In [1]:
```
import numpy as np
import pandas  as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
data=pd.read_csv("E://SIMPLE LEARN//python//DATASET//Walmart_Store_sales.csv")
```

In [3]:
```
data['Date'] = pd.to_datetime(data['Date'])
data['year_month'] = data['Date'].dt.strftime('%Y-%m')
data['year'] = data['Date'].dt.strftime('%Y')
data['day'] = data['Date'].dt.day


data = data.sort_values(by = 'Date')
data
```

Out[3]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployme
606	5	2010-01-10	283178.12	0	71.10	2.603	212.226946	6.76
2036	15	2010-01-10	566945.95	0	59.69	2.840	132.756800	8.06
5897	42	2010-01-10	481523.93	0	86.01	3.001	126.234600	9.00
4610	33	2010-01-10	224294.39	0	91.45	3.001	126.234600	9.26
5039	36	2010-01-10	422169.47	0	74.66	2.567	210.440443	8.47
...	...	...	...	...	...	...	...	
5860	41	2012-12-10	1409544.97	0	39.38	3.760	199.053937	6.19
2285	16	2012-12-10	491817.19	0	43.26	3.760	199.053937	5.84
1427	10	2012-12-10	1713889.11	0	76.03	4.468	131.108333	6.94
3572	25	2012-12-10	697317.41	0	43.74	4.000	216.115057	7.29
283	2	2012-12-10	1900745.13	0	60.97	3.601	223.015426	6.17

6435 rows × 11 columns

In [5]:
```
data.shape
```

Out[5]:
```
(6435, 11)
```

In [6]:
```
gp=data.groupby(['Store'])['Weekly_Sales'].sum()
gp.sort_values(ascending = False).head()
```

Out[6]:
```
Store
20    3.013978e+08
4     2.995440e+08
14    2.889999e+08
13    2.865177e+08
2     2.753824e+08
Name: Weekly_Sales, dtype: float64
```

In [7]:
```
gt= data.groupby(['Store'])['Weekly_Sales'].mean()
gt.sort_values(ascending = False).head()
```

Out[7]:
```
Store
20    2.107677e+06
4     2.094713e+06
14    2.020978e+06
13    2.003620e+06
2     1.925751e+06
Name: Weekly_Sales, dtype: float64
```

In [8]:
```
gi=data.groupby(['Store'])['Weekly_Sales'].std()
gi.sort_values(ascending = False).head()
```

Out[8]:
```
Store
14    317569.949476
10    302262.062504
20    275900.562742
4     266201.442297
13    265506.995776
Name: Weekly_Sales, dtype: float64
```

In [9]:
```
data.groupby('Store').apply(lambda x: np.std(x) / np.mean(x))
```

Out[9]:

Store	CPI	Fuel_Price	Holiday_Flag	Store	Temperature	Unemployment	Weekly_Sales	
1	0.020073	0.132253	3.646917	0.0	0.207894	0.050248	0.099941	0
2	0.020066	0.132253	3.646917	0.0	0.226317	0.080440	0.122992	0
3	0.020141	0.132253	3.646917	0.0	0.176408	0.062098	0.114619	0
4	0.014391	0.129161	3.646917	0.0	0.258996	0.237445	0.126637	0
5	0.020084	0.132253	3.646917	0.0	0.204228	0.061324	0.118253	0
6	0.020104	0.132253	3.646917	0.0	0.202256	0.079806	0.135347	0
7	0.016485	0.132194	3.646917	0.0	0.415581	0.049845	0.196614	0
8	0.020142	0.132253	3.646917	0.0	0.255442	0.059337	0.116543	0
9	0.020145	0.132253	3.646917	0.0	0.229577	0.075651	0.126451	0
10	0.014391	0.124553	3.646917	0.0	0.193552	0.100530	0.158576	0
11	0.020141	0.132253	3.646917	0.0	0.173291	0.062098	0.121834	0
12	0.014391	0.123767	3.646917	0.0	0.235858	0.091784	0.137442	0
13	0.014391	0.119091	3.646917	0.0	0.326449	0.125455	0.132049	0
14	0.019230	0.129177	3.646917	0.0	0.280574	0.017451	0.156586	0
15	0.017547	0.126467	3.646917	0.0	0.325456	0.024172	0.192707	0
16	0.016485	0.132194	3.646917	0.0	0.369547	0.054398	0.164602	0
17	0.014391	0.119091	3.646917	0.0	0.369791	0.049525	0.125081	0
18	0.017547	0.131796	3.646917	0.0	0.311771	0.049280	0.162275	0
19	0.017547	0.126467	3.646917	0.0	0.316532	0.024172	0.132215	0
20	0.019911	0.129177	3.646917	0.0	0.297304	0.041240	0.130444	0
21	0.020066	0.132253	3.646917	0.0	0.226799	0.080440	0.169696	0
22	0.017484	0.131796	3.646917	0.0	0.284348	0.045486	0.156234	0
23	0.017547	0.131796	3.646917	0.0	0.359139	0.113298	0.179092	0
24	0.017547	0.126467	3.646917	0.0	0.302405	0.034074	0.123204	0
25	0.019911	0.129177	3.646917	0.0	0.318678	0.041240	0.159300	0
26	0.017547	0.131796	3.646917	0.0	0.382351	0.051949	0.109725	0
27	0.017484	0.126467	3.646917	0.0	0.278553	0.020508	0.134682	0
28	0.014391	0.123767	3.646917	0.0	0.235858	0.091784	0.136849	0
29	0.017547	0.131796	3.646917	0.0	0.284348	0.052334	0.183099	0
30	0.020066	0.132253	3.646917	0.0	0.226799	0.080440	0.051826	0
31	0.020066	0.132253	3.646917	0.0	0.226799	0.080440	0.089845	0
32	0.016485	0.132194	3.646917	0.0	0.318616	0.049845	0.117896	0
33	0.014391	0.124553	3.646917	0.0	0.199710	0.112550	0.092543	0

Store	CPI	Fuel_Price	Holiday_Flag	Store	Temperature	Unemployment	Weekly_Sales	
34	0.014391	0.129161	3.646917	0.0	0.273658	0.048145	0.107846	0
35	0.017484	0.129177	3.646917	0.0	0.278553	0.020946	0.228877	0
36	0.020026	0.139120	3.646917	0.0	0.169360	0.088985	0.162009	0
37	0.020026	0.132253	3.646917	0.0	0.169360	0.088985	0.041937	0
38	0.014391	0.123767	3.646917	0.0	0.235858	0.091784	0.110487	0
39	0.020026	0.132253	3.646917	0.0	0.174963	0.088985	0.149383	0
40	0.017547	0.131796	3.646917	0.0	0.369121	0.113298	0.122997	0
41	0.016485	0.132194	3.646917	0.0	0.334744	0.056656	0.147658	0
42	0.014391	0.124553	3.646917	0.0	0.193552	0.100530	0.090019	0
43	0.019759	0.132253	3.646917	0.0	0.213030	0.048145	0.063879	0
44	0.014391	0.119091	3.646917	0.0	0.326449	0.141753	0.081507	0
45	0.019230	0.129177	3.646917	0.0	0.280574	0.017451	0.165033	0

In [10]:
```
cvr=data.groupby(['Store'])['Weekly_Sales'].apply(lambda x: np.std(x) / np.mean(x
cvr
```

Out[10]:
```
Store
1     0.099941
2     0.122992
3     0.114619
4     0.126637
5     0.118253
6     0.135347
7     0.196614
8     0.116543
9     0.126451
10    0.158576
11    0.121834
12    0.137442
13    0.132049
14    0.156586
15    0.192707
16    0.164602
17    0.125081
18    0.162275
19    0.132215
20    0.130444
21    0.169696
22    0.156234
23    0.179092
24    0.123204
25    0.159300
26    0.109725
27    0.134682
28    0.136849
29    0.183099
30    0.051826
31    0.089845
32    0.117896
33    0.092543
34    0.107846
35    0.228877
36    0.162009
37    0.041937
38    0.110487
39    0.149383
40    0.122997
41    0.147658
42    0.090019
43    0.063879
44    0.081507
45    0.165033
Name: Weekly_Sales, dtype: float64
```

In [164]:
```
cr=data.groupby(['Store'])['Weekly_Sales'].apply(lambda x: np.mean(x) / np.std(x)
cr
```

Out[164]:
```
Store
1     10.005920
2      8.130639
3      8.724593
4      7.896560
5      8.456460
6      7.388410
7      5.086118
8      8.580510
9      7.908202
10     6.306120
11     8.207917
12     7.275785
13     7.572919
14     6.386253
15     5.189235
16     6.075257
17     7.994818
18     6.162375
19     7.563416
20     7.666114
21     5.892894
22     6.400667
23     5.583733
24     8.116599
25     6.277445
26     9.113695
27     7.424895
28     7.307339
29     5.461530
30    19.295382
31    11.130249
32     8.482045
33    10.805780
34     9.272467
35     4.369166
36     6.172479
37    23.845455
38     9.050831
39     6.694215
40     8.130250
41     6.772402
42    11.108776
43    15.654573
44    12.268912
45     6.059407
Name: Weekly_Sales, dtype: float64
```

In [11]:
```
data['quarter'] = pd.PeriodIndex(data['Date'] ,freq='Q')
data['quarter']
```

Out[11]:
```
606     2010Q1
2036    2010Q1
5897    2010Q1
4610    2010Q1
5039    2010Q1
         ...
5860    2012Q4
2285    2012Q4
1427    2012Q4
3572    2012Q4
283     2012Q4
Name: quarter, Length: 6435, dtype: period[Q-DEC]
```

In [63]:
```
Q3=data.loc[data['quarter'] == '2012Q3']
Q3
Q3_store=Q3.groupby(['Store','quarter'])['Weekly_Sales'].sum()
Q3_store=Q3_store.reset_index(name='Q3_sales')
Q3_store
```

Out[63]:

	Store	quarter	Q3_sales
0	1	2012Q3	18633209.98
1	2	2012Q3	22396867.61
2	3	2012Q3	4966495.93
3	4	2012Q3	25652119.35
4	5	2012Q3	3880621.88
5	6	2012Q3	18341221.11
6	7	2012Q3	7322393.92
7	8	2012Q3	10873860.34
8	9	2012Q3	6528239.56
9	10	2012Q3	21169356.45
10	11	2012Q3	16094363.07
11	12	2012Q3	11777508.50
12	13	2012Q3	24319994.35
13	14	2012Q3	20140430.40
14	15	2012Q3	6909374.37
15	16	2012Q3	6441311.11
16	17	2012Q3	11533998.38
17	18	2012Q3	12507521.72
18	19	2012Q3	16644341.31
19	20	2012Q3	24665938.11
20	21	2012Q3	8403507.99
21	22	2012Q3	11818544.33
22	23	2012Q3	17103654.36
23	24	2012Q3	16125999.86
24	25	2012Q3	8309440.44
25	26	2012Q3	12417575.35
26	27	2012Q3	20191238.11
27	28	2012Q3	15055659.67
28	29	2012Q3	6127862.07
29	30	2012Q3	5181974.44
30	31	2012Q3	16454328.46
31	32	2012Q3	14142164.84

	Store	quarter	Q3_sales
32	33	2012Q3	3177072.43
33	34	2012Q3	11476258.98
34	35	2012Q3	10252122.68
35	36	2012Q3	3578123.58
36	37	2012Q3	6250524.08
37	38	2012Q3	5129297.64
38	39	2012Q3	18899955.17
39	40	2012Q3	11647661.37
40	41	2012Q3	16373588.44
41	42	2012Q3	6830839.86
42	43	2012Q3	7376726.03
43	44	2012Q3	4020486.01
44	45	2012Q3	8851242.32

In [62]:
```
Q2=data.loc[data['quarter'] == '2012Q2']
Q2
Q2_store=Q2.groupby(['Store','quarter'])['Weekly_Sales'].sum()
Q2_store=Q2_store.reset_index(name='Q2_sales')
Q2_store
```

Out[62]:

	Store	quarter	Q2_sales
0	1	2012Q2	21036965.58
1	2	2012Q2	25085123.61
2	3	2012Q2	5562668.16
3	4	2012Q2	28384185.16
4	5	2012Q2	4427262.21
5	6	2012Q2	20728970.16
6	7	2012Q2	7613593.92
7	8	2012Q2	11934275.61
8	9	2012Q2	7431320.13
9	10	2012Q2	23598433.93
10	11	2012Q2	17879095.77
11	12	2012Q2	13193365.04
12	13	2012Q2	26803225.55
13	14	2012Q2	24427769.06
14	15	2012Q2	7867952.23
15	16	2012Q2	6626133.44
16	17	2012Q2	12918892.02
17	18	2012Q2	13834706.08
18	19	2012Q2	18315278.56
19	20	2012Q2	27550180.62
20	21	2012Q2	9226279.62
21	22	2012Q2	13329065.39
22	23	2012Q2	18283424.90
23	24	2012Q2	17768191.98
24	25	2012Q2	9247467.19
25	26	2012Q2	13218289.66
26	27	2012Q2	22593640.73
27	28	2012Q2	16985999.95
28	29	2012Q2	7034493.19
29	30	2012Q2	5786335.45
30	31	2012Q2	18249155.35
31	32	2012Q2	15415236.21

	Store	quarter	Q2_sales
32	33	2012Q2	3512138.05
33	34	2012Q2	12858027.98
34	35	2012Q2	10753570.97
35	36	2012Q2	4090378.90
36	37	2012Q2	6859777.96
37	38	2012Q2	5732362.70
38	39	2012Q2	20191585.63
39	40	2012Q2	12849747.45
40	41	2012Q2	17560035.88
41	42	2012Q2	7608247.31
42	43	2012Q2	8239792.67
43	44	2012Q2	4322555.33
44	45	2012Q2	10278900.05

In [64]:
```
result = pd.merge(Q2_store, Q3_store, on='Store')
result
result['Q3-Q2']=(result['Q3_sales']-result['Q2_sales'])#/result['new'])*100
result['Q3-Q2']
result
```

Out[64]:

	Store	quarter_x	Q2_sales	quarter_y	Q3_sales	Q3-Q2
0	1	2012Q2	21036965.58	2012Q3	18633209.98	-2403755.60
1	2	2012Q2	25085123.61	2012Q3	22396867.61	-2688256.00
2	3	2012Q2	5562668.16	2012Q3	4966495.93	-596172.23
3	4	2012Q2	28384185.16	2012Q3	25652119.35	-2732065.81
4	5	2012Q2	4427262.21	2012Q3	3880621.88	-546640.33
5	6	2012Q2	20728970.16	2012Q3	18341221.11	-2387749.05
6	7	2012Q2	7613593.92	2012Q3	7322393.92	-291200.00
7	8	2012Q2	11934275.61	2012Q3	10873860.34	-1060415.27
8	9	2012Q2	7431320.13	2012Q3	6528239.56	-903080.57
9	10	2012Q2	23598433.93	2012Q3	21169356.45	-2429077.48
10	11	2012Q2	17879095.77	2012Q3	16094363.07	-1784732.70
11	12	2012Q2	13193365.04	2012Q3	11777508.50	-1415856.54
12	13	2012Q2	26803225.55	2012Q3	24319994.35	-2483231.20
13	14	2012Q2	24427769.06	2012Q3	20140430.40	-4287338.66
14	15	2012Q2	7867952.23	2012Q3	6909374.37	-958577.86
15	16	2012Q2	6626133.44	2012Q3	6441311.11	-184822.33
16	17	2012Q2	12918892.02	2012Q3	11533998.38	-1384893.64
17	18	2012Q2	13834706.08	2012Q3	12507521.72	-1327184.36
18	19	2012Q2	18315278.56	2012Q3	16644341.31	-1670937.25
19	20	2012Q2	27550180.62	2012Q3	24665938.11	-2884242.51
20	21	2012Q2	9226279.62	2012Q3	8403507.99	-822771.63
21	22	2012Q2	13329065.39	2012Q3	11818544.33	-1510521.06
22	23	2012Q2	18283424.90	2012Q3	17103654.36	-1179770.54
23	24	2012Q2	17768191.98	2012Q3	16125999.86	-1642192.12
24	25	2012Q2	9247467.19	2012Q3	8309440.44	-938026.75
25	26	2012Q2	13218289.66	2012Q3	12417575.35	-800714.31
26	27	2012Q2	22593640.73	2012Q3	20191238.11	-2402402.62
27	28	2012Q2	16985999.95	2012Q3	15055659.67	-1930340.28
28	29	2012Q2	7034493.19	2012Q3	6127862.07	-906631.12
29	30	2012Q2	5786335.45	2012Q3	5181974.44	-604361.01
30	31	2012Q2	18249155.35	2012Q3	16454328.46	-1794826.89
31	32	2012Q2	15415236.21	2012Q3	14142164.84	-1273071.37

	Store	quarter_x	Q2_sales	quarter_y	Q3_sales	Q3-Q2
32	33	2012Q2	3512138.05	2012Q3	3177072.43	-335065.62
33	34	2012Q2	12858027.98	2012Q3	11476258.98	-1381769.00
34	35	2012Q2	10753570.97	2012Q3	10252122.68	-501448.29
35	36	2012Q2	4090378.90	2012Q3	3578123.58	-512255.32
36	37	2012Q2	6859777.96	2012Q3	6250524.08	-609253.88
37	38	2012Q2	5732362.70	2012Q3	5129297.64	-603065.06
38	39	2012Q2	20191585.63	2012Q3	18899955.17	-1291630.46
39	40	2012Q2	12849747.45	2012Q3	11647661.37	-1202086.08
40	41	2012Q2	17560035.88	2012Q3	16373588.44	-1186447.44
41	42	2012Q2	7608247.31	2012Q3	6830839.86	-777407.45
42	43	2012Q2	8239792.67	2012Q3	7376726.03	-863066.64
43	44	2012Q2	4322555.33	2012Q3	4020486.01	-302069.32
44	45	2012Q2	10278900.05	2012Q3	8851242.32	-1427657.73

In [19]:
```
e=data.groupby(['Store'])['Weekly_Sales'].sum()
e.sort_values(ascending = False).head()
e
```

Out[19]:
```
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 [20]:
```
dum=pd.get_dummies(data, columns=['Holiday_Flag'])
dum
```

Out[20]:

	Store	Date	Weekly_Sales	Temperature	Fuel_Price	CPI	Unemployment	year_month
606	5	2010-01-10	283178.12	71.10	2.603	212.226946	6.768	2010-0
2036	15	2010-01-10	566945.95	59.69	2.840	132.756800	8.067	2010-0
5897	42	2010-01-10	481523.93	86.01	3.001	126.234600	9.003	2010-0
4610	33	2010-01-10	224294.39	91.45	3.001	126.234600	9.265	2010-0
5039	36	2010-01-10	422169.47	74.66	2.567	210.440443	8.476	2010-0
...	...	...	...	...	...	...	...	..
5860	41	2012-12-10	1409544.97	39.38	3.760	199.053937	6.195	2012-1
2285	16	2012-12-10	491817.19	43.26	3.760	199.053937	5.847	2012-1
1427	10	2012-12-10	1713889.11	76.03	4.468	131.108333	6.943	2012-1
3572	25	2012-12-10	697317.41	43.74	4.000	216.115057	7.293	2012-1
283	2	2012-12-10	1900745.13	60.97	3.601	223.015426	6.170	2012-1

6435 rows × 13 columns

In [93]:
```
ed=dum.groupby(['Store','Holiday_Flag_0'])['Weekly_Sales'].mean()
ed.reset_index(name='new')
```

Out[93]:

	Store	Holiday_Flag_0	new
0	1	0	1.665748e+06
1	1	1	1.546957e+06
2	2	0	2.079267e+06
3	2	1	1.914209e+06
4	3	0	4.378110e+05
...	...	...	...
85	43	1	6.331276e+05
86	44	0	2.960356e+05
87	44	1	3.032536e+05
88	45	0	8.362937e+05
89	45	1	7.821985e+05

90 rows × 3 columns

In [24]:
```
dum.groupby(['Store','Holiday_Flag_1'])['Weekly_Sales'].mean()
```

Out[24]:
```
Store  Holiday_Flag_1
1      0                1.546957e+06
       1                1.665748e+06
2      0                1.914209e+06
       1                2.079267e+06
3      0                4.000648e+05
                            ...
43     1                6.359463e+05
44     0                3.032536e+05
       1                2.960356e+05
45     0                7.821985e+05
       1                8.362937e+05
Name: Weekly_Sales, Length: 90, dtype: float64
```

In [100]:
```
holiday=data.loc[data['Holiday_Flag'] == 1]
holiday=holiday.groupby(['Store','Holiday_Flag','Date'])['Weekly_Sales'].mean().r
holiday
holiday
```

Out[100]:

	Store	Holiday_Flag	Date	sales on HOLIDAY
0	1	1	2010-10-09	1507460.69
1	1	1	2010-11-26	1955624.11
2	1	1	2010-12-02	1641957.44
3	1	1	2010-12-31	1367320.01
4	1	1	2011-09-09	1540471.24
...	...	...	...	...
445	45	1	2011-11-02	766456.00
446	45	1	2011-11-25	1170672.94
447	45	1	2011-12-30	869403.63
448	45	1	2012-07-09	766512.66
449	45	1	2012-10-02	803657.12

450 rows × 4 columns

In [86]:
```
holiday=data.loc[data['Holiday_Flag'] == 1]
holiday
holiday.groupby(['Store','Holiday_Flag','Date'])['Weekly_Sales'].mean()
```

Out[86]:
```
Store  Holiday_Flag  Date
1      1             2010-10-09    1507460.69
                     2010-11-26    1955624.11
                     2010-12-02    1641957.44
                     2010-12-31    1367320.01
                     2011-09-09    1540471.24
                                       ...
45     1             2011-11-02     766456.00
                     2011-11-25    1170672.94
                     2011-12-30     869403.63
                     2012-07-09     766512.66
                     2012-10-02     803657.12
Name: Weekly_Sales, Length: 450, dtype: float64
```

In [101]:
```
non_holiday=data.loc[data['Holiday_Flag'] == 0]
non_holiday=non_holiday.groupby(['Store','Holiday_Flag','Date'])['Weekly_Sales'].
non_holiday
```

Out[101]:

	Store	Holiday_Flag	Date	sales on NON HOLIDAY
0	1	0	2010-01-10	1453329.50
1	1	0	2010-02-04	1594968.28
2	1	0	2010-02-07	1492418.14
3	1	0	2010-02-19	1611968.17
4	1	0	2010-02-26	1409727.59
...	...	...	...	...
5980	45	0	2012-10-08	733037.32
5981	45	0	2012-10-19	718125.53
5982	45	0	2012-10-26	760281.43
5983	45	0	2012-11-05	770487.37
5984	45	0	2012-12-10	734464.36

5985 rows × 4 columns

In [109]:
```
rt =  pd.concat([non_holiday, holiday], axis=1)

rt
```

Out[109]:

	Store	Holiday_Flag	Date	sales on NON HOLIDAY	Store	Holiday_Flag	Date	sales on HOLIDAY
0	1	0	2010-01-10	1453329.50	1.0	1.0	2010-10-09	1507460.69
1	1	0	2010-02-04	1594968.28	1.0	1.0	2010-11-26	1955624.11
2	1	0	2010-02-07	1492418.14	1.0	1.0	2010-12-02	1641957.44
3	1	0	2010-02-19	1611968.17	1.0	1.0	2010-12-31	1367320.01
4	1	0	2010-02-26	1409727.59	1.0	1.0	2011-09-09	1540471.24
...	...	...	...	...	...	...	...	...
5980	45	0	2012-10-08	733037.32	NaN	NaN	NaT	NaN
5981	45	0	2012-10-19	718125.53	NaN	NaN	NaT	NaN
5982	45	0	2012-10-26	760281.43	NaN	NaN	NaT	NaN
5983	45	0	2012-11-05	770487.37	NaN	NaN	NaT	NaN
5984	45	0	2012-12-10	734464.36	NaN	NaN	NaT	NaN

5985 rows × 8 columns

In [139]:
```
rst = pd.merge(non_holiday, holiday, on=['Store'])
rst
```

Out[139]:

	Store	Holiday_Flag_x	Date_x	sales on NON HOLIDAY	Holiday_Flag_y	Date_y	sales on HOLIDAY
0	1	0	2010-01-10	1453329.50	1	2010-10-09	1507460.69
1	1	0	2010-01-10	1453329.50	1	2010-11-26	1955624.11
2	1	0	2010-01-10	1453329.50	1	2010-12-02	1641957.44
3	1	0	2010-01-10	1453329.50	1	2010-12-31	1367320.01
4	1	0	2010-01-10	1453329.50	1	2011-09-09	1540471.24
...	...	...	...	...	...	...	...
59845	45	0	2012-12-10	734464.36	1	2011-11-02	766456.00
59846	45	0	2012-12-10	734464.36	1	2011-11-25	1170672.94
59847	45	0	2012-12-10	734464.36	1	2011-12-30	869403.63
59848	45	0	2012-12-10	734464.36	1	2012-07-09	766512.66
59849	45	0	2012-12-10	734464.36	1	2012-10-02	803657.12

59850 rows × 7 columns

In [105]:
```
rst.columns
```

Out[105]:
```
Index(['Store', 'Holiday_Flag_x', 'Date_x', 'sales on NON HOLIDAY',
       'Holiday_Flag_y', 'Date_y', 'sales on HOLIDAY'],
      dtype='object')
```

In [122]:
```
for index, row in rst.iterrows():
    if row["sales on HOLIDAY"] > row["sales on NON HOLIDAY"]:
        print(row['Store'] , row['Date_y'] )
```

```
1 2010-10-09 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-09-09 00:00:00
1 2011-11-02 00:00:00
1 2011-11-25 00:00:00
1 2011-12-30 00:00:00
1 2012-07-09 00:00:00
1 2012-10-02 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-11-02 00:00:00
1 2011-11-25 00:00:00
1 2012-07-09 00:00:00
1 2012-10-02 00:00:00
1 2010-10-09 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-09-09 00:00:00
```

In [152]:
```
for index, row in rst.iterrows():
    if row["sales on HOLIDAY"] > row["sales on NON HOLIDAY"]:
        row["sales on NON HOLIDAY"]+1
        print(row['Store'] , row['Date_y'])
```

```
1 2010-10-09 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-09-09 00:00:00
1 2011-11-02 00:00:00
1 2011-11-25 00:00:00
1 2011-12-30 00:00:00
1 2012-07-09 00:00:00
1 2012-10-02 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-11-02 00:00:00
1 2011-11-25 00:00:00
1 2012-07-09 00:00:00
1 2012-10-02 00:00:00
1 2010-10-09 00:00:00
1 2010-11-26 00:00:00
1 2010-12-02 00:00:00
1 2011-09-09 00:00:00
```

In [25]:
```
table = pd.pivot_table(data, values ='Weekly_Sales', index =['Store'],columns ['
table
```

Out[25]:

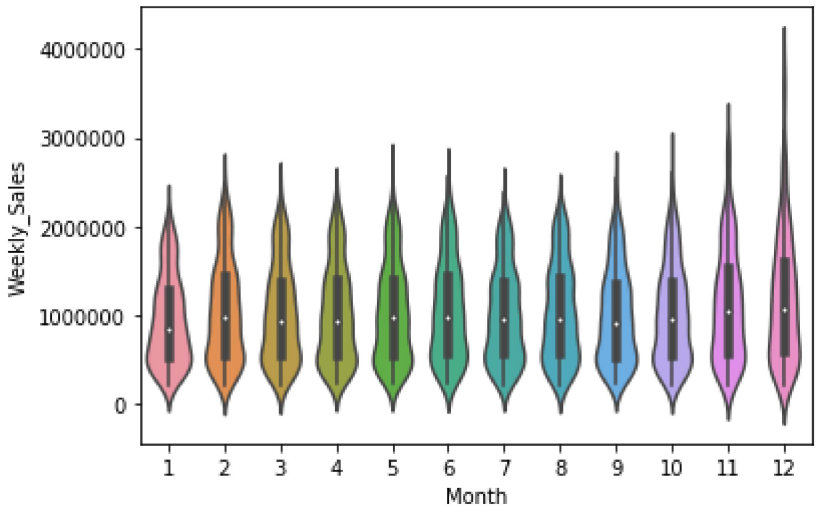
Holiday_Flag	0	1
Store		
1	1.546957e+06	1665747.656
2	1.914209e+06	2079266.900
3	4.000648e+05	437811.050
4	2.083556e+06	2243102.624
5	3.148923e+05	359501.607
6	1.555993e+06	1680907.927
7	5.629645e+05	672400.265
8	9.037434e+05	975330.860
9	5.405993e+05	588950.821
10	1.883309e+06	2113755.949
11	1.349465e+06	1448394.485
12	9.992919e+05	1138140.420
13	1.995393e+06	2113043.806
14	2.013489e+06	2120582.998
15	6.170648e+05	706406.018
16	5.156774e+05	566733.646
17	8.870990e+05	979796.971
18	1.078350e+06	1169422.161
19	1.435071e+06	1577046.734
20	2.097048e+06	2249035.081
21	7.507742e+05	826491.309
22	1.024262e+06	1084874.656
23	1.384400e+06	1462542.294
24	1.347857e+06	1475098.251
25	7.042437e+05	739676.842
26	9.977137e+05	1072046.849
27	1.766413e+06	1892299.278
28	1.311889e+06	1478244.605
29	5.343758e+05	606957.889
30	4.387090e+05	436859.307
31	1.388073e+06	1500026.030
32	1.163770e+06	1203784.083
33	2.596562e+05	262594.519

Holiday_Flag	0	1
Store		
34	9.611277e+05	1041978.089
35	9.080992e+05	1074348.457
36	3.739534e+05	367640.630
37	5.197556e+05	507525.050
38	3.860491e+05	381509.878
39	1.443115e+06	1551127.480
40	9.608268e+05	1008034.075
41	1.263101e+06	1334947.856
42	5.555550e+05	567694.158
43	6.331276e+05	635946.278
44	3.032536e+05	296035.601
45	7.821985e+05	836293.713

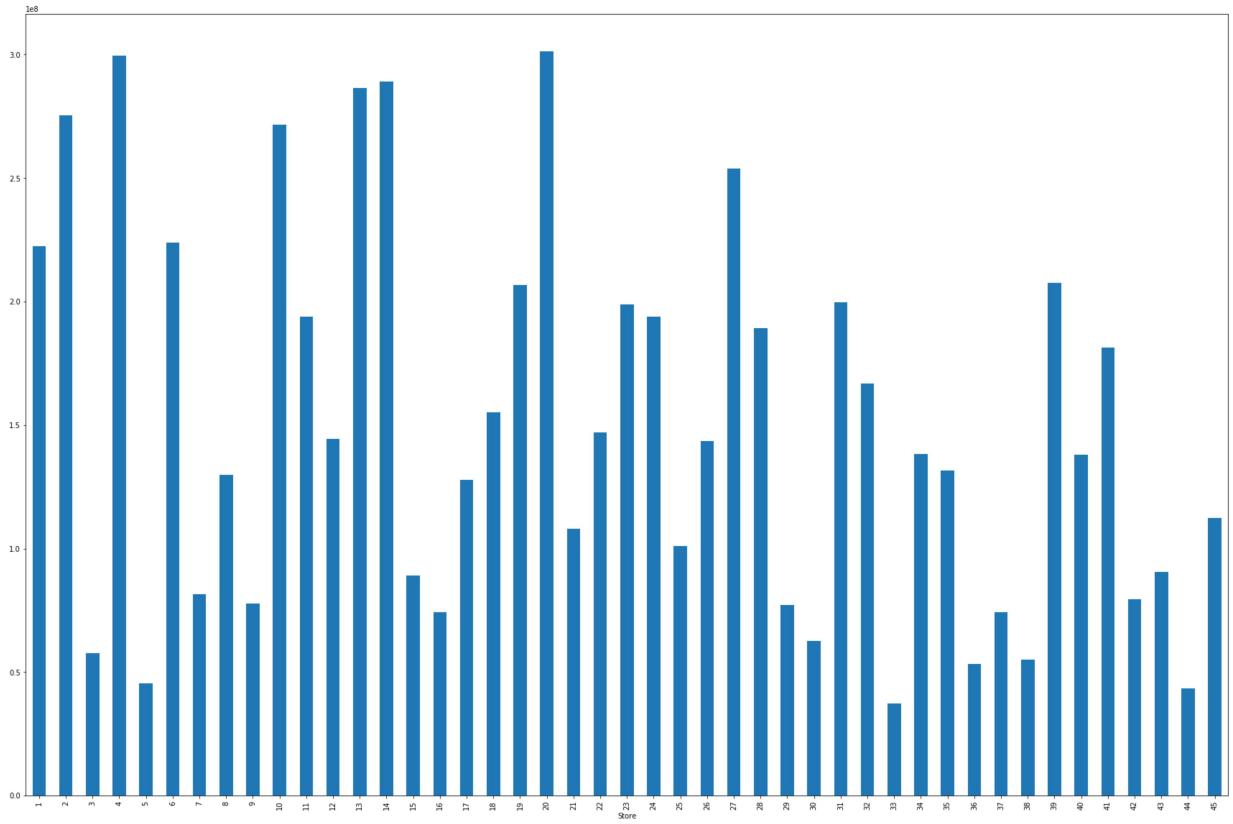
In [26]:
```
wrd=table[1]-table[0]
wrd.sort_values(ascending = False)
```

Out[26]:
```
Store
10    230446.517421
28    166355.623571
35    166249.302789
2     165058.088120
4     159546.780842
20    151986.647541
19    141976.096707
12    138848.495564
24    127240.816639
27    125886.314842
6     124915.059556
1     118790.270361
13    117650.830511
31    111953.509699
7     109435.810188
39    108012.685489
14    107093.664466
11     98929.507932
17     92697.955060
18     91072.442203
15     89341.177850
34     80850.403962
23     78142.332271
21     75717.129827
26     74333.124940
29     72582.133962
41     71846.680586
8      71587.457519
22     60612.234872
45     54095.184504
16     51056.285549
9      48351.491827
40     47207.240113
5      44609.330083
32     40014.118113
3      37746.203609
25     35433.152150
42     12139.189203
33      2938.302684
43      2818.664241
30     -1849.655782
38     -4539.202000
36     -6312.818647
44     -7218.021782
37    -12230.512556
dtype: float64
```

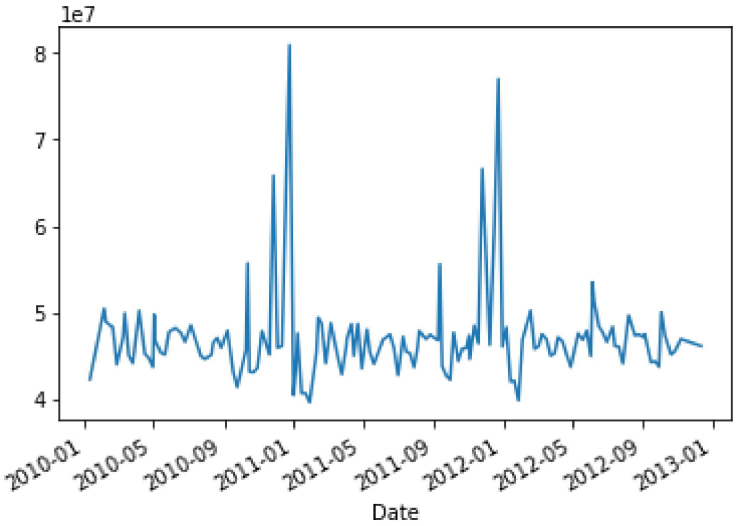
In [127]:
```
data['Month']=data['Date'].dt.month

sns.violinplot(x="Month", y="Weekly_Sales",data=data)

plt.savefig('The distribution of sales depending months.pdf')
```



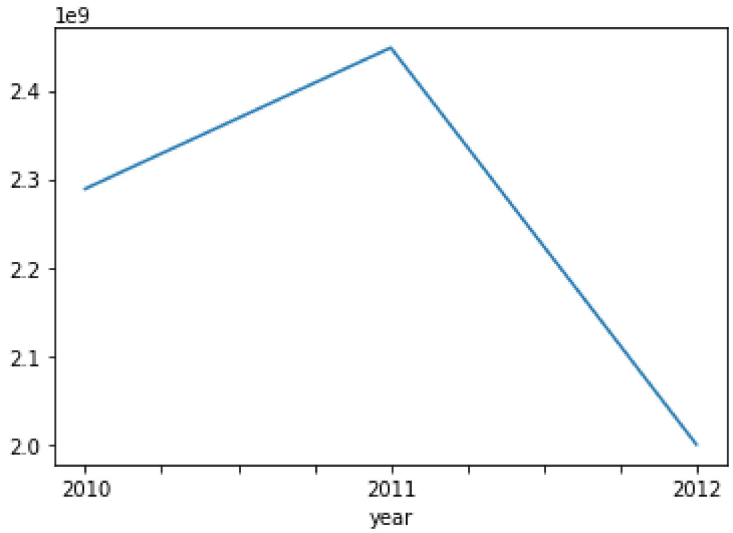
In [128]:
```
tt=data.groupby(['Store'])['Weekly_Sales'].sum()
plt.figure(figsize=(30, 20))
tt.plot(kind='bar')
plt.savefig("The distribution of sales per sales.pdf")
```



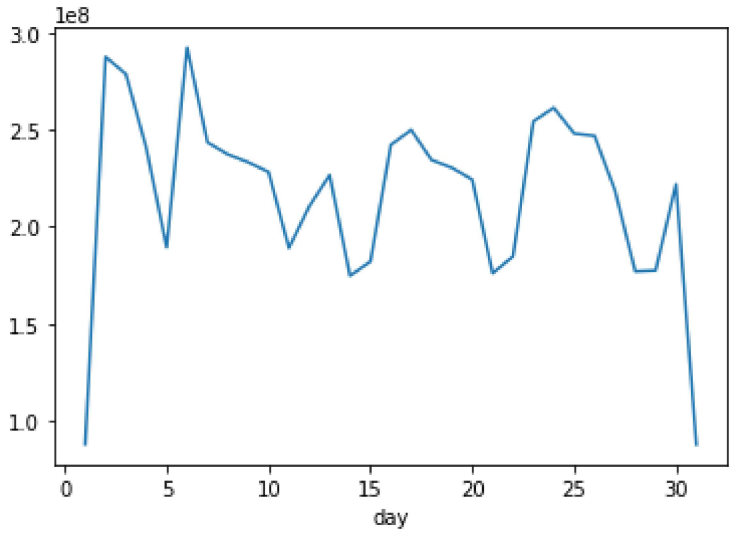
In [129]:
```
ts=data.groupby(['Date'])['Weekly_Sales'].sum()
ts.plot()
plt.savefig("The distribution of sales semester wise.pdf")
```



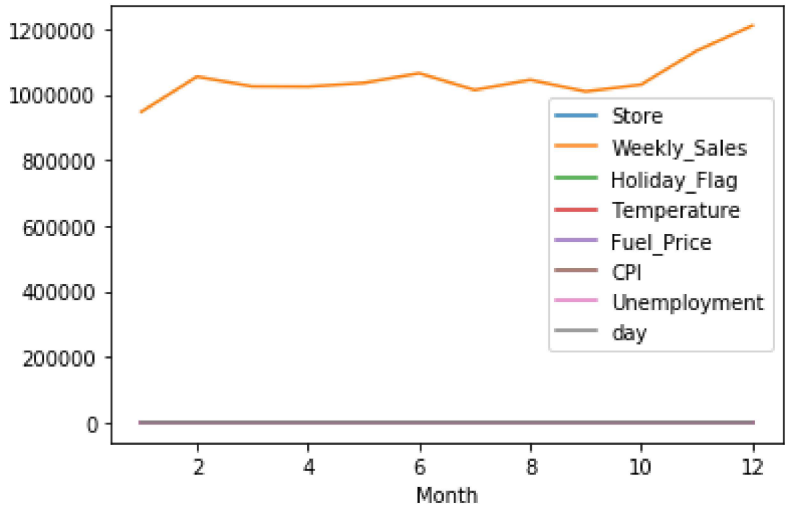
In [130]:
```
es=data.groupby(['year'])['Weekly_Sales'].sum()
es.plot()
plt.savefig("The distribution of sales yearly.pdf")
```



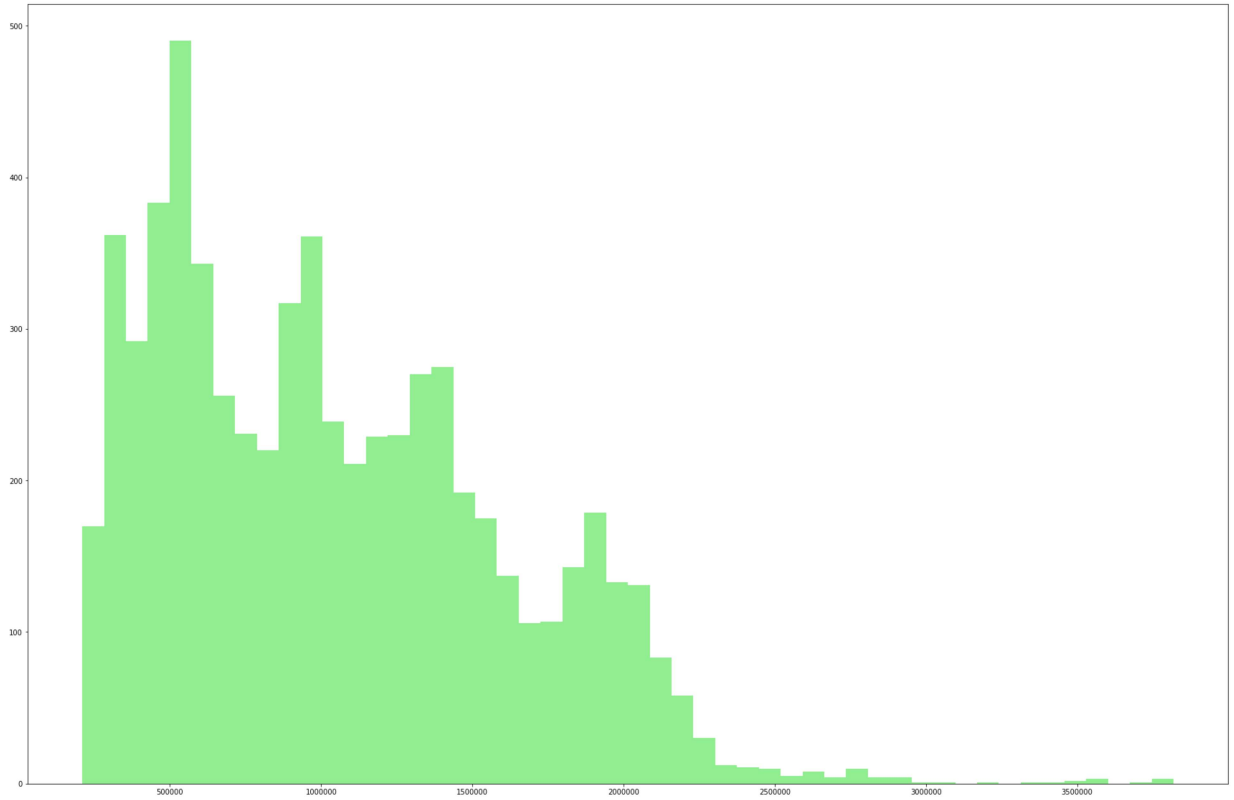
In [131]:
```
eq=data.groupby(['day'])['Weekly_Sales'].sum()
eq.plot()
plt.savefig("The distribution of sales Day wise.pdf")
```



In [133]:
```
mn=data.groupby(['Month']).mean()
mn.plot()
plt.savefig("the monthly sales.pdf")
```



In [134]:
```
x=data['Weekly_Sales']
plt.figure(figsize=(30, 20))

plt.hist(x, bins = 50,color=['lightgreen'])
plt.show()
plt.savefig(" the sales.pdf")
```




```
<Figure size 432x288 with 0 Axes>
```

In [158]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
```

In [170]:
```
store1= data[data['Store']==1]
store1
```

Out[170]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemploymen
34	1	2010-01-10	1453329.50	0	71.89	2.603	211.671989	7.838
8	1	2010-02-04	1594968.28	0	62.27	2.719	210.820450	7.808
21	1	2010-02-07	1492418.14	0	80.91	2.669	211.223533	7.787
2	1	2010-02-19	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	2010-02-26	1409727.59	0	46.63	2.561	211.319643	8.106
...	...	...	...	...	...	...	...	..
131	1	2012-10-08	1592409.97	0	85.05	3.494	221.958433	6.908
141	1	2012-10-19	1508068.77	0	67.97	3.594	223.425723	6.573
142	1	2012-10-26	1493659.74	0	69.16	3.506	223.444251	6.573
118	1	2012-11-05	1611096.05	0	73.77	3.688	221.725663	7.143
140	1	2012-12-10	1573072.81	0	62.99	3.601	223.381296	6.573

143 rows × 13 columns

In [185]:
```
X = store1[['Fuel_Price','CPI','Unemployment']].values
Y = store1['Weekly_Sales'].values
```

In [186]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_s
```

In [187]:
```
regressor = LinearRegression()
regressor.fit(X_train, Y_train)
```

Out[187]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [188]:
```
Y_pred = regressor.predict(X_test)
Y_pred
```

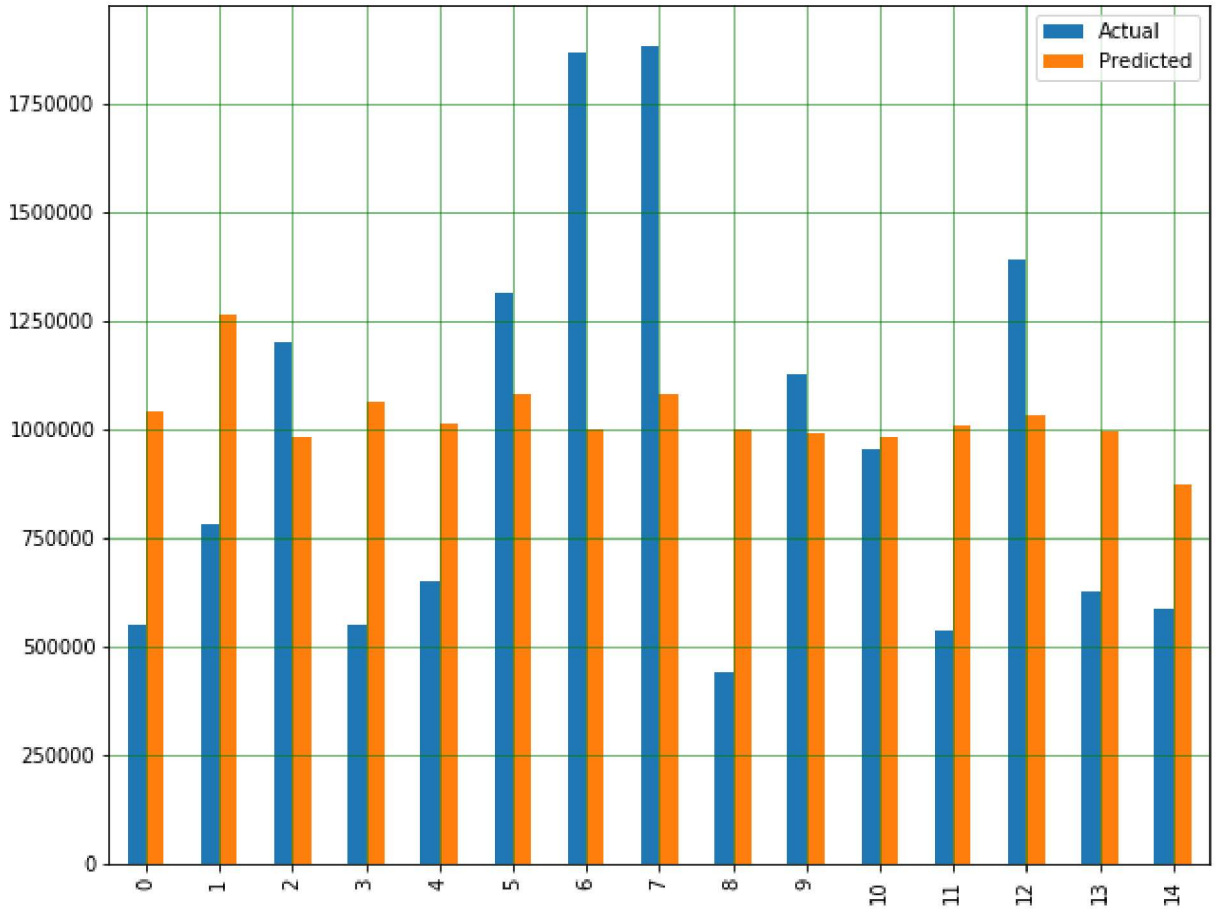
Out[188]:
```
array([1039337.26229671, 1263064.3291724 ,  982982.37943634, ...,
        907987.82187244,  977142.88747212, 1012111.43249565])
```

In [189]:
```
df = pd.DataFrame({'Actual': Y_test, 'Predicted': Y_pred})
df1 = df.head(15)
df1
```

Out[189]:

	Actual	Predicted
0	550076.32	1.039337e+06
1	780607.52	1.263064e+06
2	1199845.29	9.829824e+05
3	550414.99	1.061850e+06
4	648606.13	1.014881e+06
5	1311950.16	1.079856e+06
6	1869110.55	9.989614e+05
7	1882393.40	1.082346e+06
8	440491.33	9.987054e+05
9	1126962.44	9.919636e+05
10	955463.84	9.797191e+05
11	534847.96	1.010388e+06
12	1388973.65	1.033847e+06
13	629152.06	9.937291e+05
14	586781.78	8.746962e+05

In [190]:
```
df1.plot(kind='bar',figsize=(10,8))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



In [191]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pr
```

```
Mean Absolute Error: 460965.562636205
Mean Squared Error: 291554733460.479
Root Mean Squared Error: 539958.0849107447
```

In [192]:
```
regressor.score(X_test,Y_test)
```

Out[192]:
```
0.028380008148666813
```

In [193]:
```
data['day'] = data['Date'].dt.day
```

In [194]:
```
store1= data[data['Store']==1]

store1=pd.get_dummies(data, columns=['Holiday_Flag','Store'])
store1
```

Out[194]:

	Date	Weekly_Sales	Temperature	Fuel_Price	CPI	Unemployment	year_month	year
606	2010-01-10	283178.12	71.10	2.603	212.226946	6.768	2010-01	2010
2036	2010-01-10	566945.95	59.69	2.840	132.756800	8.067	2010-01	2010
5897	2010-01-10	481523.93	86.01	3.001	126.234600	9.003	2010-01	2010
4610	2010-01-10	224294.39	91.45	3.001	126.234600	9.265	2010-01	2010
5039	2010-01-10	422169.47	74.66	2.567	210.440443	8.476	2010-01	2010
...	...	...	...	...	...	...	...	...
5860	2012-12-10	1409544.97	39.38	3.760	199.053937	6.195	2012-12	2012
2285	2012-12-10	491817.19	43.26	3.760	199.053937	5.847	2012-12	2012
1427	2012-12-10	1713889.11	76.03	4.468	131.108333	6.943	2012-12	2012
3572	2012-12-10	697317.41	43.74	4.000	216.115057	7.293	2012-12	2012
283	2012-12-10	1900745.13	60.97	3.601	223.015426	6.170	2012-12	2012

6435 rows × 58 columns

In [195]:
```
store1.columns
```

Out[195]:
```
Index(['Date', 'Weekly_Sales', 'Temperature', 'Fuel_Price', 'CPI',
       'Unemployment', 'year_month', 'year', 'day', 'quarter', 'Month',
       'Holiday_Flag_0', 'Holiday_Flag_1', 'Store_1', 'Store_2', 'Store_3',
       'Store_4', 'Store_5', 'Store_6', 'Store_7', 'Store_8', 'Store_9',
       'Store_10', 'Store_11', 'Store_12', 'Store_13', 'Store_14', 'Store_15',
       'Store_16', 'Store_17', 'Store_18', 'Store_19', 'Store_20', 'Store_21',
       'Store_22', 'Store_23', 'Store_24', 'Store_25', 'Store_26', 'Store_27',
       'Store_28', 'Store_29', 'Store_30', 'Store_31', 'Store_32', 'Store_33',
       'Store_34', 'Store_35', 'Store_36', 'Store_37', 'Store_38', 'Store_39',
       'Store_40', 'Store_41', 'Store_42', 'Store_43', 'Store_44', 'Store_45'],
      dtype='object')
```

In [197]:
```
import statsmodels.formula.api as smf
lm=smf.ols(formula='Weekly_Sales~Fuel_Price+CPI+Unemployment',data=store1).fit()
```

In [198]:
```
lm.summary()
```

Out[198]:

OLS Regression Results

Dep. Variable:	Weekly_Sales	**R-squared:**	0.024
Model:	OLS	**Adj. R-squared:**	0.023
Method:	Least Squares	**F-statistic:**	51.75
Date:	Mon, 18 May 2020	**Prob (F-statistic):**	4.81e-33
Time:	23:41:00	**Log-Likelihood:**	-94275.
No. Observations:	6435	**AIC:**	1.886e+05
Df Residuals:	6431	**BIC:**	1.886e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.746e+06	7.96e+04	21.938	0.000	1.59e+06	1.9e+06
Fuel_Price	-1.927e+04	1.54e+04	-1.248	0.212	-4.95e+04	1.1e+04
CPI	-1696.8760	188.793	-8.988	0.000	-2066.973	-1326.779
Unemployment	-4.286e+04	3905.197	-10.975	0.000	-5.05e+04	-3.52e+04

Omnibus:	370.117	**Durbin-Watson:**	1.974
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	436.792
Skew:	0.638	**Prob(JB):**	1.42e-95
Kurtosis:	3.051	**Cond. No.**	2.04e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.04e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

In [125]:
```
xx=store1[['CPI','Temperature','Unemployment','Holiday_Flag_0','Holiday_Flag_1',
       'Store_7', 'Store_8', 'Store_9', 'Store_10', 'Store_11', 'Store_12',
       'Store_13', 'Store_14', 'Store_15', 'Store_16', 'Store_17', 'Store_18',
       'Store_19', 'Store_20', 'Store_21', 'Store_22', 'Store_23', 'Store_24',
       'Store_25', 'Store_26', 'Store_27', 'Store_28', 'Store_29', 'Store_30',
       'Store_31', 'Store_32', 'Store_33', 'Store_34', 'Store_35', 'Store_36',
       'Store_37', 'Store_38', 'Store_39', 'Store_40', 'Store_41', 'Store_42',
       'Store_43', 'Store_44', 'Store_45']]
yy=store1[['Weekly_Sales']]
xx_train, xx_test, yy_train, yy_test = train_test_split(xx, yy, test_size=0.2, ra
```

In [126]:
```
regre = LinearRegression()
regre.fit(xx_train, yy_train)
```

Out[126]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [127]:
```
yy_pred = regre.predict(xx_test)
yy_pred
```

Out[127]:
```
array([[1313562.],
       [1160240.],
       [ 515644.],
       ...,
       [ 795916.],
       [1246952.],
       [ 916482.]])
```

In [132]:
```
regre.score(xx_test,yy_test)
```

Out[132]:
```
0.916387902281158
```

In [129]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pr
```

```
Mean Absolute Error: 118810.20204491162
Mean Squared Error: 25457357661.953217
Root Mean Squared Error: 159553.62002146244
```

In [130]:
```
from sklearn.metrics import r2_score
r2_score(yy_test, yy_pred)
```

Out[130]:
```
0.916387902281158
```