import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

data=pd.read\_csv("../input/retail-analysis-with-walmart-data/Walmart\_Store\_sales.csv")

data.head() #Fetches default first 5 rows

|  | **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 |

#basic Functions

data.isnull().sum() #Null value check

Store 0

Date 0

Weekly\_Sales 0

Holiday\_Flag 0

Temperature 0

Fuel\_Price 0

CPI 0

Unemployment 0

dtype: int64

data.shape

(6435, 8)

data.describe()

|  | **Store** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 6435.000000 | 6.435000e+03 | 6435.000000 | 6435.000000 | 6435.000000 | 6435.000000 | 6435.000000 |
| **mean** | 23.000000 | 1.046965e+06 | 0.069930 | 60.663782 | 3.358607 | 171.578394 | 7.999151 |
| **std** | 12.988182 | 5.643666e+05 | 0.255049 | 18.444933 | 0.459020 | 39.356712 | 1.875885 |
| **min** | 1.000000 | 2.099862e+05 | 0.000000 | -2.060000 | 2.472000 | 126.064000 | 3.879000 |
| **25%** | 12.000000 | 5.533501e+05 | 0.000000 | 47.460000 | 2.933000 | 131.735000 | 6.891000 |
| **50%** | 23.000000 | 9.607460e+05 | 0.000000 | 62.670000 | 3.445000 | 182.616521 | 7.874000 |
| **75%** | 34.000000 | 1.420159e+06 | 0.000000 | 74.940000 | 3.735000 | 212.743293 | 8.622000 |
| **max** | 45.000000 | 3.818686e+06 | 1.000000 | 100.140000 | 4.468000 | 227.232807 | 14.313000 |

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

**1. Which store has maximum sales ?**

max\_sales = data.groupby('Store')['Weekly\_Sales'].sum()

max\_sales.idxmax()

20

#plotting the max sales in the Bar chart

plt.figure(figsize=(15,5))

sns.barplot(x=data.Store, y = data.Weekly\_Sales)

<AxesSubplot:xlabel='Store', ylabel='Weekly\_Sales'>

Store 20 has maximum Sales

**2. Which store has maximum standard deviation i.e., the sales vary a lot. Also, find out the coefficient of mean to standard deviation.**

# maximum Standard deviation

max\_std = data.groupby('Store')['Weekly\_Sales'].std()

max\_std.idxmax()

14

# maximum coefficient of variation

max\_cov = ((data.groupby('Store')['Weekly\_Sales'].std())/(data.groupby('Store')['Weekly\_Sales'].mean()))\*100

max\_cov.idxmax()

35

#plotting the max sales in the Bar chart

stores = data.groupby('Store')

store\_35 = stores.get\_group(35)

plt.figure(figsize=(10,5))

sns.distplot(store\_35.Weekly\_Sales, color='green', label='Weekly Sales for Store 35')

/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Weekly\_Sales', ylabel='Density'>

**Analysis- It is rightly skewed graph which is giving the intutions that sales are centred around 800000 for weekly sales for store 35.**

# Identify Outliers in weekly\_sales for store 35

sns.boxplot(store\_35.Weekly\_Sales, color='cyan') #less outliers

/opt/conda/lib/python3.7/site-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<AxesSubplot:xlabel='Weekly\_Sales'>

Store 14 has maximum standard deviation and store 35 has maximum Coefficient of Variance.

**3. Which store/s has good quarterly growth rate in Q3’2012 ?**

# Grouping data by year and month

growth = data.copy()

growth['Date'] = pd.to\_datetime(growth.Date,format='%d-%m-%Y')

growth['Year'] = growth['Date'].dt.year

growth['Month'] = growth['Date'].dt.month

growth

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

6435 rows × 10 columns

# Group data with year = 2012

growth\_rate = growth.groupby('Year')

growth\_rate\_2012 = growth\_rate.get\_group(2012)

growth\_rate\_2012.head()

|  | **Store** | **Date** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** | **Year** | **Month** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **100** | 1 | 2012-01-06 | 1550369.92 | 0 | 49.01 | 3.157 | 219.714258 | 7.348 | 2012 | 1 |
| **101** | 1 | 2012-01-13 | 1459601.17 | 0 | 48.53 | 3.261 | 219.892526 | 7.348 | 2012 | 1 |
| **102** | 1 | 2012-01-20 | 1394393.84 | 0 | 54.11 | 3.268 | 219.985689 | 7.348 | 2012 | 1 |
| **103** | 1 | 2012-01-27 | 1319325.59 | 0 | 54.26 | 3.290 | 220.078852 | 7.348 | 2012 | 1 |
| **104** | 1 | 2012-02-03 | 1636339.65 | 0 | 56.55 | 3.360 | 220.172015 | 7.348 | 2012 | 2 |

# Getting data for 4 quaters for year 2012

growth\_rate\_2012\_Quaters = growth\_rate\_2012.groupby('Month')

growth\_rate\_2012\_Q1\_1 = growth\_rate\_2012\_Quaters.get\_group(1)

growth\_rate\_2012\_Q1\_2 = growth\_rate\_2012\_Quaters.get\_group(2)

growth\_rate\_2012\_Q1\_3 = growth\_rate\_2012\_Quaters.get\_group(3)

Quater\_1 = growth\_rate\_2012\_Q1\_1.append(growth\_rate\_2012\_Q1\_2)

Quater\_1 = Quater\_1.append(growth\_rate\_2012\_Q1\_3) #Q1 data of 2012

display(Quater\_1.head())

growth\_rate\_2012\_Q2\_4 = growth\_rate\_2012\_Quaters.get\_group(4)

growth\_rate\_2012\_Q2\_5 = growth\_rate\_2012\_Quaters.get\_group(5)

growth\_rate\_2012\_Q2\_6 = growth\_rate\_2012\_Quaters.get\_group(6)

Quater\_2 = growth\_rate\_2012\_Q2\_4.append(growth\_rate\_2012\_Q2\_5)

Quater\_2 = Quater\_2.append(growth\_rate\_2012\_Q2\_6) #Q2 data of 2012

display(Quater\_2.head())

growth\_rate\_2012\_Q3\_7 = growth\_rate\_2012\_Quaters.get\_group(7)

growth\_rate\_2012\_Q3\_8 = growth\_rate\_2012\_Quaters.get\_group(8)

growth\_rate\_2012\_Q3\_9 = growth\_rate\_2012\_Quaters.get\_group(9)

Quater\_3 = growth\_rate\_2012\_Q3\_7.append(growth\_rate\_2012\_Q3\_8)

Quater\_3 = Quater\_3.append(growth\_rate\_2012\_Q3\_9) #Q3 data of 2012

display(Quater\_3.head())

# Q4 data of 2012

growth\_rate\_2012\_Q4\_10 = growth\_rate\_2012\_Quaters.get\_group(10)

Quater\_4 = growth\_rate\_2012\_Q4\_10

display(Quater\_4.head())

|  | **Store** | **Date** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** | **Year** | **Month** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **100** | 1 | 2012-01-06 | 1550369.92 | 0 | 49.01 | 3.157 | 219.714258 | 7.348 | 2012 | 1 |
| **101** | 1 | 2012-01-13 | 1459601.17 | 0 | 48.53 | 3.261 | 219.892526 | 7.348 | 2012 | 1 |
| **102** | 1 | 2012-01-20 | 1394393.84 | 0 | 54.11 | 3.268 | 219.985689 | 7.348 | 2012 | 1 |
| **103** | 1 | 2012-01-27 | 1319325.59 | 0 | 54.26 | 3.290 | 220.078852 | 7.348 | 2012 | 1 |
| **243** | 2 | 2012-01-06 | 1799520.14 | 0 | 46.75 | 3.157 | 219.355063 | 7.057 | 2012 | 1 |
|  | **Store** | **Date** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** | **Year** | **Month** |
| **113** | 1 | 2012-04-06 | 1899676.88 | 0 | 70.43 | 3.891 | 221.435611 | 7.143 | 2012 | 4 |
| **114** | 1 | 2012-04-13 | 1621031.70 | 0 | 69.07 | 3.891 | 221.510210 | 7.143 | 2012 | 4 |
| **115** | 1 | 2012-04-20 | 1521577.87 | 0 | 66.76 | 3.877 | 221.564074 | 7.143 | 2012 | 4 |
| **116** | 1 | 2012-04-27 | 1468928.37 | 0 | 67.23 | 3.814 | 221.617937 | 7.143 | 2012 | 4 |
| **256** | 2 | 2012-04-06 | 2129035.91 | 0 | 68.43 | 3.891 | 221.073764 | 6.891 | 2012 | 4 |

|  | **Store** | **Date** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** | **Year** | **Month** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **126** | 1 | 2012-07-06 | 1769854.16 | 0 | 81.57 | 3.227 | 221.883779 | 6.908 | 2012 | 7 |
| **127** | 1 | 2012-07-13 | 1527014.04 | 0 | 77.12 | 3.256 | 221.924158 | 6.908 | 2012 | 7 |
| **128** | 1 | 2012-07-20 | 1497954.76 | 0 | 80.42 | 3.311 | 221.932727 | 6.908 | 2012 | 7 |
| **129** | 1 | 2012-07-27 | 1439123.71 | 0 | 82.66 | 3.407 | 221.941295 | 6.908 | 2012 | 7 |
| **269** | 2 | 2012-07-06 | 2041507.40 | 0 | 84.20 | 3.227 | 221.521506 | 6.565 | 2012 | 7 |
|  | **Store** | **Date** | **Weekly\_Sales** | **Holiday\_Flag** | **Temperature** | **Fuel\_Price** | **CPI** | **Unemployment** | **Year** | **Month** |
| **139** | 1 | 2012-10-05 | 1670785.97 | 0 | 68.55 | 3.617 | 223.181477 | 6.573 | 2012 | 10 |
| **140** | 1 | 2012-10-12 | 1573072.81 | 0 | 62.99 | 3.601 | 223.381296 | 6.573 | 2012 | 10 |
| **141** | 1 | 2012-10-19 | 1508068.77 | 0 | 67.97 | 3.594 | 223.425723 | 6.573 | 2012 | 10 |
| **142** | 1 | 2012-10-26 | 1493659.74 | 0 | 69.16 | 3.506 | 223.444251 | 6.573 | 2012 | 10 |
| **282** | 2 | 2012-10-05 | 1998321.04 | 0 | 70.27 | 3.617 | 222.815930 | 6.170 | 2012 | 10 |

# Grouping the data "Store" wise each Quarter

df2 = pd.DataFrame(Quater\_1.groupby('Store')['Weekly\_Sales'].sum())

df2["Quater1\_Sales"] = pd.DataFrame(Quater\_1.groupby('Store')['Weekly\_Sales'].sum())

df2["Quater2\_Sales"] = pd.DataFrame(Quater\_2.groupby('Store')['Weekly\_Sales'].sum())

df2["Quater3\_Sales"] = pd.DataFrame(Quater\_3.groupby('Store')['Weekly\_Sales'].sum())

df2["Quater4\_Sales"] = pd.DataFrame(Quater\_4.groupby('Store')['Weekly\_Sales'].sum())

df2.drop('Weekly\_Sales', axis = 1, inplace = True)

df2

|  | **Quater1\_Sales** | **Quater2\_Sales** | **Quater3\_Sales** | **Quater4\_Sales** |
| --- | --- | --- | --- | --- |
| **Store** |  |  |  |  |
| **1** | 20723762.83 | 20978760.12 | 20253947.78 | 6245587.29 |
| **2** | 24528220.70 | 25083604.88 | 24303354.86 | 7581514.93 |
| **3** | 5421809.72 | 5620316.49 | 5298005.47 | 1684307.82 |
| **4** | 27930310.30 | 28454363.67 | 27796792.46 | 8589722.81 |
| **5** | 4237380.83 | 4466363.69 | 4163790.99 | 1301302.62 |
| **6** | 19467939.96 | 20833909.92 | 20167312.24 | 5845884.88 |
| **7** | 7792647.21 | 7290859.27 | 8262787.39 | 2021262.60 |
| **8** | 11869407.28 | 11919630.95 | 11748952.70 | 3695929.20 |
| **9** | 7209983.86 | 7484935.11 | 7022149.56 | 2256961.05 |
| **10** | 24488944.65 | 23750369.17 | 23037258.76 | 6952044.36 |
| **11** | 17713050.31 | 17787371.95 | 17516081.44 | 5167561.98 |
| **12** | 13585746.35 | 13362388.58 | 12536324.37 | 3850386.42 |
| **13** | 25182790.59 | 27009207.14 | 26421259.30 | 8094197.99 |
| **14** | 24476492.09 | 25155535.41 | 21187560.65 | 6621810.11 |
| **15** | 7020781.97 | 7955243.07 | 7612081.03 | 2239424.64 |
| **16** | 6400479.72 | 6564335.98 | 7121541.64 | 2016067.98 |
| **17** | 11460215.01 | 12592400.93 | 12459453.05 | 3773309.64 |
| **18** | 13190110.09 | 13896194.65 | 13489765.27 | 4342506.79 |
| **19** | 17237532.28 | 18367300.24 | 18203554.85 | 5404045.91 |
| **20** | 26971607.79 | 27524197.32 | 26891526.98 | 8440377.29 |
| **21** | 9308307.68 | 9294596.35 | 9027599.32 | 2621383.36 |
| **22** | 12236244.62 | 13487894.06 | 12845139.71 | 4086377.84 |
| **23** | 16049454.58 | 18488882.82 | 18641489.15 | 5588152.20 |
| **24** | 16130180.70 | 17684218.91 | 17976377.72 | 5395618.84 |
| **25** | 8287084.85 | 9323012.09 | 9109081.84 | 2771326.93 |
| **26** | 12071075.04 | 13155335.57 | 13675691.91 | 4074341.77 |
| **27** | 20293011.81 | 22744012.75 | 22307711.41 | 6575320.15 |
| **28** | 17715246.14 | 16506893.13 | 16080704.97 | 5026062.83 |
| **29** | 6361590.26 | 7125307.50 | 6671234.14 | 2086249.72 |
| **30** | 5700327.43 | 5742314.29 | 5594701.86 | 1758306.50 |
| **31** | 18327012.80 | 18267238.50 | 17806714.45 | 5483441.47 |
| **32** | 14596588.10 | 15489271.05 | 15396528.95 | 4798727.90 |
| **33** | 3387560.76 | 3549000.39 | 3433620.36 | 1065369.52 |
| **34** | 12561536.03 | 12853618.02 | 12485995.94 | 3838014.16 |
| **35** | 9642858.59 | 10838313.00 | 11322421.12 | 3434129.81 |
| **36** | 4165563.14 | 4151991.58 | 3831691.64 | 1137224.17 |
| **37** | 6905516.43 | 6824549.37 | 6728068.24 | 2154640.65 |
| **38** | 5645775.74 | 5637918.82 | 5605482.38 | 1741896.51 |
| **39** | 18740604.09 | 20214128.46 | 20715116.23 | 6215814.07 |
| **40** | 11680404.16 | 12727737.53 | 12873195.37 | 3891070.28 |
| **41** | 15681607.44 | 17659942.73 | 18093844.01 | 5452445.75 |
| **42** | 7823655.75 | 7568239.27 | 7296759.34 | 2261705.49 |
| **43** | 8332318.07 | 8168836.35 | 8000572.16 | 2473507.39 |
| **44** | 4109696.37 | 4306405.78 | 4411251.16 | 1360020.41 |
| **45** | 9805267.57 | 10390767.83 | 9581268.38 | 2946326.39 |

# Growth rate formula- ((Present value — Past value )/Past value )\*100

df2['Q3 - Q2'] = df2['Quater3\_Sales'] - df2['Quater2\_Sales']

df2['Overall Growth Rate in 2012 Q3 %'] = (df2['Q3 - Q2']/df2['Quater2\_Sales'])\*100

df2['Overall Growth Rate in 2012 Q3 %'].idxmax() # Store which has good growth in Q3-2012

7

# Plotting the data in Bar chart

plt.figure(figsize=(15,5))

sns.barplot(x=df2.index, y = 'Overall Growth Rate in 2012 Q3 %', data = df2)

<AxesSubplot:xlabel='Store', ylabel='Overall Growth Rate in 2012 Q3 %'>

Store 7 has good growth in Q3-2012

**4. 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.**

#finding the mean sales of non holiday and holiday

data.groupby('Holiday\_Flag')['Weekly\_Sales'].mean()

Holiday\_Flag

0 1.041256e+06

1 1.122888e+06

Name: Weekly\_Sales, dtype: float64

# Marking the holiday dates

data['Date'] = pd.to\_datetime(data['Date'])

Christmas1 = pd.Timestamp(2010,12,31)

Christmas2 = pd.Timestamp(2011,12,30)

Christmas3 = pd.Timestamp(2012,12,28)

Christmas4 = pd.Timestamp(2013,12,27)

Thanksgiving1=pd.Timestamp(2010,11,26)

Thanksgiving2=pd.Timestamp(2011,11,25)

Thanksgiving3=pd.Timestamp(2012,11,23)

Thanksgiving4=pd.Timestamp(2013,11,29)

LabourDay1=pd.Timestamp(2010,9,10)

LabourDay2=pd.Timestamp(2011,9,9)

LabourDay3=pd.Timestamp(2012,9,7)

LabourDay4=pd.Timestamp(2013,9,6)

SuperBowl1=pd.Timestamp(2010,2,12)

SuperBowl2=pd.Timestamp(2011,2,11)

SuperBowl3=pd.Timestamp(2012,2,10)

SuperBowl4=pd.Timestamp(2013,2,8)

#Calculating the mean sales during the holidays

Christmas\_mean\_sales=data[(data['Date'] == Christmas1) | (data['Date'] == Christmas2) | (data['Date'] == Christmas3) | (data['Date'] == Christmas4)]

Thanksgiving\_mean\_sales=data[(data['Date'] == Thanksgiving1) | (data['Date'] == Thanksgiving2) | (data['Date'] == Thanksgiving3) | (data['Date'] == Thanksgiving4)]

LabourDay\_mean\_sales=data[(data['Date'] == LabourDay1) | (data['Date'] == LabourDay2) | (data['Date'] == LabourDay3) | (data['Date'] == LabourDay4)]

SuperBowl\_mean\_sales=data[(data['Date'] == SuperBowl1) | (data['Date'] == SuperBowl2) | (data['Date'] == SuperBowl3) | (data['Date'] == SuperBowl4)]

Christmas\_mean\_sales

list\_of\_mean\_sales = {'Christmas\_mean\_sales' : round(Christmas\_mean\_sales['Weekly\_Sales'].mean(),2),

'Thanksgiving\_mean\_sales': round(Thanksgiving\_mean\_sales['Weekly\_Sales'].mean(),2),

'LabourDay\_mean\_sales' : round(LabourDay\_mean\_sales['Weekly\_Sales'].mean(),2),

'SuperBowl\_mean\_sales':round(SuperBowl\_mean\_sales['Weekly\_Sales'].mean(),2),

'Non holiday weekly sales' : round(data[data['Holiday\_Flag'] == 0 ]['Weekly\_Sales'].mean(),2)}

list\_of\_mean\_sales

{'Christmas\_mean\_sales': 960833.11,

'Thanksgiving\_mean\_sales': 1471273.43,

'LabourDay\_mean\_sales': 1039182.83,

'SuperBowl\_mean\_sales': nan,

'Non holiday weekly sales': 1041256.38}

"Thanksgiving Day" has much high sale than mean sales in Non-Holiday season.

**5. Provide a monthly and semester view of sales in units and give insights**

#Monthly sales

monthly = data.groupby(pd.Grouper(key='Date', freq='1M')).sum() # groupby each 1 month

monthly=monthly.reset\_index()

fig, ax = plt.subplots(figsize=(10,5))

X = monthly['Date']

Y = monthly['Weekly\_Sales']

plt.plot(X,Y)

plt.title('Month Wise Sales')

plt.xlabel('Monthly')

plt.ylabel('Weekly\_Sales')

# Analysis- highest sum of sales is recorded in between jan-2011 to march-2011.

Text(0, 0.5, 'Weekly\_Sales')

#Semester Sales

Semester = data.groupby(pd.Grouper(key='Date', freq='6M')).sum()

Semester = Semester.reset\_index()

fig, ax = plt.subplots(figsize=(10,5))

X = Semester['Date']

Y = Semester['Weekly\_Sales']

plt.plot(X,Y)

plt.title('Semester Wise Sales')

plt.xlabel('Semester')

plt.ylabel('Weekly\_Sales')

# ANalysis- sales are lowest in beginning of 1st sem of 2010 and 1st sem of 2013

Text(0, 0.5, 'Weekly\_Sales')

**For Store 1 – Build prediction models to forecast demand**

Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010 (starting from the earliest date in order). Hypothesize if CPI, unemployment, and fuel price have any impact on sales.

hypothesis = growth.groupby('Store')[['Fuel\_Price','Unemployment', 'CPI','Weekly\_Sales', 'Holiday\_Flag']]

factors = hypothesis.get\_group(1) #Filter by Store 1

day\_arr = [1]

for i in range (1,len(factors)):

day\_arr.append(i\*7)

factors['Day'] = day\_arr.copy()

factors

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:7: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

import sys

|  | **Fuel\_Price** | **Unemployment** | **CPI** | **Weekly\_Sales** | **Holiday\_Flag** | **Day** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 2.572 | 8.106 | 211.096358 | 1643690.90 | 0 | 1 |
| **1** | 2.548 | 8.106 | 211.242170 | 1641957.44 | 1 | 7 |
| **2** | 2.514 | 8.106 | 211.289143 | 1611968.17 | 0 | 14 |
| **3** | 2.561 | 8.106 | 211.319643 | 1409727.59 | 0 | 21 |
| **4** | 2.625 | 8.106 | 211.350143 | 1554806.68 | 0 | 28 |
| **...** | ... | ... | ... | ... | ... | ... |
| **138** | 3.666 | 6.908 | 222.981658 | 1437059.26 | 0 | 966 |
| **139** | 3.617 | 6.573 | 223.181477 | 1670785.97 | 0 | 973 |
| **140** | 3.601 | 6.573 | 223.381296 | 1573072.81 | 0 | 980 |
| **141** | 3.594 | 6.573 | 223.425723 | 1508068.77 | 0 | 987 |
| **142** | 3.506 | 6.573 | 223.444251 | 1493659.74 | 0 | 994 |

143 rows × 6 columns

sns.heatmap(factors.corr(), annot = True)

<AxesSubplot:>

Few variables which are positive and have value greater than zero are correlated with Weekly\_Sales. We can also see CPI and Holiday\_Flag is fairly strongly correlated to Weekly\_Sales. Holiday\_Flag = 1 means it's holiday\_week we have sales more than the non\_holiday\_weeks.

sns.lmplot(x='Fuel\_Price', y = 'Unemployment', data = factors)

#plt.figure()

sns.lmplot(x='CPI', y = 'Unemployment', data = factors)

<seaborn.axisgrid.FacetGrid at 0x7f61f16ebad0>

As the Fuel\_price and Cpi goes high, rate of Unemployment Fairly Decreases (shown above in Line Regression plot).