```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

data=pd.read_csv('Walmart_Store_sales.csv')
print('data loaded')
```

```
data loaded
```

```
data.head()
```

```
   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
```

```
data.tail()
```

```
      Store        Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
6430     45  28-09-2012     713173.95             0        64.88
3.997
6431     45  05-10-2012     733455.07             0        64.89
3.985
6432     45  12-10-2012     734464.36             0        54.47
4.000
6433     45  19-10-2012     718125.53             0        56.47
3.969
6434     45  26-10-2012     760281.43             0        58.85
3.882
```

```
              CPI  Unemployment
6430  192.013558         8.684
6431  192.170412         8.667
6432  192.327265         8.667
6433  192.330854         8.667
6434  192.308899         8.667
```

```
duplicate=data[data.duplicated()]
duplicate
```

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

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

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

```
data.dtypes
```

```
Store             int64
Date             object
Weekly_Sales    float64
Holiday_Flag      int64
Temperature     float64
Fuel_Price      float64
CPI             float64
Unemployment    float64
dtype: object
```

```
data.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   object
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64
```

```
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

```
data['Date']=pd.to_datetime(data['Date'],dayfirst=True)
```

```
dupdata=data.copy()
```

```
data.set_index('Date',drop=True,inplace=True)
```

```
data['Store'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of Date
2010-02-05     1
2010-02-12     1
2010-02-19     1
2010-02-26     1
2010-03-05     1
              ..
2012-09-28    45
2012-10-05    45
2012-10-12    45
2012-10-19    45
2012-10-26    45
Name: Store, Length: 6435, dtype: int64>
```

```
grpdata=data.groupby('Store')['Weekly_Sales'].sum().round()
grpdata
```

```
Store
1     222402809.0
2     275382441.0
3      57586735.0
4     299543953.0
5      45475689.0
6     223756131.0
7      81598275.0
8     129951181.0
9      77789219.0
10    271617714.0
11    193962787.0
12    144287230.0
13    286517704.0
14    288999911.0
15     89133684.0
16     74252425.0
17    127782139.0
18    155114734.0
```

```
19    206634862.0
20    301397792.0
21    108117879.0
22    147075649.0
23    198750618.0
24    194016021.0
25    101061179.0
26    143416394.0
27    253855917.0
28    189263681.0
29     77141554.0
30     62716885.0
31    199613906.0
32    166819246.0
33     37160222.0
34    138249763.0
35    131520672.0
36     53412215.0
37     74202740.0
38     55159626.0
39    207445542.0
40    137870310.0
41    181341935.0
42     79565752.0
43     90565435.0
44     43293088.0
45    112395341.0
Name: Weekly_Sales, dtype: float64
```

```
grpdata.sort_values(ascending=0)
```

```
Store
20    301397792.0
4     299543953.0
14    288999911.0
13    286517704.0
2     275382441.0
10    271617714.0
27    253855917.0
6     223756131.0
1     222402809.0
39    207445542.0
19    206634862.0
31    199613906.0
23    198750618.0
24    194016021.0
11    193962787.0
28    189263681.0
41    181341935.0
32    166819246.0
```

```
18    155114734.0
22    147075649.0
12    144287230.0
26    143416394.0
34    138249763.0
40    137870310.0
35    131520672.0
8     129951181.0
17    127782139.0
45    112395341.0
21    108117879.0
25    101061179.0
43     90565435.0
15     89133684.0
7      81598275.0
42     79565752.0
9      77789219.0
29     77141554.0
16     74252425.0
37     74202740.0
30     62716885.0
3      57586735.0
38     55159626.0
36     53412215.0
5      45475689.0
44     43293088.0
33     37160222.0
Name: Weekly_Sales, dtype: float64
```

Findings

1: Store 20 has the max sales of 301397792.0 per week

```
grpstddata=data.groupby('Store')
['Weekly_Sales'].std().round().sort_values(ascending=0)
grpstddata
```

```
Store
14    317570.0
10    302262.0
20    275901.0
4     266201.0
13    265507.0
23    249788.0
27    239930.0
2     237684.0
39    217466.0
6     212526.0
35    211243.0
```

```
19    191723.0
41    187907.0
28    181759.0
18    176642.0
24    167746.0
11    165834.0
22    161251.0
1     155981.0
12    139167.0
32    138017.0
45    130169.0
21    128753.0
31    125856.0
15    120539.0
40    119002.0
25    112977.0
7     112585.0
17    112163.0
26    110431.0
8     106281.0
34    104630.0
29     99120.0
16     85770.0
9      69029.0
36     60725.0
42     50263.0
3      46320.0
38     42768.0
43     40598.0
5      37738.0
44     24763.0
33     24133.0
30     22810.0
37     21837.0
Name: Weekly_Sales, dtype: float64

Store14data=data[data.Store==14]
Store14data
```

Date	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price
2010-02-05	14	2623469.95	0	27.31	2.784
2010-02-12	14	1704218.84	1	27.73	2.773
2010-02-19	14	2204556.70	0	31.27	2.745
2010-02-26	14	2095591.63	0	34.89	2.754

```
2010-03-05     14    2237544.75             0        37.13        2.777

...           ...           ...           ...          ...          ...

2012-09-28     14    1522512.20             0        64.88        3.997

2012-10-05     14    1687592.16             0        64.89        3.985

2012-10-12     14    1639585.61             0        54.47        4.000

2012-10-19     14    1590274.72             0        56.47        3.969

2012-10-26     14    1704357.62             0        58.85        3.882


                    CPI  Unemployment
Date
2010-02-05  181.871190         8.992
2010-02-12  181.982317         8.992
2010-02-19  182.034782         8.992
2010-02-26  182.077469         8.992
2010-03-05  182.120157         8.992
...                ...           ...
2012-09-28  192.013558         8.684
2012-10-05  192.170412         8.667
2012-10-12  192.327265         8.667
2012-10-19  192.330854         8.667
2012-10-26  192.308899         8.667

[143 rows x 7 columns]
```

```
Store14mean=Store14data.groupby('Store')
['Weekly_Sales'].mean().round()

Store14mean
```

```
Store
14    2020978.0
Name: Weekly_Sales, dtype: float64
```

```
Store14std=Store14data.groupby('Store')['Weekly_Sales'].std().round()
Store14std
```

```
Store
14    317570.0
Name: Weekly_Sales, dtype: float64
```

```
CV=Store14std/Store14mean*100
CV
```

```
Store
14    15.713679
Name: Weekly_Sales, dtype: float64
```

```
CV.round()
```

```
Store
14    16.0
Name: Weekly_Sales, dtype: float64
```

Findings:

1: Store 14 has max standard deviation. 2: The coefficient of mean to standard deviation is 15.713679 (round 16.0)

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6435 entries, 2010-02-05 to 2012-10-26
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Store         6435 non-null   int64  
 1   Weekly_Sales  6435 non-null   float64
 2   Holiday_Flag  6435 non-null   int64  
 3   Temperature   6435 non-null   float64
 4   Fuel_Price    6435 non-null   float64
 5   CPI           6435 non-null   float64
 6   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2)
memory usage: 402.2 KB
```

```
data
```

```
            Store  Weekly_Sales  Holiday_Flag  Temperature  Fuel_Price
\
Date

2010-02-05      1    1643690.90             0        42.31       2.572

2010-02-12      1    1641957.44             1        38.51       2.548

2010-02-19      1    1611968.17             0        39.93       2.514

2010-02-26      1    1409727.59             0        46.63       2.561

2010-03-05      1    1554806.68             0        46.50       2.625

...           ...           ...           ...          ...         ...
```

```
2012-09-28     45     713173.95             0        64.88        3.997

2012-10-05     45     733455.07             0        64.89        3.985

2012-10-12     45     734464.36             0        54.47        4.000

2012-10-19     45     718125.53             0        56.47        3.969

2012-10-26     45     760281.43             0        58.85        3.882


                    CPI  Unemployment
Date
2010-02-05  211.096358         8.106
2010-02-12  211.242170         8.106
2010-02-19  211.289143         8.106
2010-02-26  211.319643         8.106
2010-03-05  211.350143         8.106
...                ...           ...
2012-09-28  192.013558         8.684
2012-10-05  192.170412         8.667
2012-10-12  192.327265         8.667
2012-10-19  192.330854         8.667
2012-10-26  192.308899         8.667

[6435 rows x 7 columns]
```

```
copydata=data.copy()
qtrdata=copydata.groupby([copydata.Store,pd.Grouper(freq='QE')])
['Weekly_Sales'].sum().round().reset_index()
qtrdata=qtrdata.rename(columns={'Date':'Quarter'})
print(qtrdata)
```

```
     Store    Quarter  Weekly_Sales
0        1 2010-03-31    12178638.0
1        1 2010-06-30    19436822.0
2        1 2010-09-30    19150230.0
3        1 2010-12-31    22513142.0
4        1 2011-03-31    18187314.0
..     ...        ...           ...
535     45 2011-12-31    11917228.0
536     45 2012-03-31     9805268.0
537     45 2012-06-30    10390768.0
538     45 2012-09-30     9581268.0
539     45 2012-12-31     2946326.0

[540 rows x 3 columns]
```

```
qtrdata['GrowthRate']=qtrdata.Weekly_Sales.pct_change().mul(100)
qtrdata
```

```
     Store    Quarter  Weekly_Sales  GrowthRate
0        1 2010-03-31    12178638.0         NaN
1        1 2010-06-30    19436822.0   59.597666
2        1 2010-09-30    19150230.0   -1.474480
3        1 2010-12-31    22513142.0   17.560687
4        1 2011-03-31    18187314.0  -19.214679
..     ...        ...           ...         ...
535     45 2011-12-31    11917228.0   14.715061
536     45 2012-03-31     9805268.0  -17.721906
537     45 2012-06-30    10390768.0    5.971280
538     45 2012-09-30     9581268.0   -7.790569
539     45 2012-12-31     2946326.0  -69.249101

[540 rows x 4 columns]
```

```
data2012=qtrdata[(qtrdata.Quarter>='2012-04-
1')&(qtrdata.Quarter>='2012-09-30')]
data2012
```

```
     Store    Quarter  Weekly_Sales  GrowthRate
10       1 2012-09-30    20253948.0   -3.454980
11       1 2012-12-31     6245587.0  -69.163607
22       2 2012-09-30    24303355.0   -3.110598
23       2 2012-12-31     7581515.0  -68.804657
34       3 2012-09-30     5298005.0   -5.734749
..     ...        ...           ...         ...
515     43 2012-12-31     2473507.0  -69.083373
526     44 2012-09-30     4411251.0    2.434629
527     44 2012-12-31     1360020.0  -69.169290
538     45 2012-09-30     9581268.0   -7.790569
539     45 2012-12-31     2946326.0  -69.249101

[90 rows x 4 columns]
```

```
Q32012=data2012.loc[data2012['GrowthRate']>=0]
Q32012
```

```
     Store    Quarter  Weekly_Sales  GrowthRate
82       7 2012-09-30     8262787.0   13.330775
190     16 2012-09-30     7121542.0    8.488383
274     23 2012-09-30    18641489.0    0.825393
286     24 2012-09-30    17976378.0    1.652089
310     26 2012-09-30    13675692.0    3.955475
418     35 2012-09-30    11322421.0    4.466636
466     39 2012-09-30    20715116.0    2.478405
478     40 2012-09-30    12873195.0    1.142835
490     41 2012-09-30    18093844.0    2.456978
526     44 2012-09-30     4411251.0    2.434629
```

Findings:

1- Store 7 has max of all in 13.330775 growth rate in Q32012

```
SuperBowl= ['12-2-2010', '11-2-2011', '10-2-2012', '8-2-2013']
LabourDay= ['10-9-2010', '9-9-2011', '7-9-2012', '6-9-2013']
Thanksgiving= ['26-11-2010', '25-11-2011', '23-11-2012', '29-11-2013']
Christmas= ['31-12-2010', '30-12-2011', '28-12-2012', '27-12-2013']

SuperBowl=pd.to_datetime(SuperBowl,dayfirst=True)
LabourDay=pd.to_datetime(LabourDay,dayfirst=True)
Thanksgiving=pd.to_datetime(Thanksgiving,dayfirst=True)
Christmas=pd.to_datetime(Christmas,dayfirst=True)

SuperBowl
```

```
DatetimeIndex(['2010-02-12', '2011-02-11', '2012-02-10', '2013-02-
08'], dtype='datetime64[ns]', freq=None)
```

```
hdata=data.copy()

hdata.head(87)
```

Date	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
2010-02-05	1	1643690.90	0	42.31	2.572		
2010-02-12	1	1641957.44	1	38.51	2.548		
2010-02-19	1	1611968.17	0	39.93	2.514		
2010-02-26	1	1409727.59	0	46.63	2.561		
2010-03-05	1	1554806.68	0	46.50	2.625		
...	...	...	...	...	...		
2011-09-02	1	1550229.22	0	87.83	3.533		
2011-09-09	1	1540471.24	1	76.00	3.546		
2011-09-16	1	1514259.78	0	79.94	3.526		
2011-09-23	1	1380020.27	0	75.80	3.467		
2011-09-30	1	1394561.83	0	79.69	3.355		

```
2010-02-05  211.096358         8.106
2010-02-12  211.242170         8.106
2010-02-19  211.289143         8.106
2010-02-26  211.319643         8.106
2010-03-05  211.350143         8.106
...                ...           ...
2011-09-02  215.797141         7.962
2011-09-09  215.861056         7.962
2011-09-16  216.041053         7.962
2011-09-23  216.375825         7.962
2011-09-30  216.710597         7.962

[87 rows x 7 columns]
```

```
hdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6435 entries, 2010-02-05 to 2012-10-26
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Store         6435 non-null   int64  
 1   Weekly_Sales  6435 non-null   float64
 2   Holiday_Flag  6435 non-null   int64  
 3   Temperature   6435 non-null   float64
 4   Fuel_Price    6435 non-null   float64
 5   CPI           6435 non-null   float64
 6   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2)
memory usage: 402.2 KB
```

```
hdata.reset_index(inplace=True)
```

```
hdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   Date          6435 non-null   datetime64[ns]
 1   Store         6435 non-null   int64         
 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
```

```
SuperBowlSales=hdata.loc[hdata.Date.isin(SuperBowl)]
['Weekly_Sales'].mean().round(2)

SuperBowlSales

np.float64(1079127.99)

LabourDaySales=hdata.loc[hdata.Date.isin(LabourDay)]
['Weekly_Sales'].mean().round(2)
LabourDaySales

np.float64(1042427.29)

ThanksgivingSales=hdata.loc[hdata.Date.isin(Thanksgiving)]
['Weekly_Sales'].mean().round(2)
ThanksgivingSales

np.float64(1471273.43)

ChristmasSales=hdata.loc[hdata.Date.isin(Christmas)]
['Weekly_Sales'].mean().round(2)
ChristmasSales

np.float64(960833.11)

nonholiday=hdata[hdata.Holiday_Flag==0]
['Weekly_Sales'].mean().round(2)
nonholiday

np.float64(1041256.38)

analysisdf=pd.DataFrame([{'Super Bowl': SuperBowlSales,'Labour
Day':LabourDaySales,'Christmas':ChristmasSales,
                          'Thanksgiving':ThanksgivingSales,'Non-
holiday':nonholiday}]).T
analysisdf

                         0
Super Bowl    1079127.99
Labour Day    1042427.29
Christmas      960833.11
Thanksgiving  1471273.43
Non-holiday   1041256.38

SuperBowlholiday=hdata.loc[hdata.Date.isin(SuperBowl)]
SuperBowlholiday=SuperBowlholiday.groupby('Date').agg(WeeklySales_Supe
rBowl=('Weekly_Sales','mean'))
SuperBowlholiday=SuperBowlholiday.round(2)

LabourDayholiday=hdata.loc[hdata.Date.isin(LabourDay)]
LabourDayholiday=LabourDayholiday.groupby('Date').agg(WeeklySales_Labo
```

```
urDay=('Weekly_Sales','mean'))
LabourDayholiday=LabourDayholiday.round(2)

Christmasholiday=hdata.loc[hdata.Date.isin(Christmas)]
Christmasholiday=Christmasholiday.groupby('Date').agg(WeeklySales_Chri
stmas=('Weekly_Sales','mean'))
Christmasholiday=Christmasholiday.round(2)

Thanksgivingholiday=hdata.loc[hdata.Date.isin(Thanksgiving)]
Thanksgivingholiday=Thanksgivingholiday.groupby('Date').agg(WeeklySale
s_Thanksgiving=('Weekly_Sales','mean'))
Thanksgivingholiday=Thanksgivingholiday.round(2)

merge=pd.concat([SuperBowlholiday,LabourDayholiday,Christmasholiday,Th
anksgivingholiday])
merge
```

Date	WeeklySales_SuperBowl	WeeklySales_LabourDay	\
2010-02-12	1074148.39	NaN	
2011-02-11	1051915.40	NaN	
2012-02-10	1111320.18	NaN	
2010-09-10	NaN	1014097.73	
2011-09-09	NaN	1039182.83	
2012-09-07	NaN	1074001.32	
2010-12-31	NaN	NaN	
2011-12-30	NaN	NaN	
2010-11-26	NaN	NaN	
2011-11-25	NaN	NaN	

Date	WeeklySales_Christmas	WeeklySales_Thanksgiving
2010-02-12	NaN	NaN
2011-02-11	NaN	NaN
2012-02-10	NaN	NaN
2010-09-10	NaN	NaN
2011-09-09	NaN	NaN
2012-09-07	NaN	NaN
2010-12-31	898500.42	NaN
2011-12-30	1023165.80	NaN
2010-11-26	NaN	1462688.96
2011-11-25	NaN	1479857.89

Findings

1: Super Bowl, Labour Day and Thanksgiving has more sale than mean of non-holiday sale 2: 2010-02-12,2011-02-11,2012-02-10,2012-09-07,2010-11-26,2011-11-25 these dates have more sales than mean of non holiday sales

```
salesdata=data.copy()
salesdata.tail(20)
```

Date	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
2012-06-15	45	821498.18	0	71.93	3.620	191.029973	8.567
2012-06-22	45	822569.16	0	74.22	3.564	191.064610	8.567
2012-06-29	45	773367.71	0	75.22	3.506		
2012-07-06	45	843361.10	0	82.99	3.475		
2012-07-13	45	749817.08	0	79.97	3.523		
2012-07-20	45	737613.65	0	78.89	3.567		
2012-07-27	45	711671.58	0	77.20	3.647		
2012-08-03	45	725729.51	0	76.58	3.654		
2012-08-10	45	733037.32	0	78.65	3.722		
2012-08-17	45	722496.93	0	75.71	3.807		
2012-08-24	45	718232.26	0	72.62	3.834		
2012-08-31	45	734297.87	0	75.09	3.867		
2012-09-07	45	766512.66	1	75.70	3.911		
2012-09-14	45	702238.27	0	67.87	3.948		
2012-09-21	45	723086.20	0	65.32	4.038		
2012-09-28	45	713173.95	0	64.88	3.997		
2012-10-05	45	733455.07	0	64.89	3.985		
2012-10-12	45	734464.36	0	54.47	4.000		
2012-10-19	45	718125.53	0	56.47	3.969		
2012-10-26	45	760281.43	0	58.85	3.882		

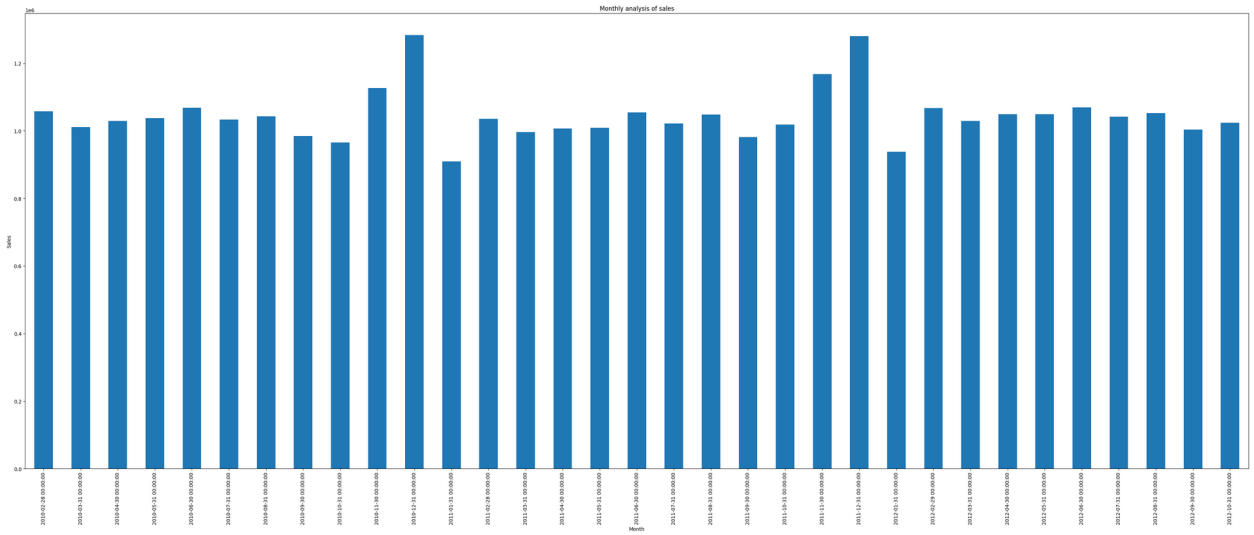
```
2012-06-29  191.099246          8.567
2012-07-06  191.133883          8.684
2012-07-13  191.168519          8.684
2012-07-20  191.167043          8.684
2012-07-27  191.165566          8.684
2012-08-03  191.164090          8.684
2012-08-10  191.162613          8.684
2012-08-17  191.228492          8.684
2012-08-24  191.344887          8.684
2012-08-31  191.461281          8.684
2012-09-07  191.577676          8.684
2012-09-14  191.699850          8.684
2012-09-21  191.856704          8.684
2012-09-28  192.013558          8.684
2012-10-05  192.170412          8.667
2012-10-12  192.327265          8.667
2012-10-19  192.330854          8.667
2012-10-26  192.308899          8.667
```

```
monthdata=salesdata.groupby(pd.Grouper(freq='ME'))
['Weekly_Sales'].mean().round()
monthdata
```

```
Date
2010-02-28    1057405.0
2010-03-31    1010666.0
2010-04-30    1028499.0
2010-05-31    1037283.0
2010-06-30    1068034.0
2010-07-31    1033689.0
2010-08-31    1042445.0
2010-09-30     984822.0
2010-10-31     965164.0
2010-11-30    1126963.0
2010-12-31    1283380.0
2011-01-31     909466.0
2011-02-28    1035174.0
2011-03-31     996425.0
2011-04-30    1006784.0
2011-05-31    1009156.0
2011-06-30    1054297.0
2011-07-31    1021828.0
2011-08-31    1047774.0
2011-09-30     981546.0
2011-10-31    1018118.0
2011-11-30    1167569.0
2011-12-31    1280347.0
2012-01-31     938303.0
2012-02-29    1067020.0
2012-03-31    1028932.0
```

```
2012-04-30    1049561.0
2012-05-31    1048703.0
2012-06-30    1069379.0
2012-07-31    1041719.0
2012-08-31    1052670.0
2012-09-30    1003586.0
2012-10-31    1024232.0
Freq: ME, Name: Weekly_Sales, dtype: float64
```

```
fig, ax = plt.subplots(figsize=(35,15))
monthdata.plot(kind='bar',ax=ax)
plt.title('Monthly analysis of sales')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.tight_layout()
plt.savefig('monthlyanalysisofsales.png')
```



Findings:

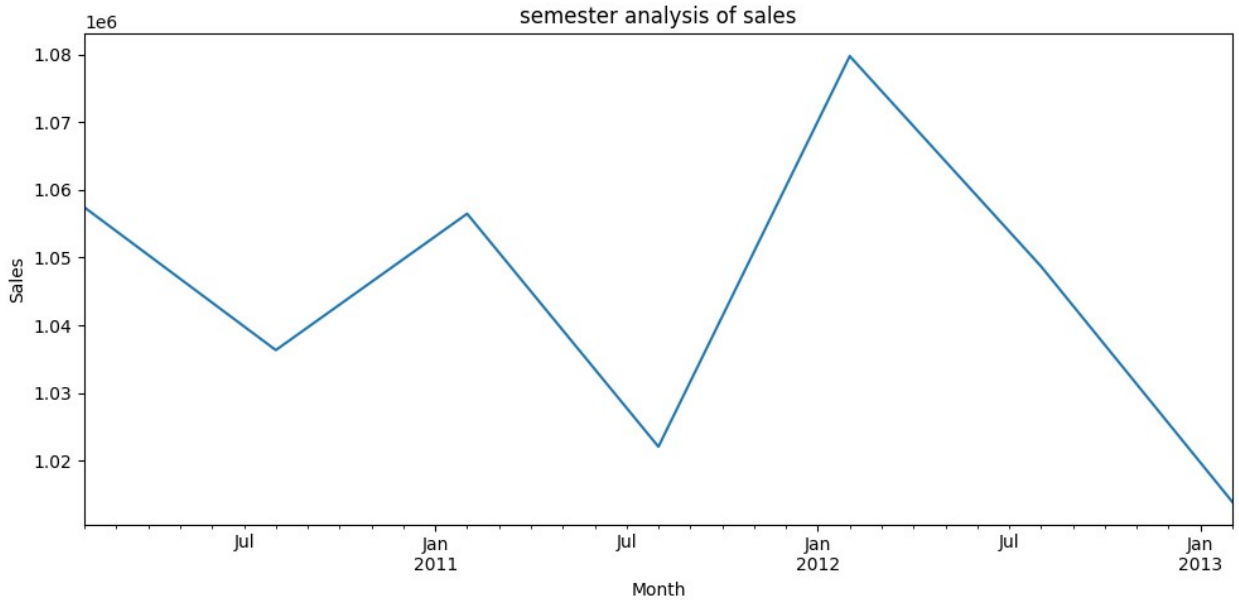
1: Due to Christmas and New year holidays, the sales are higher in Dec-31-2010 and Dec-31-2011. 2: Post holiday month, it can been seen the Sales are least of all in 2010 and 2011.

```
semdata=salesdata.groupby(pd.Grouper(freq='6ME'))
['Weekly_Sales'].mean().round()
semdata
```

```
Date
2010-02-28    1057405.0
2010-08-31    1036333.0
2011-02-28    1056478.0
2011-08-31    1022064.0
2012-02-29    1079750.0
2012-08-31    1048698.0
```

```
2013-02-28    1013909.0
Freq: 6ME, Name: Weekly_Sales, dtype: float64
```

```
fig, ax = plt.subplots(figsize=(10,5))
semdata.plot(kind='line',ax=ax)
plt.title('semester analysis of sales')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.tight_layout()
plt.savefig('semanalysisofsales.png')
```



Findings:

1: Max sales (1079750.0) happen in sem-5 (2012-02-29) due to holidays. 2: It appear in last sem (2013-02-28) the sales was least compare to previous year sem for the same time because of higher CPI and unemployment rate.

```
modeldata=dupdata.copy()
modeldata=modeldata[modeldata.Store==1]
modeldata.head()
```

```
   Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0      1 2010-02-05    1643690.90             0        42.31
2.572
1      1 2010-02-12    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 2010-03-05    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

modeldata.info()

<class 'pandas.core.frame.DataFrame'>
Index: 143 entries, 0 to 142
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         143 non-null    int64
 1   Date          143 non-null    datetime64[ns]
 2   Weekly_Sales  143 non-null    float64
 3   Holiday_Flag  143 non-null    int64
 4   Temperature   143 non-null    float64
 5   Fuel_Price    143 non-null    float64
 6   CPI           143 non-null    float64
 7   Unemployment  143 non-null    float64
dtypes: datetime64[ns](1), float64(5), int64(2)
memory usage: 10.1 KB

modeldata['Ref_date']=(modeldata['Date']-
modeldata['Date'].min()).dt.days+1
modeldata

     Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0        1 2010-02-05    1643690.90             0        42.31
2.572
1        1 2010-02-12    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 2010-03-05    1554806.68             0        46.50
2.625
..     ...        ...           ...           ...          ...
...
138      1 2012-09-28    1437059.26             0        76.08
3.666
139      1 2012-10-05    1670785.97             0        68.55
```

```
3.617
140      1 2012-10-12    1573072.81             0        62.99
3.601
141      1 2012-10-19    1508068.77             0        67.97
3.594
142      1 2012-10-26    1493659.74             0        69.16
3.506

             CPI  Unemployment  Ref_date
0     211.096358         8.106         1
1     211.242170         8.106         8
2     211.289143         8.106        15
3     211.319643         8.106        22
4     211.350143         8.106        29
..           ...           ...       ...
138   222.981658         6.908       967
139   223.181477         6.573       974
140   223.381296         6.573       981
141   223.425723         6.573       988
142   223.444251         6.573       995

[143 rows x 9 columns]
```

```
column='Weekly_Sales'
modeldata[column]=MinMaxScaler().fit_transform(np.array(modeldata[colu
mn]).reshape(-1,1))
modeldata
```

```
     Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0        1 2010-02-05      0.305113             0        42.31
2.572
1        1 2010-02-12      0.303495             1        38.51
2.548
2        1 2010-02-19      0.275495             0        39.93
2.514
3        1 2010-02-26      0.086670             0        46.63
2.561
4        1 2010-03-05      0.222125             0        46.50
2.625
..     ...        ...           ...           ...          ...
...
138      1 2012-09-28      0.112189             0        76.08
3.666
139      1 2012-10-05      0.330411             0        68.55
3.617
140      1 2012-10-12      0.239180             0        62.99
3.601
141      1 2012-10-19      0.178488             0        67.97
3.594
```

```
142      1 2012-10-26      0.165035             0        69.16
3.506

             CPI  Unemployment  Ref_date
0    211.096358         8.106         1
1    211.242170         8.106         8
2    211.289143         8.106        15
3    211.319643         8.106        22
4    211.350143         8.106        29
..          ...           ...       ...
138  222.981658         6.908       967
139  223.181477         6.573       974
140  223.381296         6.573       981
141  223.425723         6.573       988
142  223.444251         6.573       995

[143 rows x 9 columns]
```

```
column='CPI'
modeldata[column]=MinMaxScaler().fit_transform(np.array(modeldata[colu
mn]).reshape(-1,1))
modeldata
```

```
     Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0        1 2010-02-05      0.305113             0        42.31
2.572
1        1 2010-02-12      0.303495             1        38.51
2.548
2        1 2010-02-19      0.275495             0        39.93
2.514
3        1 2010-02-26      0.086670             0        46.63
2.561
4        1 2010-03-05      0.222125             0        46.50
2.625
..     ...        ...           ...           ...          ...
...
138      1 2012-09-28      0.112189             0        76.08
3.666
139      1 2012-10-05      0.330411             0        68.55
3.617
140      1 2012-10-12      0.239180             0        62.99
3.601
141      1 2012-10-19      0.178488             0        67.97
3.594
142      1 2012-10-26      0.165035             0        69.16
3.506

          CPI  Unemployment  Ref_date
0    0.057904         8.106         1
```

```
1    0.069028         8.106         8
2    0.072612         8.106        15
3    0.074939         8.106        22
4    0.077266         8.106        29
..        ...           ...       ...
138  0.964706         6.908       967
139  0.979951         6.573       974
140  0.995197         6.573       981
141  0.998586         6.573       988
142  1.000000         6.573       995

[143 rows x 9 columns]
```

```
column='Temperature'
modeldata[column]=MinMaxScaler().fit_transform(np.array(modeldata[colu
mn]).reshape(-1,1))
modeldata
```

```
     Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0        1 2010-02-05      0.305113             0     0.122844
2.572
1        1 2010-02-12      0.303495             1     0.055289
2.548
2        1 2010-02-19      0.275495             0     0.080533
2.514
3        1 2010-02-26      0.086670             0     0.199644
2.561
4        1 2010-03-05      0.222125             0     0.197333
2.625
..     ...        ...           ...           ...          ...
...
138      1 2012-09-28      0.112189             0     0.723200
3.666
139      1 2012-10-05      0.330411             0     0.589333
3.617
140      1 2012-10-12      0.239180             0     0.490489
3.601
141      1 2012-10-19      0.178488             0     0.579022
3.594
142      1 2012-10-26      0.165035             0     0.600178
3.506

          CPI  Unemployment  Ref_date
0    0.057904         8.106         1
1    0.069028         8.106         8
2    0.072612         8.106        15
3    0.074939         8.106        22
4    0.077266         8.106        29
..        ...           ...       ...
```

```
138  0.964706         6.908       967
139  0.979951         6.573       974
140  0.995197         6.573       981
141  0.998586         6.573       988
142  1.000000         6.573       995

[143 rows x 9 columns]
```

```
column='Unemployment'
modeldata[column]=MinMaxScaler().fit_transform(np.array(modeldata[colu
mn]).reshape(-1,1))
modeldata
```

```
     Store       Date  Weekly_Sales  Holiday_Flag  Temperature
Fuel_Price  \
0        1 2010-02-05      0.305113             0     0.122844
2.572
1        1 2010-02-12      0.303495             1     0.055289
2.548
2        1 2010-02-19      0.275495             0     0.080533
2.514
3        1 2010-02-26      0.086670             0     0.199644
2.561
4        1 2010-03-05      0.222125             0     0.197333
2.625
..     ...        ...           ...           ...          ...
...
138      1 2012-09-28      0.112189             0     0.723200
3.666
139      1 2012-10-05      0.330411             0     0.589333
3.617
140      1 2012-10-12      0.239180             0     0.490489
3.601
141      1 2012-10-19      0.178488             0     0.579022
3.594
142      1 2012-10-26      0.165035             0     0.600178
3.506

          CPI  Unemployment  Ref_date
0    0.057904      1.000000         1
1    0.069028      1.000000         8
2    0.072612      1.000000        15
3    0.074939      1.000000        22
4    0.077266      1.000000        29
..        ...           ...       ...
138  0.964706      0.218526       967
139  0.979951      0.000000       974
140  0.995197      0.000000       981
141  0.998586      0.000000       988
142  1.000000      0.000000       995
```

```
[143 rows x 9 columns]
```

```
dependent=modeldata[['Ref_date','Holiday_Flag','Temperature','Fuel_Price','CPI','Unemployment']]
independent=modeldata['Weekly_Sales']

X_train,X_test,y_train,y_test=train_test_split(dependent,independent,test_size=0.8,random_state=42)

lr=LinearRegression()
lr.fit(X_train,y_train)
```

```
LinearRegression()
```

```
predict=lr.predict(X_test)

r2=r2_score(y_test,predict)

meanSquareError= mean_squared_error(y_test, predict)
print(f"R-squared: {r2}")
print(f"Mean Squared Error: {meanSquareError}")
print(f"Model Coefficients: {lr.coef_}")
```

```
R-squared: 0.027171864887741037
Mean Squared Error: 0.022840286048703435
Model Coefficients: [-2.41669631e-05  2.80993043e-02 -7.67687351e-02
5.79237977e-02
  2.88007152e-02  1.87248310e-03]
```

Findings:

1- R-squared: 0.0272 — The R-squared value explains 2.72% of the variance in sales. However, this is still a low value, suggesting the model doesn't fully capture the relationships in the data.

2- Mean Squared Error (MSE): 0.0228 — This indicates that the data likely went through normalization or scaling, which affected the scale of the output, making the errors look smaller.

3-Model Coefficients:

a) Ref_date: -0.000024 — A very small negative impact of days since the start on sales.

b) Holiday_Flag: 0.0281 — Holidays are associated with an increase in sales by about 2.81%.

c) Temperature: -0.0768 — Higher temperatures have a negative impact on sales, though small.

d) Fuel_Price: 0.0579 — Higher fuel prices have a positive relationship with sales, which could suggest some external or macroeconomic factors at play.

e) CPI: 0.0288 — A slight positive relationship between CPI and sales.

f) Unemployment: 0.00187 — An extremely small positive impact from unemployment on sales.

Insights:

1-Low R-squared Value: Despite the normalization, the R-squared is still very low, indicating that the linear model still does not capture the patterns well.

2- Model Coefficients: The impact of all variables is small, suggesting that none of the predictors have a strong influence on sales in this model.

3- Interpretation of Scaled Coefficients: Since the data is likely scaled, the coefficients represent the change in the target variable (scaled sales) in response to a one-unit change in the standardized predictors.