

In [7]:

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

<Frozen Importlib\_Bootstrap>:241: RuntimeWarning: scipy.\_lib.messagestream.MessageStream size changed, may indicate binary incompatibility. Expected 56 from C header, got 64 from PyObject

In [2]:

```
data = pd.read_csv("Walmart Data Analysis and Forecasting.csv")
```

In [3]:

```
data.head(6)
```

Out[3]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106

In [4]:

```
data.tail(6)
```

Out[4]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
6429	45	21-09-2012	723086.20	0	65.32	4.038	191.856704	8.684
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	8.684
6431	45	05-10-2012	733455.07	0	64.89	3.985	192.170412	8.667
6432	45	12-10-2012	734464.36	0	54.47	4.000	192.327265	8.667
6433	45	19-10-2012	718125.53	0	56.47	3.969	192.330854	8.667
6434	45	26-10-2012	760281.43	0	58.85	3.882	192.308899	8.667

In [5]:

```
data.describe()
```

Out[5]:

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
count	6435.000000	6.435000e+03	6435.000000	6435.000000	6435.000000	6435.000000	6435.000000
mean	23.000000	1.046965e+06	0.069930	60.663782	3.358607	171.578394	7.999151
std	12.988182	5.643666e+05	0.255049	18.444933	0.459020	39.356712	1.875885
min	1.000000	2.099862e+05	0.000000	-2.060000	2.472000	126.064000	3.879000
25%	12.000000	5.533501e+05	0.000000	47.460000	2.933000	131.735000	6.891000
50%	23.000000	9.607460e+05	0.000000	62.670000	3.445000	182.616521	7.874000
75%	34.000000	1.420159e+06	0.000000	74.940000	3.735000	212.743293	8.622000
max	45.000000	3.818686e+06	1.000000	100.140000	4.468000	227.232807	14.313000

In [6]:

```
data.isna().sum()
```

Out[6]:

Store	0
Date	0
Weekly_Sales	0
Holiday_Flag	0
Temperature	0
Fuel_Price	0
CPI	0
Unemployment	0

dtype: int64

In [9]:

```
data.shape
```

Out[9]:

(6435, 8)

In [13]:

```
data.info()
```

Out[13]:

```
<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
```

In [17]:

```
grouped = data.groupby("Store").agg({"Weekly_Sales":"sum"}).sort_values("Weekly_Sales",ascending=True).head(5)
```

In [18]:

```
grouped
```

Out[18]:

	Weekly_Sales
Store	
33	37160221.96
44	43293087.84
5	45475688.90
36	53412214.97
38	55159626.42

In [23]:

```
#temp and weekly sales
plt.scatter(data["Temperature"], data["Weekly_Sales"], alpha=0.5)
plt.xlabel("Temperature")
plt.ylabel("Weekly Sales")
plt.title("Scatter plot of weekly sales vs Temperature")
```

Out[23]:

Text(0.5, 1.0, 'Scatter plot of weekly sales vs Temperature')

Figure 1: Scatter plot of weekly sales vs Temperature

In [24]:

```
#trend line weekly sales
data["Date"] = pd.to_datetime(data["Date"], format = '%d-%m-%Y')
```

In [26]:

```
WeeklySales = data.groupby(["Date"])["Weekly_Sales"].sum().reset_index()
```

In [29]:

```
plt.figure(figsize=(12, 8))
plt.plot(WeeklySales["Date"], WeeklySales["Weekly_Sales"],color="red")
plt.xlabel("Date")
plt.ylabel("Weekly Sales")
plt.title("Weekly Sales trend line")
plt.show()
```

Figure 2: Weekly Sales trend line

In [33]:

```
# distribution of weekly sales
sns.histplot(data['Weekly_Sales'], bins=50, color="blue", edgecolor="black", kde=True)
plt.title("Distribution of Weekly Sales")
plt.show()
```

Figure 3: Distribution of Weekly Sales

In [35]:

```
#cor plot
plt.figure(figsize=(12, 7))
sns.heatmap(data.corr(), annot = True, vmin = -1, vmax = 1)
plt.show()
```

Figure 4: Correlation matrix heatmap