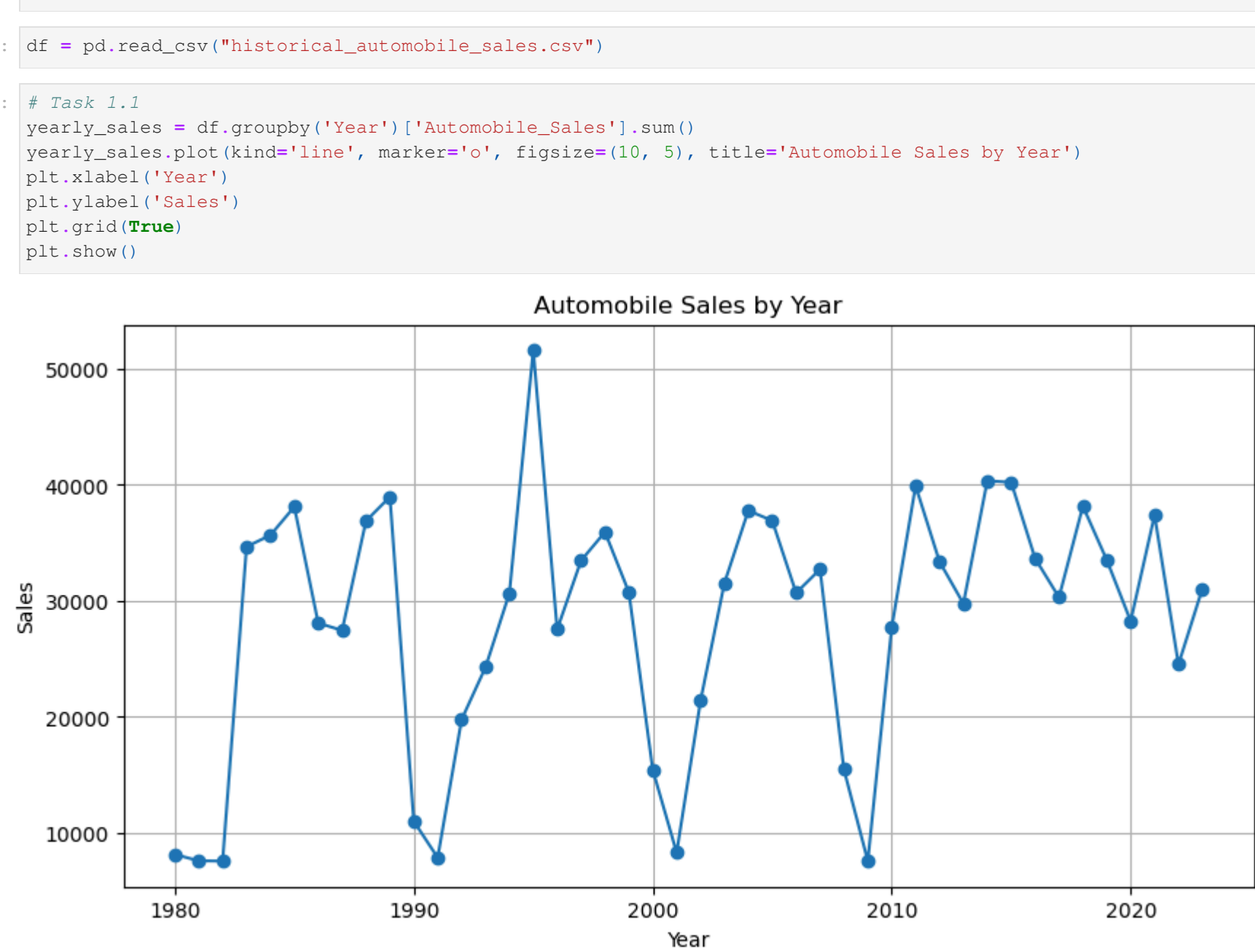


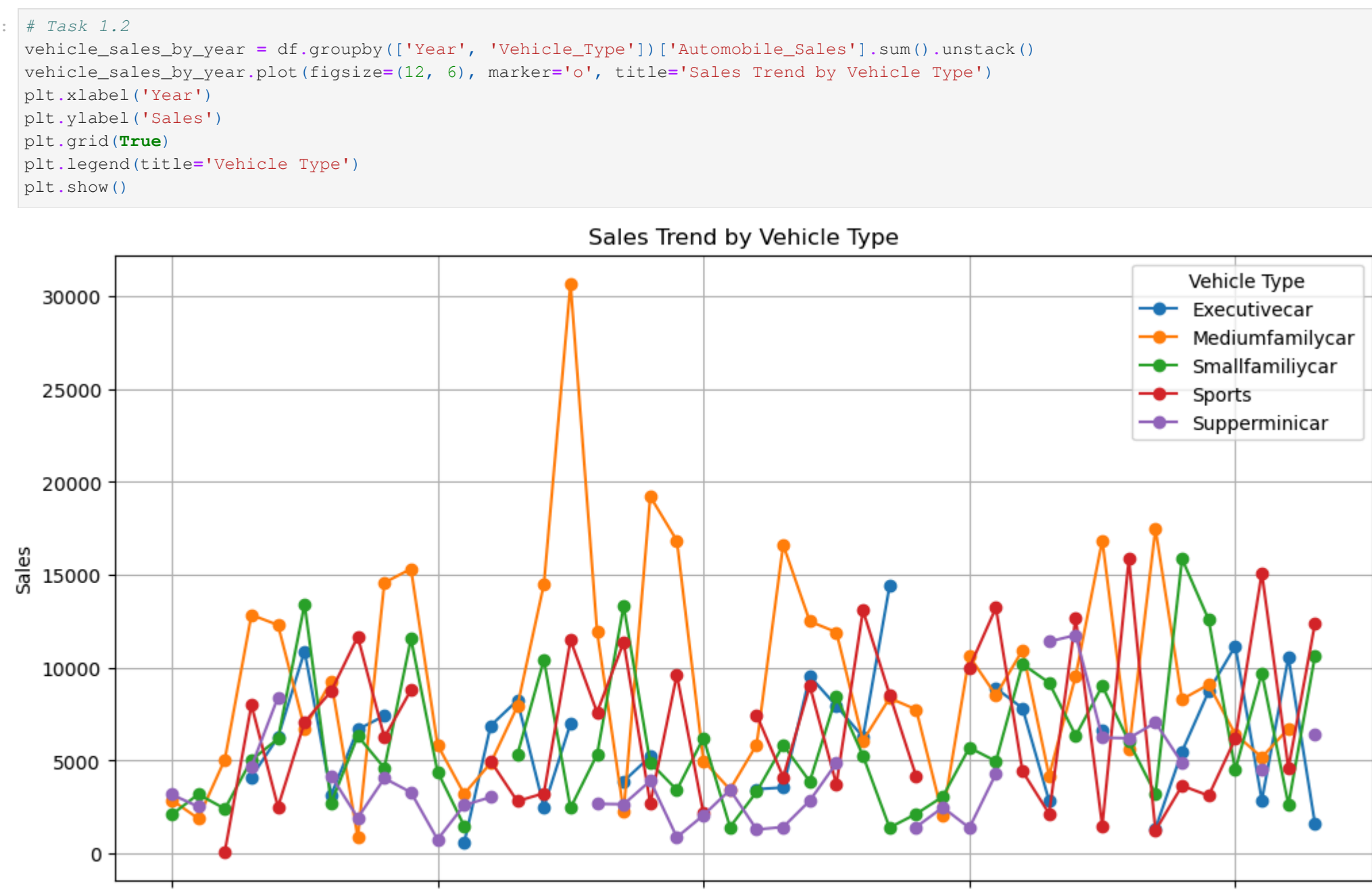
```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [9]: df = pd.read_csv('historical_automobile_sales.csv')
```

```
In [11]: # Task 1.1
yearly_sales = df.groupby('Year')['Automobile_Sales'].sum()
yearly_sales.plot(kind='line', marker='o', figsize=(10, 5), title='Automobile Sales by Year')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.grid(True)
plt.show()
```

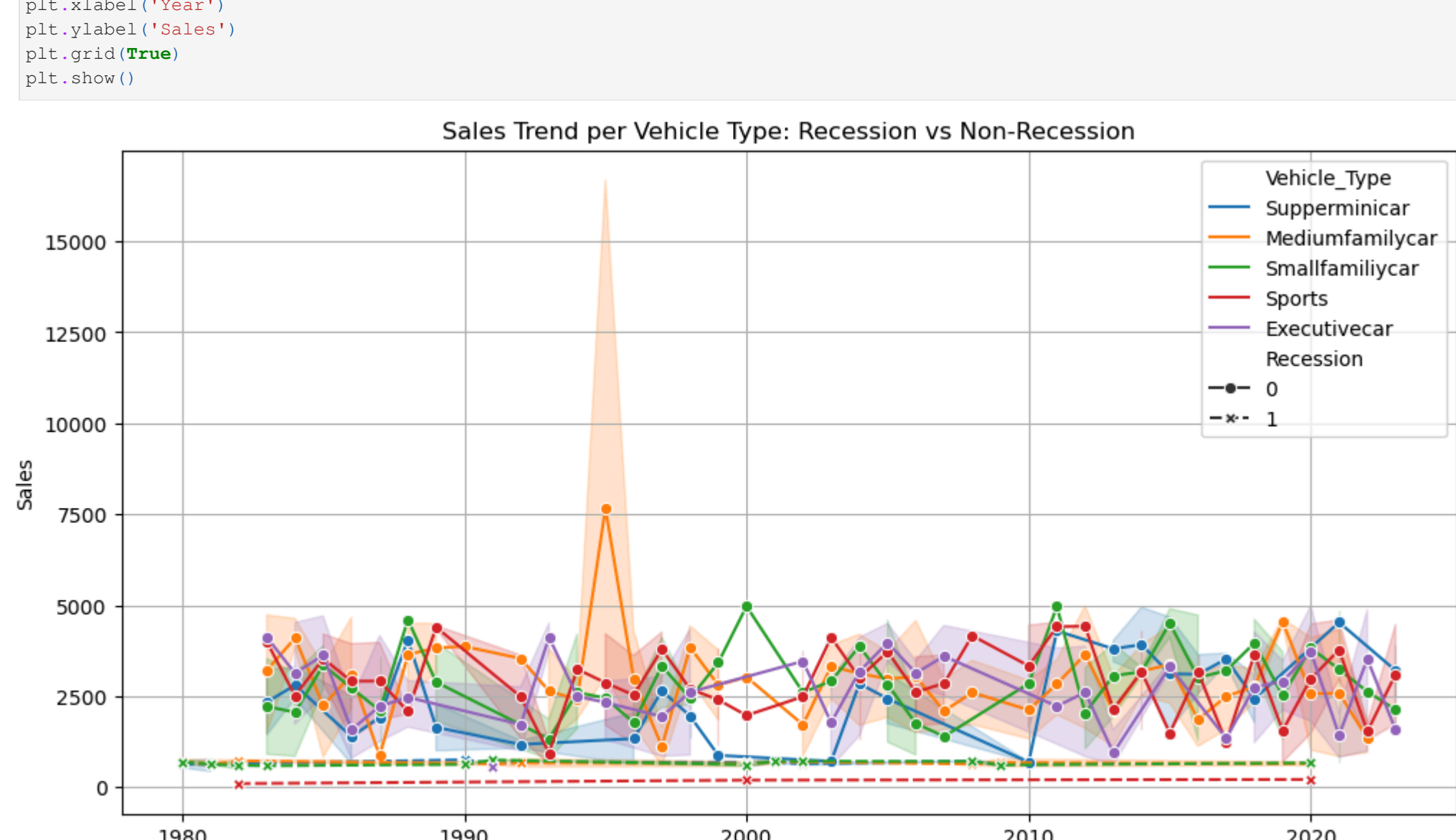


```
In [13]: # Task 1.2
vehicle_sales_by_year = df.groupby(['Year', 'Vehicle_Type'])['Automobile_Sales'].sum().unstack()
vehicle_sales_by_year.plot(figsize=(12, 6), marker='o', title='Sales Trend by Vehicle Type')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.grid(True)
plt.legend(title='Vehicle Type')
plt.show()
```



```
In [15]: import seaborn as sns
```

```
# Task 1.3
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='Year', y='Automobile_Sales', hue='Vehicle_Type', style='Recession', markers=True)
plt.title('Sales Trend per Vehicle Type: Recession vs Non-Recession')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.grid(True)
plt.show()
```

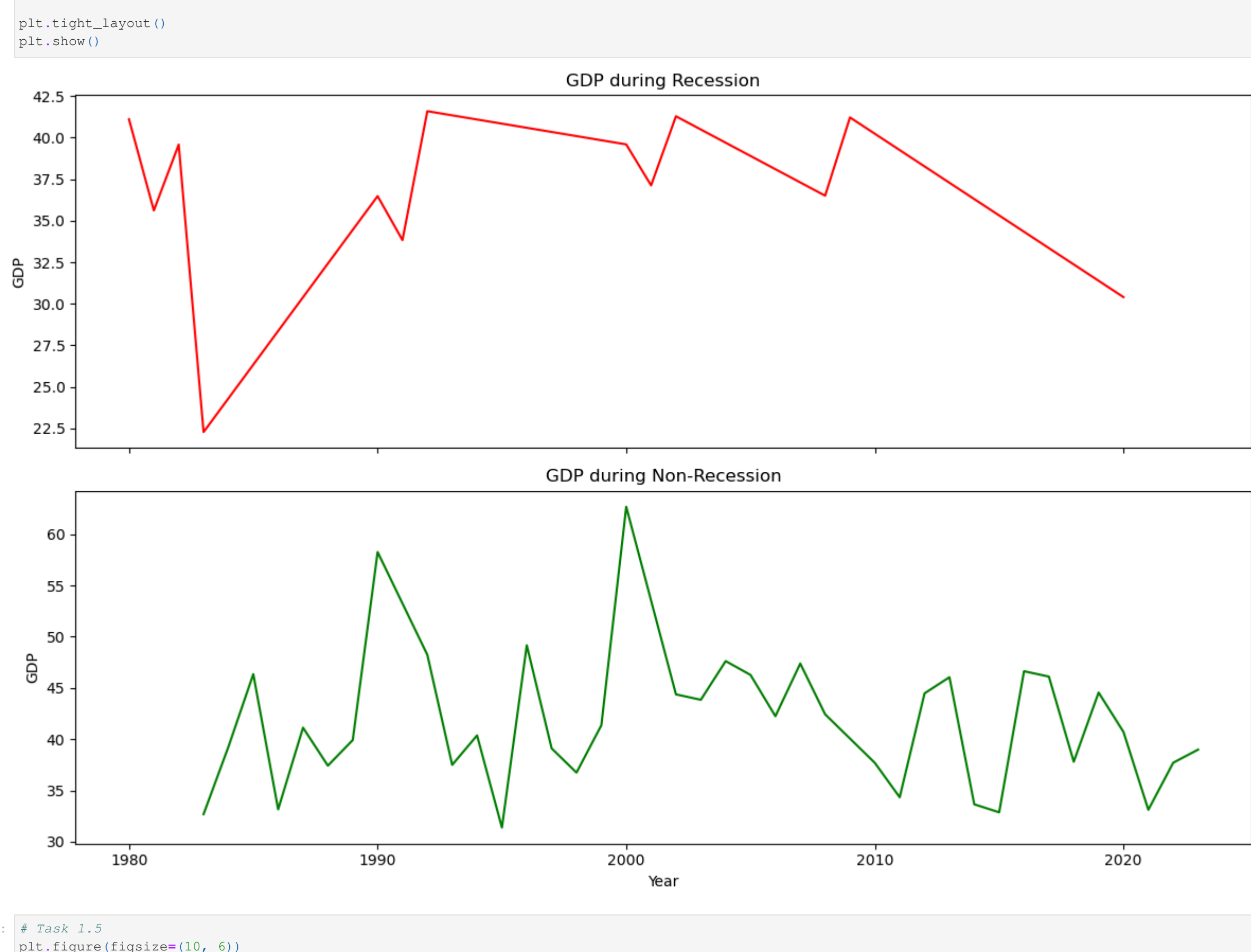


```
In [17]: # Task 1.4
fig, axes = plt.subplots(2, 1, figsize=(12, 8), sharex=True)
```

```
recession_data = df[df['Recession'] == 1]
non_recession_data = df[df['Recession'] == 0]

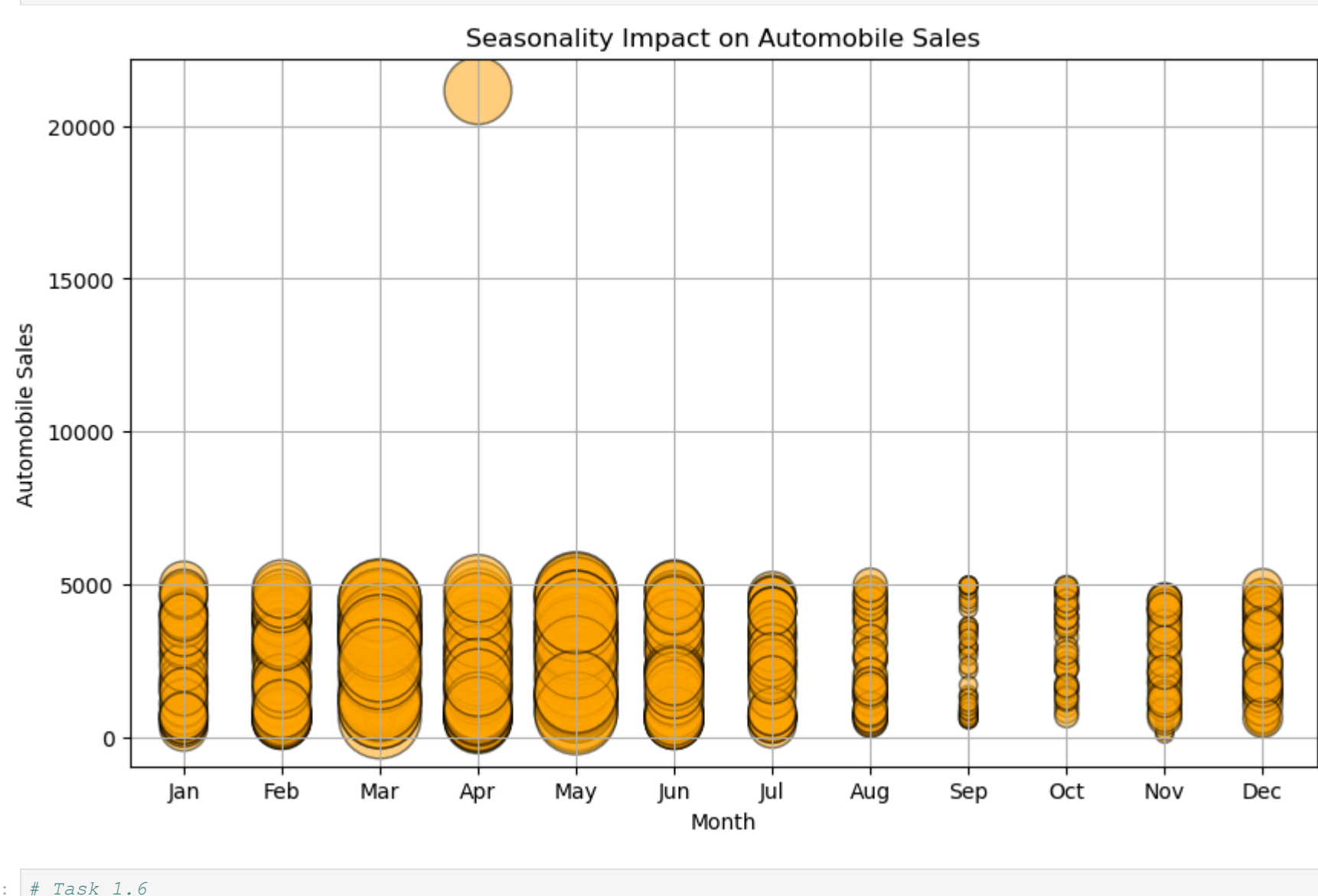
recession_data.groupby('Year')['GDP'].mean().plot(ax=axes[0], title='GDP during Recession', color='red')
axes[0].set_ylabel('GDP')

non_recession_data.groupby('Year')['GDP'].mean().plot(ax=axes[1], title='GDP during Non-Recession', color='green')
axes[1].set_ylabel('GDP')
axes[1].set_xlabel('Year')
```



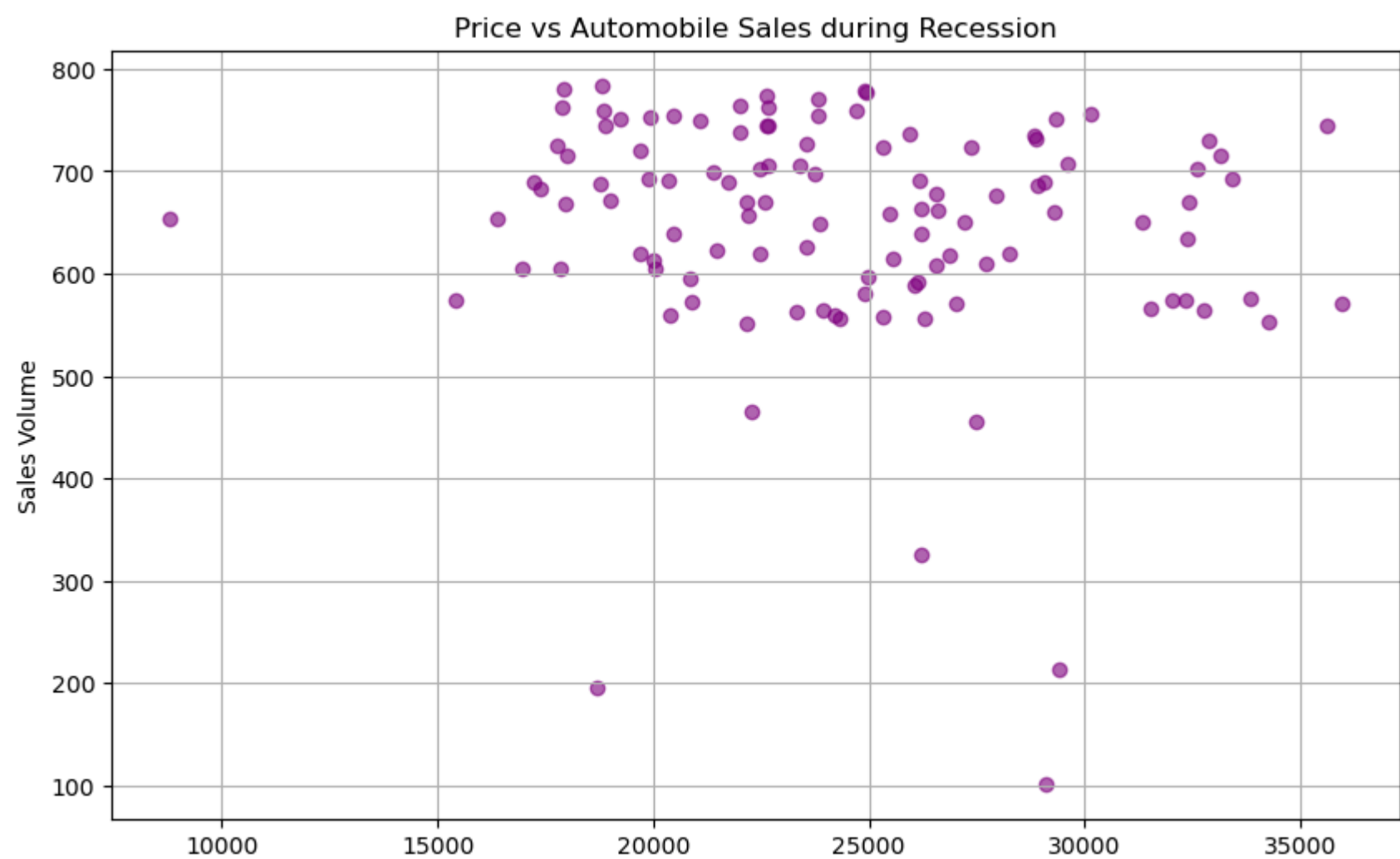
```
In [19]: # Task 1.5
plt.figure(figsize=(10, 6))
```

```
plt.scatter(df['Month'], df['Automobile_Sales'], s=df['Seasonality_Weight'] * 1000, alpha=0.5, c='orange', edgecolors='black')
plt.title('Seasonality Impact on Automobile Sales')
plt.xlabel('Month')
plt.ylabel('Automobile Sales')
plt.grid(True)
plt.show()
```



```
In [21]: # Task 1.6
recession_df = df[df['Recession'] == 1]
```

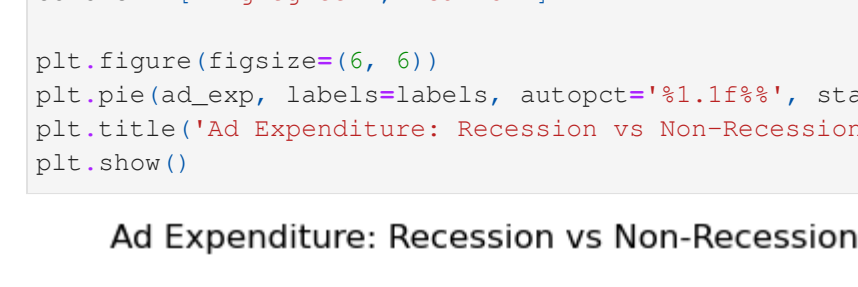
```
plt.figure(figsize=(10, 6))
plt.scatter(recession_df['Price'], recession_df['Automobile_Sales'], color='purple', alpha=0.6)
plt.title('Price vs Automobile Sales during Recession')
plt.xlabel('Average Vehicle Price')
plt.ylabel('Sales Volume')
plt.grid(True)
plt.show()
```



```
In [23]: # Task 1.7
ad_exp = df.groupby('Recession')['Advertising_Expenditure'].sum()
```

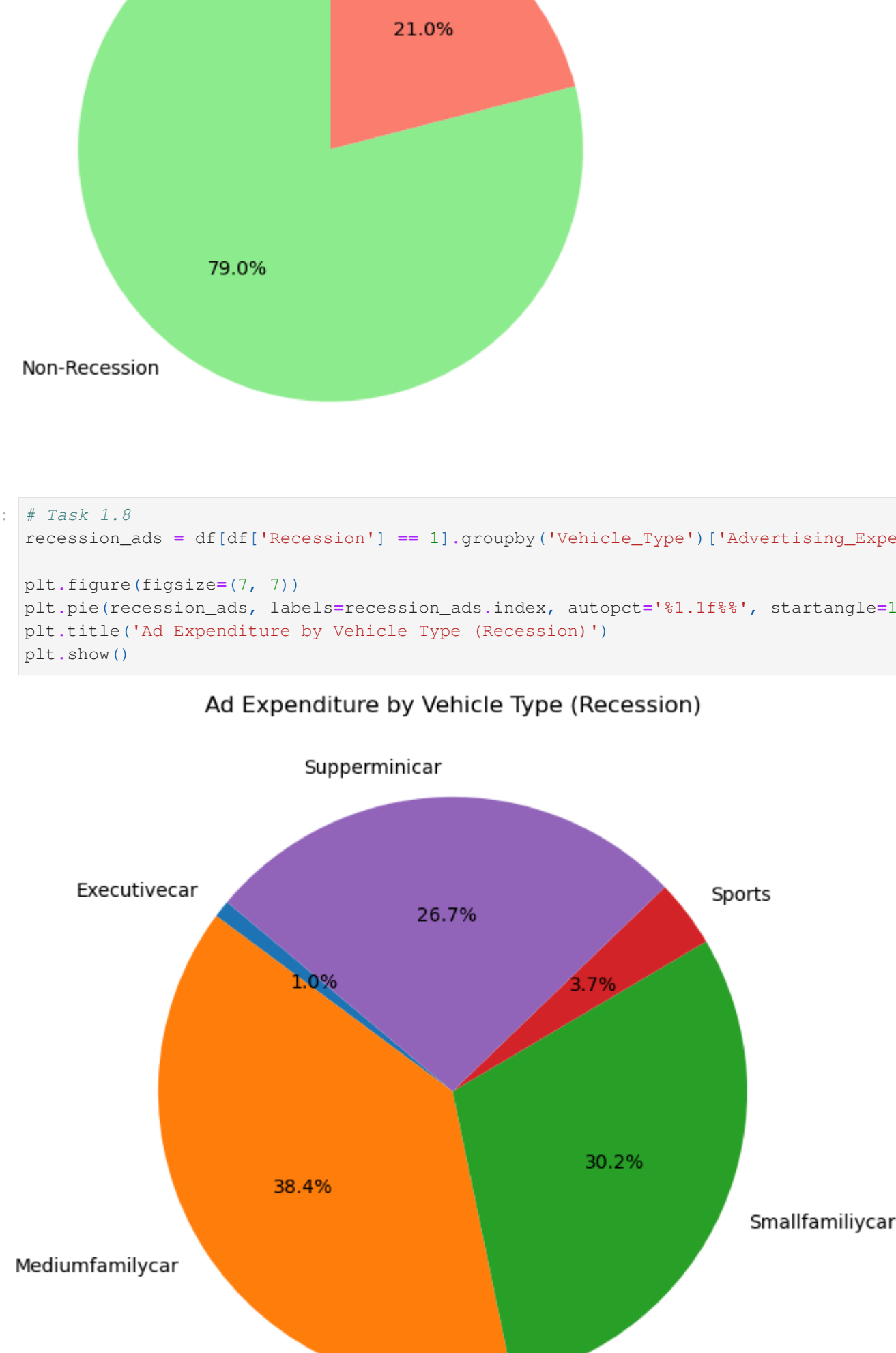
```
labels = ['Non-Recession', 'Recession']
colors = ['lightgreen', 'salmon']

plt.figure(figsize=(6, 6))
plt.pie(ad_exp, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors)
plt.title('Ad Expenditure: Recession vs Non-Recession')
```



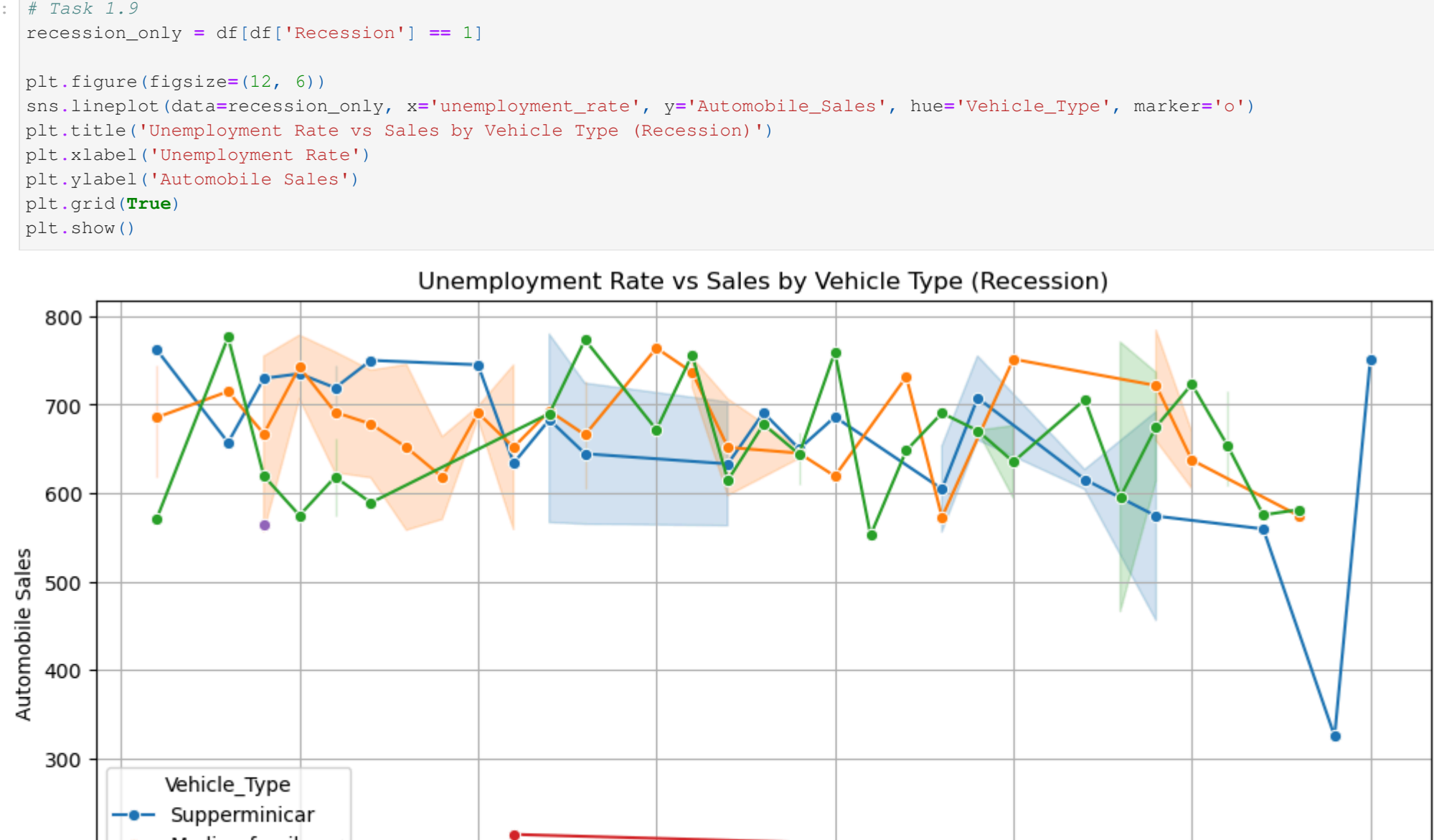
```
In [25]: # Task 1.8
recession_ads = df[df['Recession'] == 1].groupby('Vehicle_Type')['Advertising_Expenditure'].sum()
```

```
plt.figure(figsize=(7, 7))
plt.pie(recession_ads, labels=recession_ads.index, autopct='%1.1f%%', startangle=140)
plt.title('Ad Expenditure by Vehicle Type (Recession)')
```



```
In [27]: # Task 1.9
recession_only = df[df['Recession'] == 1]
```

```
plt.figure(figsize=(12, 6))
sns.lineplot(data=recession_only, x='unemployment_rate', y='Automobile_Sales', hue='Vehicle_Type', marker='o')
plt.title('Unemployment Rate vs Sales by Vehicle Type (Recession)')
```



```
In [31]: import dash
```

```
from dash import html, dcc, Output, Input
import pandas as pd
import plotly.express as px
```

```
In [ ]: app = dash.Dash(__name__)
```

```
app.title = 'XYZ Automotives Sales Dashboard'

app.layout = html.Div([
    html.H1('XYZ Automotives Sales Analysis', style={'textAlign': 'center'})
])
```

```
if __name__ == '__main__':
    app.run_server(debug=True)
```

```
In [ ]: from dash import dcc
```

```
vehicle_types = ['Suppeminicar', 'Mediumfamilycar', 'Smallfamilycar'] # Example list
years = [1980, 1981, 1982] # Example list
```

```
app.layout = html.Div([
    html.H1('XYZ Automotives Sales Analysis', style={'textAlign': 'center'}),
```

```
    html.Div([
        html.Label('Select Vehicle Type:'),
        dcc.Dropdown(
            id='vehicle-dropdown',
            options=[{'label': v, 'value': v} for v in vehicle_types],
            value=vehicle_types[0]
        ),
        style={'width': '45%', 'display': 'inline-block'}),
```

```
    html.Div([
        html.Label('Select Year:'),
        dcc.Dropdown(
            id='year-dropdown',
            options=[{'label': y, 'value': y} for y in years],
            value=years[0]
        ),
        style={'width': '45%', 'display': 'inline-block', 'marginLeft': '5%'}
    )
])
```

```
In [ ]: app.layout = html.Div([
```

```
    html.H1('XYZ Automotives Sales Analysis', style={'textAlign': 'center'}),

    # Dropdowns here...

    html.Div(id='output-container', className='output-class', style={'marginTop': 30})
])
```

```
In [ ]: from dash import Output, Input
```

```
@app.callback(
    Output('output-container', 'children'),
    Input('vehicle-dropdown', 'value'),
    Input('year-dropdown', 'value')
)
```

```
def update_output(vehicle_type, year):
    return f'Selected Vehicle: {vehicle_type}, Year: {year}'
```

```
In [ ]: import plotly.express as px
```

```
import pandas as pd

df = pd.read_csv('historical_automobile_sales.csv')
```

```
@app.callback(
    Output('recession-graph', 'figure'),
    Input('recession-dropdown', 'value')
)
```

```
def update_recession_graph(vehicle_type):
    filtered = df[(df['Recession'] == 1) & (df['Vehicle_Type'] == vehicle_type)]
    fig = px.line(filtered, x='Year', y='Automobile_Sales', title='Recession Period Sales')
    return fig
```

```
# Add this to layout
dcc.Graph(id='recession-graph')
```

```
In [ ]: @app.callback(
```

```
    Output('yearly-graph', 'figure'),
    Input('year-dropdown', 'value')
)
```

```
def update_yearly_graph(year):
    filtered = df[(df['Year'] == year)]
    group_data = filtered.groupby('Vehicle_Type')['Automobile_Sales'].sum().reset_index()
    fig = px.bar(group_data, x='Vehicle_Type', y='Automobile_Sales', title=f'Sales in {year}')
```



```
return fig

# Add this in layout
doc.Graph(id='yearly-graph')
```