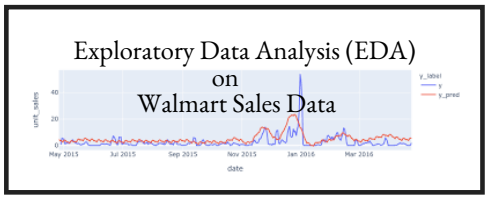


[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:
```
df=pd.read_csv("/content/drive/MyDrive/DATASETS/Walmart.csv")
```

[ ]:
```
df
```

[ ]:
```
      Store        Date  Weekly_Sales  Holiday_Flag  Temperature  Fuel_Price  \
0         1  05-02-2010    1643690.90             0        42.31       2.572
1         1  12-02-2010    1641957.44             1        38.51       2.548
2         1  19-02-2010    1611968.17             0        39.93       2.514
3         1  26-02-2010    1409727.59             0        46.63       2.561
4         1  05-03-2010    1554806.68             0        46.50       2.625
…       …           …             …             …            …           …
6430     45  28-09-2012     713173.95             0        64.88       3.997
6431     45  05-10-2012     733455.07             0        64.89       3.985
6432     45  12-10-2012     734464.36             0        54.47       4.000
6433     45  19-10-2012     718125.53             0        56.47       3.969
6434     45  26-10-2012     760281.43             0        58.85       3.882

             CPI  Unemployment
0     211.096358         8.106
1     211.242170         8.106
2     211.289143         8.106
3     211.319643         8.106
4     211.350143         8.106
…            …             …
6430  192.013558         8.684
6431  192.170412         8.667
6432  192.327265         8.667
6433  192.330854         8.667
6434  192.308899         8.667

[6435 rows x 8 columns]
```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   object
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

```
[ ]: df['Date']=df['Date'].astype('datetime64[ns]')
```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   datetime64[ns]
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(2)
memory usage: 402.3 KB
```

--The dataset consists of 6,435 rows and 8 columns. Each row represents a week's sales data for a specific store. The columns are as follows:

1.**Store**: Identifies the store number (from 1 to 45).

2.**Date**: The week's end date for the data, in the format YYYY-MM-DD.

3.**Weekly_Sales**: The total sales amount for the store in that week (in dollars).

4.**Holiday_Flag**: A binary flag indicating whether the week corresponds to a holiday (1 = Holiday, 0 = Non-Holiday).

5.**Temperature**: The average temperature for the week in Fahrenheit.

6.**Fuel_Price**: The average fuel price for the week (in dollars per gallon).

7.**CPI (Consumer Price Index)**: An economic indicator showing the relative price level of goods and services in the economy.

7.**Unemployment**: The unemployment rate for the week (as a percentage).

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

[ ]:
```
              Store                           Date  Weekly_Sales  Holiday_Flag  \
count  6435.000000                           6435  6.435000e+03   6435.000000
mean     23.000000  2011-06-17 20:18:27.692307712  1.046965e+06      0.069930
min       1.000000            2010-01-10 00:00:00  2.099862e+05      0.000000
25%      12.000000            2010-10-12 00:00:00  5.533501e+05      0.000000
50%      23.000000            2011-06-17 00:00:00  9.607460e+05      0.000000
75%      34.000000            2012-03-02 00:00:00  1.420159e+06      0.000000
max      45.000000            2012-12-10 00:00:00  3.818686e+06      1.000000
std      12.988182                            NaN  5.643666e+05      0.255049

       Temperature   Fuel_Price          CPI  Unemployment
count  6435.000000  6435.000000  6435.000000   6435.000000
mean     60.663782     3.358607   171.578394      7.999151
min      -2.060000     2.472000   126.064000      3.879000
25%      47.460000     2.933000   131.735000      6.891000
50%      62.670000     3.445000   182.616521      7.874000
75%      74.940000     3.735000   212.743293      8.622000
max     100.140000     4.468000   227.232807     14.313000
std      18.444933     0.459020    39.356712      1.875885
```

Key Observations from the Descriptive Statistics:

1.**Store**:

There are 45 unique stores (ranging from 1 to 45). The mean store number is around 23, which suggests the data is roughly evenly distributed across the stores.

2.**Date**:

The dataset spans from January 10, 2010, to December 10, 2012. The dates are evenly distributed across this time period, with the mean date being June 17, 2011.

3.**Weekly Sales**:

The mean weekly sales are approximately 1,046,965. The min value is around 209,986, and the max value is 3,818,686, indicating that there is a large variation in sales across different weeks. The standard deviation (std) is quite high at 564,366, showing significant variability in weekly sales.

4.**Holiday Flag**:

The mean of the holiday flag is 0.07, suggesting that holidays are relatively rare, but present occasionally in the dataset.

5.**Temperature**:

The temperature ranges from -2.06°C to 100.14°C, with an average temperature of 60.66°C. There seems to be some extreme values (like the -2.06°C), which could be worth investigating further for data quality.

6.**Fuel Price**:

The fuel price ranges from 2.472 to 4.468 with an average value of 3.36. The variability is not very high here, with a standard deviation of 0.46.

7.**CPI (Consumer Price Index)**:

The CPI ranges from 126.06 to 227.23, with an average of 171.58. The spread of CPI values is wider, indicating variation in economic conditions during the dataset period.

8.**Unemployment**:

The unemployment rate ranges from 3.88% to 14.31%, with an average value of 8.00%. This aligns with typical unemployment rates seen over the given period

DATA SUMMARY

Time Range: The data spans from January 10, 2010, to December 10, 2012.

Stores: The dataset includes 45 unique stores, each with weekly sales recorded.

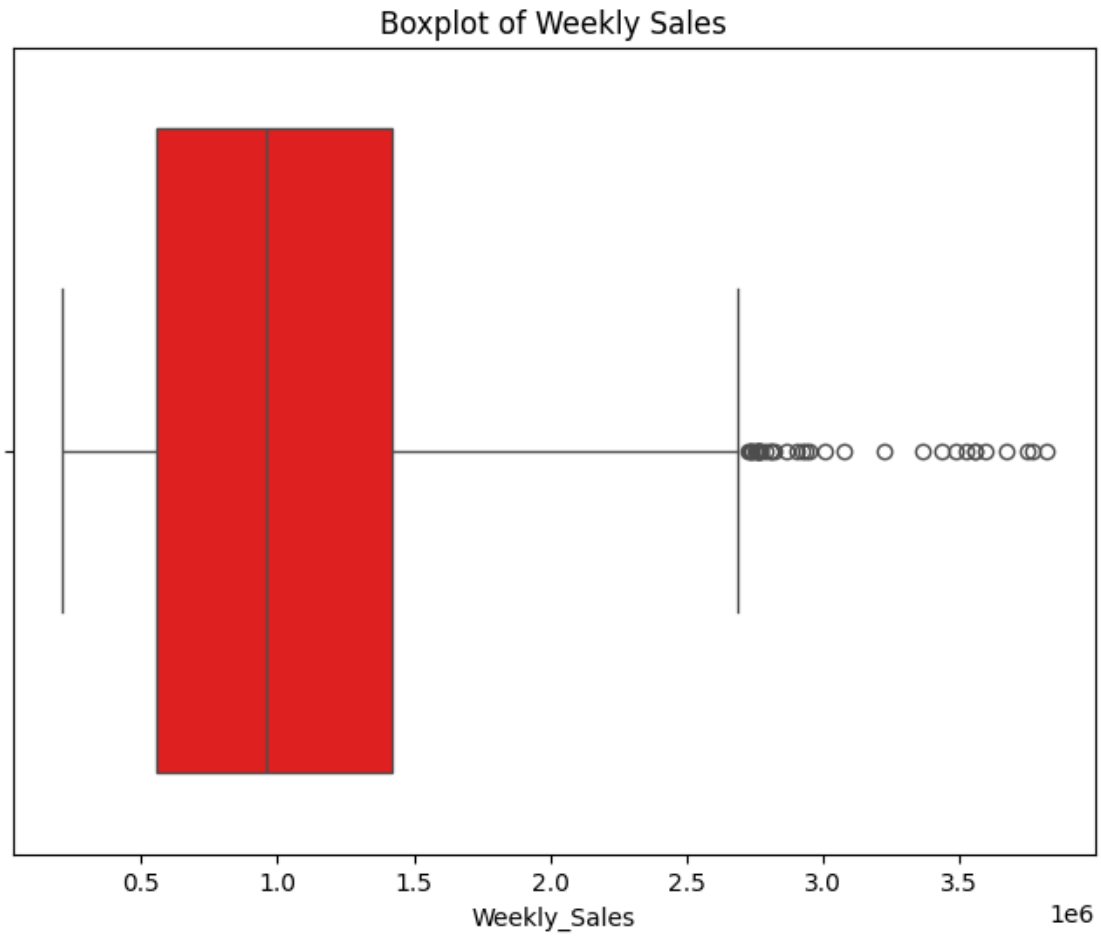
Target Variable: The primary target variable is Weekly_Sales, which we aim to analyze and forecast based on the other features.

[ ]:
```
df.isna().sum()
```

[ ]:
```
Store           0
Date            0
Weekly_Sales    0
Holiday_Flag    0
Temperature     0
Fuel_Price      0
CPI             0
Unemployment    0
dtype: int64
```

--There are no missing values in the dataset, Lets go forward with the analysis

[ ]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Weekly_Sales'],color='red')
plt.title('Boxplot of Weekly Sales')
plt.show()
```

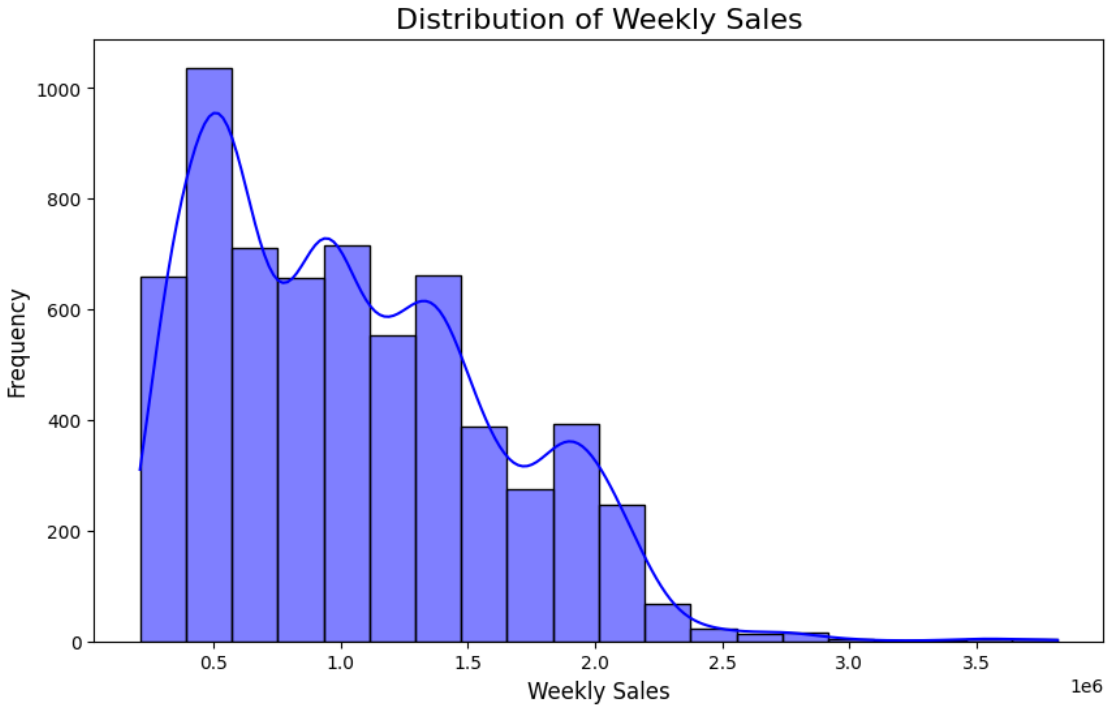


--Key Observations:

Sales Distribution: Most weekly sales are centered around the 1 million mark, with some weeks showing much higher or lower sales.

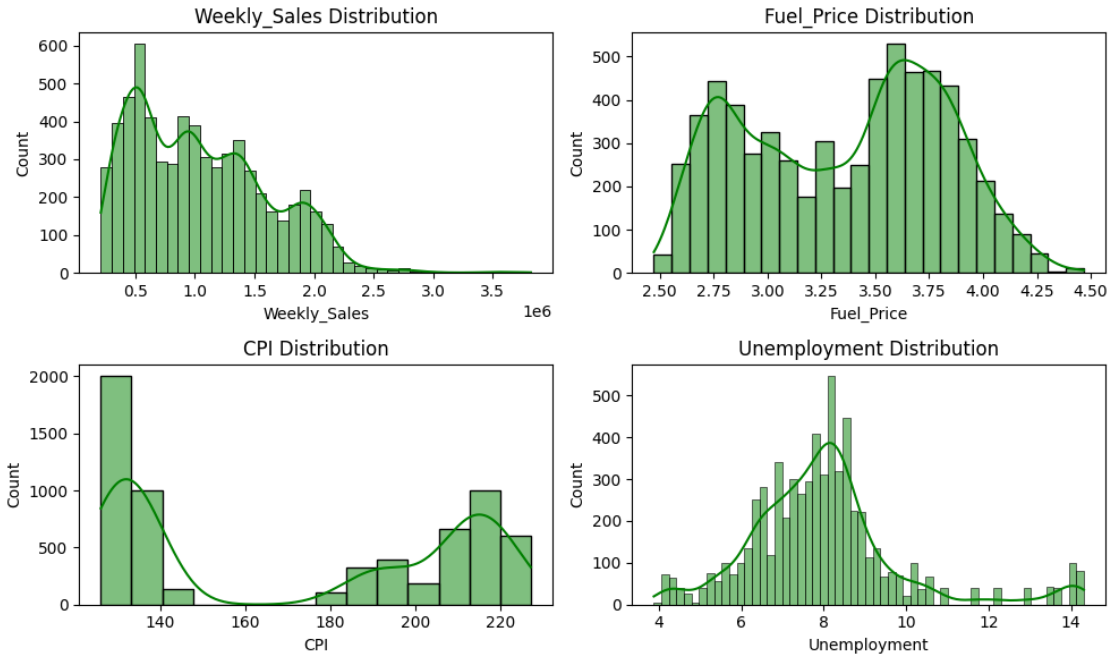
Potential Outliers: There are several weeks with exceptionally high sales, which are considered outliers.

[ ]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['Weekly_Sales'], bins=20, kde=True, color='blue')

plt.title('Distribution of Weekly Sales', fontsize=16)
plt.xlabel('Weekly Sales', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

plt.show()
```



--Subplots with Histograms

[ ]:
```
columns_to_plot = ["Weekly_Sales", "Fuel_Price", "CPI", "Unemployment"]

fig, axes = plt.subplots(2, 2, figsize=(10, 6))
for ax, col in zip(axes.ravel(), columns_to_plot):
    sns.histplot(df[col], kde=True, ax=ax, color="green")
    ax.set_title(f'{col} Distribution')

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



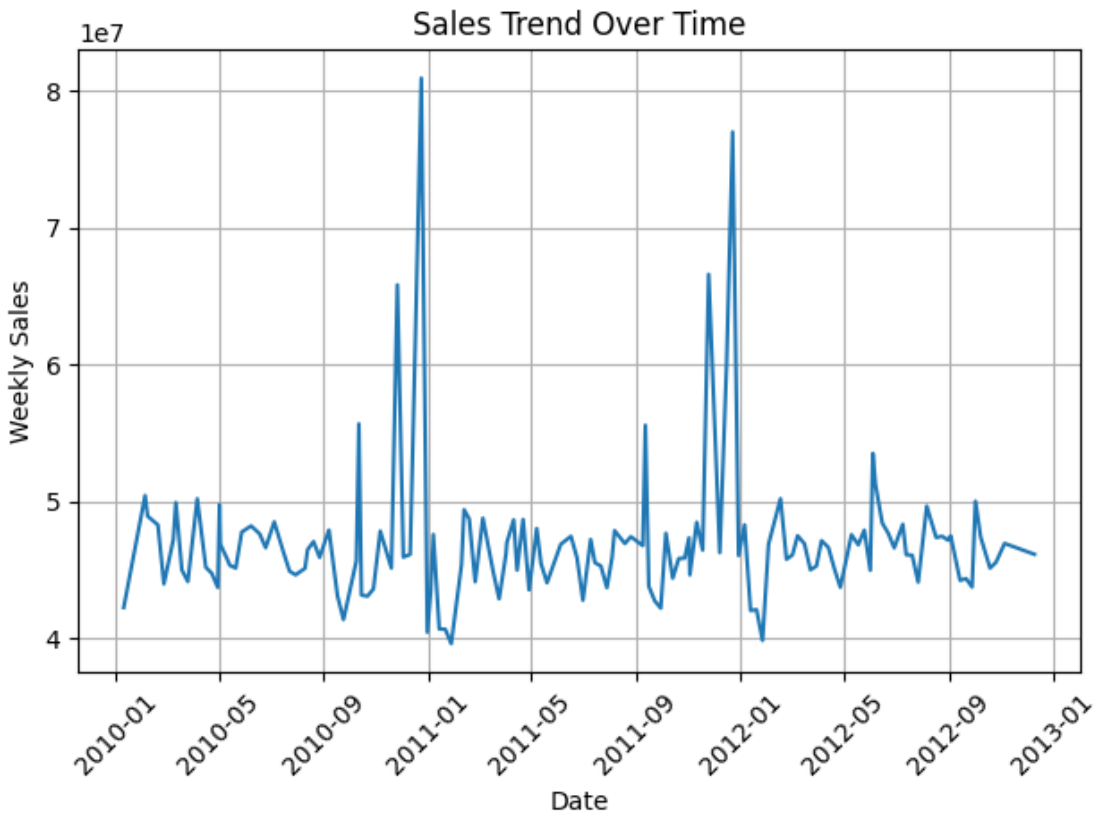
As you can see in these histograms, Weekly Sales tend to be concentrated towards the lower end, while Fuel Price, CPI, and Unemployment exhibit more bell-shaped distributions.

This suggests that most weeks have relatively lower sales, while fuel prices, inflation, and unemployment rates tend to cluster around a central value.

The absence of significant outliers indicates that the data is relatively consistent and representative of typical values.

LINE PLOT

[ ]:
```
weekly_sales = df.groupby('Date')['Weekly_Sales'].sum()

# Plot the sales trend over time
plt.plot(weekly_sales.index, weekly_sales.values)
plt.title('Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Weekly Sales')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```



-Key Features and Observations for the Sales Trend Over Time:

Peaks and Troughs:

The high peaks in sales correspond to periods of increased activity. These peaks may indicate promotional events, holidays, or seasonal sales periods that significantly boost sales.

The end of 2010 and end of 2011 show notably high sales, suggesting that the retailer might have experienced holiday shopping spikes, special promotions, or other events that led to increased sales during those times.

Seasonality:

The regular fluctuations in sales might suggest seasonal patterns. Sales could be higher during certain times of the year (e.g., holidays or seasonal promotions like Black Friday, Christmas, etc.) and lower during off-peak months.

Sales Growth or Decline :

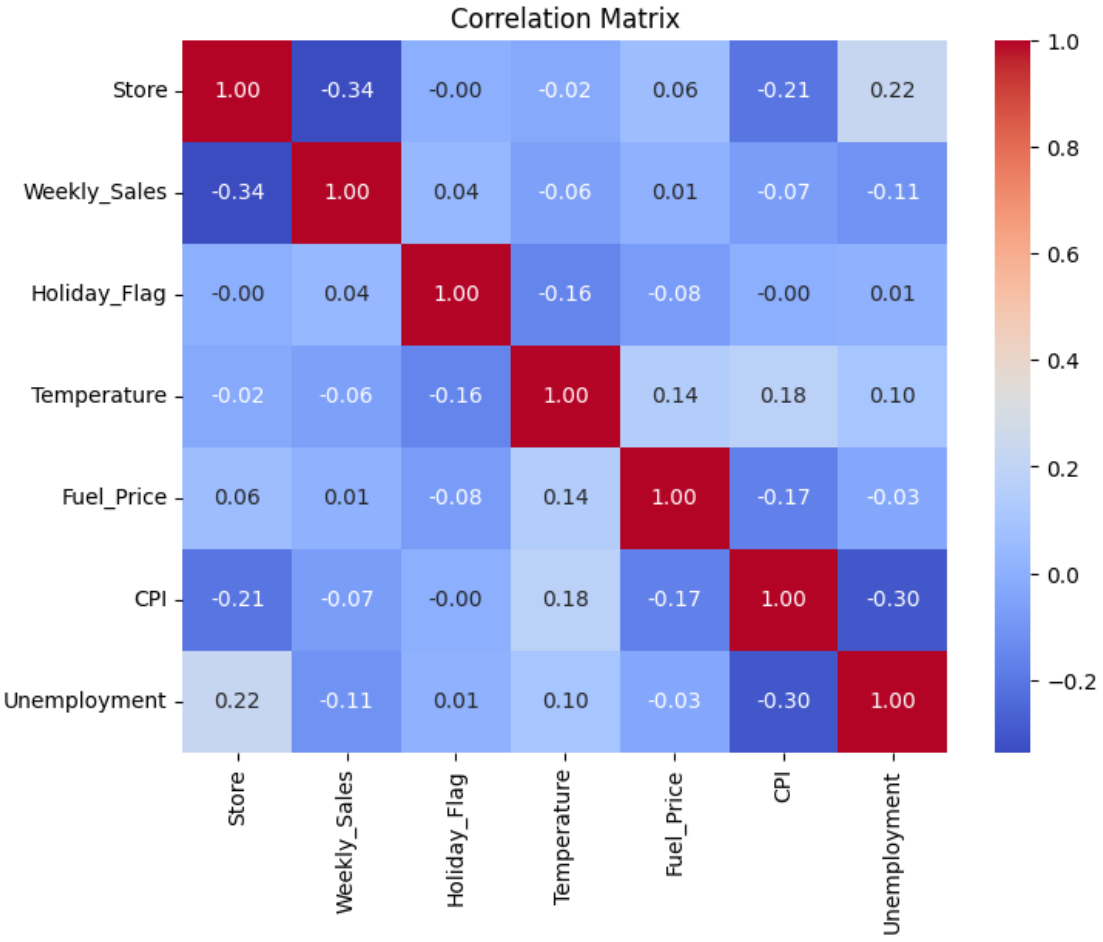
If there is a steady increase or decrease in sales over time, it might suggest changes in business strategy, market demand, or external factors like economic conditions (e.g., changing unemployment rates, fuel prices).

Outliers:

Sharp, isolated peaks could be outliers, which we should investigate. These might represent one-off events like major sales, special promotions, or extraordinary circumstances affecting consumer behavior during that week.

-Correlation Analysis

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

numeric_data = df.select_dtypes(include=['float64', 'int64'])

correlation_matrix = numeric_data.corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



--Correlation of Weekly_Sales with Other Variables:

Store: Shows a moderate negative correlation (-0.34), suggesting that higher store numbers tend to have lower average weekly sales. This could indicate variations in store size, location, or customer base.

Holiday_Flag: A very weak positive correlation (0.04). This implies that holidays have a minimal effect on sales overall, though some specific holidays might still influence sales spikes.

Temperature: Weak negative correlation (-0.06), indicating that temperature changes have little impact on sales trends.

Fuel_Price: Very weak positive correlation (0.01). Changes in fuel prices seem to have no significant impact on sales patterns.

CPI (Consumer Price Index): Weak negative correlation (-0.07). CPI changes do not strongly affect sales, though this might vary by time period or economic conditions.

Unemployment: Weak negative correlation (-0.11), implying that higher unemployment rates slightly reduce weekly sales, which could reflect reduced consumer spending during economic downturns.

-The values inside the heatmap represent the correlation coefficients, which measure the linear relationship between variables. These range from -1 to 1:

1.0: Perfect positive correlation (e.g., a variable with itself).

0.0: No correlation.

-1.0: Perfect negative correlation.

The colors visually represent the strength of the correlation:

Red: Strong positive correlation.

Blue: Strong negative correlation.

White or Pale Colors: Weak or no correlation.

Summary: Key Relationships with Weekly_Sales (Target):

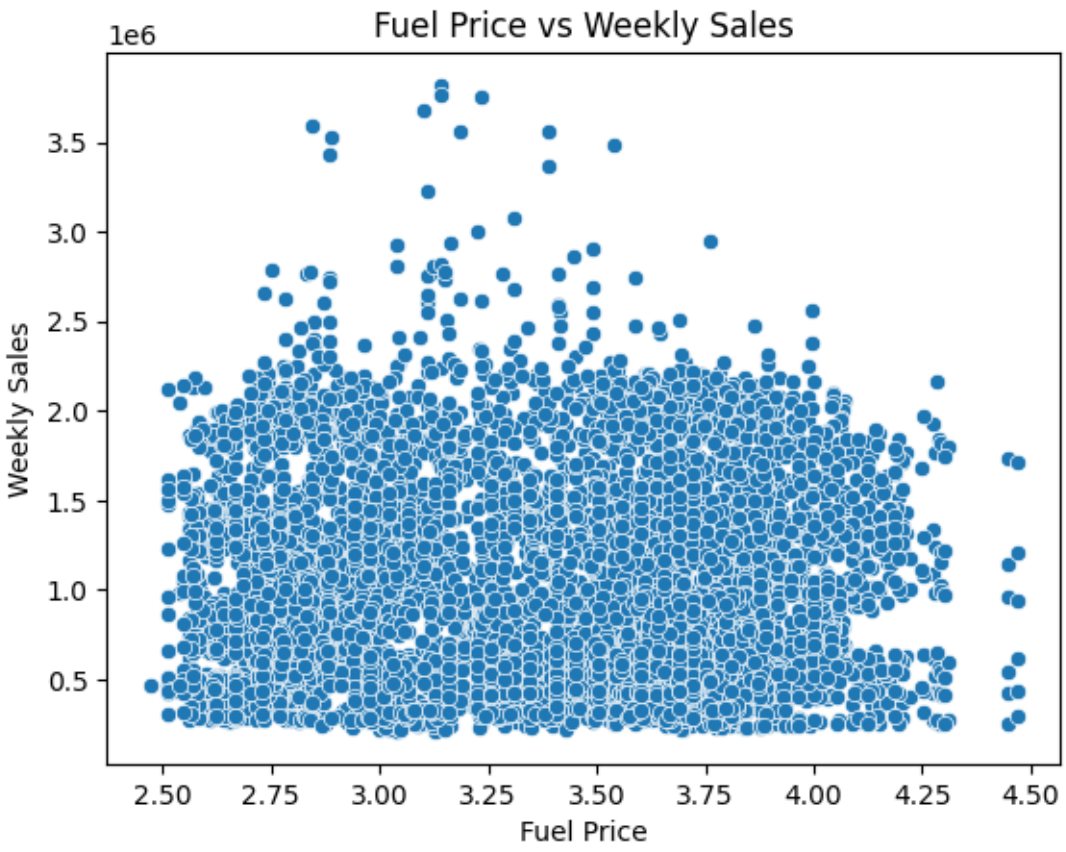
The strongest correlation is with Store (-0.34), indicating a slight negative trend.

Other variables like Holiday_Flag, Temperature, Fuel_Price, and CPI have minimal impact.

General Observations

There is no strong linear correlation between most variables in this dataset.

[ ]:
```
sns.scatterplot(x='Fuel_Price', y='Weekly_Sales', data=df)
plt.title('Fuel Price vs Weekly Sales')
plt.xlabel('Fuel Price')
plt.ylabel('Weekly Sales')
plt.show()
```

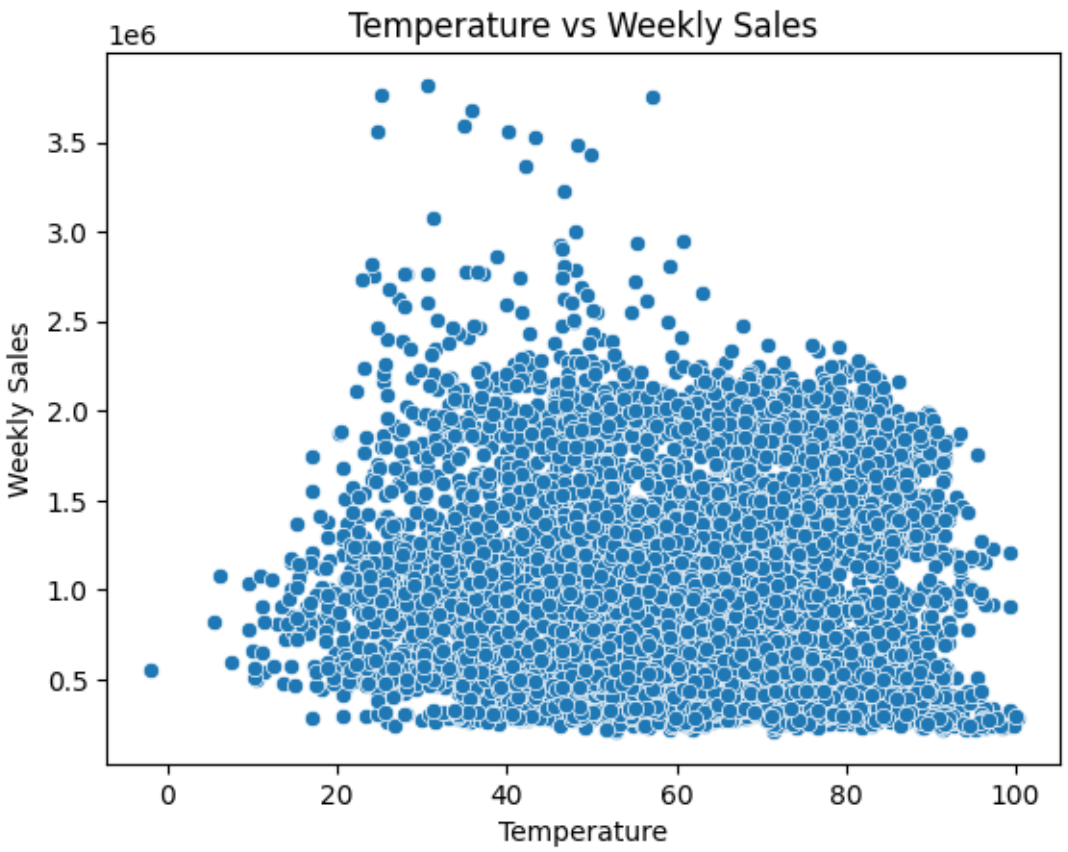


Possible Insights:

Seasonality: The chart doesn't clearly reveal any strong seasonal patterns. However, it could be interesting to analyze the data further to see if there are any subtle seasonal trends.

Year-over-Year Differences: The chart highlights the differences in sales performance between the three years. For instance, 2012 experienced a significant increase in sales in the first quarter compared to the other two years.

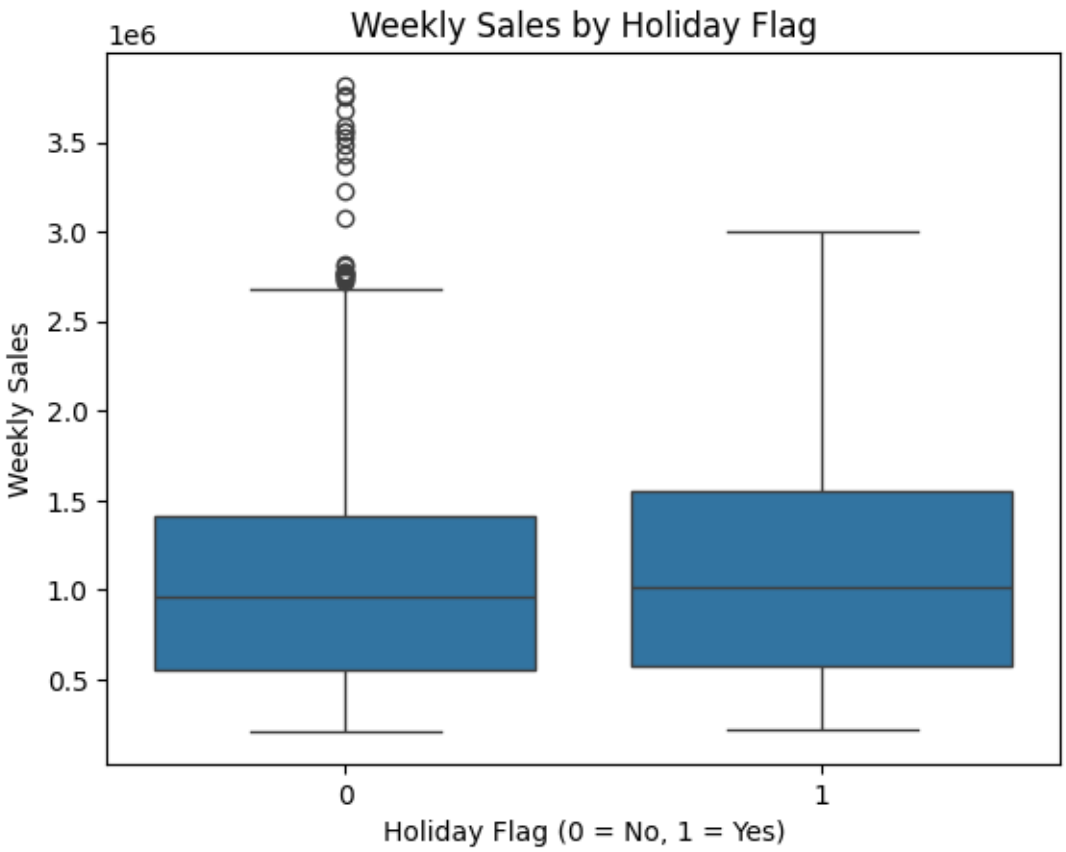
[ ]:
```
sns.scatterplot(x='Temperature', y='Weekly_Sales', data=df)
plt.title('Temperature vs Weekly Sales')
plt.xlabel('Temperature')
plt.ylabel('Weekly Sales')
plt.show()
```



-No Insights

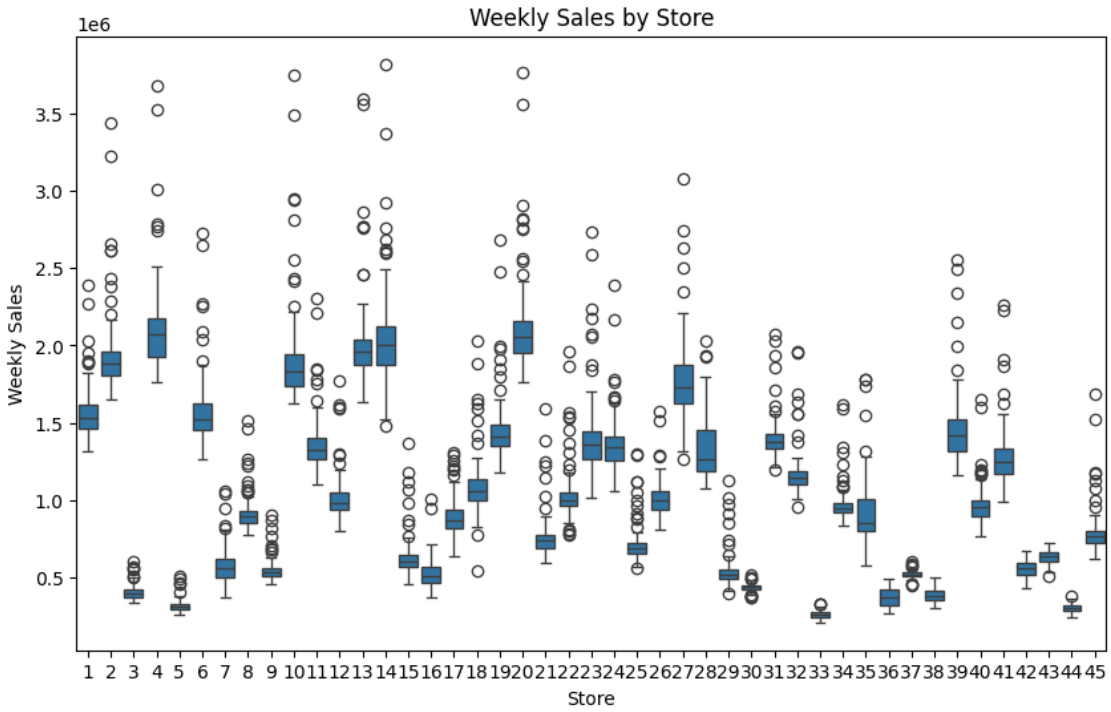
Weekly Sales over Holiday

[ ]:
```
sns.boxplot(x='Holiday_Flag', y='Weekly_Sales', data=df)
plt.title('Weekly Sales by Holiday Flag')
plt.xlabel('Holiday Flag (0 = No, 1 = Yes)')
plt.ylabel('Weekly Sales')
plt.show()
```

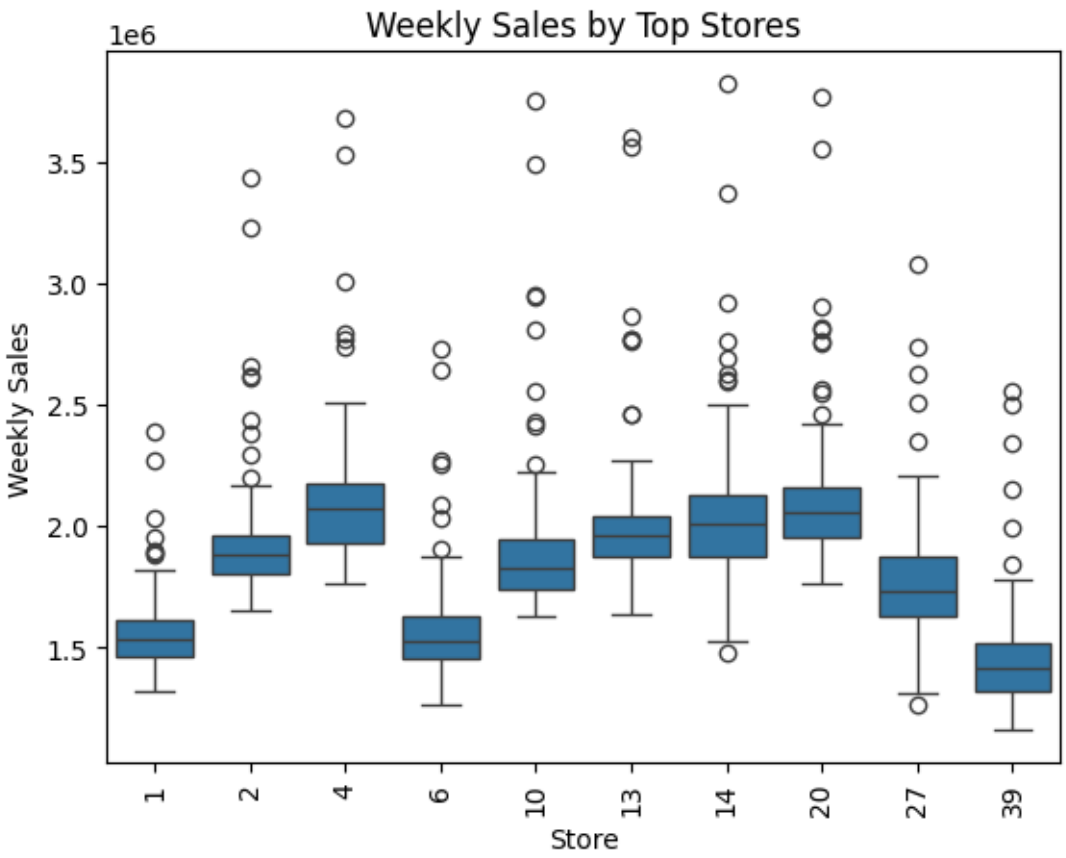


#-Box Plot For Store and Weekly Sales

[ ]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Store', y='Weekly_Sales', data=df)
plt.title('Weekly Sales by Store')
plt.xlabel('Store')
plt.ylabel('Weekly Sales')
plt.show()
```



[ ]:
```
top_stores = df.groupby('Store')['Weekly_Sales'].median().
  ↪sort_values(ascending=False).head(10).index
filtered_df = df[df['Store'].isin(top_stores)]

sns.boxplot(x='Store', y='Weekly_Sales', data=filtered_df)
plt.title('Weekly Sales by Top Stores')
plt.xlabel('Store')
plt.ylabel('Weekly Sales')
plt.xticks(rotation=90)
plt.show()
```



-This box plot analysis is a great way to understand how weekly sales vary across different stores.

Store 10 stands out with higher median weekly sales and is likely a high-performing store.

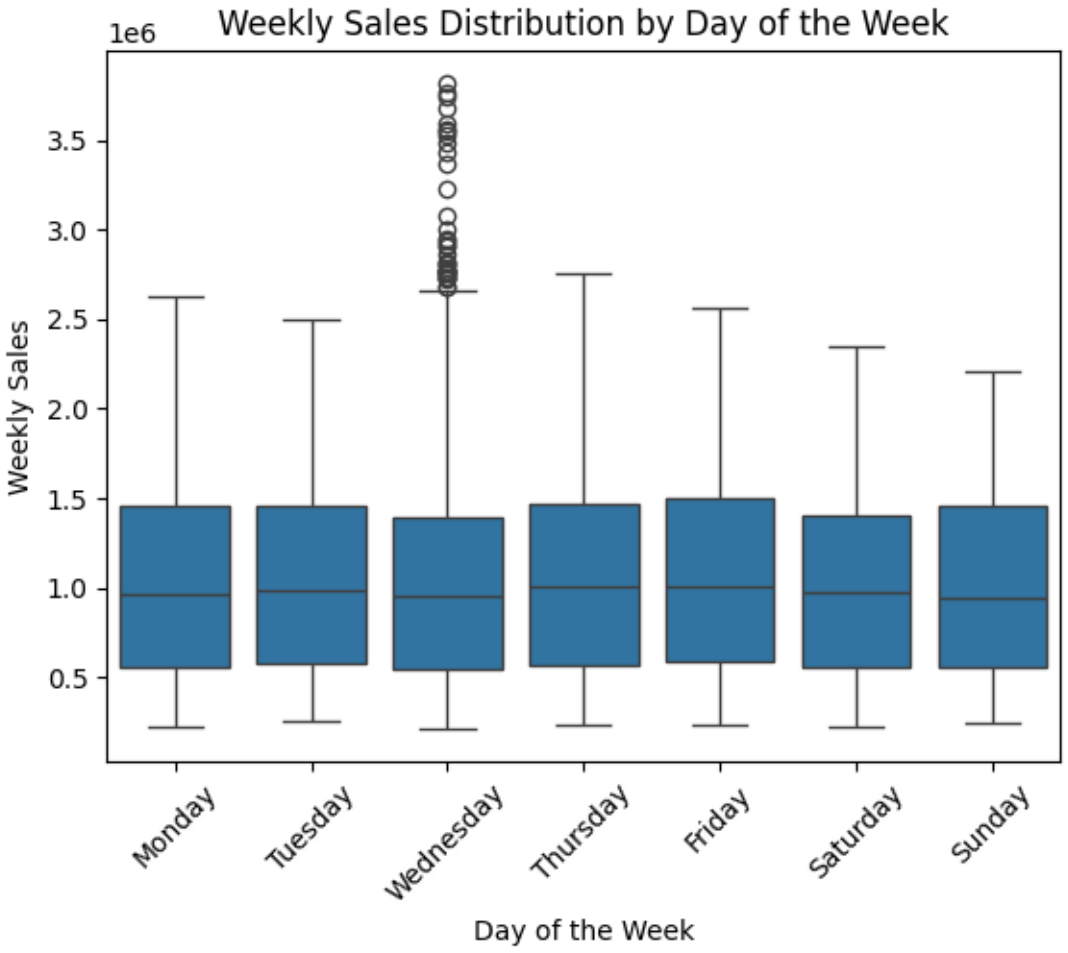
Store 2 and **Store 3** show consistent sales patterns, making them reliable for forecasting.

Store 1, Store 6, and Store 27 have outliers in their sales, which might require a deeper dive to understand what causes the spikes in sales.

-Sales by Weekday (Day of the Week vs Weekly Sales):

[ ]:
```
df['Day_of_Week'] = df['Date'].dt.strftime('%A')

df = df.sort_values(by='Date')
sns.boxplot(x='Day_of_Week', y='Weekly_Sales', data=df)

plt.xticks([0, 1, 2, 3, 4, 5, 6], ['Monday', 'Tuesday', 'Wednesday',␣
 ↪'Thursday', 'Friday', 'Saturday', 'Sunday'])

plt.title('Weekly Sales Distribution by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Weekly Sales')
```

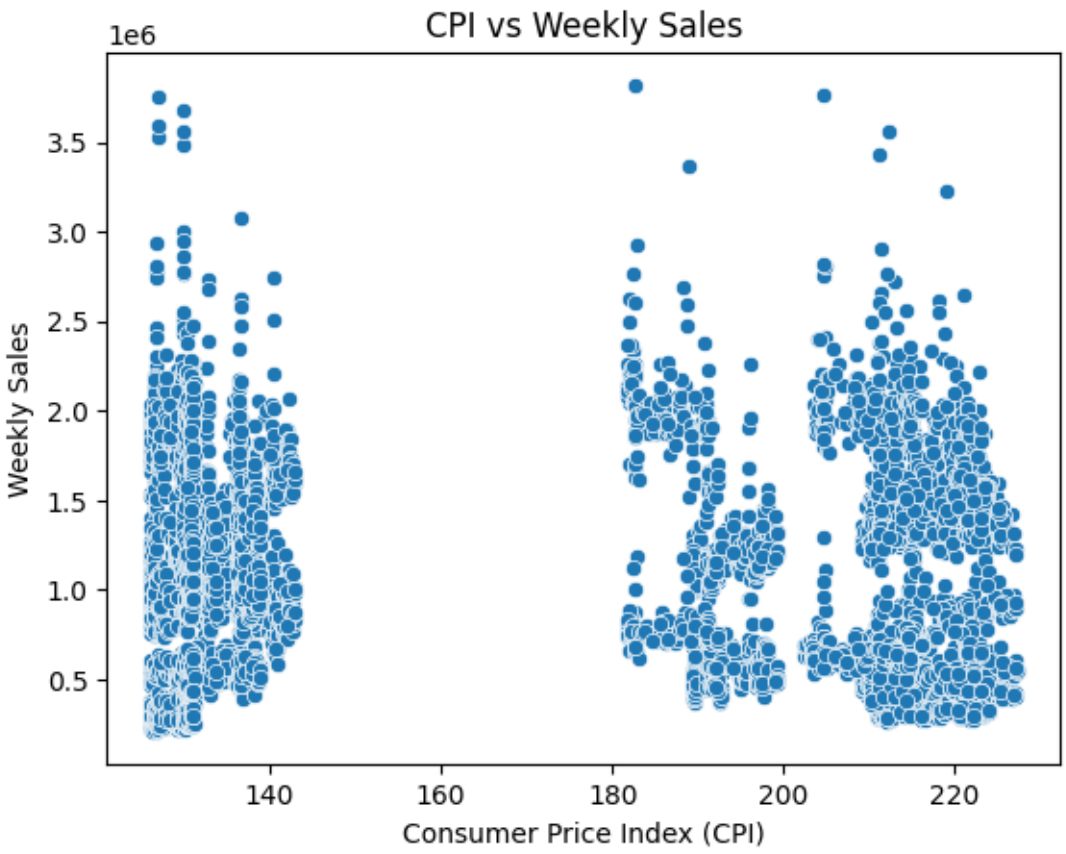
```
plt.xticks(rotation=45)

plt.show()
```



Wednesday, it could indicate that there are specific events or circumstances on Wednesdays that lead to unusually change of Sales

1 -Scatter plot: CPI vs Weekly Sales

[ ]:
```
sns.scatterplot(x=df['CPI'], y=df['Weekly_Sales'])
plt.title('CPI vs Weekly Sales')
plt.xlabel('Consumer Price Index (CPI)')
plt.ylabel('Weekly Sales')
plt.show()
```



No Strong Linear Correlation:

There doesn't appear to be a clear linear relationship between CPI and Weekly Sales.

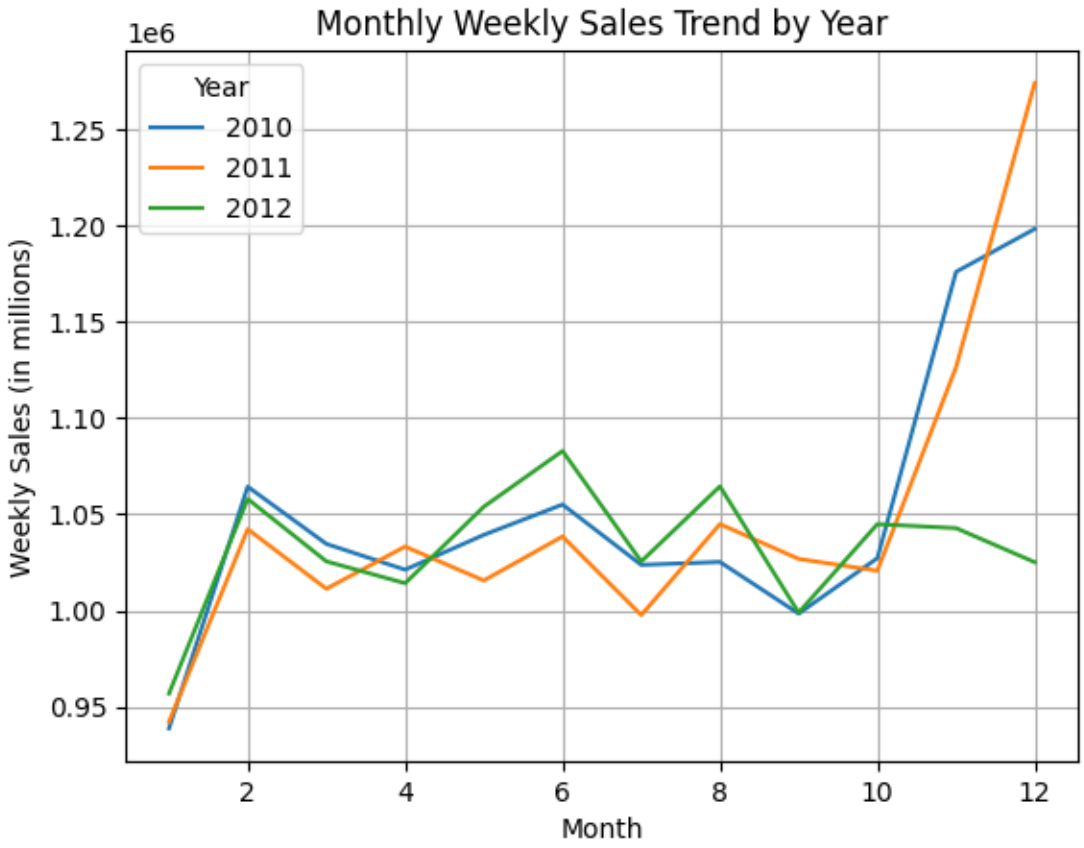
The points are scattered across the plot without a distinct upward or downward trend

[ ]:
```
df["Year"] = df["Date"].dt.year
df["Month"] = df["Date"].dt.month
```

--Monthly Weekly Sales Trend by Year

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

weeklysales = pd.pivot_table(data=df, values="Weekly_Sales", columns="Year",␣
 ↪index="Month")
weeklysales.plot()

plt.title('Monthly Weekly Sales Trend by Year')
plt.xlabel('Month')
plt.ylabel('Weekly Sales (in millions)')
```

```
plt.grid(True)
plt.show()
```



Observations:

Year-to-Year Fluctuations: The sales figures exhibit considerable variability from year to year.

2010: The sales for 2010 begin at a relatively high point in January and then show a gradual decline throughout the year.

2011: In contrast, 2011 displays a more consistent pattern. Sales show a slight upward trend from January to July, followed by a subsequent decline.

2012: Sales in 2012 commence at a lower level compared to the previous two years. However, there is a substantial increase from February to April, after which the sales remain relatively stable with some fluctuations.

#Possible Insights:

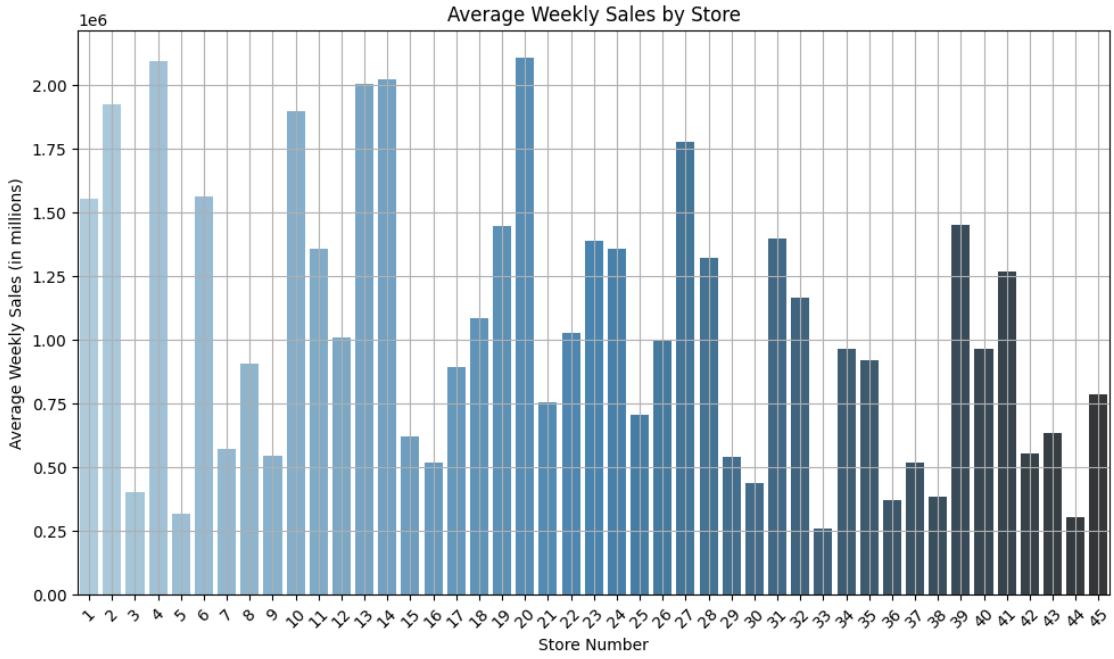
Seasonality: While not immediately apparent, there might be subtle seasonal patterns within the data that could be revealed through further analysis.

Year-over-Year Comparisons: The chart highlights the differences in sales performance across the three years. For instance, 2012 experienced a significant surge in sales during the first quarter

compared to the other two years

[ ]:
```
store_sales = df.groupby('Store')['Weekly_Sales'].mean().
 ↪sort_values(ascending=False)

plt.figure(figsize=(10,6))
sns.barplot(x=store_sales.index, y=store_sales.values, palette='Blues_d')

plt.title('Average Weekly Sales by Store')
plt.xlabel('Store Number')
plt.ylabel('Average Weekly Sales (in millions)')
plt.xticks(rotation=45)
plt.grid(True)

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

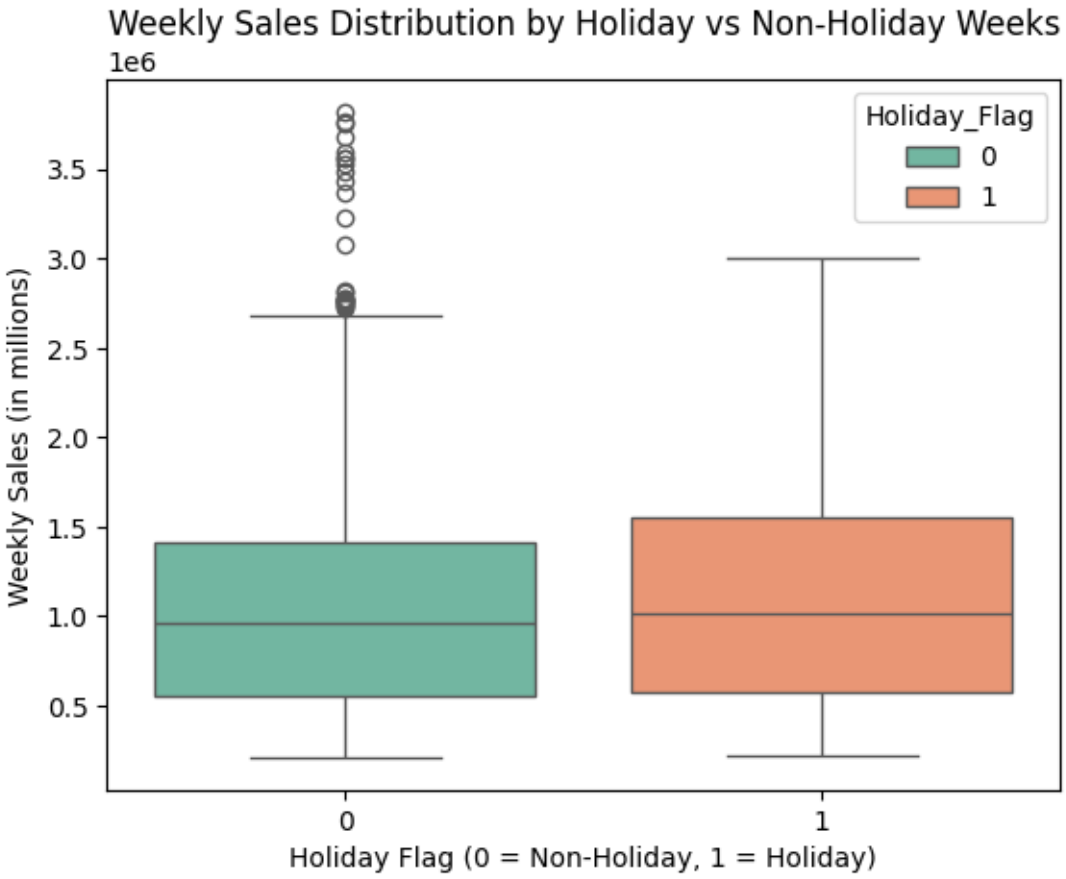
```
<ipython-input-172-408991525211>:9: FutureWarning:

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_sales.index, y=store_sales.values, palette='Blues_d')
```



The chart provides a clear visual comparison of the average sales performance across different stores.

This can help identify high-performing stores and low-performing stores.

```
[ ]: sns.boxplot(x='Holiday_Flag', y='Weekly_Sales', data=df, hue='Holiday_Flag',␣
      ↪palette='Set2')

     plt.title('Weekly Sales Distribution by Holiday vs Non-Holiday Weeks')
     plt.xlabel('Holiday Flag (0 = Non-Holiday, 1 = Holiday)')
     plt.ylabel('Weekly Sales (in millions)')

     plt.show()
```



INSIGHTS :

Impact of Holidays: The box plot suggests that, on average, Weekly Sales tend to be higher during holiday weeks compared to non-holiday weeks.

Variability: The similar box widths indicate that the variability in Weekly Sales is comparable between holiday and non-holiday weeks.

Outliers: The presence of outliers suggests that there might be some weeks with unusually high or low sales, even during non-holiday periods.

1.1 Exploratory Data Analysis (EDA) Overview

In this project, I conducted a thorough **Exploratory Data Analysis (EDA)** on a dataset containing sales data from multiple stores. The goal was to uncover valuable insights about the relationship between weekly sales and various factors, including store-specific attributes, economic indicators, and external influences such as holidays.

Key EDA Steps:

1. **Data Cleaning and Preprocessing:**
   - Checked for and handled missing data.
   - Converted date-related columns into separate year and month columns for better trend analysis.
   - Ensured all data types were appropriately formatted for analysis.
2. **Univariate Analysis:**
   - **Box Plots** were used to visualize the distribution and identify outliers for `Weekly_Sales`, and to compare sales between holiday and non-holiday weeks.
   - **Histograms** provided a deeper understanding of the distribution of `Weekly_Sales`, `Fuel_Price`, `CPI`, and `Unemployment` variables.
   - Summary statistics were computed for each variable to understand their central tendency and spread.
3. **Bivariate Analysis:**
   - **Scatter Plots** were used to examine the relationships between `Weekly_Sales` and other key variables such as `Fuel_Price`, `CPI`, and `Unemployment`. This helped identify any linear or non-linear relationships.
   - **Correlation Heatmap** showed the strength of relationships between numerical variables, revealing moderate correlations between `Weekly_Sales` and variables like `Fuel_Price` and `CPI`.
4. **Trend Analysis:**
   - **Line Plots** demonstrated the trend of weekly sales over time and highlighted how sales fluctuated across different months and years.
   - A **pivot table** was created to show the trend of monthly sales across multiple years, providing insights into how different years performed.
5. **Comparative Analysis:**
   - **Bar Plots** compared the average weekly sales across different stores, revealing the performance variance between stores.
   - **Box Plots** comparing holiday and non-holiday weeks provided insights into the impact of holidays on sales.
6. **Seasonality and Insights:**
   - Analyzed weekly sales patterns over time, identified potential seasonal trends, and assessed outliers that may warrant further investigation.
   - Gleaned insights about the performance of stores based on average sales and outlier detection.

Key Findings:

- **Holiday Impact:** Holiday weeks tended to have similar median sales to non-holiday weeks, but with more consistent values.

- **Store Performance:** There was a noticeable variation in sales across stores, with some outperforming others significantly.
- **Economic Indicators:** Although variables like `Fuel_Price` and `CPI` were analyzed, they didn't show a strong linear relationship with weekly sales, suggesting that other external factors might be influencing sales trends.
- **Trends Over Time:** Sales showed significant fluctuations across the years, indicating potential seasonality, promotions, or other business-related factors.

Limitations:

- **Limited Data:** The analysis was based on data spanning a limited number of years, which may not provide a full picture of long-term trends.
- **External Factors:** Variables such as marketing efforts, store location, and competition were not included in the dataset, limiting the scope of the analysis.

Conclusion: This EDA provided valuable insights into the sales data, highlighting key patterns, trends, and areas for further investigation. The next steps could involve feature engineering, building predictive models, or optimizing business strategies based on the identified insights.
