

Walmart Project

```
In [89]: #Sun 14 May 2023 09:11:09 PM EDT original date  
Date
```

Sun 25 Aug 2024 03:20:11 PM EDT

```
In [1]: %time
```

CPU times: user 5 µs, sys: 1 µs, total: 6 µs
Wall time: 14.8 µs

Import Libraries

```
In [101]:
```

```
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import cufflinks as cf  
%matplotlib inline
```

Using Pandas to read the csv file

```
In [102]:
```

```
wa = pd.read_csv('/home/harlohutch77/python/python_master/Walmart.csv')
```

Walmart Statistics

```
In [103]:
```

```
# 6435 entries 8 columns  
wa.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
```

```
In [105]:
```

```
#This gives a description of the dataset  
wa.describe()
```

```
Out[105]:
```

	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

```
In [106]:
```

```
#This fills in any na with a 0  
wa.fillna(0)
```

```
Out[106]:
```

	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
...
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	8.684
6431	45	05-10-2012	733455.07	0	64.89	3.985	192.170412	8.667
6432	45	12-10-2012	734464.36	0	54.47	4.000	192.327265	8.667
6433	45	19-10-2012	718125.53	0	56.47	3.969	192.330854	8.667
6434	45	26-10-2012	760281.43	0	58.85	3.882	192.308899	8.667

6435 rows × 8 columns

```
In [112]:
```

```
print('Temperature median:',wa['Temperature'].median())  
print('Unemployment minimum percent:',wa['Unemployment'].min())  
print('Unemployment median percent:',wa['Unemployment'].median())  
print('Unemployment maximum percent:',wa['Unemployment'].max())
```

Temperature median: 62.67

Unemployment minimum percent: 3.879

Unemployment median percent: 7.874

Unemployment maximum percent: 14.313

```
In [145]:
```

```
#How many stores are there, fuel price from minimum to maximum, and weekly sales from minimum to maximum  
print('Store maximum:',wa['Store'].max())  
print('Fuel Price minimum: $',wa['Fuel_Price'].min())  
print('Fuel Price maximum: $',wa['Fuel_Price'].max())  
print('Weekly Sales minimum: $',wa['Weekly_Sales'].min())  
print('Weekly Sales median: $',wa['Weekly_Sales'].median())  
print('Weekly Sales maximum: $',wa['Weekly_Sales'].max())
```

Store maximum: 45

Fuel Price minimum: \$ 2.472

Fuel Price maximum: \$ 4.468

Weekly Sales minimum: \$ 209986.25

Weekly Sales median: \$ 960746.04

Weekly Sales maximum: \$ 3818686.45

```
In [117]:
```

```
#Consumer Price Index - CPI
```

```
print('CPI minimum:',wa['CPI'].min())
```

```
print('CPI median:',wa['CPI'].median())
```

```
print('CPI maximum:',wa['CPI'].max())
```

CPI minimum: 126.064

CPI median: 182.6165205

CPI maximum: 227.2328068

```
In [151]:
```

```
#450 stores that had holiday_flag
```

```
wa[wa['Holiday_Flag']==1].sum()
```

```
Out[151]:
```

Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
10350	12-02-2010-09-2010-10-11-2010-31-12-2010-11-02-...	505299551.56	450	22604.42	1452.359	77070.641224	3633.357

```
In [126]:
```

```
wa[wa['Fuel_Price']==2.472][['Store', 'CPI']]
```

```
Out[126]:
```

Store	CPI
5007	210.045102

```
In [127]:
```

```
wa[wa['Fuel_Price']==4.468][['Store', 'CPI']]
```

```
Out[127]:
```

Store	CPI
1427	10 131.108333
1713	12 131.108333
4001	28 131.108333
4716	33 131.108333
5431	38 131.108333
6003	42 131.108333

```
In [9]:
```

```
wa[['Store','Date','Fuel_Price','Weekly_Sales','Unemployment']]
```

```
Out[9]:
```

Store	Date	Fuel_Price	Weekly_Sales	Unemployment
0	1 05-02-2010	2.572	1643690.90	8.106
1	1 12-02-2010	2.548	1641957.44	8.106
2	1 19-02-2010	2.514	1611968.17	8.106
3	1 26-02-2010	2.561	1409727.59	8.106
4	1 05-03-2010	2.625	1554806.68	8.106
...
6430	45 28-09-2012	3.997	713173.95	8.684
6431	45 05-10-2012	3.985	733455.07	8.667
6432	45 12-10-2012	4.000	734464.36	8.667
6433	45 19-10-2012	3.969	718125.53	8.667
6434	45 26-10-2012	3.882	760281.43	8.667

6435 rows × 5 columns

Cufflinks Visualizations

```
In [129]:
```

```
#this import assists with cufflinks  
from plotly import __version__  
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

```
print(__version__) # requires version >= 1.9.0
```

5.7.0

```
In [130]:
```

```
# This is for Jupyter Notebook  
init_notebook_mode(connected=True)
```

```
In [13]:
```

```
#For offline use with Jupyter Notebook  
cf.go_offline()
```

```
In [153]:
```

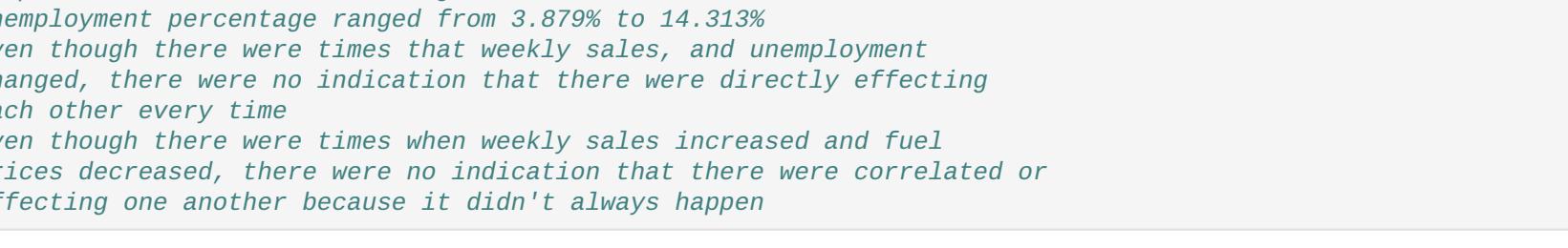
```
wa[['Store','Fuel_Price','Weekly_Sales']].iplot(kind='box', title='Walmart Inventory')
```



Export to plotly »

```
In [152]:
```

```
wa[['Store','Date','Weekly_Sales']].iplot(kind='scatter', title='Walmart Inventory', bins=25)
```



Export to plotly »

```
In [144]:
```

```
sns.jointplot(x='Weekly_Sales',y='Fuel_Price',data=wa)
```

```
<seaborn.axisgrid.JointGrid at 0x7f59bfecc2e0>
```

```
print('CPI minimum:',wa['CPI'].min())
```

```
print('CPI median:',wa['CPI'].median())
```

```
print('CPI maximum:',wa['CPI'].max())
```

CPI minimum: 126.064

CPI median: 182.6165205

CPI maximum: 227.2328068

```
print('Temperature minimum:',wa['Temperature'].min())
```

```
print('Temperature median:',wa['Temperature'].median())
```

```
print('Temperature maximum:',wa['Temperature'].max())
```

Temperature minimum