

```
In [21]: # importing the libraries pandas,matplotlib and seaborn
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [23]: # import Dataset of Walmart Sales
df = pd.read_csv(r"C:\Users\user\Downloads\Walmart_Sales.csv")
```

```
In [25]: # To check data type of dataframe
type(df)
```

```
Out[25]: pandas.core.frame.DataFrame
```

```
In [27]: # to check the top 5 rows
df.head()
```

```
Out[27]:
```

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



```
In [29]: # to check the bottom 10 rows
df.tail(10)
```

Out[29]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unerr
6425	45	24-08-2012	718232.26	0	72.62	3.834	191.344887	
6426	45	31-08-2012	734297.87	0	75.09	3.867	191.461281	
6427	45	07-09-2012	766512.66	1	75.70	3.911	191.577676	
6428	45	14-09-2012	702238.27	0	67.87	3.948	191.699850	
6429	45	21-09-2012	723086.20	0	65.32	4.038	191.856704	
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	
6431	45	05-10-2012	733455.07	0	64.89	3.985	192.170412	
6432	45	12-10-2012	734464.36	0	54.47	4.000	192.327265	
6433	45	19-10-2012	718125.53	0	56.47	3.969	192.330854	
6434	45	26-10-2012	760281.43	0	58.85	3.882	192.308899	



In [31]: # total describe the statistics of the Dataset
df.describe()

Out[31]:

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



In [57]:

```
# To check correlation of the Dataframe
df.corr()
```

Out[57]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fu
Store	1.000000e+00	1.577299e-13	-0.335332	-4.386841e-16	-0.022659	0
Date	1.577299e-13	1.000000e+00	0.006949	-1.328524e-02	0.145357	0
Weekly_Sales	-3.353320e-01	6.949360e-03	1.000000	3.689097e-02	-0.063810	0
Holiday_Flag	-4.386841e-16	-1.328524e-02	0.036891	1.000000e+00	-0.155091	-0
Temperature	-2.265908e-02	1.453566e-01	-0.063810	-1.550913e-01	1.000000	0
Fuel_Price	6.002295e-02	7.714439e-01	0.009464	-7.834652e-02	0.144982	1
CPI	-2.094919e-01	7.715746e-02	-0.072634	-2.162091e-03	0.176888	-0
Unemployment	2.235313e-01	-2.482029e-01	-0.106176	1.096028e-02	0.101158	-0



In [59]:

```
# To check the null values and Data type of the dataset
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
```

```
In [61]: # total weekly sales of Walmart
total_sales = df["Weekly_Sales"].sum()
print(total_sales)
```

```
6737218987.11
```

```
In [63]: # average weekly sales of Walmart
average_sales = df["Weekly_Sales"].mean()
print(average_sales)
```

```
1046964.8775617715
```

```
In [65]: # total sales by Store number
store_sales = df.groupby("Store")["Weekly_Sales"].sum()
print(store_sales)
```

Store

```
1    2.224028e+08
2    2.753824e+08
3    5.758674e+07
4    2.995440e+08
5    4.547569e+07
6    2.237561e+08
7    8.159828e+07
8    1.299512e+08
9    7.778922e+07
10   2.716177e+08
11   1.939628e+08
12   1.442872e+08
13   2.865177e+08
14   2.889999e+08
15   8.913368e+07
16   7.425243e+07
17   1.277821e+08
18   1.551147e+08
19   2.066349e+08
20   3.013978e+08
21   1.081179e+08
22   1.470756e+08
23   1.987506e+08
24   1.940160e+08
25   1.010612e+08
26   1.434164e+08
27   2.538559e+08
28   1.892637e+08
29   7.714155e+07
30   6.271689e+07
31   1.996139e+08
32   1.668192e+08
33   3.716022e+07
34   1.382498e+08
35   1.315207e+08
36   5.341221e+07
37   7.420274e+07
38   5.515963e+07
39   2.074455e+08
40   1.378703e+08
41   1.813419e+08
42   7.956575e+07
43   9.056544e+07
44   4.329309e+07
45   1.123953e+08
Name: Weekly_Sales, dtype: float64
```

```
In [67]: df1 = df.sort_values(by="Date")
df1
```

Out[67]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Uner
0	1	2010-02-05	1643690.90	0	42.31	2.572	211.096358	
1287	10	2010-02-05	2193048.75	0	54.34	2.962	126.442065	
5148	37	2010-02-05	536006.73	0	45.97	2.572	209.852966	
2288	17	2010-02-05	789036.02	0	23.11	2.666	126.442065	
4147	30	2010-02-05	465108.52	0	39.05	2.572	210.752605	
...
3574	25	2012-10-26	688940.94	0	56.69	3.882	216.151590	
714	5	2012-10-26	319550.77	0	71.70	3.506	224.037814	
5719	40	2012-10-26	921264.52	0	49.65	3.917	138.728161	
2573	18	2012-10-26	1127516.25	0	56.09	3.917	138.728161	
6434	45	2012-10-26	760281.43	0	58.85	3.882	192.308899	

6435 rows × 8 columns



In [69]: # to check the total sales according to holiday and non holiday weeks
holiday_wise_sales_total = df.groupby("Holiday_Flag")["Weekly_Sales"].sum()
print(holiday_wise_sales_total)

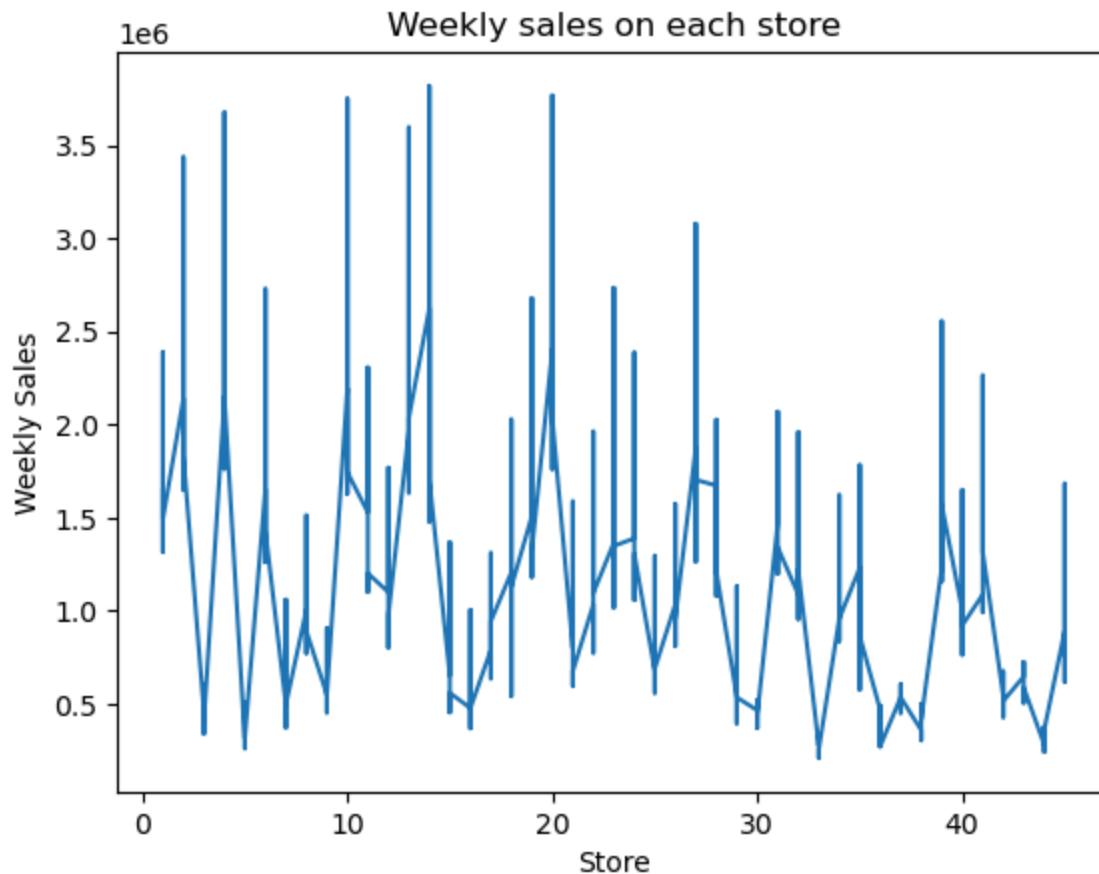
```
Holiday_Flag
0    6.231919e+09
1    5.052996e+08
Name: Weekly_Sales, dtype: float64
```

In [71]: # to check the average sales according to holiday and non holiday weeks
holiday_wise_sales_avg = df.groupby("Holiday_Flag")["Weekly_Sales"].mean()
print(holiday_wise_sales_avg)

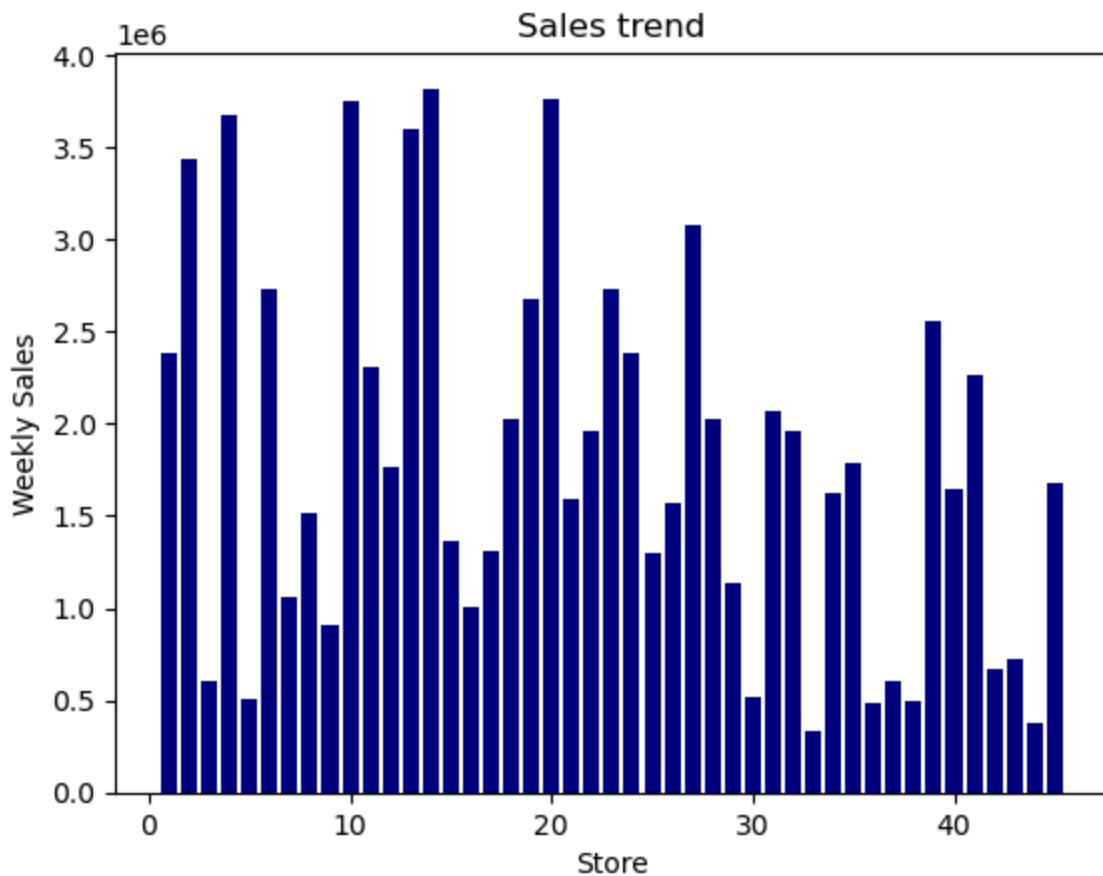
```
Holiday_Flag
0    1.041256e+06
1    1.122888e+06
Name: Weekly_Sales, dtype: float64
```

In [73]: # to convert the date format in actual time & date format that python understands
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')

```
In [77]: # create Line chart to check weekly sales on the basis of number of stores  
plt.figure()  
plt.plot(df["Store"],df["Weekly_Sales"])  
plt.xlabel("Store")  
plt.ylabel("Weekly Sales")  
plt.title("Weekly sales on each store")  
plt.show()
```

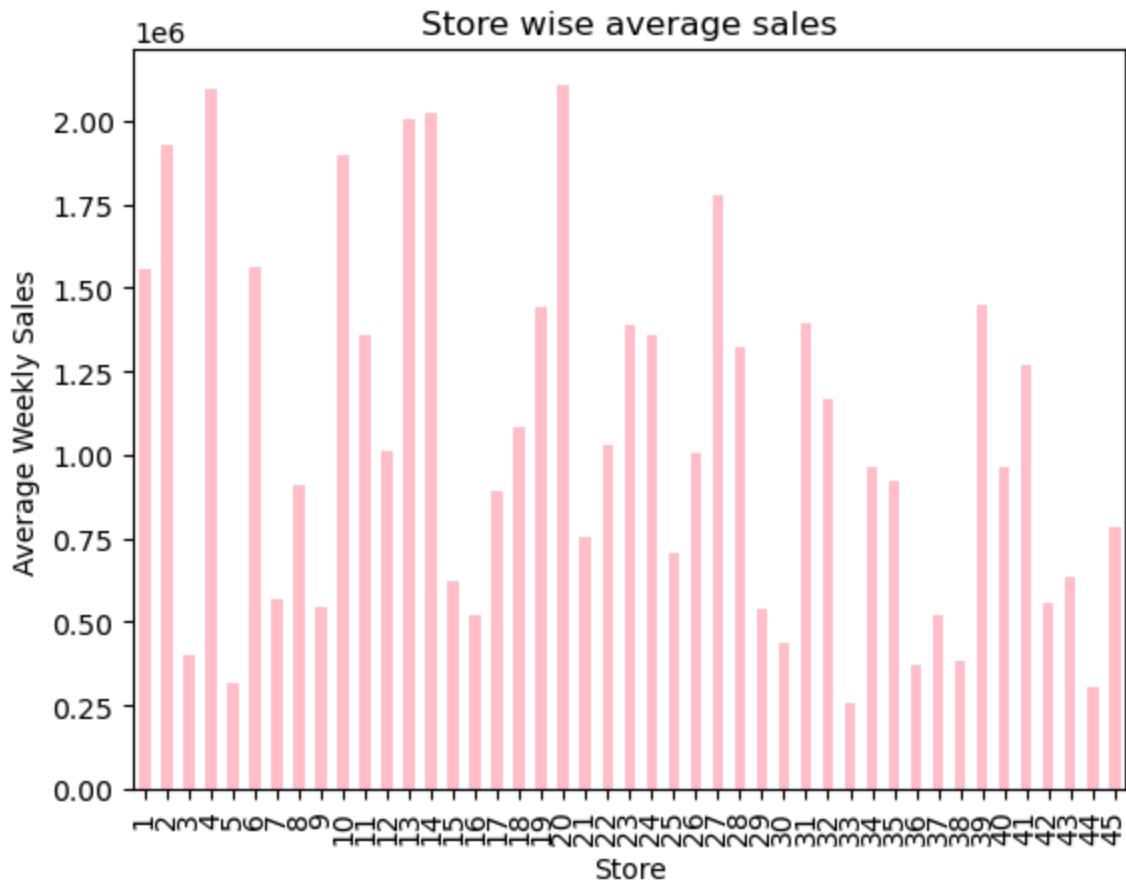


```
In [81]: plt.figure()  
plt.bar(df["Store"],df["Weekly_Sales"],color="navy")  
plt.xlabel("Store")  
plt.ylabel("Weekly Sales")  
plt.title("Sales trend")  
plt.show()
```

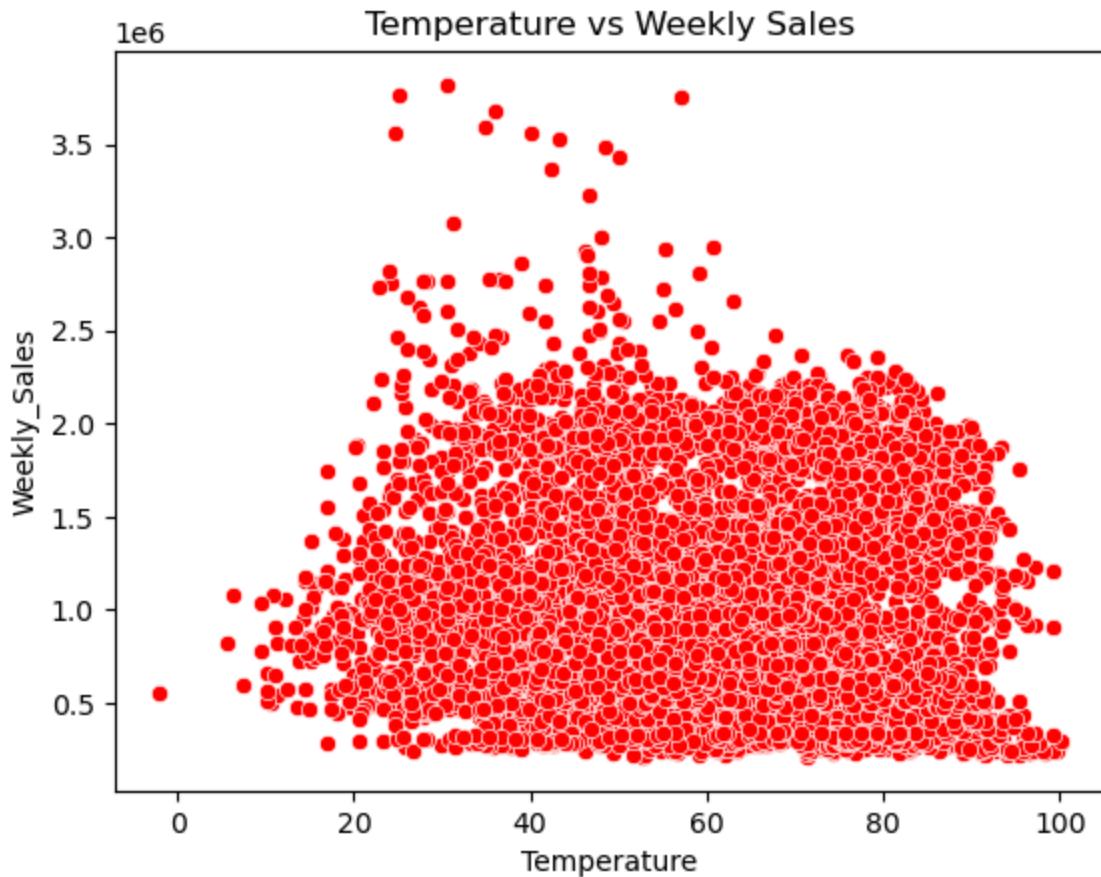


In [83]:

```
#  
store_wise_salesavg = df.groupby('Store')['Weekly_Sales'].mean()  
store_wise_salesavg.plot(kind='bar', color = 'pink')  
plt.xlabel("Store")  
plt.ylabel("Average Weekly Sales")  
plt.title("Store wise average sales")  
plt.show()
```

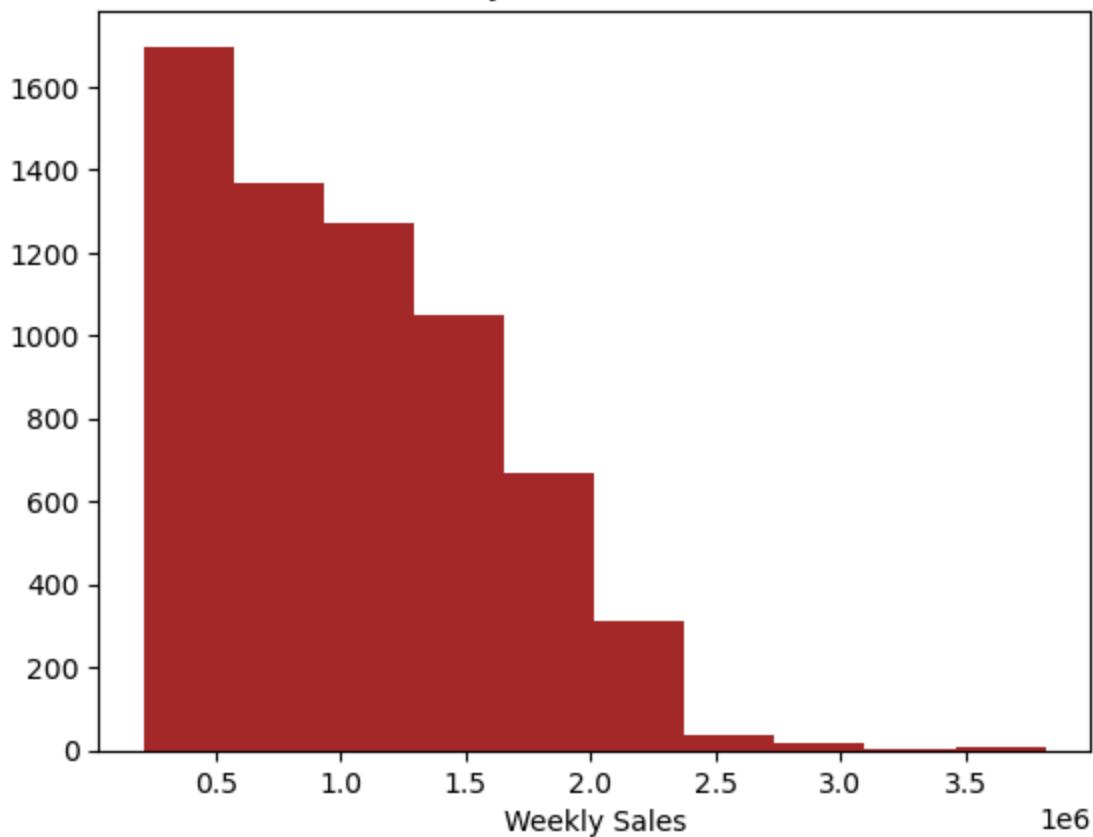


```
In [85]: sns.scatterplot(x='Temperature', y='Weekly_Sales', data=df, color = 'red')
plt.title("Temperature vs Weekly Sales")
plt.show()
```



```
In [87]: #  
plt.figure()  
plt.hist(df["Weekly_Sales"], color = 'brown')  
plt.xlabel("Weekly Sales")  
plt.title("Weekly sales Sales trend")  
plt.show()
```

Weekly sales Sales trend

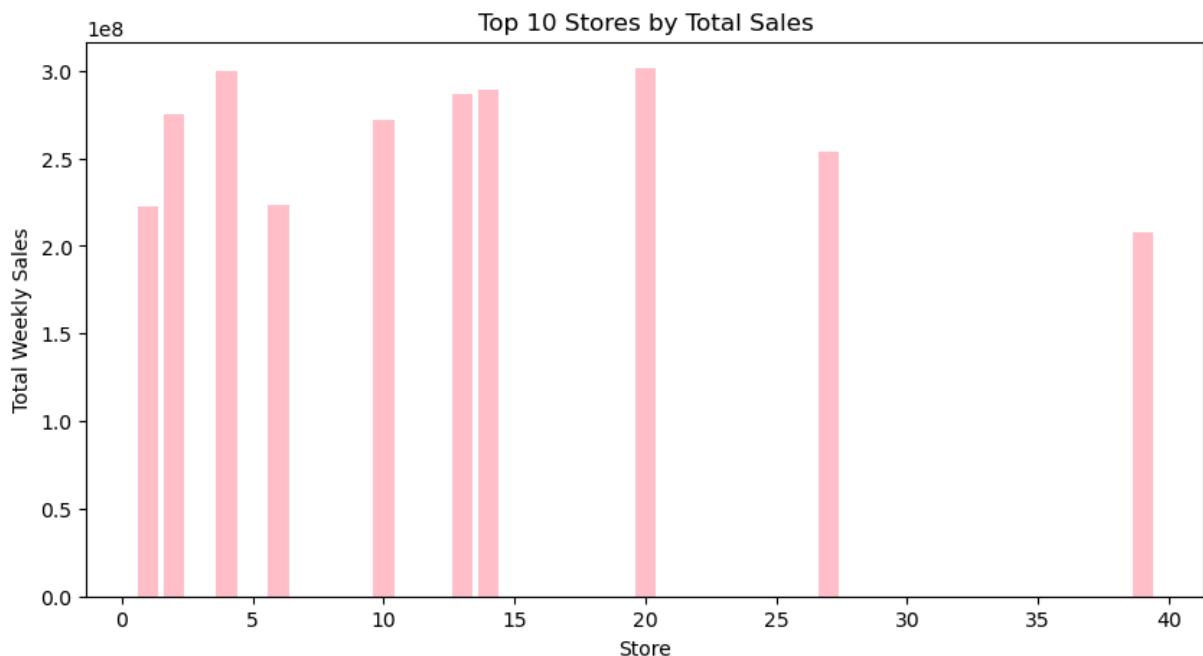


```
In [89]: store_sales = df.groupby("Store")["Weekly_Sales"].sum()

top10_stores = store_sales.sort_values(ascending=False).head(10)

plt.figure(figsize=(10,5))
plt.bar(top10_stores.index, top10_stores.values, color="pink")

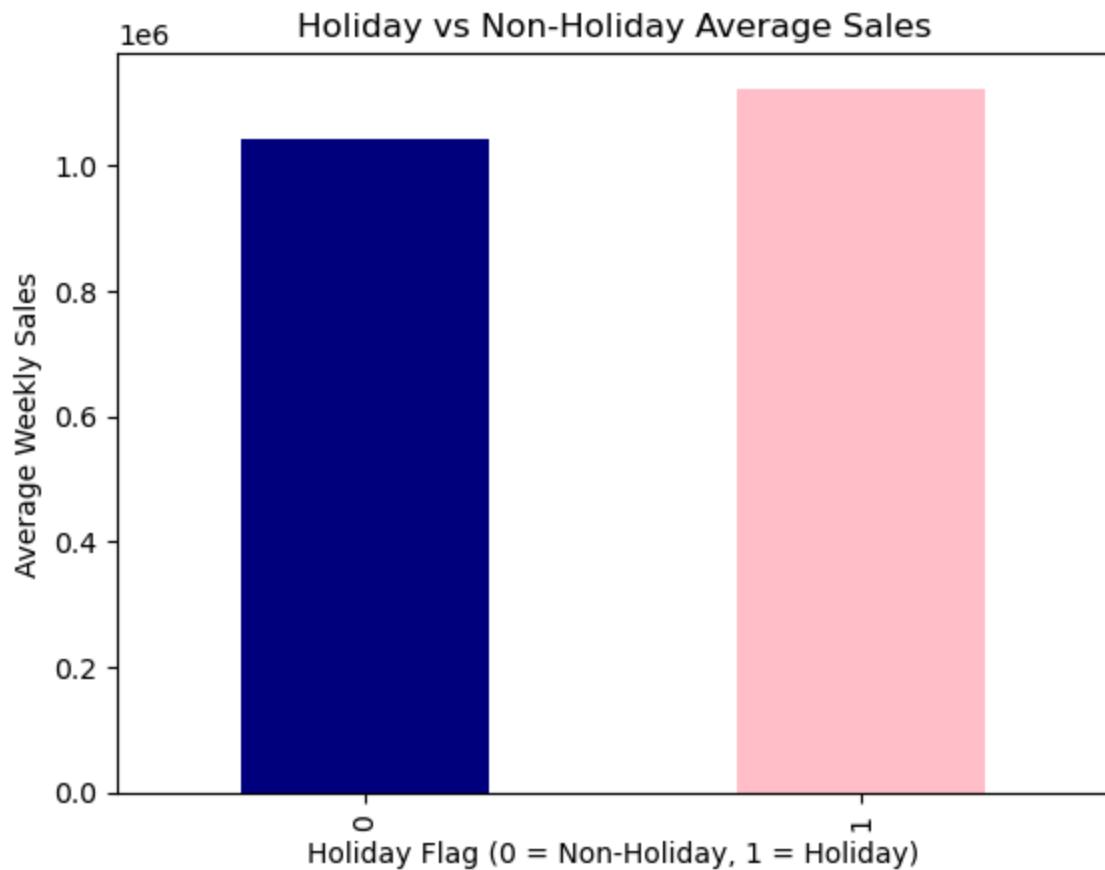
plt.xlabel("Store")
plt.ylabel("Total Weekly Sales")
plt.title("Top 10 Stores by Total Sales")
plt.show()
```



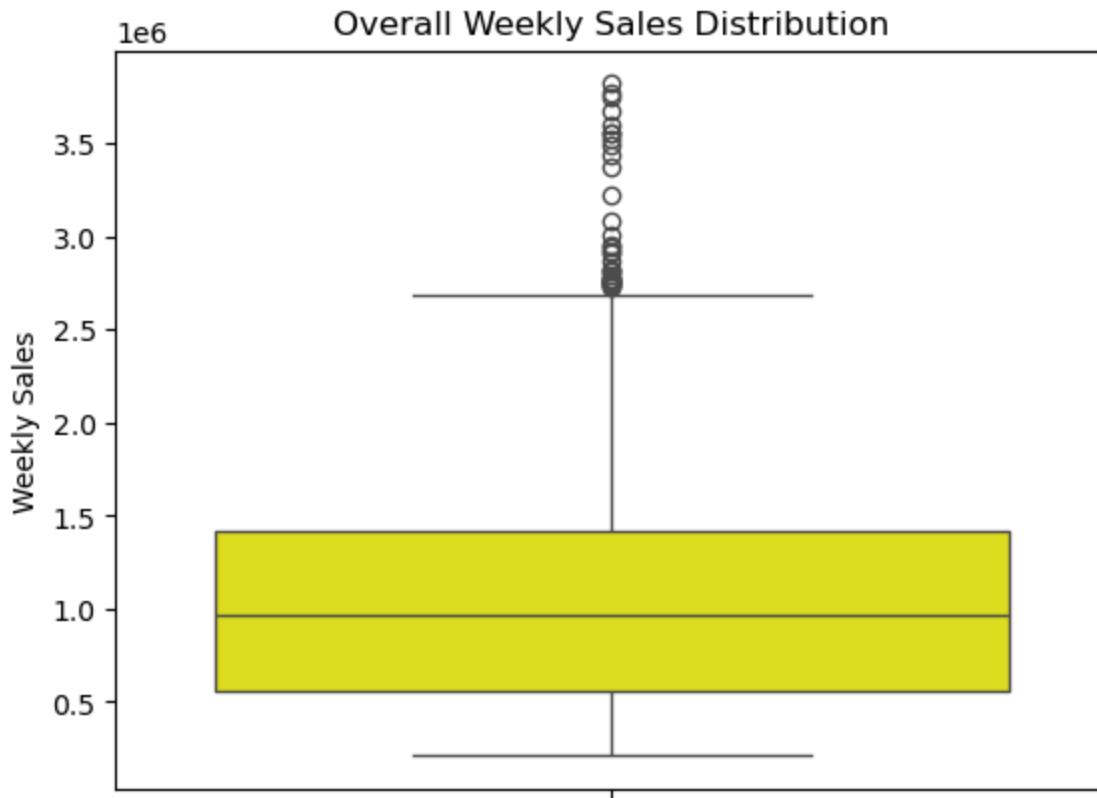
```
In [91]: holiday_wise_salesavg = df.groupby('Holiday_Flag')['Weekly_Sales'].mean()

holiday_wise_salesavg.plot(kind='bar' , color =[ 'navy' , 'pink' ] )

plt.xlabel("Holiday Flag (0 = Non-Holiday, 1 = Holiday)")
plt.ylabel("Average Weekly Sales")
plt.title("Holiday vs Non-Holiday Average Sales")
plt.show()
```



```
In [93]: plt.figure()
sns.boxplot(y=df['Weekly_Sales'], color = 'yellow')
plt.title("Overall Weekly Sales Distribution")
plt.ylabel("Weekly Sales")
plt.show()
```

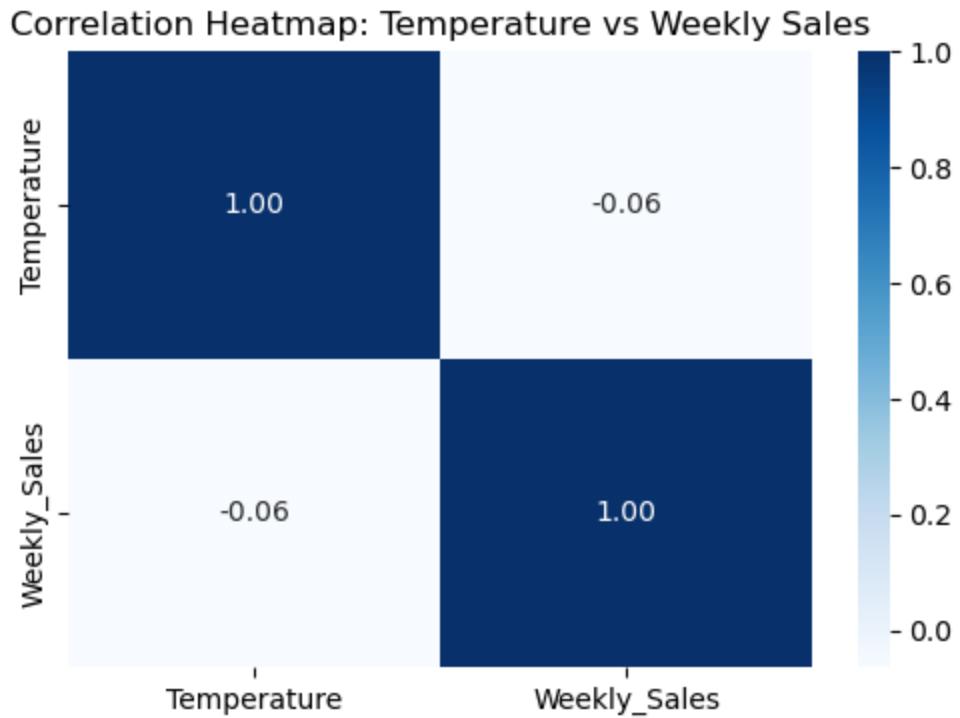


```
In [95]: import seaborn as sns
import matplotlib.pyplot as plt

# Select only numeric columns
corr = df[['Temperature', 'Weekly_Sales']].corr()

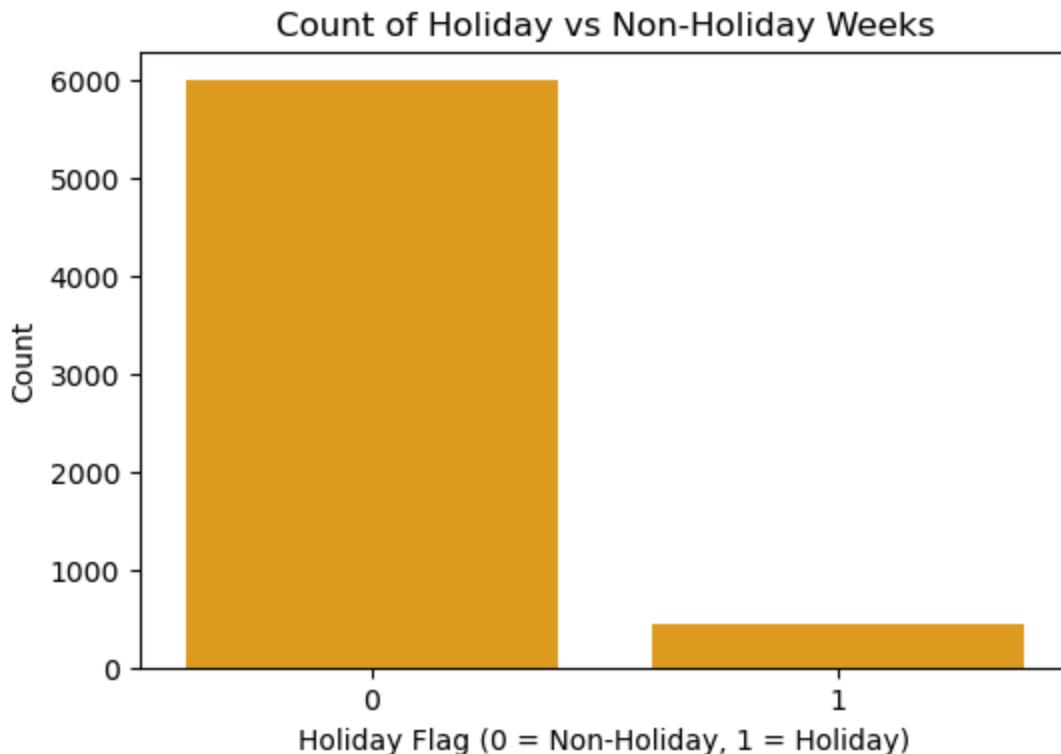
# Create heatmap
plt.figure(figsize=(6,4))
sns.heatmap(corr, annot=True, cmap='Blues', fmt='.2f')

plt.title("Correlation Heatmap: Temperature vs Weekly Sales")
plt.show()
```



```
In [97]: plt.figure(figsize=(6,4))
sns.countplot(x='Holiday_Flag', data=df, color = 'orange')

plt.xlabel("Holiday Flag (0 = Non-Holiday, 1 = Holiday)")
plt.ylabel("Count")
plt.title("Count of Holiday vs Non-Holiday Weeks")
plt.show()
```



```
In [99]: plt.hexbin(df['Temperature'], df['Weekly_Sales'], gridsize=30,color = 'pink')
plt.colorbar(label='frequency')
plt.xlabel("Temperature")
plt.ylabel("Weekly Sales")
plt.title("Density of Sales Across Temperature")
plt.show()
```

