Walmart_Project

June 9, 2023

1 Retail Analysis with Walmart Data

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

[2]:
```
pd.set_option('display.max_columns',None)
data=pd.read_csv('Walmart_Store_sales.csv')
```

[3]:
```
data.shape
```

[3]:
```
(6435, 8)
```

[4]:
```
holiday=pd.read_csv('Holiday.csv')
from datetime import date,time,datetime
holiday["Holiday"]=holiday["Day"].astype(str)+"-"+holiday["Month"].
 ↪astype(str)+"-"+holiday["Year"].astype(str)
holiday["Holiday"]=pd.to_datetime(holiday["Holiday"],format="%d-%m-%Y")
holiday=holiday.loc[:,["Event","Holiday"]]
holiday.head()
```

[4]:
```
         Event    Holiday
0  Super Bowl 2010-02-12
1  Super Bowl 2011-02-11
2  Super Bowl 2012-02-10
3  Super Bowl 2013-02-08
4  Labour Day 2010-09-10
```

[5]:
```
from datetime import date,time,datetime
data["Date"]=pd.to_datetime(data["Date"],format="%d-%m-%Y")
```

[6]:
```
data["Holiday"]=data["Date"]
```

[7]:
```
d=pd.merge(data,holiday,on="Holiday",how="left")
d.head()
```

[7]:
```
   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    Holiday       Event
0  211.096358         8.106 2010-02-05         NaN
1  211.242170         8.106 2010-02-12  Super Bowl
2  211.289143         8.106 2010-02-19         NaN
3  211.319643         8.106 2010-02-26         NaN
4  211.350143         8.106 2010-03-05         NaN
```

[8]:
```
d.tail()
```

[8]:
```
      Store       Date  Weekly_Sales  Holiday_Flag  Temperature  Fuel_Price  \
6430     45 2012-09-28     713173.95             0        64.88       3.997
6431     45 2012-10-05     733455.07             0        64.89       3.985
6432     45 2012-10-12     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    Holiday Event
6430  192.013558         8.684 2012-09-28   NaN
6431  192.170412         8.667 2012-10-05   NaN
6432  192.327265         8.667 2012-10-12   NaN
6433  192.330854         8.667 2012-10-19   NaN
6434  192.308899         8.667 2012-10-26   NaN
```

[9]:
```
# above shows the historical data that covers sales from 2010-02-05 to␣
↪2012-11-01, in the file Walmart_Store_sales.
```

[10]:
```
d.shape
```

[10]:
```
(6435, 10)
```

[11]:
```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6435 entries, 0 to 6434
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   datetime64[ns]
 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
 8   Holiday       6435 non-null   datetime64[ns]
 9   Event         315 non-null    object
dtypes: datetime64[ns](2), float64(5), int64(2), object(1)
memory usage: 553.0+ KB
```

[12]:
```
d.columns
```

[12]:
```
Index(['Store', 'Date', 'Weekly_Sales', 'Holiday_Flag', 'Temperature',
       'Fuel_Price', 'CPI', 'Unemployment', 'Holiday', 'Event'],
      dtype='object')
```

[13]:
```
d["Holiday_Flag"]=np.where(d["Event"].isnull(),"No","Yes")
#d.drop(["Holiday"],axis=1,inplace=True)
d.drop(["Holiday"],axis=1,inplace=True)
```

[14]:
```
d.head()
```

[14]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Event
0	1	2010-02-05	1643690.90	No	42.31	2.572	211.096358	8.106	NaN
1	1	2010-02-12	1641957.44	Yes	38.51	2.548	211.242170	8.106	Super Bowl
2	1	2010-02-19	1611968.17	No	39.93	2.514	211.289143	8.106	NaN
3	1	2010-02-26	1409727.59	No	46.63	2.561	211.319643	8.106	NaN
4	1	2010-03-05	1554806.68	No	46.50	2.625	211.350143	8.106	NaN

[15]:
```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6435 entries, 0 to 6434
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Date          6435 non-null   datetime64[ns]
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   object
 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
 8   Event         315 non-null    object
dtypes: datetime64[ns](1), float64(5), int64(1), object(2)
memory usage: 502.7+ KB
```

1.1 Which store has Maximum Sales?

[16]:
```
store = d.groupby("Store")["Weekly_Sales"].sum()
store = pd.DataFrame(store)
store = store.sort_values("Weekly_Sales",ascending=0)
store.head(5)
# Store No. 20 has maximum sales
```

[16]:

Store	Weekly_Sales
20	3.013978e+08
4	2.995440e+08
14	2.889999e+08
13	2.865177e+08
2	2.753824e+08

1.2 Which store has maximum standard deviatiion?

[17]:
```
store = d.groupby("Store")["Weekly_Sales"].std()
store = pd.DataFrame(store)
store = store.sort_values("Weekly_Sales",ascending=0)
store.head(5)
# Store No. 14 has maximum standard deviation
```

[17]:

Store	Weekly_Sales
14	317569.949476
10	302262.062504
20	275900.562742
4	266201.442297
13	265506.995776

1.3 Which store/s has good quarterly growth rate in Q3'2012?

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

[18]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	\
0	1	2010-02-05	1643690.90	No	42.31	2.572	
1	1	2010-02-12	1641957.44	Yes	38.51	2.548	
2	1	2010-02-19	1611968.17	No	39.93	2.514	

```
3      1 2010-02-26    1409727.59           No        46.63       2.561
4      1 2010-03-05    1554806.68           No        46.50       2.625

           CPI  Unemployment       Event
0  211.096358         8.106         NaN
1  211.242170         8.106  Super Bowl
2  211.289143         8.106         NaN
3  211.319643         8.106         NaN
4  211.350143         8.106         NaN
```

[19]:
```
d.columns
```

[19]:
```
Index(['Store', 'Date', 'Weekly_Sales', 'Holiday_Flag', 'Temperature',
       'Fuel_Price', 'CPI', 'Unemployment', 'Event'],
      dtype='object')
```

[20]:
```
import datetime as dt
import pandas as pd
d["qtr"]=d["Date"].dt.to_period('Q')
```

[21]:
```
d["qtr"].unique()
```

[21]:
```
<PeriodArray>
['2010Q1', '2010Q2', '2010Q3', '2010Q4', '2011Q1', '2011Q2', '2011Q3',
 '2011Q4', '2012Q1', '2012Q2', '2012Q3', '2012Q4']
Length: 12, dtype: period[Q-DEC]
```

[22]:
```
joyita=d.loc[(d["qtr"]=="2012Q2")|(d["qtr"]=="2012Q3")]
```

[28]:
```
joyita.head(20)
```

[28]:
```
     Store       Date  Weekly_Sales Holiday_Flag  Temperature  Fuel_Price  \
113      1 2012-04-06    1899676.88           No        70.43       3.891
114      1 2012-04-13    1621031.70           No        69.07       3.891
115      1 2012-04-20    1521577.87           No        66.76       3.877
116      1 2012-04-27    1468928.37           No        67.23       3.814
117      1 2012-05-04    1684519.99           No        75.55       3.749
118      1 2012-05-11    1611096.05           No        73.77       3.688
119      1 2012-05-18    1595901.87           No        70.33       3.630
120      1 2012-05-25    1555444.55           No        77.22       3.561
121      1 2012-06-01    1624477.58           No        77.95       3.501
122      1 2012-06-08    1697230.96           No        78.30       3.452
123      1 2012-06-15    1630607.00           No        79.35       3.393
124      1 2012-06-22    1527845.81           No        78.39       3.346
125      1 2012-06-29    1540421.49           No        84.88       3.286
126      1 2012-07-06    1769854.16           No        81.57       3.227
127      1 2012-07-13    1527014.04           No        77.12       3.256
```

```
128      1 2012-07-20    1497954.76          No        80.42       3.311
129      1 2012-07-27    1439123.71          No        82.66       3.407
130      1 2012-08-03    1631135.79          No        86.11       3.417
131      1 2012-08-10    1592409.97          No        85.05       3.494
132      1 2012-08-17    1597868.05          No        84.85       3.571

             CPI  Unemployment Event     qtr
113  221.435611         7.143   NaN  2012Q2
114  221.510210         7.143   NaN  2012Q2
115  221.564074         7.143   NaN  2012Q2
116  221.617937         7.143   NaN  2012Q2
117  221.671800         7.143   NaN  2012Q2
118  221.725663         7.143   NaN  2012Q2
119  221.742674         7.143   NaN  2012Q2
120  221.744944         7.143   NaN  2012Q2
121  221.747214         7.143   NaN  2012Q2
122  221.749484         7.143   NaN  2012Q2
123  221.762642         7.143   NaN  2012Q2
124  221.803021         7.143   NaN  2012Q2
125  221.843400         7.143   NaN  2012Q2
126  221.883779         6.908   NaN  2012Q3
127  221.924158         6.908   NaN  2012Q3
128  221.932727         6.908   NaN  2012Q3
129  221.941295         6.908   NaN  2012Q3
130  221.949864         6.908   NaN  2012Q3
131  221.958433         6.908   NaN  2012Q3
132  222.038411         6.908   NaN  2012Q3
```

[24]:
```
j=pd.
 ↪pivot_table(joyita,index="Store",columns="qtr",values="Weekly_Sales",aggfunc=np.
 ↪sum)
j["growth"]=((j["2012Q3"]-j["2012Q2"])/j["2012Q2"])*100
j=j.sort_values("growth",ascending=0)
j.head(10)
# store 7 and 16 has good growth rate of 13.3% and 8.4% respectively
```

[24]:
```
qtr         2012Q2       2012Q3     growth
Store
7       7290859.27   8262787.39  13.330776
16      6564335.98   7121541.64   8.488378
35     10838313.00  11322421.12   4.466637
26     13155335.57  13675691.91   3.955478
39     20214128.46  20715116.23   2.478404
41     17659942.73  18093844.01   2.456980
44      4306405.78   4411251.16   2.434638
24     17684218.91  17976377.72   1.652088
40     12727737.53  12873195.37   1.142841
```

```
23     18488882.82  18641489.15   0.825395
```

1.4 Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together.

[30]:
```
d.head()
```

[30]:
```
   Store       Date  Weekly_Sales Holiday_Flag  Temperature  Fuel_Price  \
0      1 2010-02-05    1643690.90           No        42.31       2.572
1      1 2010-02-12    1641957.44          Yes        38.51       2.548
2      1 2010-02-19    1611968.17           No        39.93       2.514
3      1 2010-02-26    1409727.59           No        46.63       2.561
4      1 2010-03-05    1554806.68           No        46.50       2.625

           CPI  Unemployment       Event     qtr
0  211.096358         8.106         NaN  2010Q1
1  211.242170         8.106  Super Bowl  2010Q1
2  211.289143         8.106         NaN  2010Q1
3  211.319643         8.106         NaN  2010Q1
4  211.350143         8.106         NaN  2010Q1
```

[31]:
```
d.columns
```

[31]:
```
Index(['Store', 'Date', 'Weekly_Sales', 'Holiday_Flag', 'Temperature',
       'Fuel_Price', 'CPI', 'Unemployment', 'Event', 'qtr'],
      dtype='object')
```

[32]:
```
d.shape
```

[32]:
```
(6435, 10)
```

[37]:
```
# Mean sales in non-holiday for all stores
mean_sales=d.loc[d["Holiday_Flag"]=="No","Weekly_Sales"].mean()
```

[34]:
```
ealina=d.groupby("Event")["Weekly_Sales"].mean()
ealina=pd.DataFrame(ealina)
ealina
```

[34]:
```
              Weekly_Sales
Event
Christmas     9.608331e+05
Labour Day    1.014098e+06
Super Bowl    1.079128e+06
ThanksGiving  1.462689e+06
```

[36]:
```
ealina.loc[ealina["Weekly_Sales"]>mean_sales,]
# ThanksGiving is having higher mean sales than average non-holiday sales
```

[36]:
```
              Weekly_Sales
Event
Super Bowl    1.079128e+06
ThanksGiving  1.462689e+06
```

1.5 Provide a monthly and semester view of sales in units and give insights

[38]:
```
d.head()
```

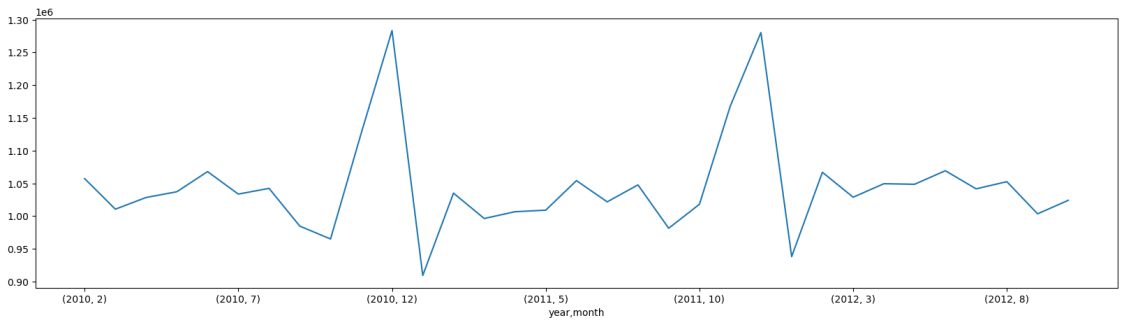
[38]:
```
   Store       Date  Weekly_Sales Holiday_Flag  Temperature  Fuel_Price  \
0      1 2010-02-05    1643690.90           No        42.31       2.572
1      1 2010-02-12    1641957.44          Yes        38.51       2.548
2      1 2010-02-19    1611968.17           No        39.93       2.514
3      1 2010-02-26    1409727.59           No        46.63       2.561
4      1 2010-03-05    1554806.68           No        46.50       2.625

           CPI  Unemployment       Event     qtr
0  211.096358         8.106         NaN  2010Q1
1  211.242170         8.106  Super Bowl  2010Q1
2  211.289143         8.106         NaN  2010Q1
3  211.319643         8.106         NaN  2010Q1
4  211.350143         8.106         NaN  2010Q1
```

[41]:
```
import datetime as dt
import pandas as pd
d["month"]=d["Date"].dt.month
d["year"]=d["Date"].dt.year
d["semester"]=np.where(d["Date"].dt.month.le(6),'H1','H2')
#assuming semester is half of a year
```

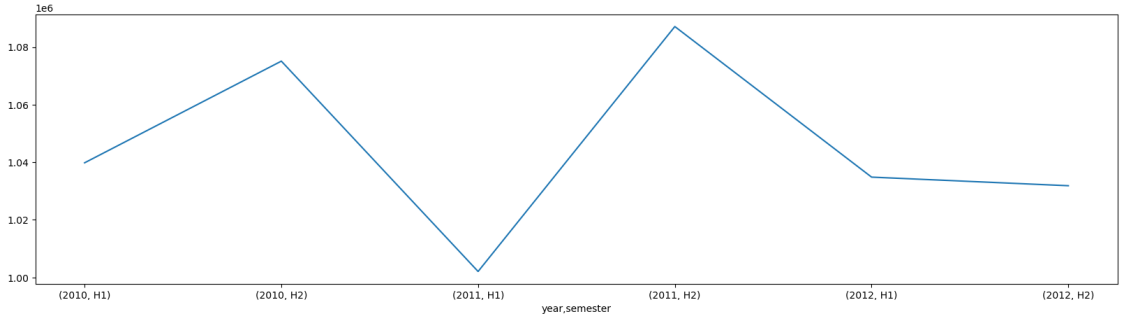
[54]:
```
#d.groupby("month")["Weekly_Sales"].sum()
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"]=(20,5)
d.groupby(["year","month"])["Weekly_Sales"].mean().plot()
# clear spikes are visible for 2 period
```

[54]:
```
<Axes: xlabel='year,month'>
```



[46]:
```
plt.rcParams["figure.figsize"]=(20,5)
d.groupby(["year","semester"])["Weekly_Sales"].mean().plot()
#seems like a seasonal pattern
```

[46]:
```
<Axes: xlabel='year,semester'>
```



1.6 Linear Regression

[47]:
```
d.head()
```

[47]:
```
   Store       Date  Weekly_Sales Holiday_Flag  Temperature  Fuel_Price  \
0      1 2010-02-05    1643690.90           No        42.31       2.572
1      1 2010-02-12    1641957.44          Yes        38.51       2.548
2      1 2010-02-19    1611968.17           No        39.93       2.514
3      1 2010-02-26    1409727.59           No        46.63       2.561
4      1 2010-03-05    1554806.68           No        46.50       2.625

           CPI  Unemployment       Event     qtr  month  year semester
0  211.096358         8.106         NaN  2010Q1      2  2010       H1
1  211.242170         8.106  Super Bowl  2010Q1      2  2010       H1
2  211.289143         8.106         NaN  2010Q1      2  2010       H1
3  211.319643         8.106         NaN  2010Q1      2  2010       H1
4  211.350143         8.106         NaN  2010Q1      3  2010       H1
```

[48]:
```
from sklearn import preprocessing
le=preprocessing.LabelEncoder()
d["new_date"]=d["Date"]
d.new_date=le.fit_transform(d.new_date)+1
d.head
```

[48]:
```
<bound method NDFrame.head of       Store       Date  Weekly_Sales Holiday_Flag
Temperature  Fuel_Price  \
```

```
0          1 2010-02-05    1643690.90           No        42.31       2.572
1          1 2010-02-12    1641957.44          Yes        38.51       2.548
2          1 2010-02-19    1611968.17           No        39.93       2.514
3          1 2010-02-26    1409727.59           No        46.63       2.561
4          1 2010-03-05    1554806.68           No        46.50       2.625
…      …         …             …          …          …         …
6430      45 2012-09-28     713173.95           No        64.88       3.997
6431      45 2012-10-05     733455.07           No        64.89       3.985
6432      45 2012-10-12     734464.36           No        54.47       4.000
6433      45 2012-10-19     718125.53           No        56.47       3.969
6434      45 2012-10-26     760281.43           No        58.85       3.882

              CPI  Unemployment       Event     qtr  month  year semester  \
0      211.096358         8.106         NaN  2010Q1      2  2010       H1
1      211.242170         8.106  Super Bowl  2010Q1      2  2010       H1
2      211.289143         8.106         NaN  2010Q1      2  2010       H1
3      211.319643         8.106         NaN  2010Q1      2  2010       H1
4      211.350143         8.106         NaN  2010Q1      3  2010       H1
…           …            …          …      …    …   …       …
6430   192.013558         8.684         NaN  2012Q3      9  2012       H2
6431   192.170412         8.667         NaN  2012Q4     10  2012       H2
6432   192.327265         8.667         NaN  2012Q4     10  2012       H2
6433   192.330854         8.667         NaN  2012Q4     10  2012       H2
6434   192.308899         8.667         NaN  2012Q4     10  2012       H2

      new_date
0            1
1            2
2            3
3            4
4            5
…        …
6430       139
6431       140
6432       141
6433       142
6434       143

[6435 rows x 14 columns]>
```

[49]:
```
import statsmodels.formula.api as sm
rock=sm.ols(formula=
"Weekly_Sales ~ CPI +  Unemployment + Fuel_Price ",data=data).fit()
rock.summary()# shows total summary
```

[49]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                            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:                Fri, 09 Jun 2023   Prob (F-statistic):           4.81e-33
Time:                        10:24:54   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
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
Fuel_Price   -1.927e+04   1.54e+04     -1.248      0.212   -4.95e+04     1.1e+04
==============================================================================
Omnibus:                      370.117   Durbin-Watson:                   0.112
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
==============================================================================

Notes:
[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.
"""
```

[50]:
```
#predicting the outcomes
data["pred"] = rock.predict()
var = pd.DataFrame(round(rock.pvalues,3))# shows p value
rock.rsquared
var["coeff"] = rock.params#coefficients

from statsmodels.stats.outliers_influence import variance_inflation_factor
variables = rock.model.exog #.if I had saved data as rock
# this it would have looked like rock.model.exog
vif = [variance_inflation_factor(variables, i) for i in range(variables.
 ↪shape[1])]
vif
var["vif"] = vif
var
```

[50]:

	0	coeff	vif
Intercept	0.000	1.745657e+06	130.951193
CPI	0.000	-1.696876e+03	1.141629
Unemployment	0.000	-4.285920e+04	1.109722
Fuel_Price	0.212	-1.926614e+04	1.038744

[57]:
```
###### mape
data["mp"] = abs((data["Weekly_Sales"] - data["pred"])/data["Weekly_Sales"])
(data.mp.mean())*100 #mape
```

[57]: 66.3671994641876

[ ]: