Walmart_Solution

April 8, 2021

[1]:
```
import numpy as np
import pandas as pd
from datetime import date
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
df = pd.read_csv("Walmart_Store_sales.csv")
```

[3]:
```
df.head()
```

[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

[4]:
```
df.isnull().sum()
```

[4]:
```
Store           0
Date            0
Weekly_Sales    0
Holiday_Flag    0
Temperature     0
Fuel_Price      0
CPI             0
Unemployment    0
dtype: int64
```

[5]:
```
df.shape
```

[5]:
```
(6435, 8)
```

[6]:
```
df.columns
```

[6]:
```
Index(['Store', 'Date', 'Weekly_Sales', 'Holiday_Flag', 'Temperature',
       'Fuel_Price', 'CPI', 'Unemployment'],
      dtype='object')
```

[7]:
```
df.Store.unique()
```

[7]:
```
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])
```

1 *Which Store has maximum values?*

[8]:
```
sales_list=df.groupby(['Store'])['Weekly_Sales'].sum()
max_sales=max(df.groupby(['Store'])['Weekly_Sales'].sum())
sales_list
```

[8]:
```
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
```

[9]:
```
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
```

2 *Which store has maximum standard deviation*

[10]:
```
std_dev=df.groupby(['Store'])['Weekly_Sales'].std()
max_std=max(df.groupby(['Store'])['Weekly_Sales'].std())
print(std_dev)
```

```
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
```

[11]:
```
for i in range(1,46):
    if max_std==std_dev[i]:
        print('Store which has maximum standard deviation of {} is {}'.
 ↪format(max_std,i))
```

```
Store which has maximum standard deviation of 317569.9494755081 is 14
```

3 *The coefficient of mean to standard deviation*

[12]:
```
#Coefficient of mean to std
co_eff_std = df.groupby('Store')['Weekly_Sales'].var().
 ↪sort_values(ascending=False)
co_eff_std
```

[12]:
```
Store
14    1.008507e+11
10    9.136235e+10
20    7.612112e+10
4     7.086321e+10
13    7.049396e+10
23    6.239406e+10
27    5.756647e+10
2     5.649354e+10
39    4.729166e+10
6     4.516724e+10
35    4.462380e+10
19    3.675757e+10
41    3.530910e+10
28    3.303632e+10
18    3.120222e+10
24    2.813861e+10
11    2.750088e+10
22    2.600200e+10
1     2.433000e+10
12    1.936742e+10
32    1.904876e+10
45    1.694385e+10
21    1.657729e+10
31    1.583972e+10
15    1.452957e+10
40    1.416150e+10
25    1.276375e+10
7     1.267549e+10
17    1.258052e+10
26    1.219507e+10
8     1.129561e+10
34    1.094747e+10
29    9.824801e+09
16    7.356438e+09
9     4.764957e+09
36    3.687547e+09
42    2.526362e+09
3     2.145508e+09
```

```
38    1.829116e+09
43    1.648231e+09
5     1.424154e+09
44    6.131978e+08
33    5.823982e+08
30    5.202808e+08
37    4.768747e+08
Name: Weekly_Sales, dtype: float64
```

4 *Which store/s has good quarterly growth rate in Q3'2012*

[13]:
```
# Import datetime package
from datetime import datetime
df.head()
```

[13]:

	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	

	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

[14]:
```
df.shape
```

[14]:
```
(6435, 8)
```

[15]:
```
df_growth=df
df["Date"]=pd.to_datetime(df["Date"])
```

[16]:
```
# Third Quartile Period
date_from=pd.Timestamp(date(2012,7,1))
date_to = pd.Timestamp(date(2012,9,30))
```

[17]:
```
df_growth = df_growth[
    (df['Date'] > date_from ) &
    (df['Date'] < date_to)]
```

[18]:
```
df_growth.head()
```

[18]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
109	1	2012-09-03	1675431.16	0	58.76	3.669	221.059189	7.348
122	1	2012-08-06	1697230.96	0	78.30	3.452	221.749484	7.143
127	1	2012-07-13	1527014.04	0	77.12	3.256	221.924158	6.908
128	1	2012-07-20	1497954.76	0	80.42	3.311	221.932727	6.908
129	1	2012-07-27	1439123.71	0	82.66	3.407	221.941295	6.908

[19]:
```
# To find which store has good growth in quarter Q3
Q3_growth=[]
Q3_growth=df_growth.groupby(['Store'])['Weekly_Sales'].sum()
max_Q3_growth=max(df_growth.groupby(['Store'])['Weekly_Sales'].sum())
print(Q3_growth)
print(max_Q3_growth)
```

```
Store
1     18633209.98
2     22396867.61
3      4966495.93
4     25652119.35
5      3880621.88
6     18341221.11
7      7322393.92
8     10873860.34
9      6528239.56
10    21169356.45
11    16094363.07
12    11777508.50
13    24319994.35
14    20140430.40
15     6909374.37
16     6441311.11
17    11533998.38
18    12507521.72
19    16644341.31
20    24665938.11
21     8403507.99
22    11818544.33
23    17103654.36
24    16125999.86
25     8309440.44
```

```
26    12417575.35
27    20191238.11
28    15055659.67
29     6127862.07
30     5181974.44
31    16454328.46
32    14142164.84
33     3177072.43
34    11476258.98
35    10252122.68
36     3578123.58
37     6250524.08
38     5129297.64
39    18899955.17
40    11647661.37
41    16373588.44
42     6830839.86
43     7376726.03
44     4020486.01
45     8851242.32
Name: Weekly_Sales, dtype: float64
25652119.35
```

[20]:
```
for i in range(1,46):
    if max_Q3_growth==Q3_growth[i]:
        print('Store which has maximum Q3 growth of {} is {}'.
 ↪format(max_Q3_growth,i))
```

```
Store which has maximum Q3 growth of 25652119.35 is 4
```

4.0.1 Some holidays have a negative impact on sales. Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together

Holiday Events:

- Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13
- Labour Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13
- Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13
- Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

[21]:
```
Christmas_sales=df.loc[(df["Date"]=="2010-12-31") | (df["Date"]=="2011-12-31")␣
 ↪| (df["Date"]=="2012-12-28") | (df["Date"]=="2013-12-")]
Christmas_sales
```

[21]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	\
47	1	2010-12-31	1367320.01	1	48.43	2.943	

190	2	2010-12-31	1750434.55	1	47.30	2.943
333	3	2010-12-31	382677.76	1	53.20	2.943
476	4	2010-12-31	1794868.74	1	38.09	2.955
619	5	2010-12-31	298180.18	1	49.79	2.943
762	6	2010-12-31	1464050.02	1	49.14	2.943
905	7	2010-12-31	729572.08	1	13.76	2.829
1048	8	2010-12-31	773586.49	1	41.47	2.943
1191	9	2010-12-31	459770.85	1	45.92	2.943
1334	10	2010-12-31	1707298.14	1	49.67	3.148
1477	11	2010-12-31	1172003.10	1	55.03	2.943
1620	12	2010-12-31	891736.91	1	45.64	3.148
1763	13	2010-12-31	1675292.00	1	26.79	2.868
1906	14	2010-12-31	1623716.46	1	29.67	3.179
2049	15	2010-12-31	543754.17	1	26.54	3.336
2192	16	2010-12-31	575317.38	1	19.66	2.829
2335	17	2010-12-31	635862.55	1	20.79	2.868
2478	18	2010-12-31	887907.01	1	26.10	3.177
2621	19	2010-12-31	1275146.94	1	28.65	3.336
2764	20	2010-12-31	1799737.79	1	28.85	3.179
2907	21	2010-12-31	672903.23	1	47.19	2.943
3050	22	2010-12-31	774262.28	1	28.49	3.177
3193	23	2010-12-31	1169773.85	1	19.05	3.177
3336	24	2010-12-31	1208600.05	1	25.90	3.336
3479	25	2010-12-31	623092.54	1	25.89	3.179
3622	26	2010-12-31	877268.29	1	18.73	3.177
3765	27	2010-12-31	1440963.00	1	29.59	3.336
3908	28	2010-12-31	1090558.57	1	45.64	3.148
4051	29	2010-12-31	465992.02	1	28.49	3.177
4194	30	2010-12-31	397631.02	1	47.19	2.943
4337	31	2010-12-31	1198071.60	1	47.19	2.943
4480	32	2010-12-31	955463.84	1	27.70	2.829
4623	33	2010-12-31	219804.85	1	52.91	3.148
4766	34	2010-12-31	902109.69	1	34.11	2.955
4909	35	2010-12-31	576332.05	1	29.59	3.179
5052	36	2010-12-31	359310.65	1	52.88	2.949
5195	37	2010-12-31	460331.70	1	52.88	2.943
5338	38	2010-12-31	303908.81	1	45.64	3.148
5481	39	2010-12-31	1230012.16	1	52.45	2.943
5624	40	2010-12-31	811318.30	1	19.29	3.177
5767	41	2010-12-31	1001790.16	1	25.19	2.829
5910	42	2010-12-31	428953.60	1	49.67	3.148
6053	43	2010-12-31	534740.30	1	48.61	2.943
6196	44	2010-12-31	241937.11	1	26.79	2.868
6339	45	2010-12-31	679156.20	1	29.67	3.179

	CPI	Unemployment
47	211.404932	7.838

```
190   211.064774         8.163
333   214.698647         7.564
476   127.087677         7.127
619   211.956714         6.768
762   212.914967         7.007
905   191.255700         9.137
1048  214.744730         6.433
1191  214.926813         6.560
1334  127.087677         9.003
1477  214.698647         7.564
1620  127.087677        14.313
1763  127.087677         7.795
1906  182.571448         8.724
2049  132.815032         8.067
2192  191.255700         6.986
2335  127.087677         6.885
2478  132.815032         9.331
2621  132.815032         8.067
2764  204.643227         7.484
2907  211.064774         8.163
3050  136.665265         8.572
3193  132.815032         5.287
3336  132.815032         8.275
3479  204.643227         7.484
3622  132.815032         8.149
3765  136.665265         8.021
3908  127.087677        14.313
4051  132.815032        10.524
4194  211.064774         8.163
4337  211.064774         8.163
4480  191.255700         9.137
4623  127.087677         9.265
4766  127.087677        10.210
4909  136.665265         8.763
5052  210.182398         8.476
5195  210.182398         8.476
5338  127.087677        14.313
5481  210.182398         8.476
5624  132.815032         5.287
5767  191.255700         7.508
5910  127.087677         9.003
6053  203.417684        10.210
6196  127.087677         7.610
6339  182.571448         8.724
```

[22]:
```
print ("Total sales in Christmas holidays is {}" .
 ↪format(Christmas_sales["Weekly_Sales"].sum()))
```

```
Total sales in Christmas holidays is 40432519.0
```

[23]:
```
Thanksgivings=df.loc[(df["Date"]=="2010-11-26") | (df["Date"]=="2011-11-25") |␣
↪(df["Date"]=="2012-11-23") | (df["Date"]=="2013-11-29")]
Thanksgivings
```

[23]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
42	1	2010-11-26	1955624.11	1	64.52	2.735	211.748433	7.838
94	1	2011-11-25	2033320.66	1	60.14	3.236	218.467621	7.866
185	2	2010-11-26	2658725.29	1	62.98	2.735	211.406287	8.163
237	2	2011-11-25	2614202.30	1	56.36	3.236	218.113027	7.441
328	3	2010-11-26	565567.84	1	68.71	2.735	215.061402	7.564
…	…	…	…	…	…	…	…	…
6100	43	2011-11-25	669965.22	1	55.70	3.236	210.088857	10.148
6191	44	2010-11-26	307646.50	1	28.22	2.830	126.669267	7.610
6243	44	2011-11-25	309129.01	1	38.89	3.445	129.836400	6.078
6334	45	2010-11-26	1182500.16	1	46.15	3.039	182.783277	8.724
6386	45	2011-11-25	1170672.94	1	48.71	3.492	188.350400	8.523

[90 rows x 8 columns]

[24]:
```
print ("Total sales in Thanksgiving holidays is {}" .
↪format(Thanksgivings["Weekly_Sales"].sum()))
```

```
Total sales in Thanksgiving holidays is 132414608.5
```

[25]:
```
Labour_Day=df.loc[(df["Date"]=="2010-09-10") | (df["Date"]=="2011-09-09") |␣
↪(df["Date"]=="2012-09-07") | (df["Date"]=="2013-09-06")]
Labour_Day
```

[25]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price
83	1	2011-09-09	1540471.24	1	76.00	3.546
226	2	2011-09-09	1748000.65	1	77.97	3.546
369	3	2011-09-09	377347.49	1	81.72	3.546

```
512       4 2011-09-09    2093139.01             1        73.34       3.554
655       5 2011-09-09     321110.22             1        79.04       3.546
798       6 2011-09-09    1483574.38             1        80.21       3.546
941       7 2011-09-09     613135.23             1        45.61       3.566
1084      8 2011-09-09     848358.09             1        69.01       3.546
1227      9 2011-09-09     528784.86             1        75.65       3.546
1370     10 2011-09-09    1670579.82             1        89.06       3.771
1513     11 2011-09-09    1249439.95             1        84.91       3.546
1656     12 2011-09-09     922850.57             1        88.00       3.913
1799     13 2011-09-09    1872921.31             1        70.19       3.619
1942     14 2011-09-09    2202742.90             1        71.48       3.738
2085     15 2011-09-09     607593.51             1        67.59       3.930
2228     16 2011-09-09     574622.56             1        56.99       3.566
2371     17 2011-09-09    1161900.18             1        61.94       3.619
2514     18 2011-09-09     951549.61             1        68.11       3.809
2657     19 2011-09-09    1566712.79             1        68.28       3.930
2800     20 2011-09-09    2050542.56             1        68.74       3.738
2943     21 2011-09-09     653989.65             1        78.87       3.546
3086     22 2011-09-09    1004434.54             1        69.14       3.809
3229     23 2011-09-09    1423289.90             1        66.04       3.809
3372     24 2011-09-09    1527455.19             1        68.32       3.930
3515     25 2011-09-09     673248.48             1        67.51       3.738
3658     26 2011-09-09    1069710.97             1        60.98       3.809
3801     27 2011-09-09    1911470.84             1        70.93       3.930
3944     28 2011-09-09    1310087.00             1        88.00       3.913
4087     29 2011-09-09     505406.72             1        69.14       3.809
4230     30 2011-09-09     370897.82             1        78.87       3.546
4373     31 2011-09-09    1376670.27             1        78.87       3.546
4516     32 2011-09-09    1128237.30             1        61.24       3.566
4659     33 2011-09-09     281842.28             1        96.22       3.771
4802     34 2011-09-09     930506.14             1        70.05       3.554
4945     35 2011-09-09     922440.64             1        70.93       3.738
5088     36 2011-09-09     352960.64             1        77.94       3.499
5231     37 2011-09-09     506273.74             1        77.94       3.546
5374     38 2011-09-09     397771.68             1        88.00       3.913
5517     39 2011-09-09    1429345.86             1        79.15       3.546
5660     40 2011-09-09    1021391.99             1        64.83       3.809
5803     41 2011-09-09    1280958.97             1        58.31       3.566
5946     42 2011-09-09     608390.94             1        89.06       3.771
6089     43 2011-09-09     649128.23             1        79.29       3.546
6232     44 2011-09-09     295811.25             1        70.19       3.619
6375     45 2011-09-09     746129.56             1        71.48       3.738

             CPI  Unemployment
83    215.861056         7.962
226   215.514829         7.852
369   219.213531         7.567
```

```
512   129.368613         5.644
655   216.422682         6.529
798   217.398030         6.925
941   194.638785         8.622
1084  219.260435         6.425
1227  219.445767         6.404
1370  129.368613         8.257
1513  219.213531         7.567
1656  129.368613        13.503
1799  129.368613         6.877
1942  186.673738         8.625
2085  136.274581         7.806
2228  194.638785         6.338
2371  129.368613         6.745
2514  136.274581         8.890
2657  136.274581         7.806
2800  209.022556         7.274
2943  215.514829         7.852
3086  140.231017         8.023
3229  136.274581         4.584
3372  136.274581         8.358
3515  209.022556         7.274
3658  136.274581         7.767
3801  140.231017         7.850
3944  129.368613        13.503
4087  136.274581         9.863
4230  215.514829         7.852
4373  215.514829         7.852
4516  194.638785         8.622
4659  129.368613         8.442
4802  129.368613        10.641
4945  140.231017         8.684
5088  214.615538         8.177
5231  214.615538         8.177
5374  129.368613        13.503
5517  214.615538         8.177
5660  136.274581         4.584
5803  194.638785         6.901
5946  129.368613         8.257
6089  207.683389        10.641
6232  129.368613         6.560
6375  186.673738         8.625
```

[26]:
```
print ("Total sales in Labour day is {}" .format(Labour_Day["Weekly_Sales"].
↪sum()))
```

```
Total sales in Labour day is 46763227.529999994
```

[27]:
```
Super_Bowl=df.loc[(df["Date"]=="2010-02-12") | (df["Date"]=="2011-02-11") |␣
↪(df["Date"]=="2012-02-10") | (df["Date"]=="2013-02-08")]
Super_Bowl
```

[27]:
```
Empty DataFrame
Columns: [Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI,
Unemployment]
Index: []
```

[28]:
```
print ("Total sales in Super Bowl is {}" .format(Super_Bowl["Weekly_Sales"].
↪sum()))
```

```
Total sales in Super Bowl is 0.0
```

[29]:
```
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
```

5 *Monthly View of Sales*

[30]:
```
df["Year"]= pd.DatetimeIndex(df['Date']).year
df["Month"]= pd.DatetimeIndex(df['Date']).month
```

[31]:
```
df
```

[31]:

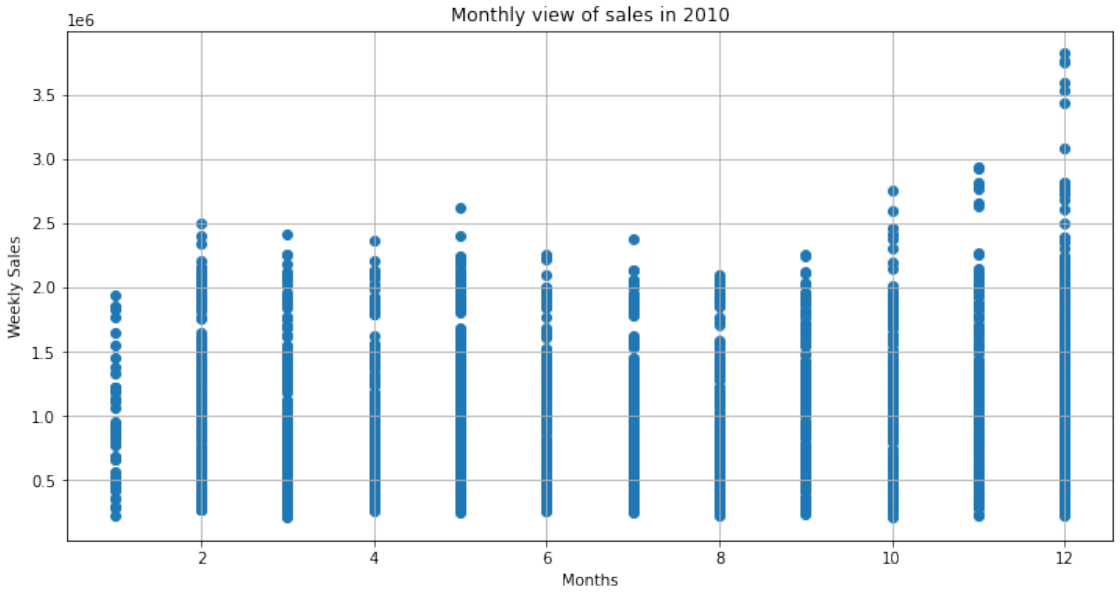
	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	\
0	1	2010-05-02	1643690.90	0	42.31	2.572	
1	1	2010-12-02	1641957.44	1	38.51	2.548	
2	1	2010-02-19	1611968.17	0	39.93	2.514	
3	1	2010-02-26	1409727.59	0	46.63	2.561	
4	1	2010-05-03	1554806.68	0	46.50	2.625	
…	…	…	…	…	…	…	
6430	45	2012-09-28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

	CPI	Unemployment	Year	Month
0	211.096358	8.106	2010	5
1	211.242170	8.106	2010	12
2	211.289143	8.106	2010	2
3	211.319643	8.106	2010	2
4	211.350143	8.106	2010	5

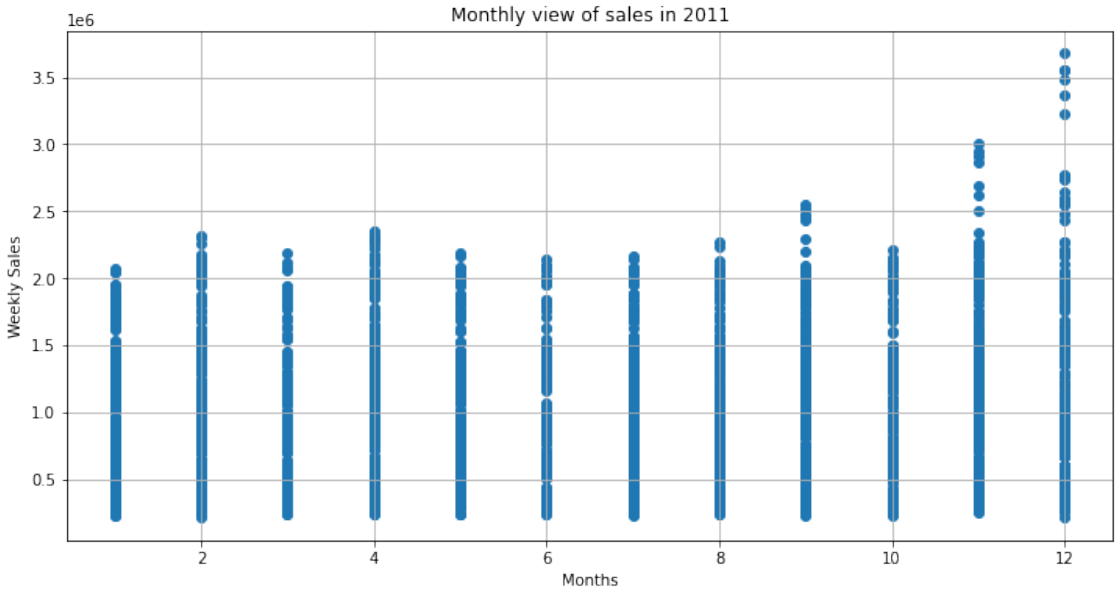
```
…           …              …    …     …
6430   192.013558          8.684  2012      9
6431   192.170412          8.667  2012      5
6432   192.327265          8.667  2012     12
6433   192.330854          8.667  2012     10
6434   192.308899          8.667  2012     10

[6435 rows x 10 columns]
```

[32]:
```
year_2010=df.loc[df["Year"]==2010]
year_2011=df.loc[df["Year"]==2011]
year_2012=df.loc[df["Year"]==2012]
```

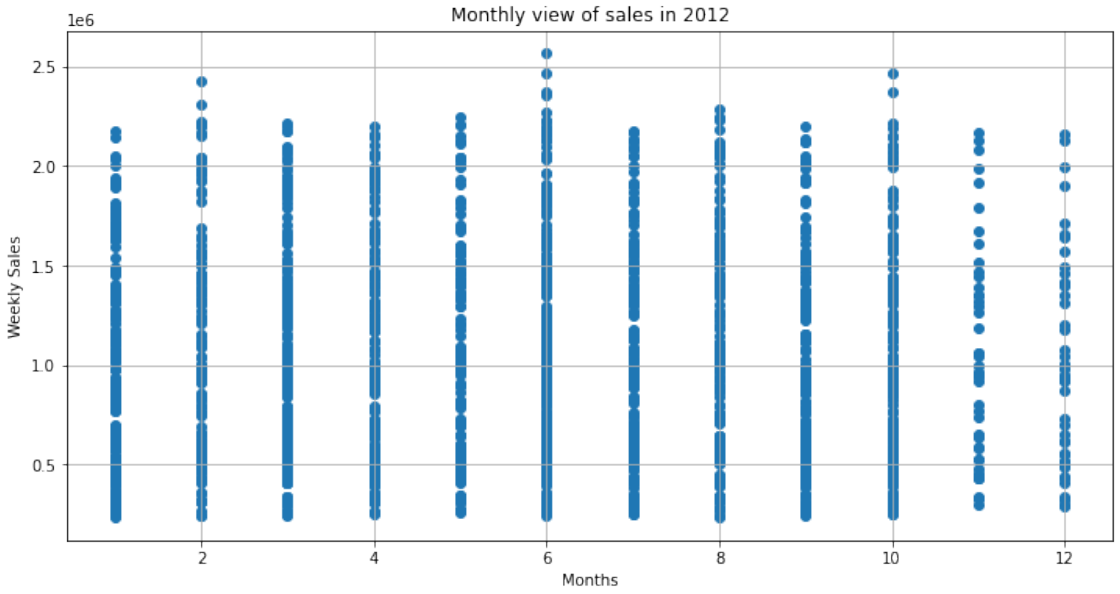
[33]:
```
plt.figure(figsize=(12,6))
plt.scatter(year_2010["Month"],year_2010["Weekly_Sales"])
plt.xlabel("Months")
plt.ylabel("Weekly Sales")
plt.title("Monthly view of sales in 2010")
plt.grid()
```



[34]:
```
plt.figure(figsize=(12,6))
plt.scatter(year_2011["Month"],year_2011["Weekly_Sales"])
plt.xlabel("Months")
plt.ylabel("Weekly Sales")
plt.title("Monthly view of sales in 2011")
plt.grid()
```

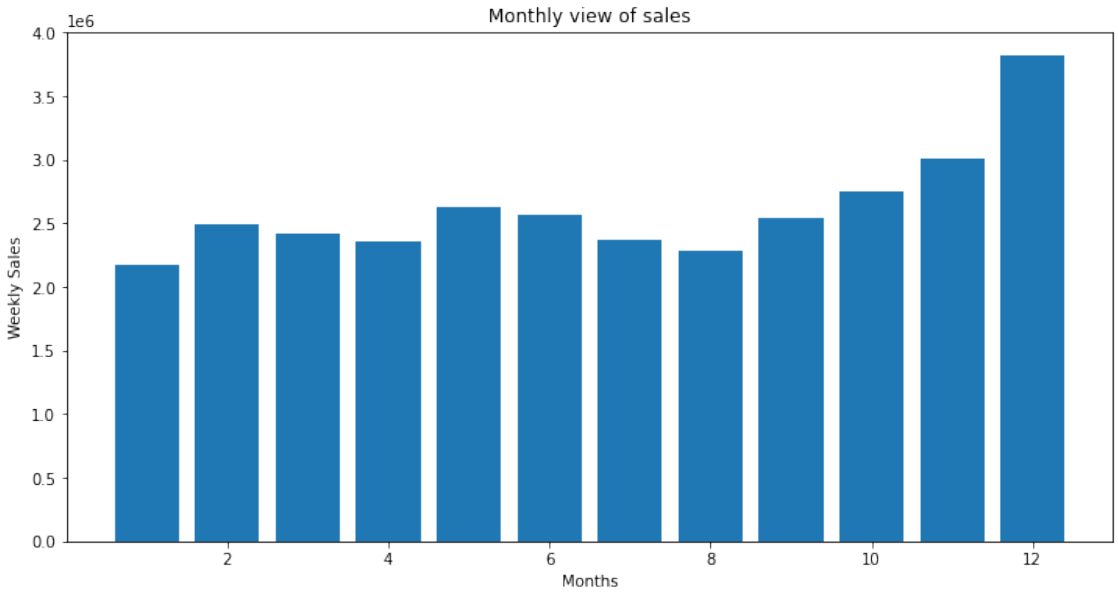


[35]:
```
plt.figure(figsize=(12,6))
plt.scatter(year_2012["Month"],year_2012["Weekly_Sales"])
plt.xlabel("Months")
plt.ylabel("Weekly Sales")
plt.title("Monthly view of sales in 2012")
plt.grid()
```



[36]:
```
plt.figure(figsize=(12,6))
plt.bar(df["Month"],df["Weekly_Sales"])
plt.xlabel("Months")
plt.ylabel("Weekly Sales")
plt.title("Monthly view of sales")
```

[36]:
```
Text(0.5, 1.0, 'Monthly view of sales')
```

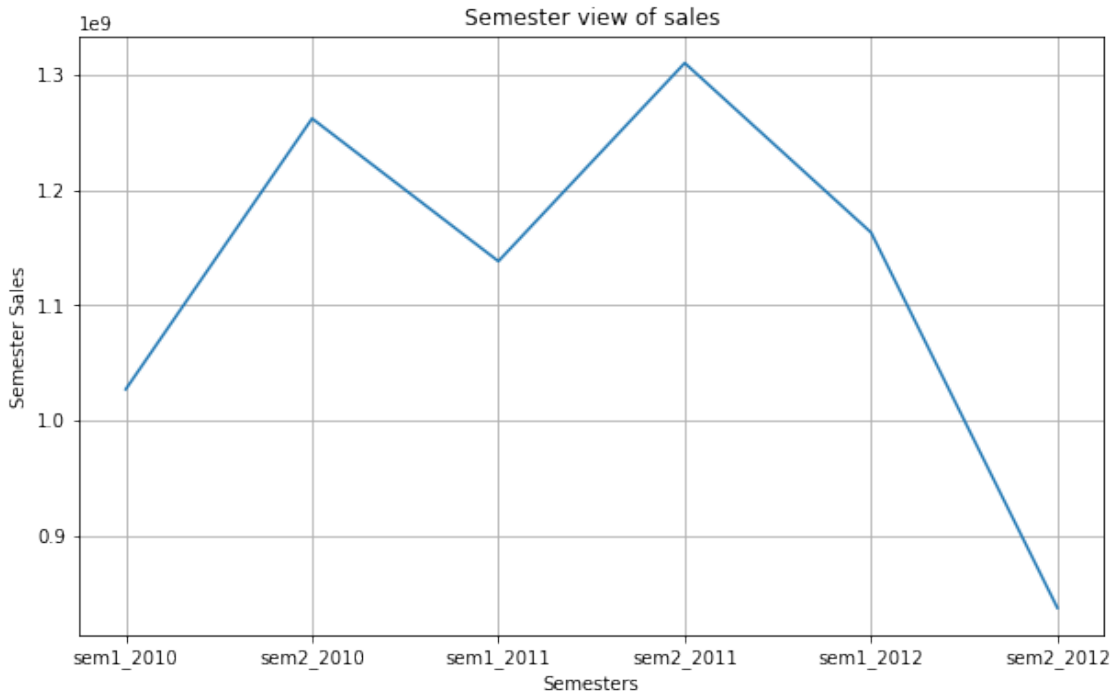


6 *Semester View of Sales*

[37]:
```
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())
```

[38]:
```
semester_names=["sem1_2010","sem2_2010","sem1_2011","sem2_2011","sem1_2012","sem2_2012"]
```

[39]:
```
plt.figure(figsize=(10,6))
plt.plot(semester_names,semester_sales)
plt.xlabel("Semesters")
plt.ylabel("Semester Sales")
plt.title("Semester view of sales")
plt.grid()
```



[40]:
```
df # Checking before Model creation
```

[40]:

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

	CPI	Unemployment	Year	Month
0	211.096358	8.106	2010	5

```
1      211.242170         8.106  2010     12
2      211.289143         8.106  2010      2
3      211.319643         8.106  2010      2
4      211.350143         8.106  2010      5
…           …           …   …     …
6430   192.013558         8.684  2012      9
6431   192.170412         8.667  2012      5
6432   192.327265         8.667  2012     12
6433   192.330854         8.667  2012     10
6434   192.308899         8.667  2012     10

[6435 rows x 10 columns]
```

7 *Prediction models to forecast demand*

8 *Linear Regression Model*

[41]:
```
from sklearn.linear_model import LinearRegression
from sklearn import model_selection
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from math import sqrt
```

[42]:
```
x=df.drop(["Weekly_Sales","Date"],axis=1)
y=df["Weekly_Sales"]
```

[43]:
```
linreg=LinearRegression(n_jobs=-1)
```

[44]:
```
xtrain,xtest,ytrain,ytest=model_selection.train_test_split(x,y,test_size=0.
↪4,random_state=21)
```

[45]:
```
linreg.fit(xtrain,ytrain)
```

[45]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=-1, normalize=False)
```

[46]:
```
print(linreg.intercept_)
print(linreg.coef_)
```

```
102262644.94582875
[-15632.43095327  28090.88125484  -1228.9182576   84114.17043215
  -1955.43396691 -22592.54398505 -50028.32491431   8770.17761978]
```

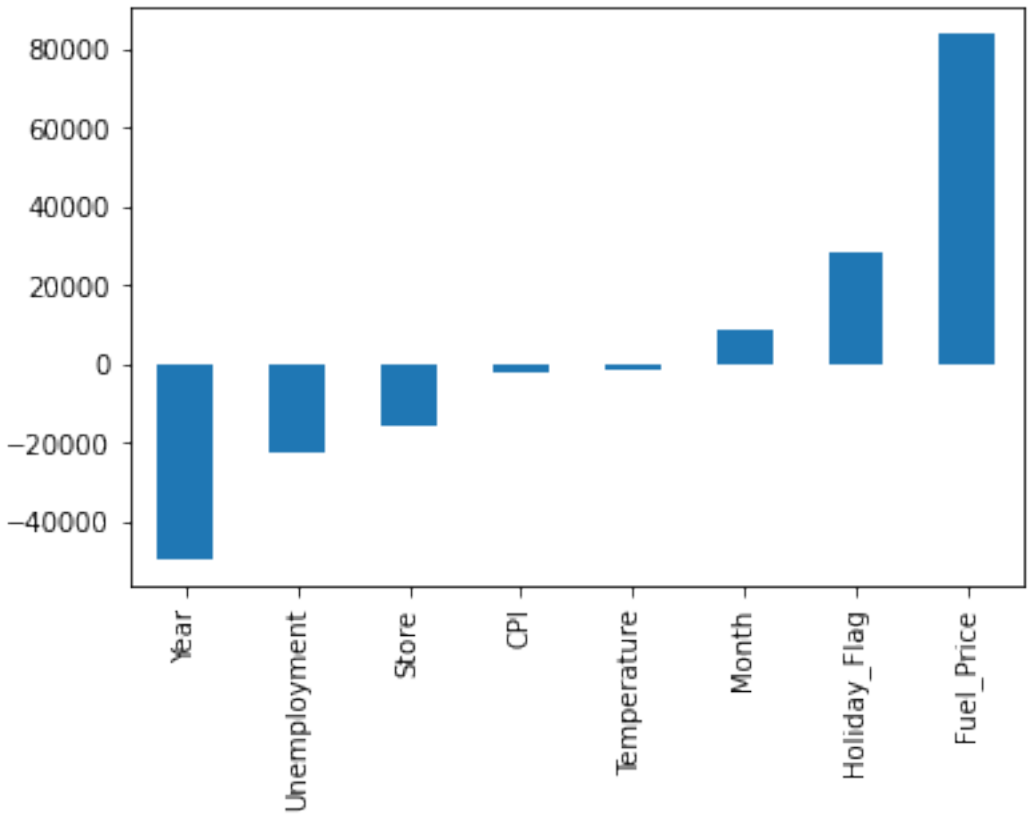
[47]:
```
x.columns
```

[47]:
```
Index(['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI',
       'Unemployment', 'Year', 'Month'],
      dtype='object')
```

[48]:
```
features=['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI',␣
↪'Unemployment', 'Year', 'Month'],
```

[49]:
```
relation=pd.Series(linreg.coef_,x.columns).sort_values()
relation.plot(kind="bar")
```

[49]:
```
<AxesSubplot:>
```



[50]:
```
# 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.
```

[51]:
```
#Hypothesize if CPI, unemployment, and fuel price have any impact on sales
print(format(linreg.score(xtest,ytest)))
```

```
0.14207157531015357
```

[52]:
```
print(sqrt(mean_squared_error(ytrain,linreg.predict(xtrain))))
```

```
515554.3957748044
```

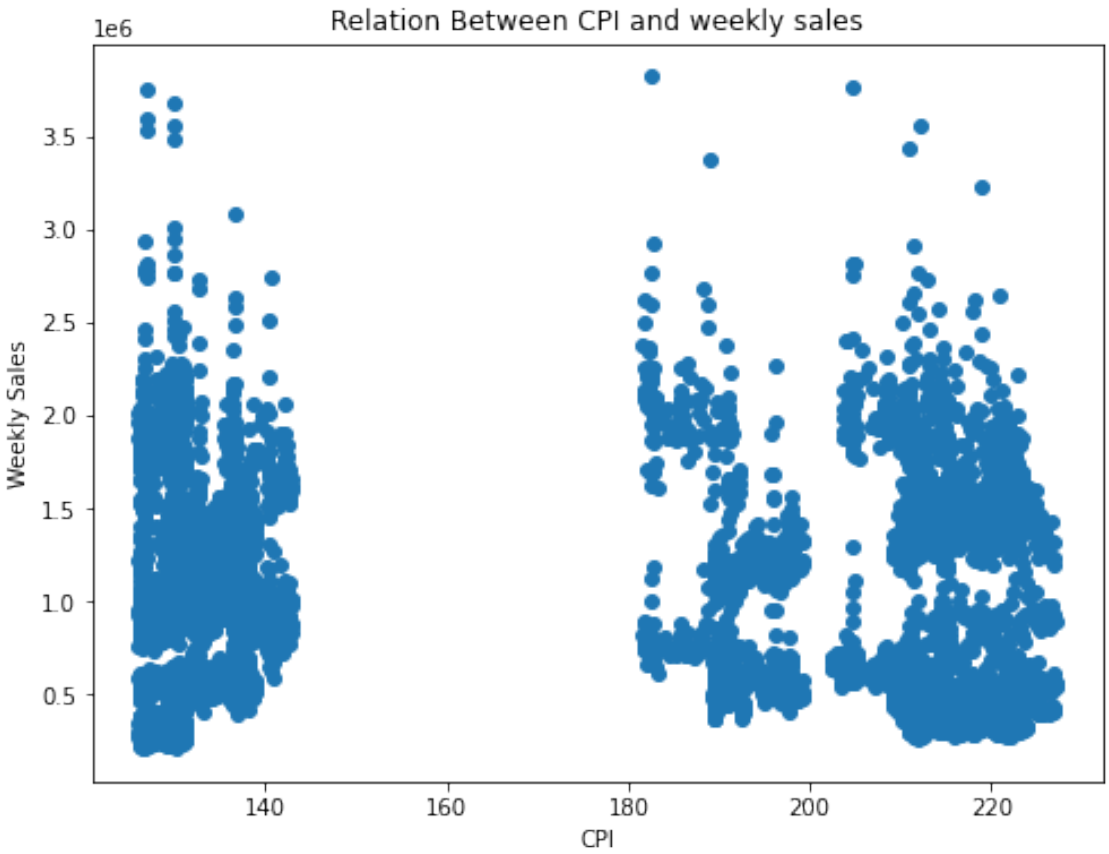
[53]:
```
print(sqrt(mean_squared_error(ytest,linreg.predict(xtest))))
```

```
529886.3465550701
```

[54]:
```
# The test error is less compared to train error. Hence the predicted model is␣
↪well good
```

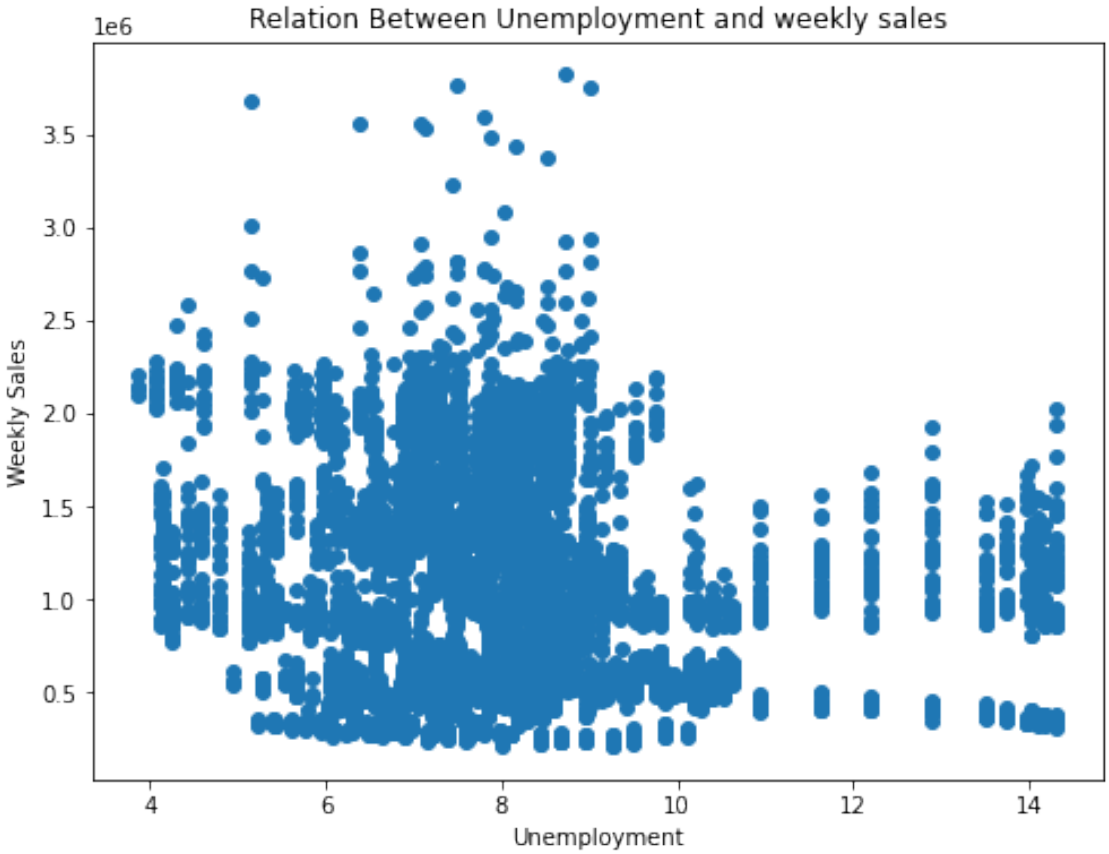
[55]:
```
plt.figure(figsize=(8,6))
plt.scatter(df["CPI"],df["Weekly_Sales"])
plt.title("Relation Between CPI and weekly sales")
plt.xlabel("CPI")
plt.ylabel("Weekly Sales")
```

[55]:
```
Text(0, 0.5, 'Weekly Sales')
```



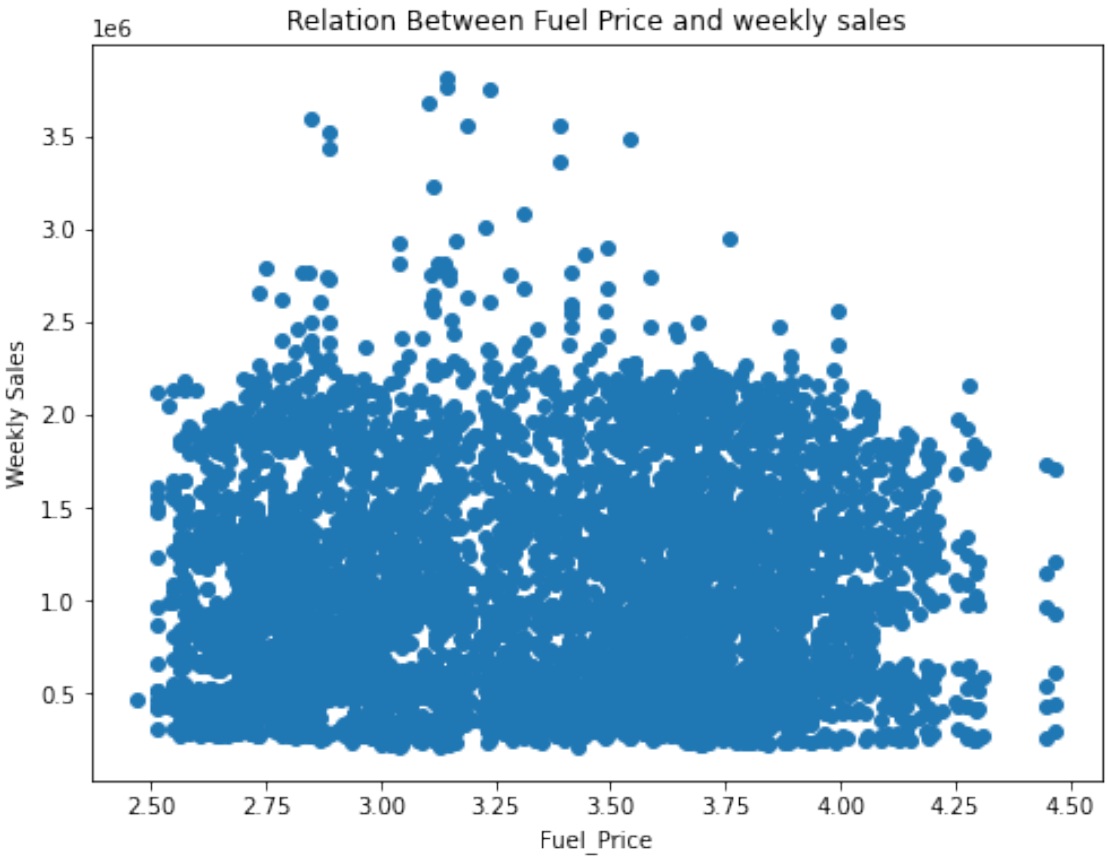
[56]:
```
plt.figure(figsize=(8,6))
plt.scatter(df["Unemployment"],df["Weekly_Sales"])
plt.title("Relation Between Unemployment and weekly sales")
plt.xlabel("Unemployment")
plt.ylabel("Weekly Sales")
```

[56]: Text(0, 0.5, 'Weekly Sales')



[57]:
```
plt.figure(figsize=(8,6))
plt.scatter(df["Fuel_Price"],df["Weekly_Sales"])
plt.title("Relation Between Fuel Price and weekly sales")
plt.xlabel("Fuel_Price")
plt.ylabel("Weekly Sales")
```

[57]: Text(0, 0.5, 'Weekly Sales')



9 *Change dates into days by creating new variable.*

[58]:
```
df1 = df['Date']
```

[59]:
```
df1
```

[59]:
```
0       2010-05-02
1       2010-12-02
2       2010-02-19
3       2010-02-26
4       2010-05-03
           ...
6430    2012-09-28
6431    2012-05-10
6432    2012-12-10
6433    2012-10-19
6434    2012-10-26
Name: Date, Length: 6435, dtype: datetime64[ns]
```

[60]:
```
df['days'] = df1.dt.day_name()
```

[61]:
```
df
```

[61]:

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

	CPI	Unemployment	Year	Month	days
0	211.096358	8.106	2010	5	Sunday
1	211.242170	8.106	2010	12	Thursday
2	211.289143	8.106	2010	2	Friday
3	211.319643	8.106	2010	2	Friday
4	211.350143	8.106	2010	5	Monday
…	…	…	…	…	…
6430	192.013558	8.684	2012	9	Friday
6431	192.170412	8.667	2012	5	Thursday
6432	192.327265	8.667	2012	12	Monday
6433	192.330854	8.667	2012	10	Friday
6434	192.308899	8.667	2012	10	Friday

[6435 rows x 11 columns]

[62]:
```
# Linear Regression Model gives the best accuracy
```