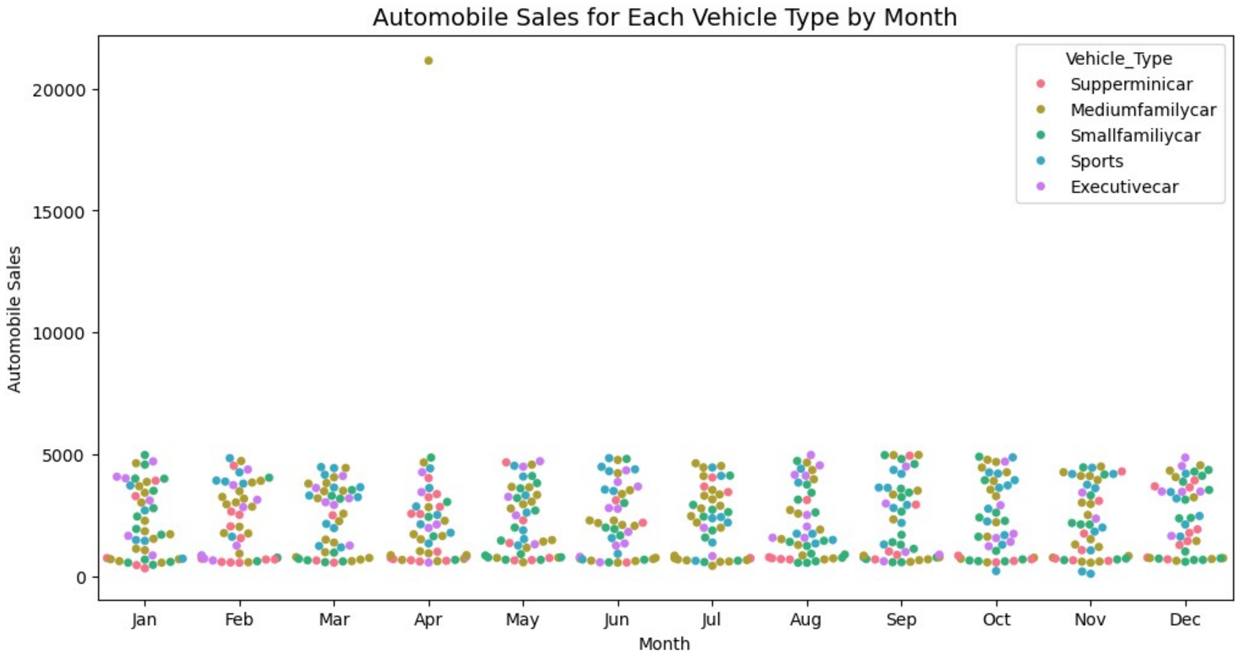
```
#Prepare a Joint Plot to understand relation between Price,
Adverstising_Expenditure and Automobile_Sales.
sns.jointplot(
    data=df,
    x="Price",
    y="Automobile_Sales",
    hue="Advertising_Expenditure",
    palette="crest",
    kind="scatter"
)
plt.suptitle("Relation Between Price, Advertising Expenditure, and
Automobile Sales", y=1.02, fontsize=14)
plt.show()
```



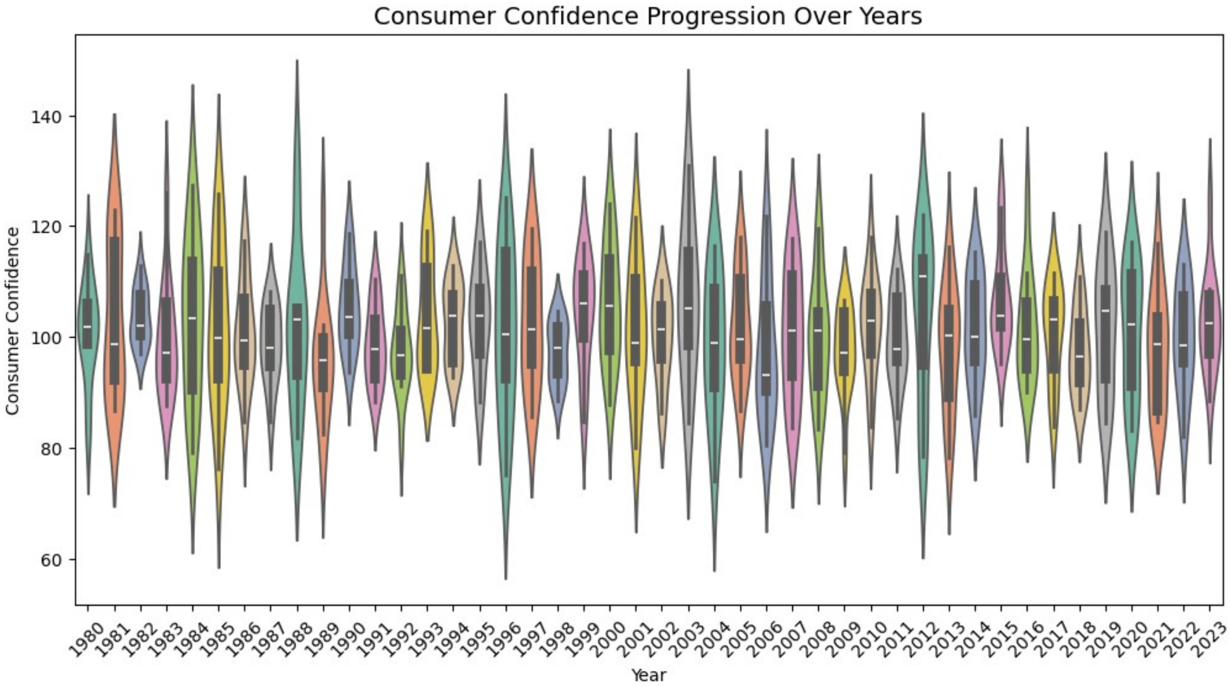
```python
#Create a heat map to understand correlation for GDP, Growth_Rate and
Unemployment_Rate.
correlation_data = df[["GDP", "Growth_Rate",
"unemployment_rate"]].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(
    correlation_data,
    annot=True,
    cmap="coolwarm",
    linewidths=0.5,
    square=True
)
plt.title("Correlation Heatmap: GDP, Growth Rate, Unemployment Rate",
fontsize=14)
plt.show()
```


Correlation Heatmap: GDP, Growth Rate, Unemployment Rate

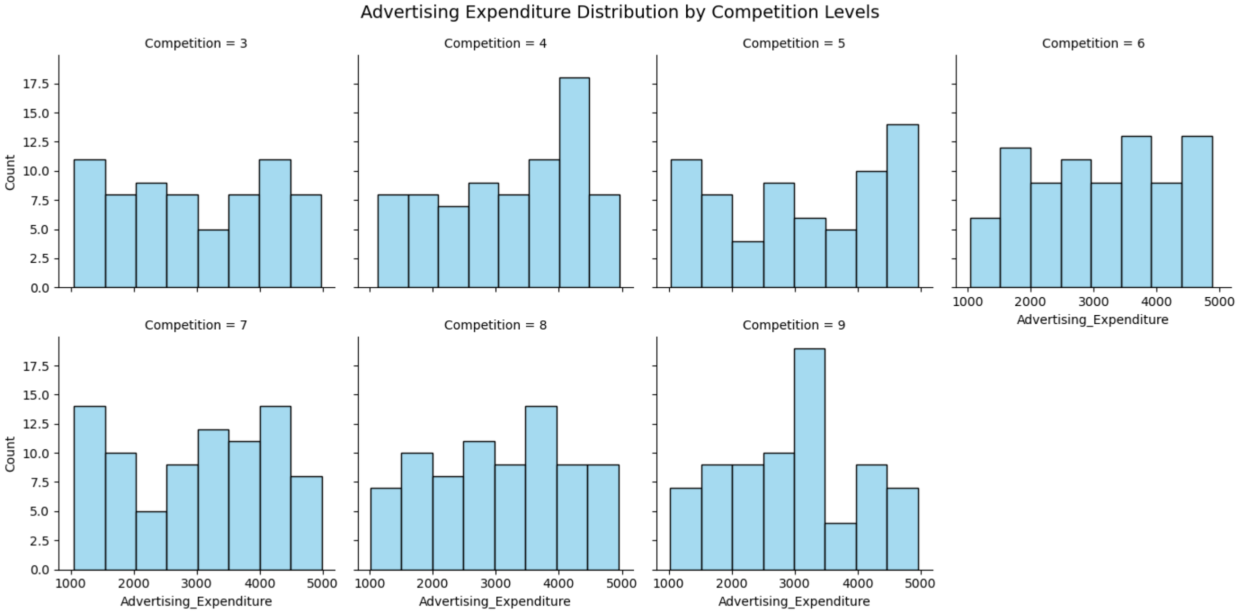
```
#Make a swarm categorical Plot to deduce the automotive sales for
every Automobile Type within every calendar month.
plt.figure(figsize=(12, 6))
sns.swarmplot(data=df, x="Month", y="Automobile_Sales",
hue="Vehicle_Type", palette="husl")
plt.title("Automobile Sales for Each Vehicle Type by Month",
fontsize=14)
plt.xlabel("Month")
plt.ylabel("Automobile Sales")
plt.show()
```



```
#Prepare a Violin Plot of how Consumer_Confidence has changed with
every progressing year.
plt.figure(figsize=(12, 6))
sns.violinplot(data=df, x="Year", y="Consumer_Confidence",
palette="Set2")
plt.title("Consumer Confidence Progression Over Years", fontsize=14)
plt.xlabel("Year")
plt.ylabel("Consumer Confidence")
plt.xticks(rotation=45)
plt.show()
```



```
#Create a facet grid of Competition to view the histogram of Adverstising_Expenditure.
facet = sns.FacetGrid(df, col="Competition", col_wrap=4, height=3.5, sharex=True, sharey=True)
facet.map(sns.histplot, "Advertising_Expenditure", color="skyblue", kde=False)
facet.fig.subplots_adjust(top=0.9)
facet.fig.suptitle("Advertising Expenditure Distribution by Competition Levels", fontsize=14)
plt.show()
```


Advertising Expenditure Distribution by Competition Levels
Competition = 3
Competition = 4
Competition = 5
Competition = 6
Competition = 7
Competition = 8
Competition = 9
Count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
1000
2000
3000
4000
5000
Advertising_Expenditure