DATA SCIENCE METHODOLOGY

Supervised By: Dr. Huda Hemdan

- Ali Mahmoud Ali Mahmoud Student Id: 20221370972
- Mohammed Ahmed Elsayed Student Id: 20221370982
- Omar Abdulkader Mohammed Student Id: 20221311487

Ali Mahmoud Ali Mahmoud

First we import pandas,numpy,matplotlib and seaborn.

Read csv file and print it.

```
In [82]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns
         import datetime as dt

In [ ]:

In [53]: dfk=pd.read_csv("E:\walmart-sales-dataset-of-45stores.csv")
         print(dfk.to_string())
```

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
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106
6	1	19-03-2010	1472515.79	0	54.58	2.720	211.215635	8.106
7	1	26-03-2010	1404429.92	0	51.45	2.732	211.018042	8.106
8	1	02-04-2010	1594968.28	0	62.27	2.719	210.820450	7.808
9	1	09-04-2010	1545418.53	0	65.86	2.770	210.622857	7.808
10	1	16-04-2010	1466058.28	0	66.32	2.808	210.488700	7.808
11	1	23-04-2010	1391256.12	0	64.84	2.795	210.439123	7.808
12	1	30-04-2010	1425100.71	0	67.41	2.780	210.389546	7.808
13	1	07-05-2010	1603955.12	0	72.55	2.835	210.339968	7.808
14	1	14-05-2010	1494251.50	0	74.78	2.854	210.337426	7.808
15	1	21-05-2010	1399662.07	0	76.44	2.826	210.617093	7.808
16	1	28-05-2010	1432069.95	0	80.44	2.759	210.896761	7.808
17	1	04-06-2010	1615524.71	0	80.69	2.705	211.176428	7.808
18	1	11-06-2010	1542561.09	0	80.43	2.668	211.456095	7.808

Then,we replace every “0” in “Holiday_flag “column with “Non_holiday_week”, and similary with “1” “Holiday_week”

```
In [54]: for x in dfk.index:
             if dfk.loc[x,"Holiday_Flag"] > 0:
                 dfk.loc[x,"Holiday_Flag"] = "Holiday_week"
             elif dfk.loc[x,"Holiday_Flag"] < 1:
                 dfk.loc[x,"Holiday_Flag"] = "Non_holiday_week"
         print(dfk.to_string())
```

```
      Store        Date  Weekly_Sales      Holiday_Flag  Temperature  Fuel_Price         CPI  Unemployment
0         1  05-02-2010   1643690.90  Non_holiday_week        42.31       2.572  211.096358         8.106
1         1  12-02-2010   1641957.44      Holiday_week        38.51       2.548  211.242170         8.106
2         1  19-02-2010   1611968.17  Non_holiday_week        39.93       2.514  211.289143         8.106
3         1  26-02-2010   1409727.59  Non_holiday_week        46.63       2.561  211.319643         8.106
4         1  05-03-2010   1554806.68  Non_holiday_week        46.50       2.625  211.350143         8.106
5         1  12-03-2010   1439541.59  Non_holiday_week        57.79       2.667  211.380643         8.106
6         1  19-03-2010   1472515.79  Non_holiday_week        54.58       2.720  211.215635         8.106
7         1  26-03-2010   1404429.92  Non_holiday_week        51.45       2.732  211.018042         8.106
8         1  02-04-2010   1594968.28  Non_holiday_week        62.27       2.719  210.820450         7.808
9         1  09-04-2010   1545418.53  Non_holiday_week        65.86       2.770  210.622857         7.808
10        1  16-04-2010   1466058.28  Non_holiday_week        66.32       2.808  210.488700         7.808
11        1  23-04-2010   1391256.12  Non_holiday_week        64.84       2.795  210.439123         7.808
12        1  30-04-2010   1425100.71  Non_holiday_week        67.41       2.780  210.389546         7.808
13        1  07-05-2010   1603955.12  Non_holiday_week        72.55       2.835  210.339968         7.808
14        1  14-05-2010   1494251.50  Non_holiday_week        74.78       2.854  210.337426         7.808
15        1  21-05-2010   1399662.07  Non_holiday_week        76.44       2.826  210.617093         7.808
16        1  28-05-2010   1432069.95  Non_holiday_week        80.44       2.759  210.896761         7.808
17        1  04-06-2010   1615524.71  Non_holiday_week        80.69       2.705  211.176428         7.808
```

We use describe function().

```
In [181]: desk=dfk.describe()
          desk
```

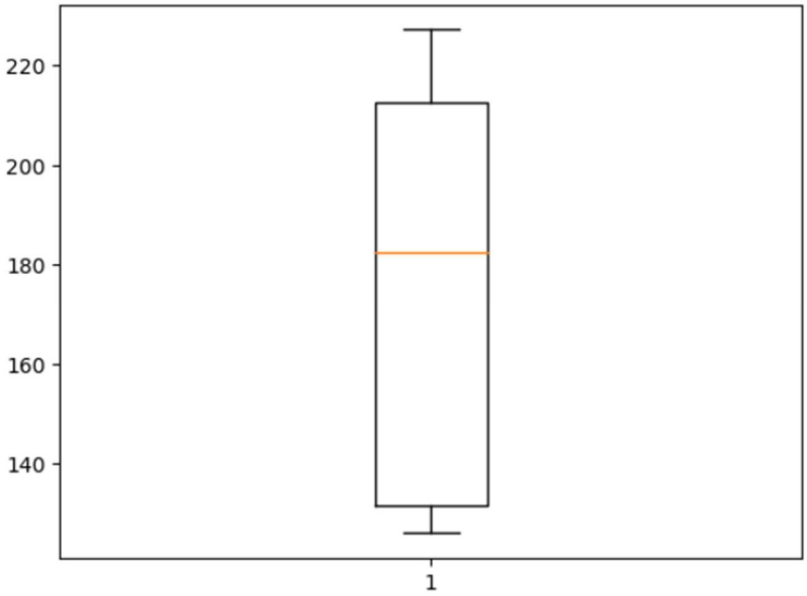
Out[181]:

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

We use boxplot as it is better for visualizing columns:

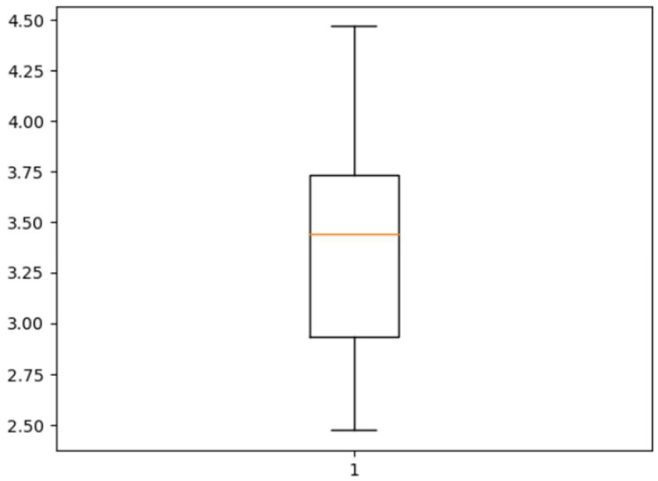
```
In [185]: plt.boxplot(dfk['CPI'])

Out[185]: {'whiskers': [<matplotlib.lines.Line2D at 0x1df8e1c8a00>,
           <matplotlib.lines.Line2D at 0x1df8e1c8cd0>],
          'caps': [<matplotlib.lines.Line2D at 0x1df8e1c8fa0>,
           <matplotlib.lines.Line2D at 0x1df8e1d72b0>],
          'boxes': [<matplotlib.lines.Line2D at 0x1df8e1c8700>],
          'medians': [<matplotlib.lines.Line2D at 0x1df8e1d7580>],
          'fliers': [<matplotlib.lines.Line2D at 0x1df8e1d7850>],
          'means': []}
```



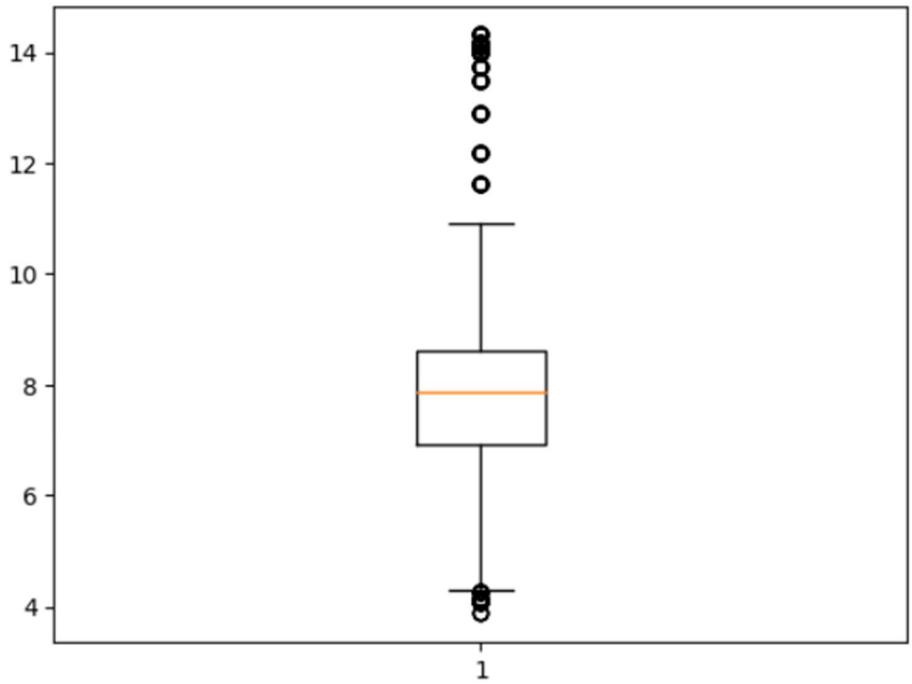
In [186]:
```
plt.boxplot(dfk['Fuel_Price'])
```

Out[186]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1df8e230520>,
  <matplotlib.lines.Line2D at 0x1df8e230820>],
 'caps': [<matplotlib.lines.Line2D at 0x1df8e230af0>,
  <matplotlib.lines.Line2D at 0x1df8e230dc0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1df8e230250>],
 'medians': [<matplotlib.lines.Line2D at 0x1df8e23e0d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1df8e23e3a0>],
 'means': []}
```



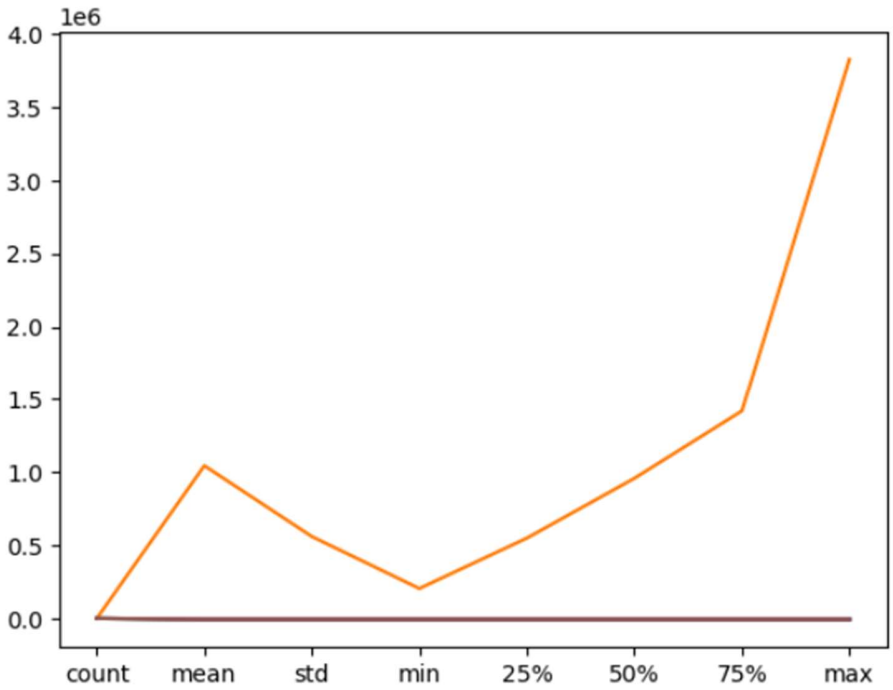
In [188]:
```
plt.boxplot(dfk['Unemployment'])
```

Out[188]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1df8e47b7c0>,
  <matplotlib.lines.Line2D at 0x1df8e47bac0>],
 'caps': [<matplotlib.lines.Line2D at 0x1df8e47bd90>,
  <matplotlib.lines.Line2D at 0x1df8e30a0a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1df8e47b4f0>],
 'medians': [<matplotlib.lines.Line2D at 0x1df8e30a3a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1df8e30a670>],
 'means': []}
```



```
In [177]: plt.plot(desk)
```

```
Out[177]: [<matplotlib.lines.Line2D at 0x1df8d0f3490>,
 <matplotlib.lines.Line2D at 0x1df8d0f3430>,
 <matplotlib.lines.Line2D at 0x1df8d0f35e0>,
 <matplotlib.lines.Line2D at 0x1df8d0f3700>,
 <matplotlib.lines.Line2D at 0x1df8d0f3820>,
 <matplotlib.lines.Line2D at 0x1df8d0f3940>]
```



Items relations:

1-Relations between weekly_sales and temperature is weak(less than 0.5).

```
In [172]: # relations items
dfk.corr()
```

Out[172]:

	Store	Weekly_Sales	Temperature	Fuel_Price	CPI	Unemployment
Store	1.000000	-0.335332	-0.022659	0.060023	-0.209492	0.223531
Weekly_Sales	-0.335332	1.000000	-0.063810	0.009464	-0.072634	-0.106176
Temperature	-0.022659	-0.063810	1.000000	0.144982	0.176888	0.101158
Fuel_Price	0.060023	0.009464	0.144982	1.000000	-0.170642	-0.034684
CPI	-0.209492	-0.072634	0.176888	-0.170642	1.000000	-0.302020
Unemployment	0.223531	-0.106176	0.101158	-0.034684	-0.302020	1.000000

#we use info function to know information about columns data and type.

#We use isnull function to know if there is a null column.

#We use duplicate function to check for duplicates.

```
In [58]: dfk.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   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
dtypes: float64(5), int64(1), object(2)
memory usage: 402.3+ KB

In [59]: col=dfk.columns
         dfk[col].isnull().sum()

Out[59]: Store           0
         Date            0
         Weekly_Sales    0
         Holiday_Flag    0
         Temperature     0
         Fuel_Price      0
         CPI             0
         Unemployment    0
         dtype: int64

In [60]: dfk.duplicated().to_string( index=False)
```

#We use groupby function to get the maximum weekly_sales by .sum and std.

```
In [62]: dfk.drop_duplicates(inplace=True)

In [64]: dfk.duplicated()

Out[64]: 0       False
         1       False
         2       False
         3       False
         4       False
                 ...
         6430    False
         6431    False
         6432    False
         6433    False
         6434    False
         Length: 6435, dtype: bool

In [65]: dfmax1=dfk.groupby('Store')['Weekly_Sales'].sum()
         dfmax1

Out[65]: Store
         1     2.224028e+08
         2     2.753824e+08
         3     5.758674e+07
         4     2.995440e+08
         Name: Weekly_Sales, dtype: float64

In [66]: print("index {} and max_value {}".format(dfmax1.idxmax(),dfmax1.max()))

         index 20 and max_value 301397792.46

In [67]: dfmax2=dfk.groupby('Store')['Weekly_Sales'].std()
         dfmax2

Out[67]: Store
         1      155980.767761
         2      237683.694682
         3       46319.631557
         4      266201.442297

In [68]: print("index {} and max_value {}".format(dfmax2.idxmax(),dfmax2.max()))

         index 14 and max_value 317569.9494755081
```

#We use filter: Holiday_week

```
In [70]: # filter
         holiday=dfk[dfk.Holiday_Flag=='Holiday_week']
         holiday
```

Out[70]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
1	1	12-02-2010	1641957.44	Holiday_week	38.51	2.548	211.242170	8.106
31	1	10-09-2010	1507460.69	Holiday_week	78.69	2.565	211.495190	7.787
42	1	26-11-2010	1955624.11	Holiday_week	64.52	2.735	211.748433	7.838
47	1	31-12-2010	1367320.01	Holiday_week	48.43	2.943	211.404932	7.838
53	1	11-02-2011	1649614.93	Holiday_week	36.39	3.022	212.936705	7.742
...	...	...	...	...	...	...	...	...

#We use filter: Non_Holiday_week

```
# filter
non_holiday=dfk[dfk.Holiday_Flag=='Non_holiday_week']
non_holiday
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	Non_holiday_week	42.31	2.572	211.096358	8.106
2	1	19-02-2010	1611968.17	Non_holiday_week	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	Non_holiday_week	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	Non_holiday_week	46.50	2.625	211.350143	8.106
5	1	12-03-2010	1439541.59	Non_holiday_week	57.79	2.667	211.380643	8.106
...	...	...	...	...	...	...	...	...

#Result:

```
holidays_higher_sales=holiday[(holiday.Weekly_Sales)>(non_holiday.Weekly_Sales.mean())]
holidays_higher_sales
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
1	1	12-02-2010	1641957.44	Holiday_week	38.51	2.548	211.242170	8.106
31	1	10-09-2010	1507460.69	Holiday_week	78.69	2.565	211.495190	7.787
42	1	26-11-2010	1955624.11	Holiday_week	64.52	2.735	211.748433	7.838
47	1	31-12-2010	1367320.01	Holiday_week	48.43	2.943	211.404932	7.838
53	1	11-02-2011	1649614.93	Holiday_week	36.39	3.022	212.936705	7.742

#To split date:

```
73]: dfk['year']=dfk['Date'].apply(lambda x:x[6:])
     dfk['month']=dfk['Date'].apply(lambda x:x[3:5])
     dfk['day']=dfk['Date'].apply(lambda x:x[0:2])
```

```
79]: dfk
```

79]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	year	month	day
0	1	05-02-2010	1643690.90	Non_holiday_week	42.31	2.572	211.096358	8.106	2010	02	05
1	1	12-02-2010	1641957.44	Holiday_week	38.51	2.548	211.242170	8.106	2010	02	12
2	1	19-02-2010	1611968.17	Non_holiday_week	39.93	2.514	211.289143	8.106	2010	02	19
3	1	26-02-2010	1409727.59	Non_holiday_week	46.63	2.561	211.319643	8.106	2010	02	26
4	1	05-03-2010	1554806.68	Non_holiday_week	46.50	2.625	211.350143	8.106	2010	03	05

#To get monthly weekly_sales of every year.

```
In [159]: dfa=dfk.groupby(['month' ,'year'])['Weekly_Sales'].sum().to_frame().reset_index()
          dfa
```

Out[159]:

	month	year	Weekly_Sales
0	01	2011	1.637040e+08
1	01	2012	1.688945e+08
2	02	2010	1.903330e+08
3	02	2011	1.863313e+08
4	02	2012	1.920636e+08

#We use filter to get monthly_sales of every year:

2010:

```
In [103]: # filter
          year2010=dfa[dfa.year=='2010']
          year2010
```

Out[103]:

	month	year	Weekly_Sales
2	02	2010	1.903330e+08
5	03	2010	1.819198e+08
8	04	2010	2.314124e+08
11	05	2010	1.867109e+08

2011:

```
# filter
year2011=dfa[dfa.year=='2011']
year2011
```

	month	year	Weekly_Sales
0	01	2011	1.637040e+08
3	02	2011	1.863313e+08
6	03	2011	1.793564e+08

2012:

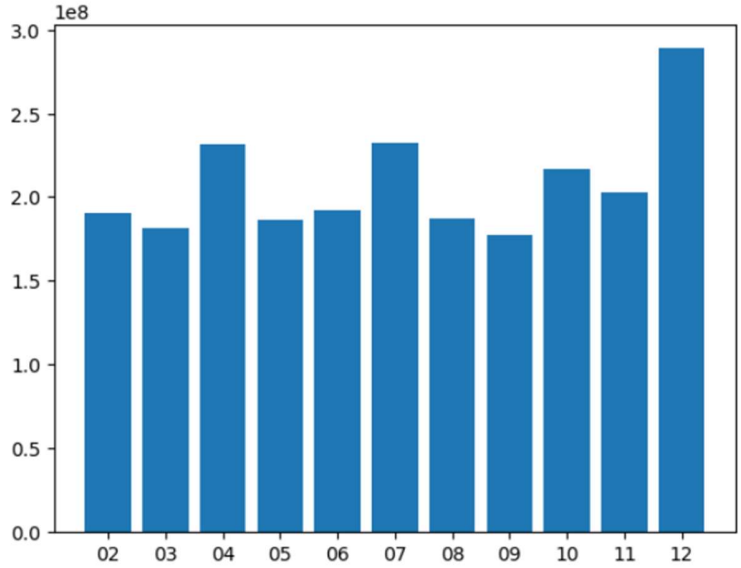
```
# filter
year2012=dfa[dfa.year=='2012']
year2012
```

	month	year	Weekly_Sales
1	01	2012	1.688945e+08
4	02	2012	1.920636e+08
7	03	2012	2.315097e+08

#bar2010:

```
plt.bar(year2010['month'],year2010['Weekly_Sales'])
#one month notfound 12month is higher sales
```

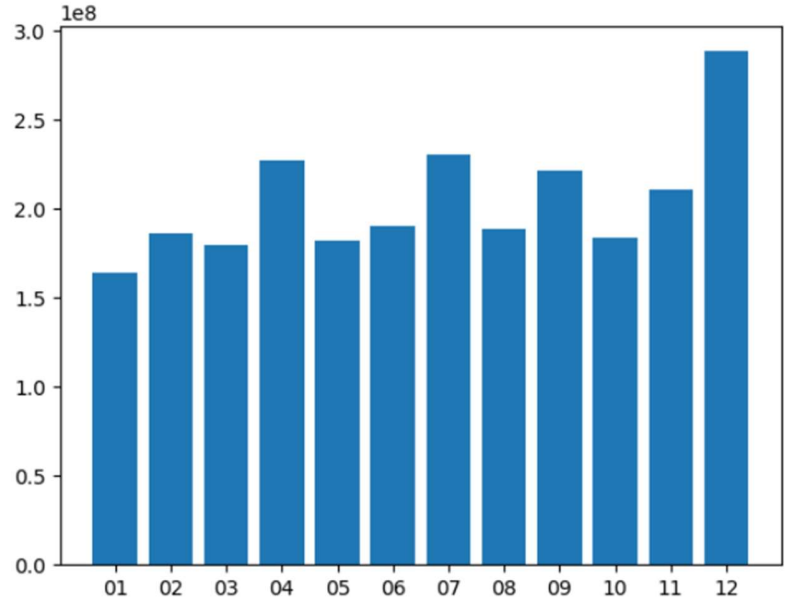
```
<BarContainer object of 11 artists>
```



#bar2011:

```
plt.bar(year2011['month'],year2011['Weekly_Sales'])
# 12month is higher sales
```

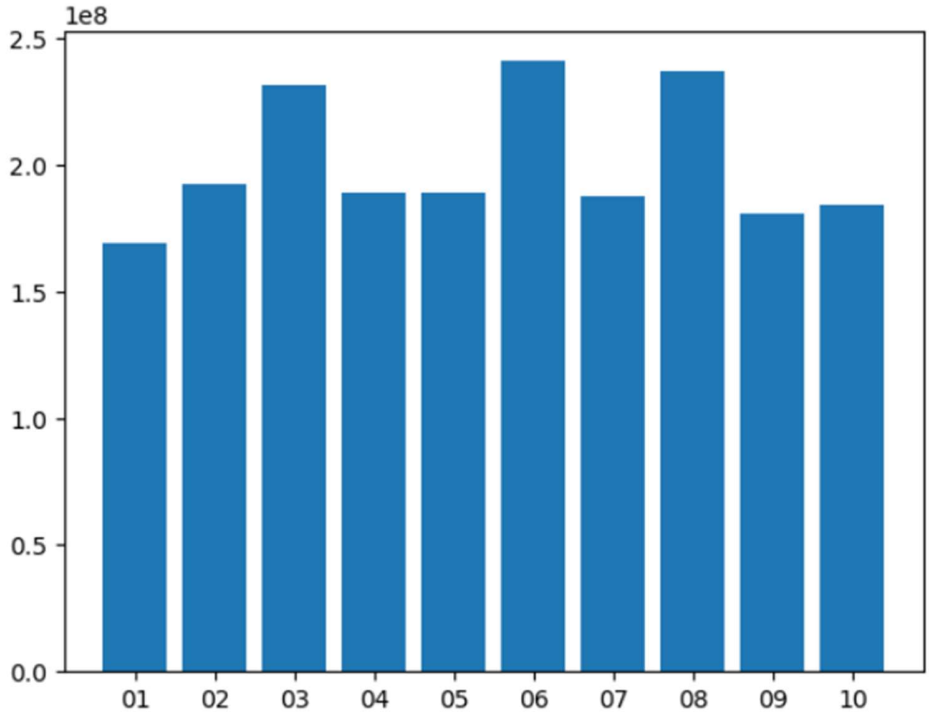
```
<BarContainer object of 12 artists>
```



#bar2012:

```
plt.bar(year2012['month'],year2012['Weekly_Sales'])
# 11 and12 months not found and 6month is higher sales
```

```
<BarContainer object of 10 artists>
```



#we get Weekly_sales of year:

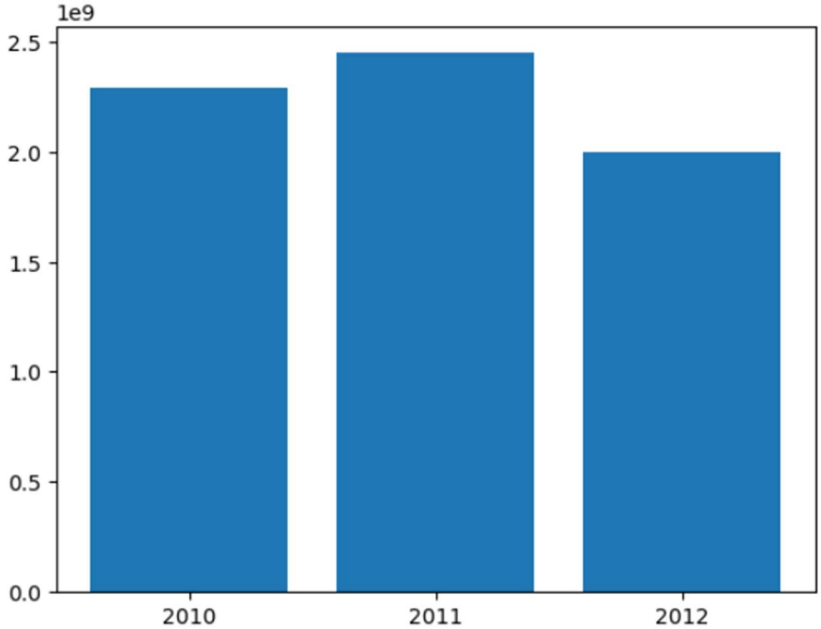
```
dfak=dfk.groupby('year')['Weekly_Sales'].sum().to_frame().reset_index()
dfak
```

	year	Weekly_Sales
0	2010	2.288886e+09
1	2011	2.448200e+09
2	2012	2.000133e+09

bar of year:

```
plt.bar(dfak['year'],dfak['Weekly_Sales'])
```

```
<BarContainer object of 3 artists>
```



#visual relation between Weekly_sales and Temperature :

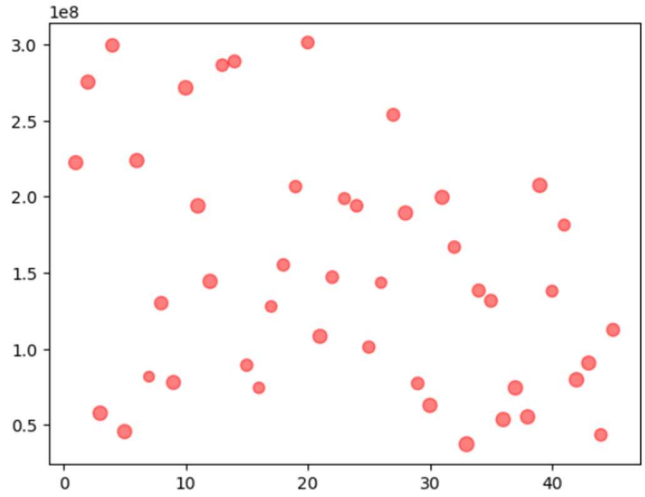
```
sales_and_temp=dfk.groupby('Store').agg({
    'Weekly_Sales' :lambda sales :sales.sum(),
    'Temperature'  :lambda temp :temp.mean()
}).reset_index()
sales_and_temp
```

	Store	Weekly_Sales	Temperature
0	1	2.224028e+08	68.306783
1	2	2.753824e+08	68.216364
2	3	5.758674e+07	71.434196
3	4	2.995440e+08	62.253357

#Scatter between sales and temp:

```
plt.scatter(sales_and_temp['Store'],sales_and_temp['Weekly_Sales'],s=sales_and_temp['Temperature'],color='red',alpha=.5)
# temperature is not effect in thr weekly_sales
```

```
<matplotlib.collections.PathCollection at 0x1df8ccdbfa0>
```



#visual relation between Weekly_sales and CPI:

```
2]: sales_and_cpi=dfk.groupby('Store').agg({
        'Weekly_Sales' :lambda sales :sales.sum(),
        'CPI'  :lambda cpi :cpi.mean()
    }).reset_index()
    sales_and_cpi
```

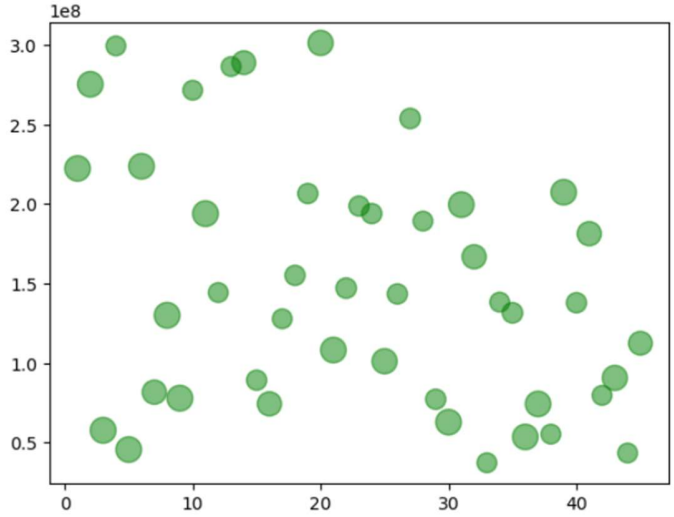
2]:

	Store	Weekly_Sales	CPI
0	1	2.224028e+08	215.996892
1	2	2.753824e+08	215.646311
2	3	5.758674e+07	219.391531
3	4	2.995440e+08	128.679669
4	5	4.547569e+07	216.565581

#Scatter between sales andcpi:

```
plt.scatter(sales_and_cpi['Store'],sales_and_cpi['Weekly_Sales'],s=sales_and_cpi['CPI'],color='g',alpha=.5)
# CPI is effect in the weekly_sales
# increase the CPI leads to increase the weeekly_sales
```

```
<matplotlib.collections.PathCollection at 0x1df8ce3af40>
```




```
# THE FINAL PROJEGT DATA SCIENCE METHODOLOGE
```