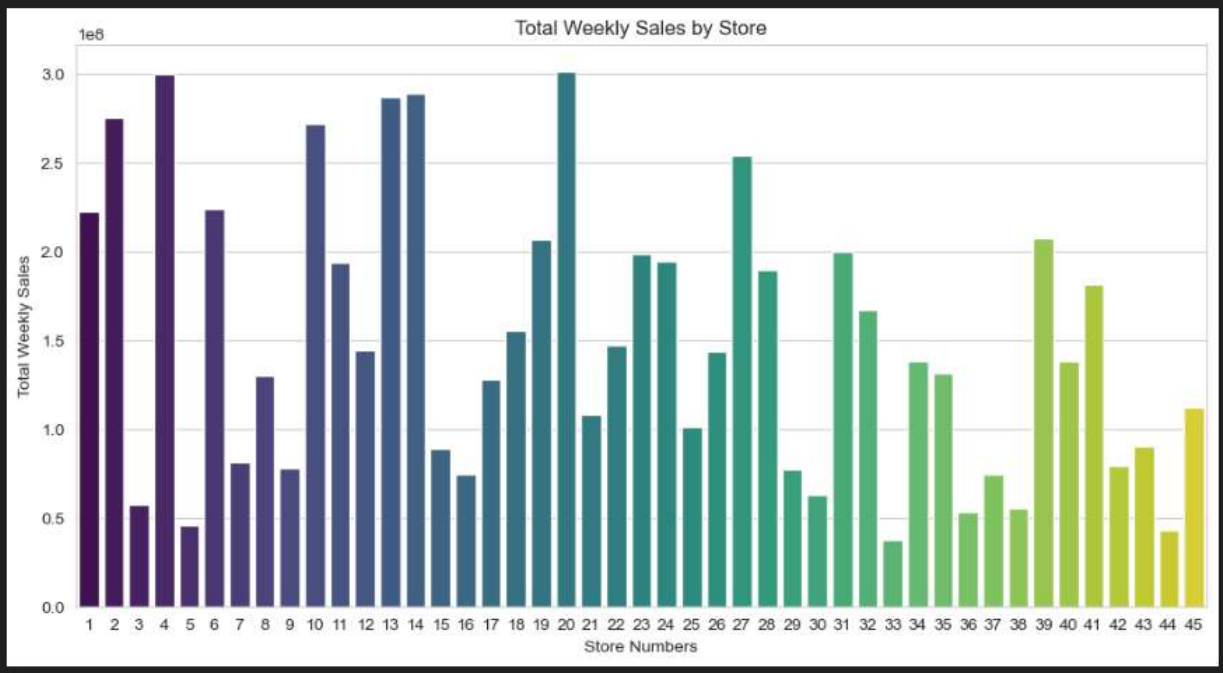
```python
# Group by store & sum weekly sales
store_sales = df.groupby('Store')['Weekly_Sales'].sum().sort_values(ascending=False).reset_index()

#plot
plt.figure(figsize=(12, 6))
sns.barplot(x='Store', y='Weekly_Sales', data=store_sales, palette='viridis')
plt.title('Total Weekly Sales by Store')
plt.xlabel('Store Numbers')
plt.ylabel('Total Weekly Sales')
plt.show()
```

✓ 1.0s

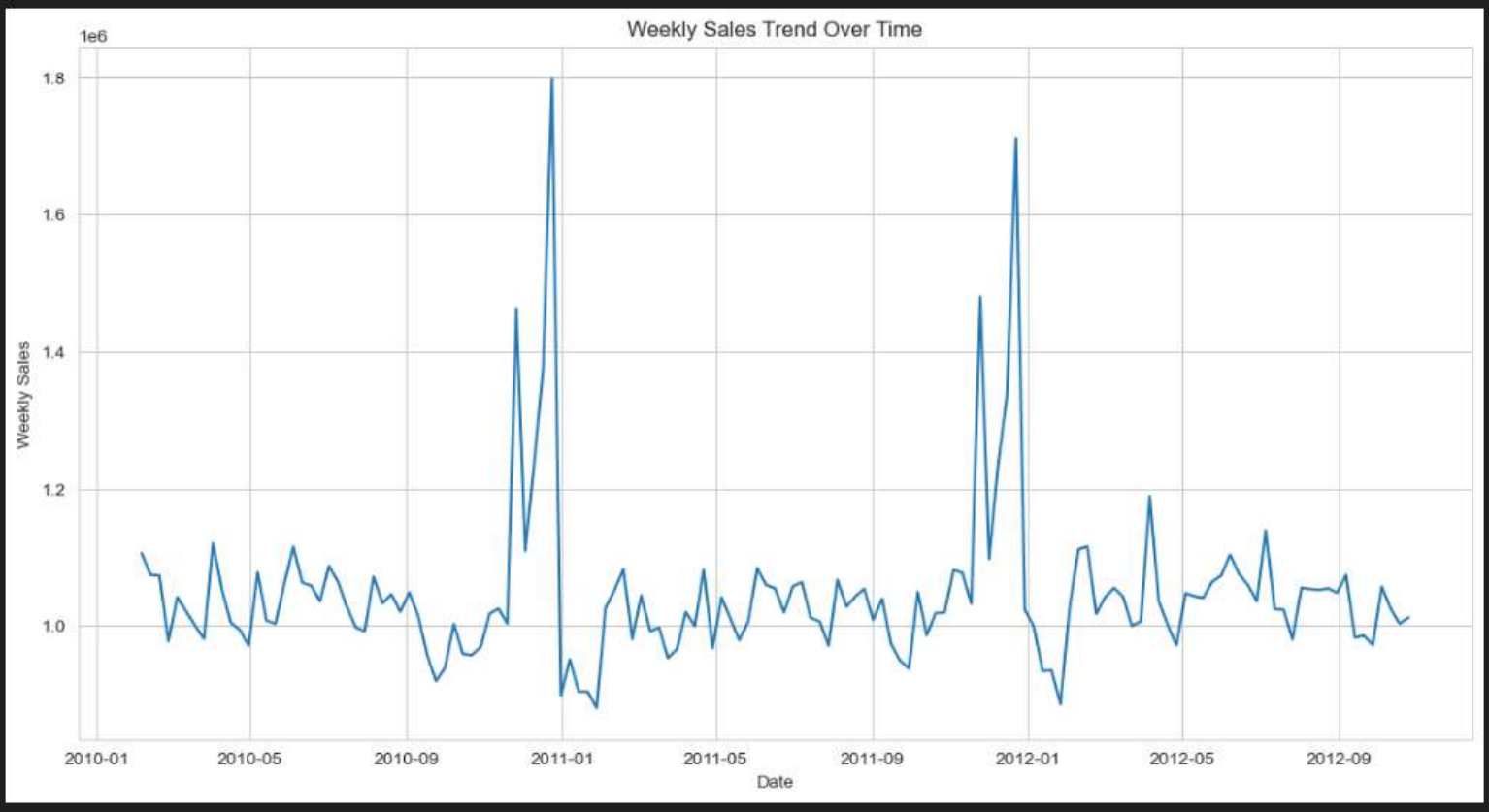
```
C:\Users\mmamt\AppData\Local\Temp\ipykernel_36652\417666500.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Store', y='Weekly_Sales', data=store_sales, palette='viridis')
```



```python
# Line chart for overall sales trends

plt.figure(figsize=(14, 7))
sns.lineplot(data=df, x='Date', y='Weekly_Sales', ci=None)
plt.title('Weekly Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Weekly Sales')
plt.show()
```

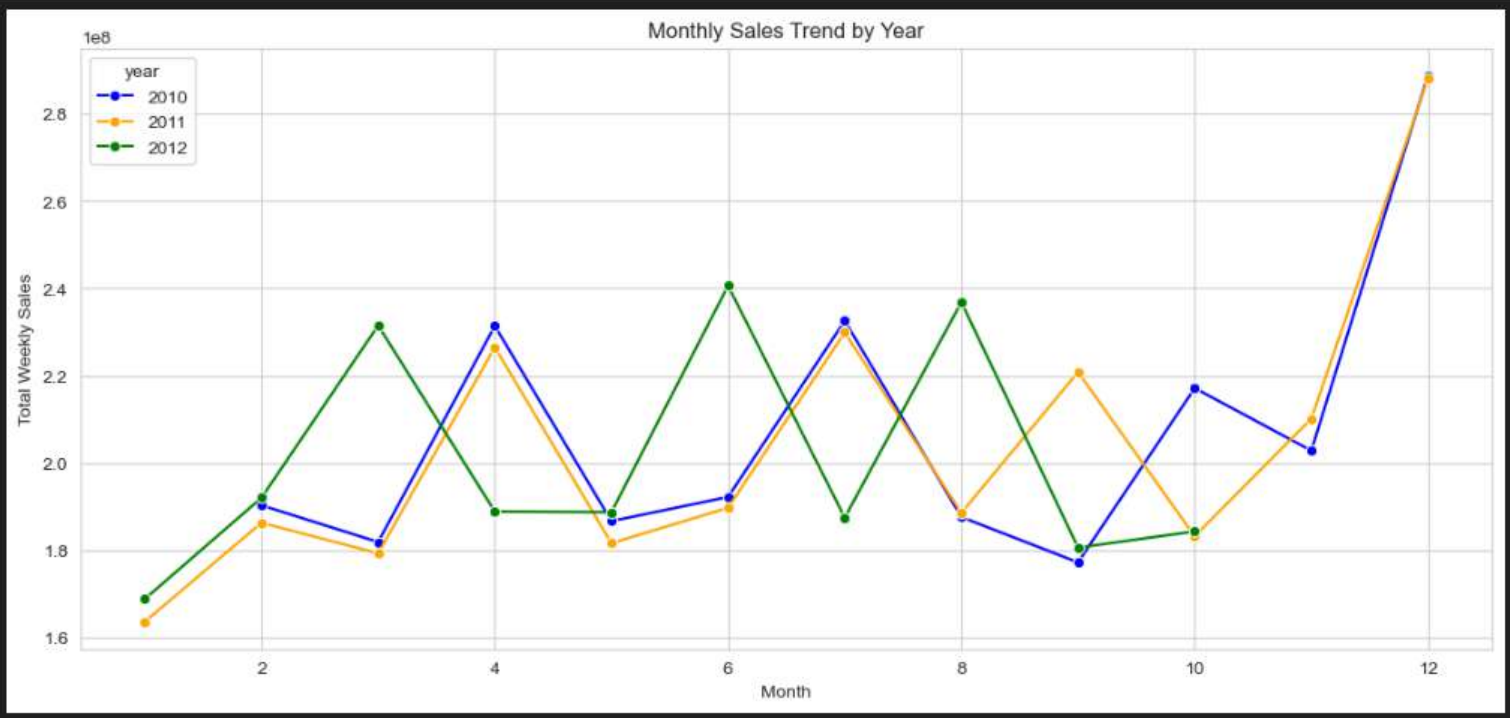
✓ 0.5s

```
C:\Users\mmamt\AppData\Local\Temp\ipykernel_36652\1836221953.py:4: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=df, x='Date', y='Weekly_Sales', ci=None)
```



```python
# Create month and year columns if not already done
df['month']= df['Date'].dt.month
df['year'] = df['Date'].dt.year

# Group By Year and Month
monthly_sales = df.groupby(['year', 'month'])['Weekly_Sales'].sum().reset_index()

#Line chart by year
custom_colors={
    2010: 'blue',
    2011: 'orange',
    2012: 'green'}

plt.figure(figsize=(14, 6))
sns.lineplot(data=monthly_sales, x='month', y='Weekly_Sales', hue='year', marker='o',palette=custom_colors)
plt.title('Monthly Sales Trend by Year')
plt.xlabel('Month')
plt.ylabel('Total Weekly Sales')
plt.show()
```

✓ 0.5s

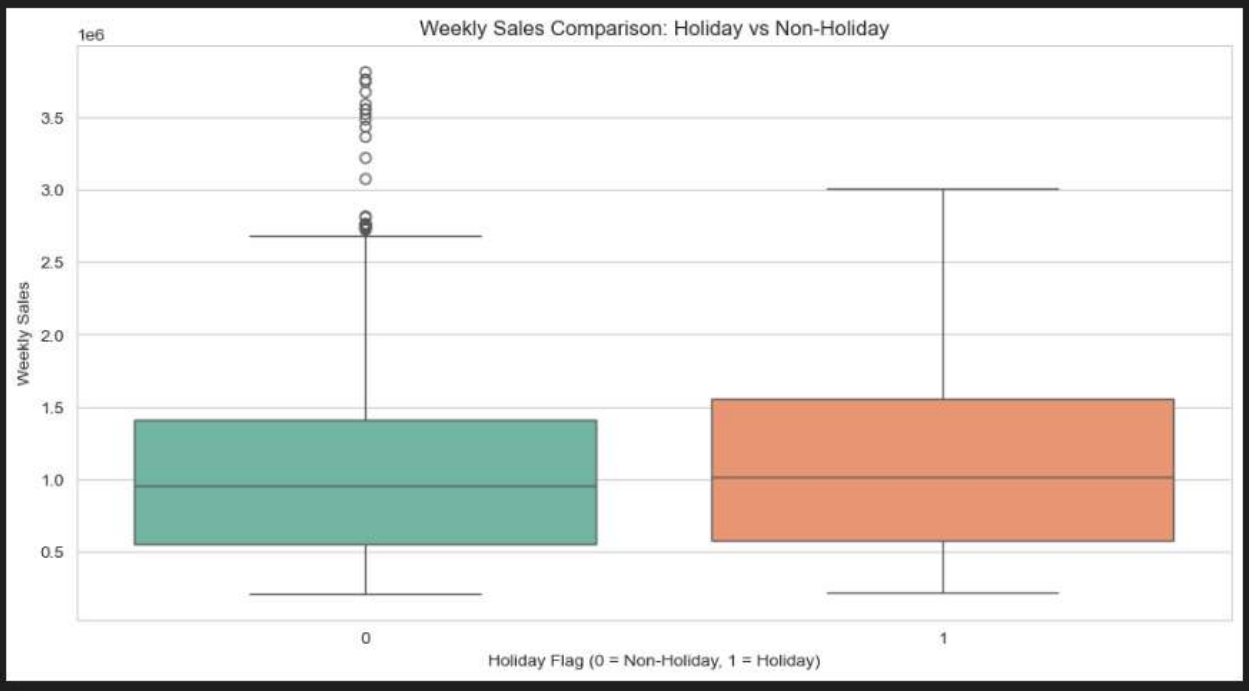


. Sales : holiday VS Non-Holiday

```
#Boxplot comparison of holiday and non-holiday sales
plt.figure(figsize=(12, 6))
sns.boxplot(x='Holiday_Flag', y='Weekly_Sales', data=df, palette='Set2')
plt.title('Weekly Sales Comparison: Holiday vs Non-Holiday')
plt.xlabel('Holiday Flag (0 = Non-Holiday, 1 = Holiday)')
plt.ylabel('Weekly Sales')
plt.show()
```

✓ 0.2s

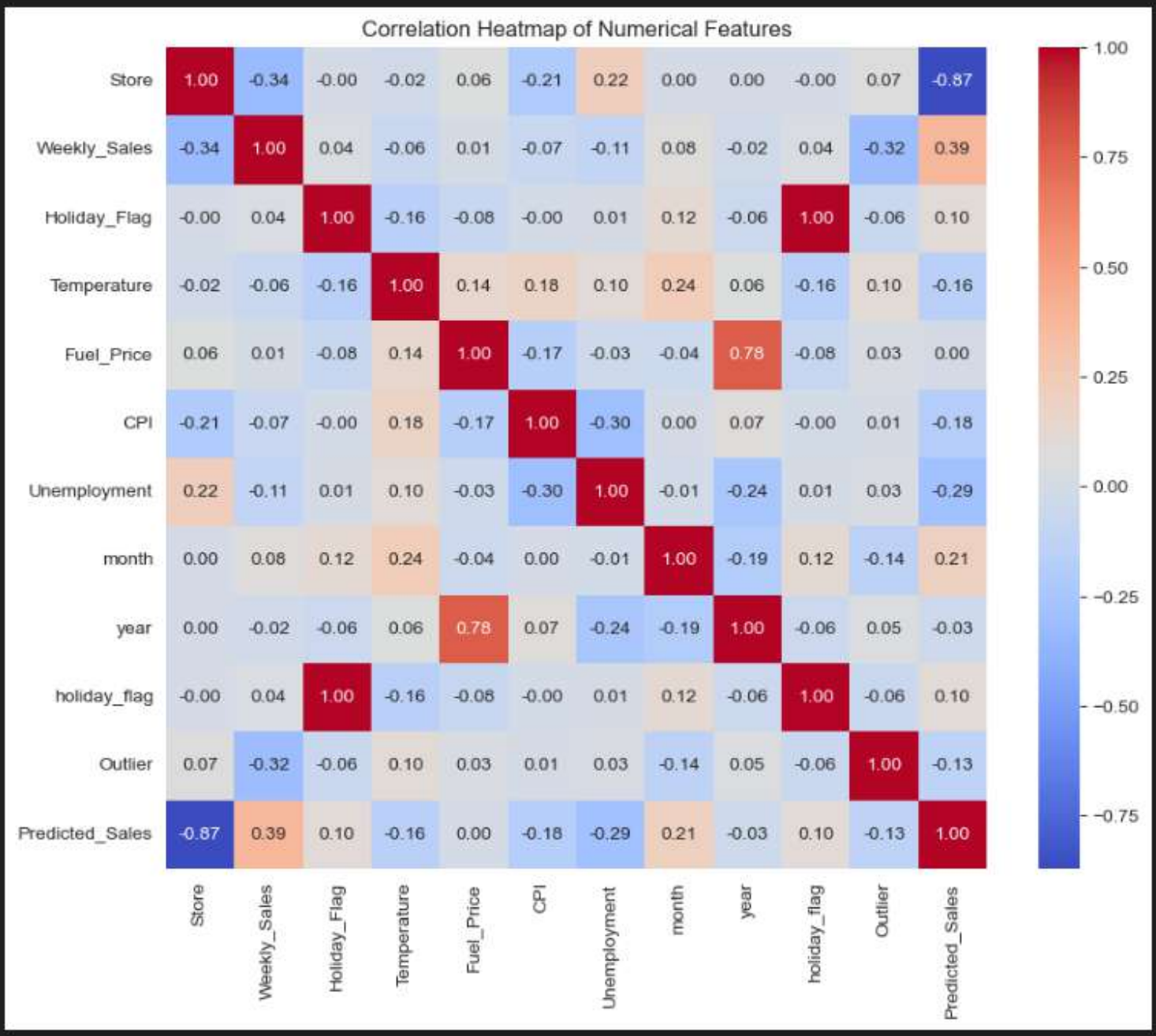
```
C:\Users\mmamt\AppData\Local\Temp\ipykernel_36652\1483077897.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x='Holiday_Flag', y='Weekly_Sales', data=df, palette='Set2')
```



vi. Corrrelation Heatmap

```
# Correlation heatmap between numerical features
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm',fmt='.2f',  square=True)
plt.title('Correlation Heatmap of Numerical Features')
plt.show()
```

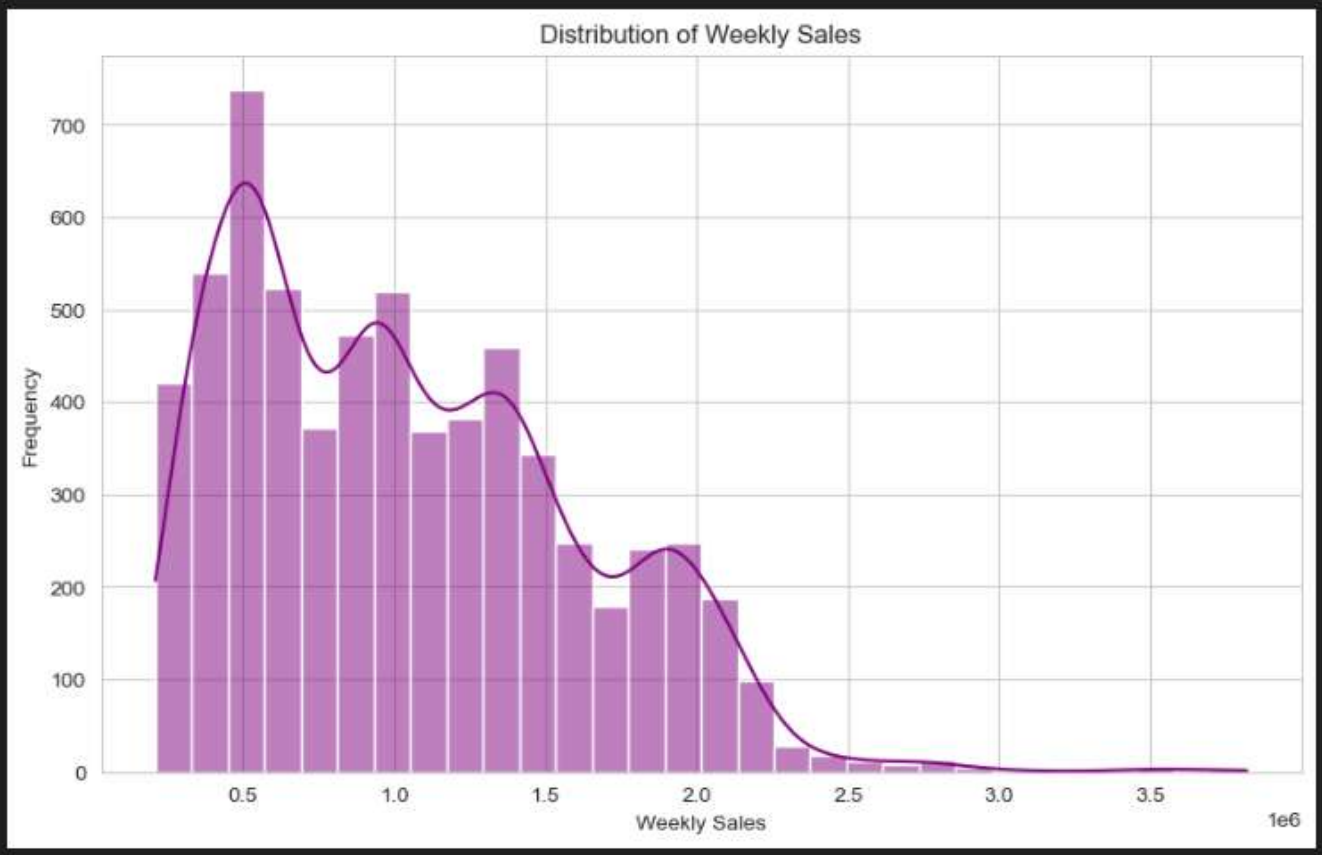
✓ 0.7s



vii. Weekly Sales Distributiion (Chech outliers)

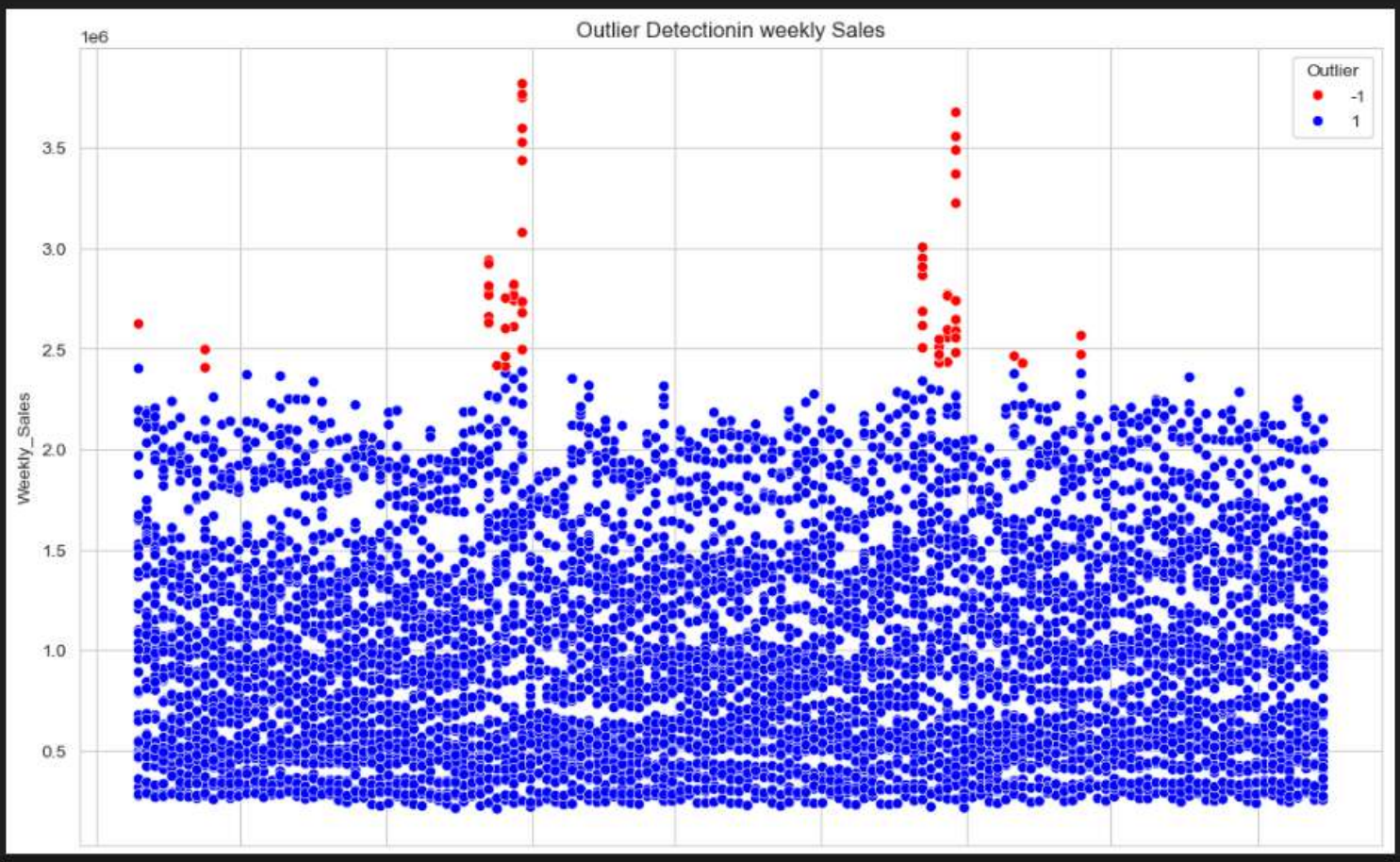
```
# Distribution plot of Weekly Sales
plt.figure(figsize=(10, 6))
sns.histplot(df['Weekly_Sales'], bins=30, kde=True, color='purple')
plt.title('Distribution of Weekly Sales')
plt.xlabel('Weekly Sales')
plt.ylabel('Frequency')
plt.show()
```

✓ 0.3s



```python
# ii. Visualization Outliers

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(13,8))
sns.scatterplot(data=df,x='Date', y='Weekly_Sales' ,hue='Outlier',palette={1:'blue',-1:'red'})
plt.title("Outlier Detectionin weekly Sales")
plt.show()
```

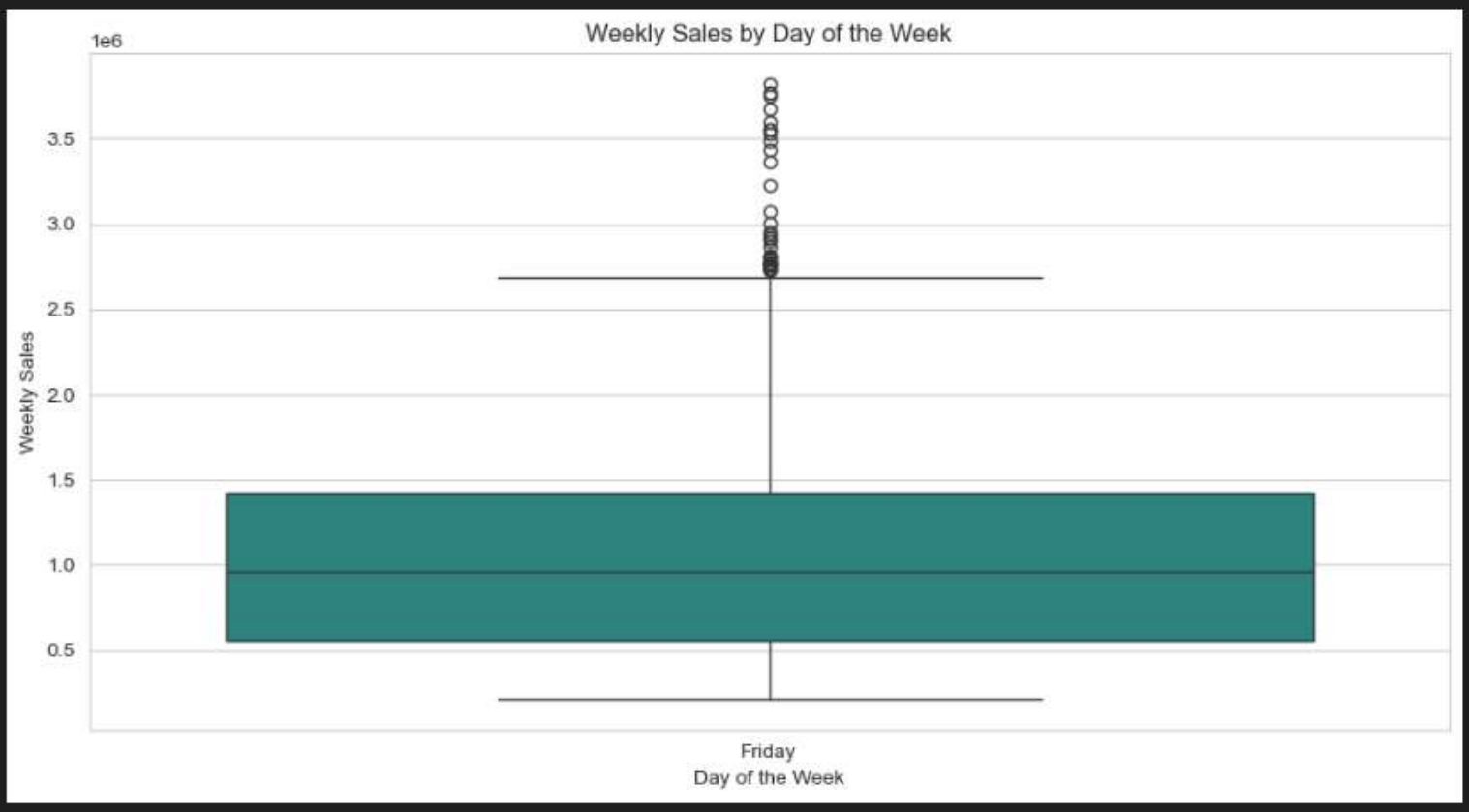
✓ 0.8s



```
plt.show()
```

✓ 0.2s

```
C:\Users\mmamt\AppData\Local\Temp\ipykernel_36652\2433753423.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.boxplot(x='day_of_week', y='Weekly_Sales', data=df, palette='viridis')
```



```python
#correct a copy for simulation
df_simulated=df.copy()


#increase sales by 10% during holidays
df_simulated['simulated_sales']=df_simulated['Weekly_Sales']
df_simulated.loc[df_simulated['Weekly_Sales']==1,'simulated_sales'] *=1.10
```

✓ 0.0s

```python
# Visualiza Before And After
plt.figure(figsize=(12,6))
sns.lineplot(data=df_simulated, x="Date",y="Weekly_Sales",label="Actual")
sns.lineplot(data=df_simulated, x="Date",y="simulated_sales",label="Simulated(10%Boost on Holidays)")
plt.title("Holiday Sales Simulations")
plt.legend()
plt.show()
```

✓ 7.5s