

# Report Analysis with Walmart Data

```
In [3]: import pandas as pd
import numpy as np
from math import sqrt
from datetime import date
import matplotlib.pyplot as plt
from matplotlib import style
from sklearn.linear_model import LinearRegression
from sklearn import model_selection
from sklearn.metrics import mean_squared_error
```

```
In [111]: datasets = pd.read_csv("D:/Riya/SIMPLILEARN/Data Science with Python/My Project/Python 3-Retail Analysis with Wal
```



In [112]: datasets

Out[112]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	5/2/2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12/2/2010	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	5/3/2010	1554806.68	0	46.50	2.625	211.350143	8.106
...	...	...	...	...	...	...	...	...
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	8.684
6431	45	5/10/2012	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

6435 rows × 8 columns

In [52]: datasets.describe()

Out[52]:

	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

## Analysing the Datasets

```
In [6]: datasets.shape
```

```
Out[6]: (6435, 8)
```

```
In [7]: datasets.isnull().sum()
```

```
Out[7]: Store          0  
Date          0  
Weekly_Sales  0  
Holiday_Flag  0  
Temperature   0  
Fuel_Price    0  
CPI           0  
Unemployment  0  
dtype: int64
```

```
In [8]: datasets.columns
```

```
Out[8]: Index(['Store', 'Date', 'Weekly_Sales', 'Holiday_Flag', 'Temperature',  
              'Fuel_Price', 'CPI', 'Unemployment'],  
             dtype='object')
```

```
In [9]: datasets.Store.unique()
```

```
Out[9]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,  
              18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,  
              35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45], dtype=int64)
```

**Which store has maximum sales..**

```
In [10]: sales_list=[]  
sales_list=datasets.groupby(["Store"])[ "Weekly_Sales"].sum()  
max_sales=max(datasets.groupby(["Store"])[ "Weekly_Sales"].sum())  
sales_list
```

```
Out[10]: 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 [11]: for i in range(1,46):
         if max_sales==sales_list[i]:
             print("Store which has maximum sales of {} is {}".format(max_sales,i))

```

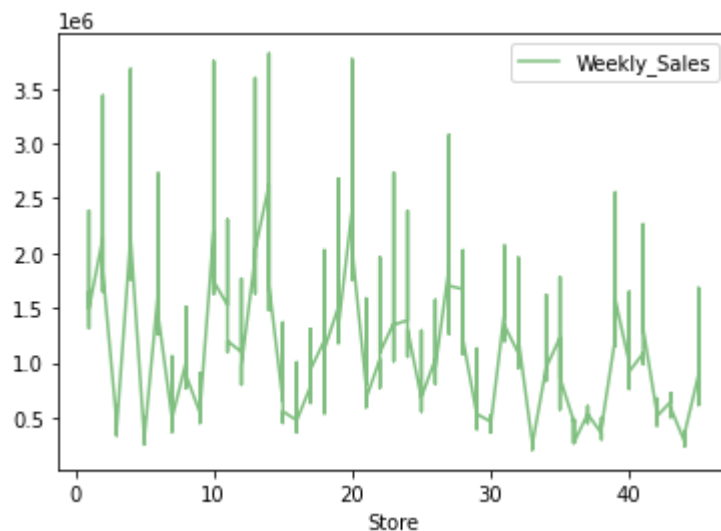
Store which has maximum sales of 301397792.46000004 is 20

```

In [35]: datasets.plot(kind='line', x='Store',y='Weekly_Sales', alpha=1.5,fig=(4,5),color='green')

```

Out[35]: <AxesSubplot:xlabel='Store'>



**Thus, the store with maximum sales of 301397792.46000004 is store no. 20**

**Which store has maximum standard deviation i.e., the sales vary a lot and the coefficient of mean**

**to standard deviation**

```
In [12]: std_dev=[]  
std_dev=datasets.groupby(["Store"])[  
max_std=max(datasets.groupby(["Store"])[  
std_dev
```

```
Out[12]: Store  
1      155980.767761  
2      237683.694682  
3       46319.631557  
4      266201.442297  
5       37737.965745  
6      212525.855862  
7      112585.469220  
8      106280.829881  
9       69028.666585  
10     302262.062504  
11     165833.887863  
12     139166.871880  
13     265506.995776  
14     317569.949476  
15     120538.652043  
16      85769.680133  
17     112162.936087  
18     176641.510839  
19     191722.638730  
20     275900.562742  
21     128752.812853  
22     161251.350631  
23     249788.038068  
24     167745.677567  
25     112976.788600  
26     110431.288141  
27     239930.135688  
28     181758.967539  
29      99120.136596  
30     22809.665590  
31     125855.942933  
32     138017.252087  
33      24132.927322  
34     104630.164676  
35     211243.457791  
36      60725.173579  
37      21837.461190
```

```
38      42768.169450
39      217466.454833
40      119002.112858
41      187907.162766
42       50262.925530
43      40598.413260
44      24762.832015
45      130168.526635
Name: Weekly_Sales, dtype: float64
```

```
In [13]: for i in range(1,46):
          if max_std==std_dev[i]:
              print("Store which has maximum standard deviation of {} is {}".format(max_sales,i))
```

Store which has maximum standard deviation of 301397792.46000004 is 14

**Thus, the store with maximum standard deviation of 301397792.46000004 is store no. 14.**



```
In [14]: mean=[]  
mean=datasets.groupby(['Store'])['Weekly_Sales'].mean()  
mean
```

```
Out[14]: Store  
1      1.555264e+06  
2      1.925751e+06  
3      4.027044e+05  
4      2.094713e+06  
5      3.180118e+05  
6      1.564728e+06  
7      5.706173e+05  
8      9.087495e+05  
9      5.439806e+05  
10     1.899425e+06  
11     1.356383e+06  
12     1.009002e+06  
13     2.003620e+06  
14     2.020978e+06  
15     6.233125e+05  
16     5.192477e+05  
17     8.935814e+05  
18     1.084718e+06  
19     1.444999e+06  
20     2.107677e+06  
21     7.560691e+05  
22     1.028501e+06  
23     1.389864e+06  
24     1.356755e+06  
25     7.067215e+05  
26     1.002912e+06  
27     1.775216e+06  
28     1.323522e+06  
29     5.394514e+05  
30     4.385796e+05  
31     1.395901e+06  
32     1.166568e+06  
33     2.598617e+05  
34     9.667816e+05  
35     9.197250e+05  
36     3.735120e+05  
37     5.189003e+05  
38     3.857317e+05
```

```
39    1.450668e+06
40    9.641280e+05
41    1.268125e+06
42    5.564039e+05
43    6.333247e+05
44    3.027489e+05
45    7.859814e+05
Name: Weekly_Sales, dtype: float64
```

## The coefficient of mean to standard deviation

```
In [20]: for i in range(1,46):  
         cv=std_dev[i]/mean[i]  
         print("Coefficient of mean to standard deviation of store {} is {}".format(i,cv))
```

```
Coefficient of mean to standard deviation of store 1 is 0.10029212268130752  
Coefficient of mean to standard deviation of store 2 is 0.12342387633191824  
Coefficient of mean to standard deviation of store 3 is 0.11502140735338794  
Coefficient of mean to standard deviation of store 4 is 0.12708253937002847  
Coefficient of mean to standard deviation of store 5 is 0.11866844091939749  
Coefficient of mean to standard deviation of store 6 is 0.13582285902663824  
Coefficient of mean to standard deviation of store 7 is 0.19730468653665717  
Coefficient of mean to standard deviation of store 8 is 0.11695283213906865  
Coefficient of mean to standard deviation of store 9 is 0.12689546764678367  
Coefficient of mean to standard deviation of store 10 is 0.15913349066639104  
Coefficient of mean to standard deviation of store 11 is 0.12226183360044687  
Coefficient of mean to standard deviation of store 12 is 0.1379253219997774  
Coefficient of mean to standard deviation of store 13 is 0.13251362792719143  
Coefficient of mean to standard deviation of store 14 is 0.1571367360094833  
Coefficient of mean to standard deviation of store 15 is 0.19338398778229554  
Coefficient of mean to standard deviation of store 16 is 0.16518065494781425  
Coefficient of mean to standard deviation of store 17 is 0.1255206714123267  
Coefficient of mean to standard deviation of store 18 is 0.1628454974226666  
Coefficient of mean to standard deviation of store 19 is 0.13268011534826385  
Coefficient of mean to standard deviation of store 20 is 0.1309026856173849  
Coefficient of mean to standard deviation of store 21 is 0.1702923921734945  
Coefficient of mean to standard deviation of store 22 is 0.1567828757814859  
Coefficient of mean to standard deviation of store 23 is 0.1797211491975351  
Coefficient of mean to standard deviation of store 24 is 0.1236373766137555  
Coefficient of mean to standard deviation of store 25 is 0.15986040240723462  
Coefficient of mean to standard deviation of store 26 is 0.11011066299216304  
Coefficient of mean to standard deviation of store 27 is 0.13515544496695323  
Coefficient of mean to standard deviation of store 28 is 0.13732974165124925  
Coefficient of mean to standard deviation of store 29 is 0.18374246746816345  
Coefficient of mean to standard deviation of store 30 is 0.05200803855544513  
Coefficient of mean to standard deviation of store 31 is 0.09016105262945949  
Coefficient of mean to standard deviation of store 32 is 0.1183104917616628  
Coefficient of mean to standard deviation of store 33 is 0.09286835290692934  
Coefficient of mean to standard deviation of store 34 is 0.1082252383219858  
Coefficient of mean to standard deviation of store 35 is 0.22968111389976448  
Coefficient of mean to standard deviation of store 36 is 0.16257891245773293  
Coefficient of mean to standard deviation of store 37 is 0.04208411895180788  
Coefficient of mean to standard deviation of store 38 is 0.11087544692154411  
Coefficient of mean to standard deviation of store 39 is 0.1499077910802601
```

Coefficient of mean to standard deviation of store 40 is 0.12342978096269436  
 Coefficient of mean to standard deviation of store 41 is 0.14817711243560774  
 Coefficient of mean to standard deviation of store 42 is 0.09033532809329818  
 Coefficient of mean to standard deviation of store 43 is 0.06410362927019238  
 Coefficient of mean to standard deviation of store 44 is 0.08179331054568031  
 Coefficient of mean to standard deviation of store 45 is 0.16561272979512937

## store/s has good quarterly growth rate in Q3'2012...

In [42]: `datasets.Date = pd.to_datetime(datasets.Date)`

In [44]: `df_pivot = datasets.pivot_table(index='Store', columns='Date', values='Weekly_Sales', aggfunc='sum')`  
`df_pivot.head()`

Out[44]:

	Date	2010-01-10	2010-02-04	2010-02-07	2010-02-19	2010-02-26	2010-03-09	2010-03-12	2010-03-19	2010-03-26	2010-04-06	...	2012
Store													
1		1453329.50	1594968.28	1492418.14	1611968.17	1409727.59	1540163.53	1548033.78	1472515.79	1404429.92	1615524.71	...	1675
2		1827440.43	2066187.72	2003940.64	2124451.54	1865097.27	1904608.09	2015781.27	1946070.88	1750197.81	2102539.93	...	1937
3		358784.10	423294.40	381151.72	421642.19	407204.86	366473.97	476420.77	375328.59	359949.27	396968.80	...	445
4		1842821.02	1979247.12	1881337.21	2049860.26	1925728.84	1935857.58	2102530.17	1897429.36	1762539.30	1903290.58	...	2202
5		283178.12	331406.00	305993.27	303447.57	270281.63	323798.00	344490.88	281706.41	273282.97	337825.89	...	347

5 rows × 143 columns

In [46]: `df_Qr = (df_pivot.groupby(pd.PeriodIndex(df_pivot.columns, freq='Q'), axis=1).apply(lambda x: x.sum(axis=1)/x.shape[0], axis=1))`  
`df_Qr.columns`

Out[46]: `PeriodIndex(['2010Q1', '2010Q2', '2010Q3', '2010Q4', '2011Q1', '2011Q2', '2011Q3', '2011Q4', '2012Q1', '2012Q2', '2012Q3', '2012Q4'], dtype='period[Q-DEC]', name='Date', freq='Q-DEC')`

```
In [50]: df_Qr3_2012 =df_Q['2012Q3']  
df_Qr3_2012.sort_values(ascending=False)
```

```
Out[50]: Store  
4      2.137677e+06  
20     2.055495e+06  
13     2.026666e+06  
2      1.866406e+06  
10     1.764113e+06  
27     1.682603e+06  
14     1.678369e+06  
39     1.574996e+06  
1      1.552767e+06  
6      1.528435e+06  
23     1.425305e+06  
19     1.387028e+06  
31     1.371194e+06  
41     1.364466e+06  
24     1.343833e+06  
11     1.341197e+06  
28     1.254638e+06  
32     1.178514e+06  
18     1.042293e+06  
26     1.034798e+06  
22     9.848787e+05  
12     9.814590e+05  
40     9.706384e+05  
17     9.611665e+05  
34     9.563549e+05  
8      9.061550e+05  
35     8.543436e+05  
45     7.376035e+05  
21     7.002923e+05  
25     6.924534e+05  
43     6.147272e+05  
7      6.101995e+05  
15     5.757812e+05  
42     5.692367e+05  
9      5.440200e+05  
16     5.367759e+05  
37     5.208770e+05  
29     5.106552e+05  
30     4.318312e+05
```

```

38    4.274415e+05
3     4.138747e+05
44    3.350405e+05
5     3.233852e+05
36    2.981770e+05
33    2.647560e+05
Name: 2012Q3, dtype: float64

```

stores have good quarterly growth rate in Q3'2012 are 4,20,13 and so on.

Top one is store no. 4 and it's quarterly growth rate in Q3'2012 is 2.137677e+06

## Holidays having higher sales...

```

In [57]: Christmas_sales=datasets.loc[(datasets["Date"]=="2010-12-31") | (datasets["Date"]=="2011-12-31") | (datasets["Date"]=="2012-12-31")]
Christmas_sales.head(10)

```

Out[57]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
<b>47</b>	1	2010-12-31	1367320.01	1	48.43	2.943	211.404932	7.838
<b>190</b>	2	2010-12-31	1750434.55	1	47.30	2.943	211.064774	8.163
<b>333</b>	3	2010-12-31	382677.76	1	53.20	2.943	214.698647	7.564
<b>476</b>	4	2010-12-31	1794868.74	1	38.09	2.955	127.087677	7.127
<b>619</b>	5	2010-12-31	298180.18	1	49.79	2.943	211.956714	6.768
<b>762</b>	6	2010-12-31	1464050.02	1	49.14	2.943	212.914967	7.007
<b>905</b>	7	2010-12-31	729572.08	1	13.76	2.829	191.255700	9.137
<b>1048</b>	8	2010-12-31	773586.49	1	41.47	2.943	214.744730	6.433
<b>1191</b>	9	2010-12-31	459770.85	1	45.92	2.943	214.926813	6.560
<b>1334</b>	10	2010-12-31	1707298.14	1	49.67	3.148	127.087677	9.003

```
In [58]: print ("Total sales in christmas holidays is {}".format(Christmas_sales["Weekly_Sales"].sum()))
```

Total sales in christmas holidays is 40432519.0

```
In [61]: Thanksgivings=datasets.loc[(datasets["Date"]=="2010-11-26") | (datasets["Date"]=="2011-11-25") |
                                     (datasets["Date"]=="2012-11-23") | (datasets["Date"]=="2013-11-29")]
Thanksgivings.head(10)
```

Out[61]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
<b>42</b>	1	2010-11-26	1955624.11	1	64.52	2.735	211.748433	7.838
<b>94</b>	1	2011-11-25	2033320.66	1	60.14	3.236	218.467621	7.866
<b>185</b>	2	2010-11-26	2658725.29	1	62.98	2.735	211.406287	8.163
<b>237</b>	2	2011-11-25	2614202.30	1	56.36	3.236	218.113027	7.441
<b>328</b>	3	2010-11-26	565567.84	1	68.71	2.735	215.061402	7.564
<b>380</b>	3	2011-11-25	556925.19	1	68.00	3.236	221.901118	7.197
<b>471</b>	4	2010-11-26	2789469.45	1	48.08	2.752	126.669267	7.127
<b>523</b>	4	2011-11-25	3004702.33	1	47.96	3.225	129.836400	5.143
<b>614</b>	5	2010-11-26	488362.61	1	66.15	2.735	212.303441	6.768
<b>666</b>	5	2011-11-25	507900.07	1	61.93	3.236	219.042820	6.300

```
In [62]: print ("Total sales in Thanksgiving holidays is {}".format(Thanksgivings["Weekly_Sales"].sum()))
```

Total sales in Thanksgiving holidays is 132414608.5

```
In [64]: Labour_Day=datasets.loc[(datasets["Date"]=="2010-09-10") | (datasets["Date"]=="2011-09-09") |
                                (datasets["Date"]=="2012-09-07") | (datasets["Date"]=="2013-09-06")]
Labour_Day.head(10)
```

Out[64]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
<b>83</b>	1	2011-09-09	1540471.24	1	76.00	3.546	215.861056	7.962
<b>226</b>	2	2011-09-09	1748000.65	1	77.97	3.546	215.514829	7.852
<b>369</b>	3	2011-09-09	377347.49	1	81.72	3.546	219.213531	7.567
<b>512</b>	4	2011-09-09	2093139.01	1	73.34	3.554	129.368613	5.644
<b>655</b>	5	2011-09-09	321110.22	1	79.04	3.546	216.422682	6.529
<b>798</b>	6	2011-09-09	1483574.38	1	80.21	3.546	217.398030	6.925
<b>941</b>	7	2011-09-09	613135.23	1	45.61	3.566	194.638785	8.622
<b>1084</b>	8	2011-09-09	848358.09	1	69.01	3.546	219.260435	6.425
<b>1227</b>	9	2011-09-09	528784.86	1	75.65	3.546	219.445767	6.404
<b>1370</b>	10	2011-09-09	1670579.82	1	89.06	3.771	129.368613	8.257

```
In [65]: print ("Total sales in Labour day is {}".format(Labour_Day["Weekly_Sales"].sum()))
Total sales in Labour day is 46763227.529999994
```

```
In [68]: Super_Bowl=datasets.loc[(datasets["Date"]=="2010-02-12") | (datasets["Date"]=="2011-02-11") |
                                (datasets["Date"]=="2012-02-10") | (datasets["Date"]=="2013-02-08")]
Super_Bowl.head(10)
```

Out[68]:

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

```
In [69]: Super_Bowl["Weekly_Sales"].sum()
```

Out[69]: 0.0



```
In [71]: print("Holidays which have higher sales is Thanksgivings. The total weekly sales of thanksgiving holidays is",
            Thanksgivings["Weekly_Sales"].sum())
```

Holidays which have higher sales is Thanksgivings. The total weekly sales of thanksgiving holidays is 132414608.5

**Holidays which have higher sales is Thanksgivings. The total weekly sales of thanksgiving holidays is 132414608.5**

## Monthly and semester view of sales in units and give insights

```
In [73]: datasets["Year"] = pd.DatetimeIndex(datasets['Date']).year
          datasets["Month"] = pd.DatetimeIndex(datasets['Date']).month
          datasets
```

Out[73]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Year	Month
0	1	2010-05-02	1643690.90	0	42.31	2.572	211.096358	8.106	2010	5
1	1	2010-12-02	1641957.44	1	38.51	2.548	211.242170	8.106	2010	12
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106	2010	2
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106	2010	2
4	1	2010-05-03	1554806.68	0	46.50	2.625	211.350143	8.106	2010	5
...	...	...	...	...	...	...	...	...	...	...
6430	45	2012-09-28	713173.95	0	64.88	3.997	192.013558	8.684	2012	9
6431	45	2012-05-10	733455.07	0	64.89	3.985	192.170412	8.667	2012	5
6432	45	2012-12-10	734464.36	0	54.47	4.000	192.327265	8.667	2012	12
6433	45	2012-10-19	718125.53	0	56.47	3.969	192.330854	8.667	2012	10
6434	45	2012-10-26	760281.43	0	58.85	3.882	192.308899	8.667	2012	10

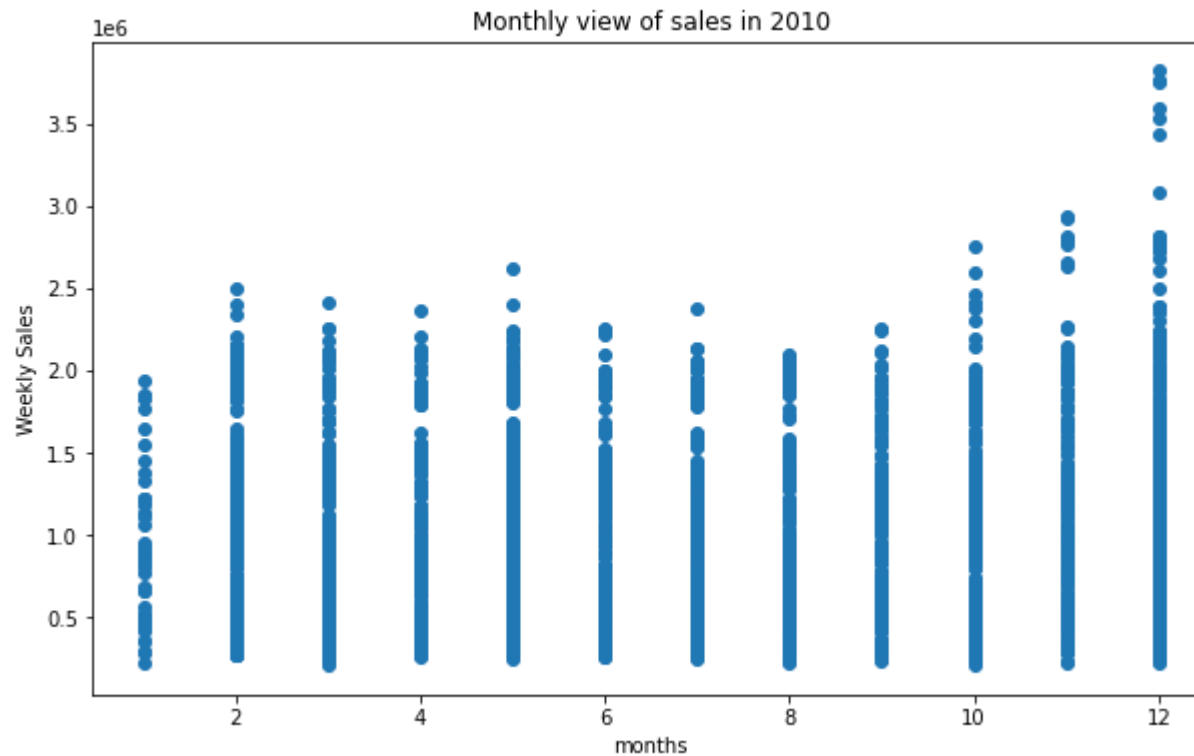
6435 rows × 10 columns

```
In [75]: year_2010=datasets.loc[datasets["Year"]==2010]
year_2011=datasets.loc[datasets["Year"]==2011]
year_2012=datasets.loc[datasets["Year"]==2012]
```

### Monthly view of sales for each year

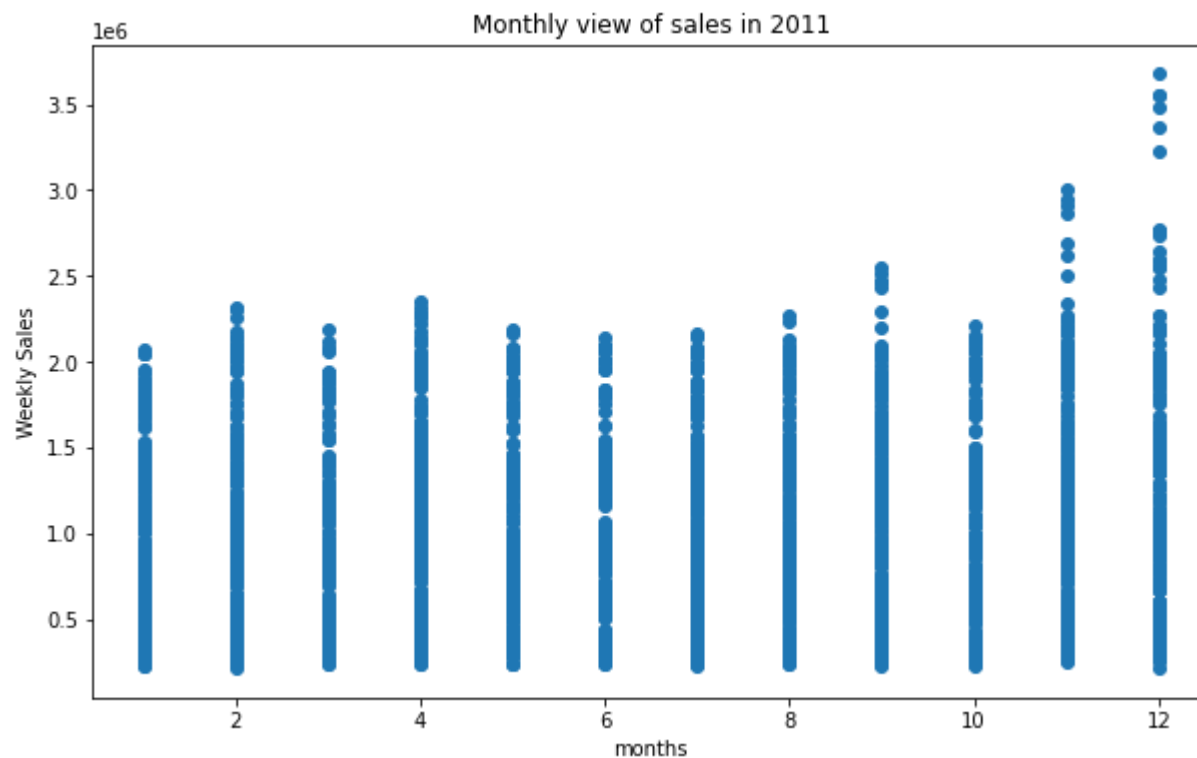
```
In [78]: plt.figure(figsize=(10,6))
plt.scatter(year_2010["Month"],year_2010["Weekly_Sales"])
plt.xlabel("months")
plt.ylabel("Weekly Sales")
plt.title("Monthly view of sales in 2010")
```

Out[78]: Text(0.5, 1.0, 'Monthly view of sales in 2010')



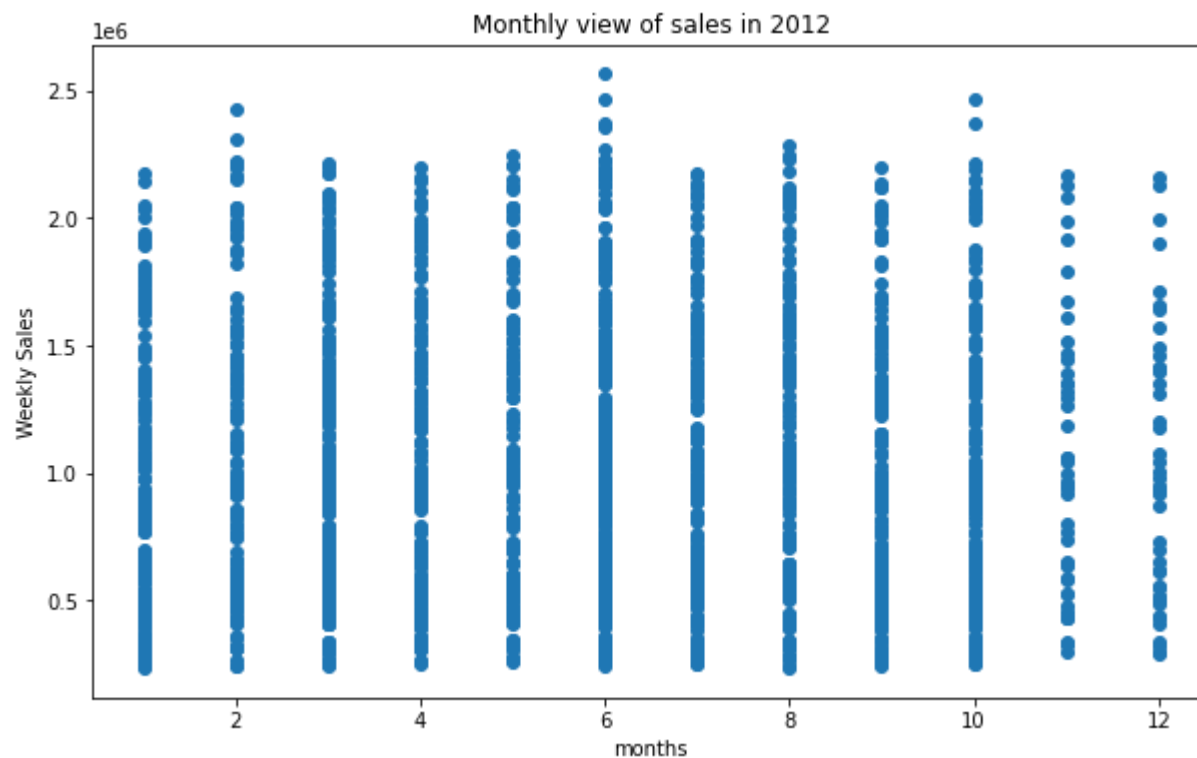
```
In [81]: plt.figure(figsize=(10,6))  
plt.scatter(year_2011["Month"],year_2011["Weekly_Sales"],cmap='green')  
plt.xlabel("months")  
plt.ylabel("Weekly Sales")  
plt.title("Monthly view of sales in 2011")
```

Out[81]: Text(0.5, 1.0, 'Monthly view of sales in 2011')



```
In [82]: plt.figure(figsize=(10,6))  
plt.scatter(year_2012["Month"],year_2012["Weekly_Sales"])  
plt.xlabel("months")  
plt.ylabel("Weekly Sales")  
plt.title("Monthly view of sales in 2012")
```

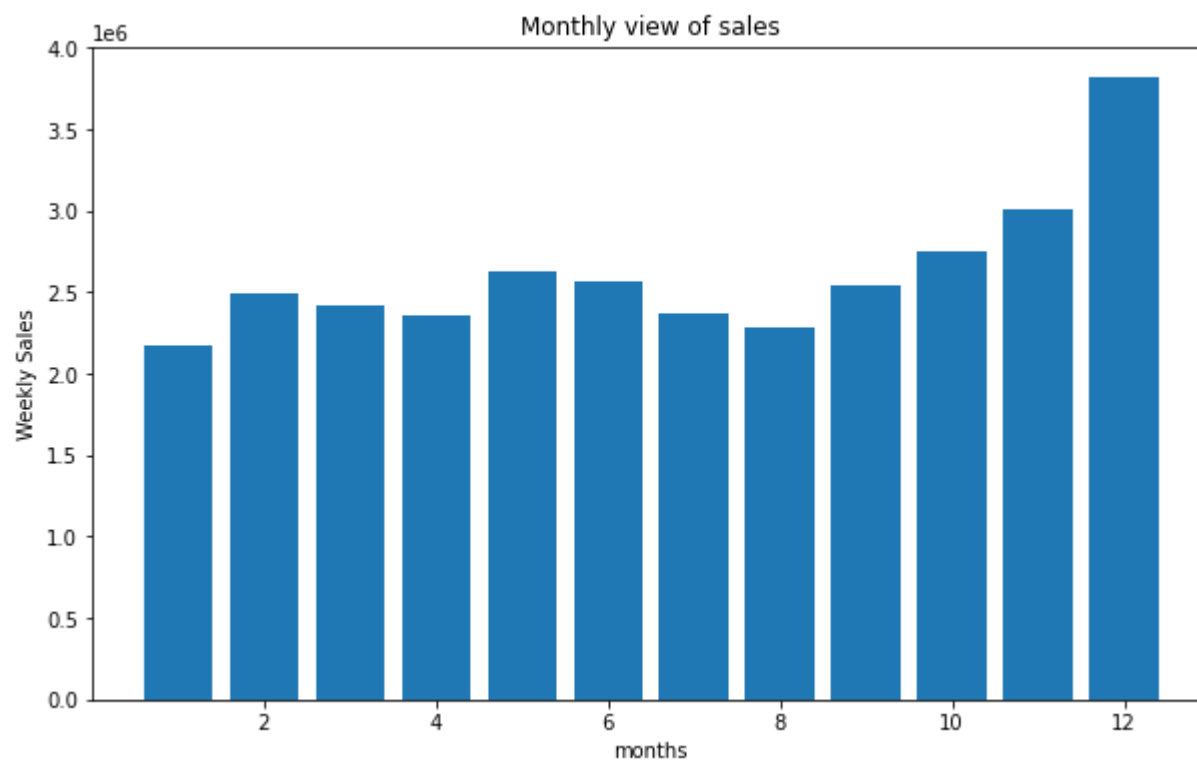
```
Out[82]: Text(0.5, 1.0, 'Monthly view of sales in 2012')
```



**Monthly view of sales...**

```
In [83]: plt.figure(figsize=(10,6))  
plt.bar(datasets["Month"],datasets["Weekly_Sales"])  
plt.xlabel("months")  
plt.ylabel("Weekly Sales")  
plt.title("Monthly view of sales")
```

```
Out[83]: Text(0.5, 1.0, 'Monthly view of sales')
```



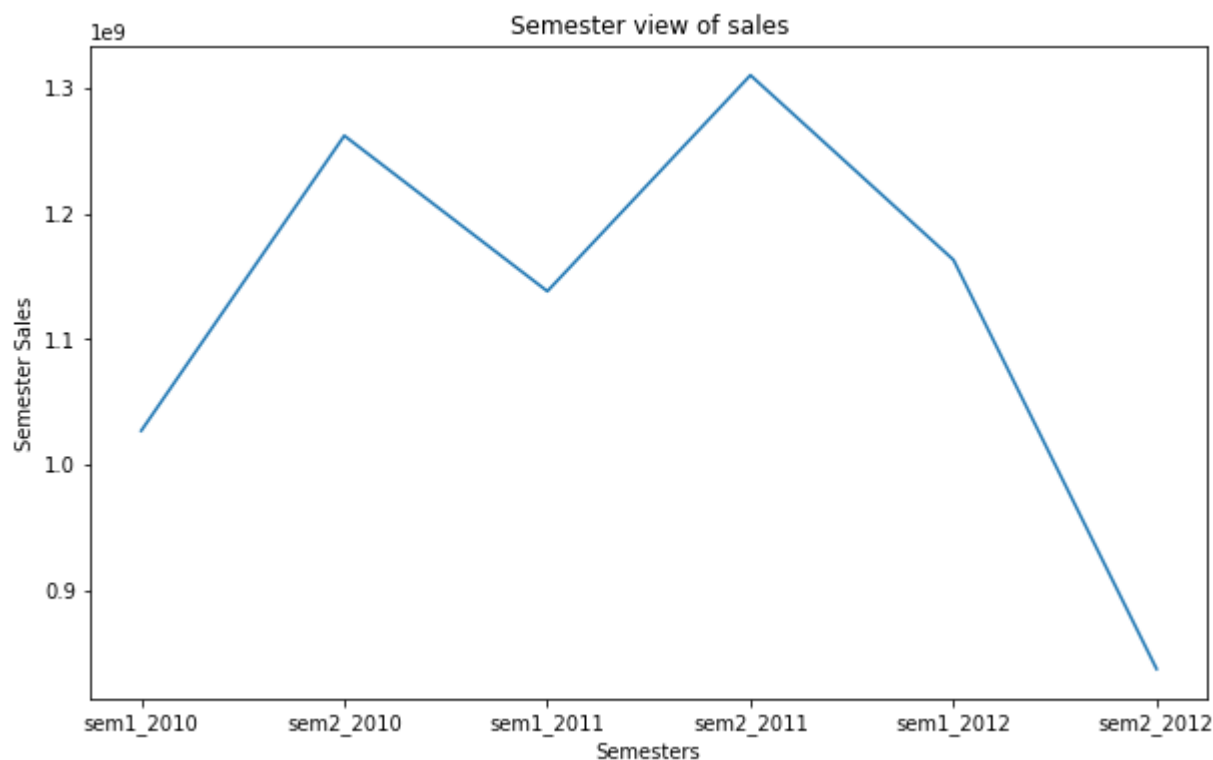
**Semester view of sales**

```
In [84]: semester_sales=[]  
semester_sales.append(year_2010.loc[year_2010["Month"]<7,["Weekly_Sales"]].sum())  
semester_sales.append(year_2010.loc[year_2010["Month"]>6,["Weekly_Sales"]].sum())  
semester_sales.append(year_2011.loc[year_2011["Month"]<7,["Weekly_Sales"]].sum())  
semester_sales.append(year_2011.loc[year_2011["Month"]>6,["Weekly_Sales"]].sum())  
semester_sales.append(year_2012.loc[year_2012["Month"]<7,["Weekly_Sales"]].sum())  
semester_sales.append(year_2012.loc[year_2012["Month"]>6,["Weekly_Sales"]].sum())
```

```
In [85]: semester_names=["sem1_2010","sem2_2010","sem1_2011","sem2_2011","sem1_2012","sem2_2012"]
```

```
In [89]: plt.figure(figsize=(10,6))  
plt.plot(semester_names,semester_sales)  
plt.xlabel("Semesters")  
plt.ylabel("Semester Sales")  
plt.title("Semester view of sales")
```

```
Out[89]: Text(0.5, 1.0, 'Semester view of sales')
```



## For Store 1 – Building prediction models to forecast demand

### Linear Regression Model

```
In [90]: x=datasets.drop(["Weekly_Sales", "Date"],axis=1)
y=datasets["Weekly_Sales"]
```

```
In [95]: linreg=LinearRegression(n_jobs=-1)
xtrain,xtest,ytrain,ytest=model_selection.train_test_split(x,y,test_size=0.4,random_state=42)

linreg.fit(xtrain,ytrain)
```

```
Out[95]: LinearRegression(n_jobs=-1)
```

```
In [96]: print(linreg.intercept_)
print(linreg.coef_)
```

```
80380267.57798947
[-15063.98405298  16701.66272831  -806.19443715  47215.71929544
 -2175.95412712 -26362.8254153  -39081.36479694  11681.52115132]
```

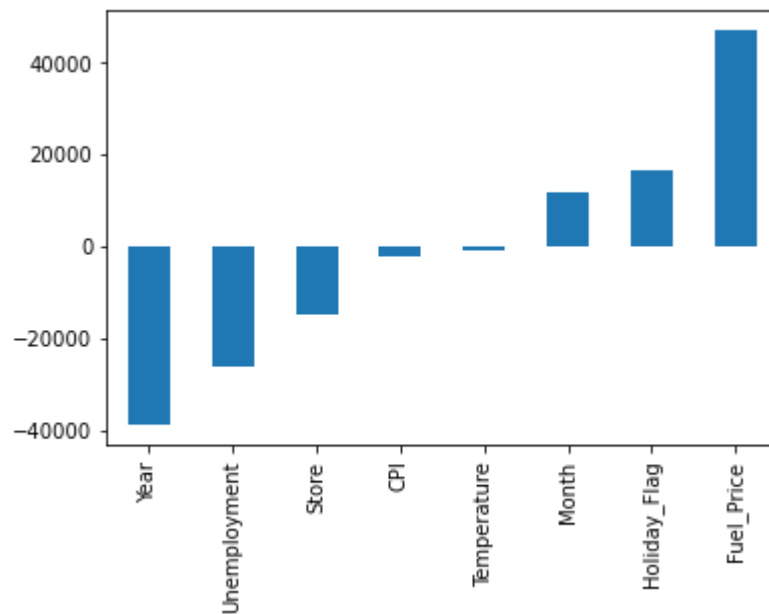
```
In [97]: x.columns
```

```
Out[97]: Index(['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI',
               'Unemployment', 'Year', 'Month'],
              dtype='object')
```

```
In [99]: features=['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Year', 'Month']
```

```
In [100]: relation=pd.Series(linreg.coef_,x.columns).sort_values()  
relation.plot(kind="bar")
```

Out[100]: <AxesSubplot:>



**The plot shows that fuel price have greater positive impact on weekly sales. Unemployment also has certain negative impact on weekly sales. CPI has least impact towards weekly sales.**



```
In [101]: print(format(linreg.score(xtest,ytest)))
```

0.14998336593087003

```
In [102]: print(sqrt(mean_squared_error(ytrain,linreg.predict(xtrain))))
```

522629.1662854094

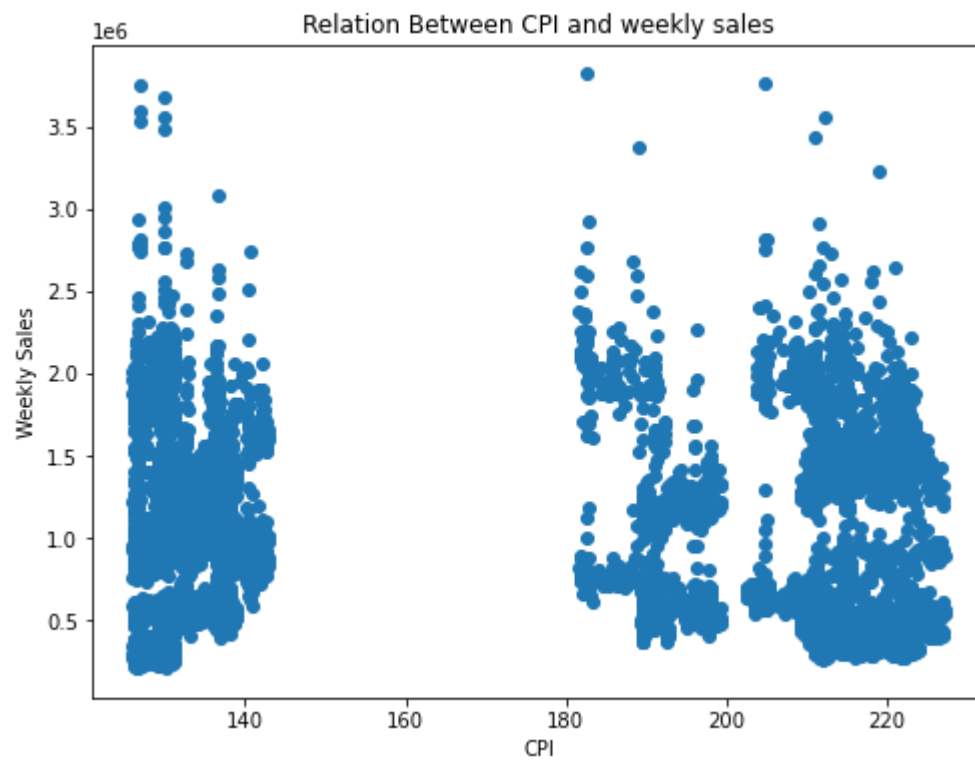
```
In [103]: print(sqrt(mean_squared_error(ytest,linreg.predict(xtest))))
```

519668.86389805033

**Thus test error is less compared to train error. Hence our predicted model is well and good.**

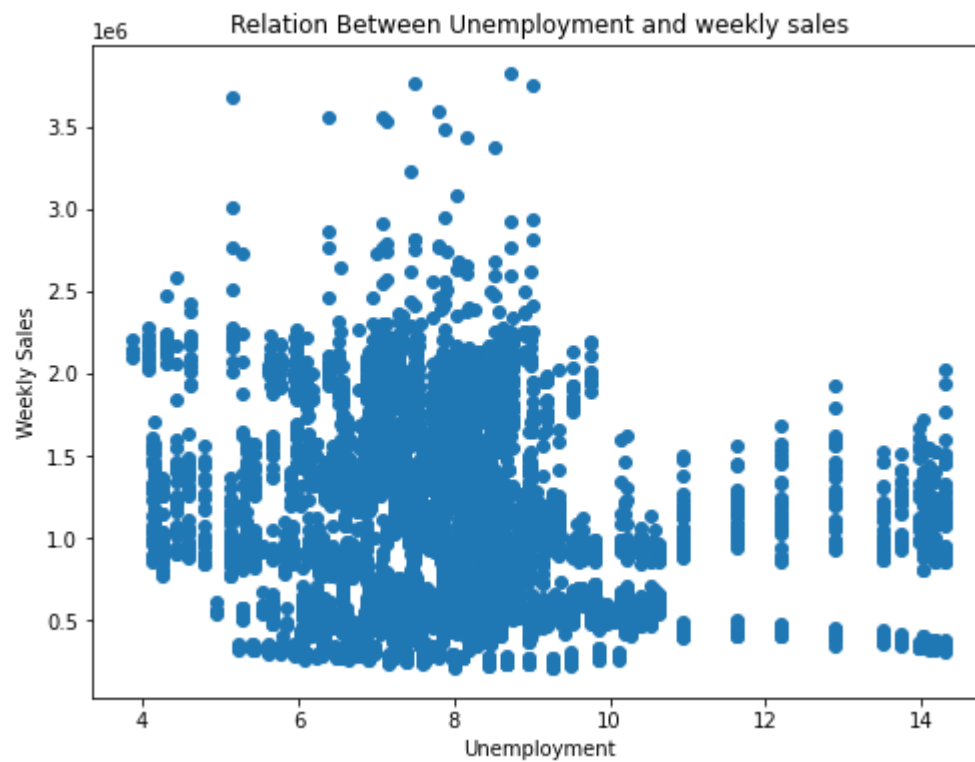
```
In [104]: plt.figure(figsize=(8,6))  
plt.scatter(datasets["CPI"],datasets["Weekly_Sales"])  
plt.title("Relation Between CPI and weekly sales")  
plt.xlabel("CPI")  
plt.ylabel("Weekly Sales")
```

```
Out[104]: Text(0, 0.5, 'Weekly Sales')
```



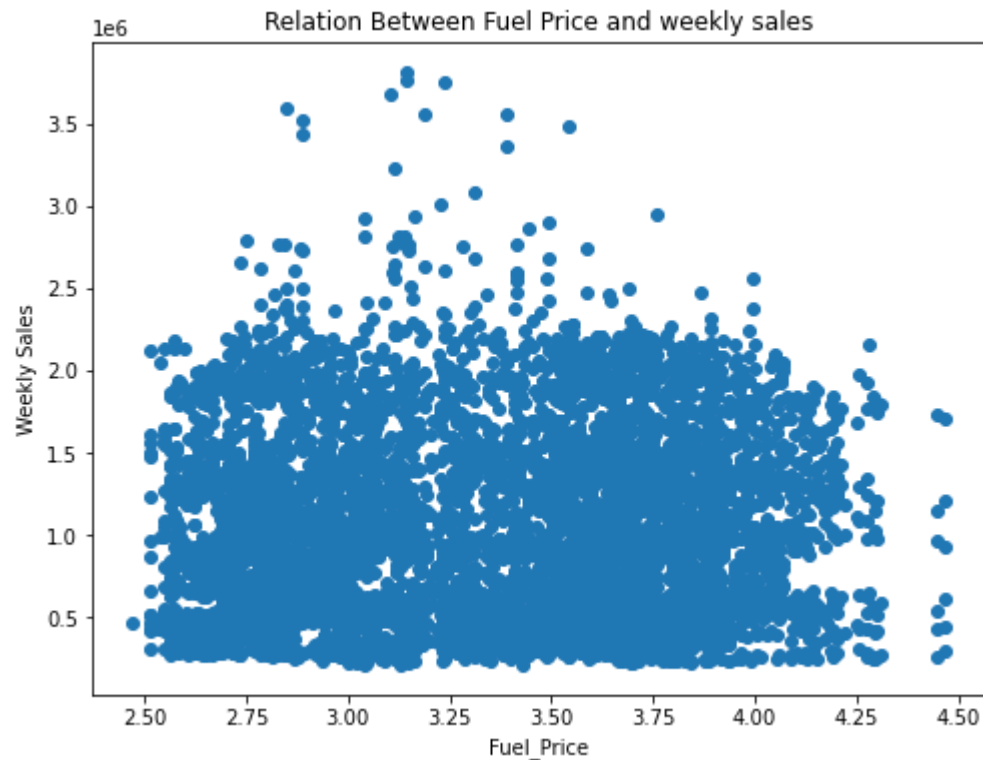
```
In [106]: plt.figure(figsize=(8,6))  
plt.scatter(datasets["Unemployment"],datasets["Weekly_Sales"])  
plt.title("Relation Between Unemployment and weekly sales")  
plt.xlabel("Unemployment")  
plt.ylabel("Weekly Sales")
```

```
Out[106]: Text(0, 0.5, 'Weekly Sales')
```



```
In [107]: plt.figure(figsize=(8,6))
plt.scatter(datasets["Fuel_Price"],datasets["Weekly_Sales"])
plt.title("Relation Between Fuel Price and weekly sales")
plt.xlabel("Fuel_Price")
plt.ylabel("Weekly Sales")
```

```
Out[107]: Text(0, 0.5, 'Weekly Sales')
```



**Changing dates into days by creating new variable.**

```
In [109]: datasets['days'] = datasets['Date'].dt.day_name()  
datasets
```

Out[109]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Year	Month	days
0	1	2010-05-02	1643690.90	0	42.31	2.572	211.096358	8.106	2010	5	Sunday
1	1	2010-12-02	1641957.44	1	38.51	2.548	211.242170	8.106	2010	12	Thursday
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106	2010	2	Friday
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106	2010	2	Friday
4	1	2010-05-03	1554806.68	0	46.50	2.625	211.350143	8.106	2010	5	Monday
...	...	...	...	...	...	...	...	...	...	...	...
6430	45	2012-09-28	713173.95	0	64.88	3.997	192.013558	8.684	2012	9	Friday
6431	45	2012-05-10	733455.07	0	64.89	3.985	192.170412	8.667	2012	5	Thursday
6432	45	2012-12-10	734464.36	0	54.47	4.000	192.327265	8.667	2012	12	Monday
6433	45	2012-10-19	718125.53	0	56.47	3.969	192.330854	8.667	2012	10	Friday
6434	45	2012-10-26	760281.43	0	58.85	3.882	192.308899	8.667	2012	10	Friday

6435 rows × 11 columns

In [ ]:

In [ ]:

In [ ]: