Impact of promotions on sales and customers

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.offline
from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)

# importing the train datadet from a CSV file
train = pd.read_csv(r"C:\Users\Vic\Documents\train.csv", low_memory=False)
```

In [2]:

```
train
```

Out[2]:

	Store	Day Of Week	Date	Sales	Customers	Open	Promo	State Holiday	School Holiday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1
...	...	...	...	...	...	...	...	...	...
1017204	1111	2	2013-01-01	0	0	0	0	a	1
1017205	1112	2	2013-01-01	0	0	0	0	a	1
1017206	1113	2	2013-01-01	0	0	0	0	a	1
1017207	1114	2	2013-01-01	0	0	0	0	a	1
1017208	1115	2	2013-01-01	0	0	0	0	a	1

1017209 rows × 9 columns

In [3]:

```
# Extracting Year, Week and Day from Date
train['Date'] = pd.to_datetime(train['Date'], errors='coerce')
train["Year"] = train["Date"].dt.year
train["Month"] = train["Date"].dt.month
train["Day"] = train["Date"].dt.day
```

In [4]:

```
train
```

Out[4]:

	Store	Day Of Week	Date	Sales	Customers	Open	Promo	State Holiday	School Holiday	Year	Month	Da
0	1	5	2015-07-31	5263	555	1	1	0	1	2015	7	
1	2	5	2015-07-31	6064	625	1	1	0	1	2015	7	
2	3	5	2015-07-31	8314	821	1	1	0	1	2015	7	
3	4	5	2015-07-31	13995	1498	1	1	0	1	2015	7	
4	5	5	2015-07-31	4822	559	1	1	0	1	2015	7	
...	...	...	...	...	...	...	...	...	...	...	...	
1017204	1111	2	2013-01-01	0	0	0	0	a	1	2013	1	
1017205	1112	2	2013-01-01	0	0	0	0	a	1	2013	1	
1017206	1113	2	2013-01-01	0	0	0	0	a	1	2013	1	
1017207	1114	2	2013-01-01	0	0	0	0	a	1	2013	1	
1017208	1115	2	2013-01-01	0	0	0	0	a	1	2013	1	

1017209 rows × 12 columns

In [68]:

```
# make a pivot table, add an aggregate function and select some columns
chart_promo = train.pivot_table(index="Customers", values=["Promo", "Year", "Sale
```

In [71]:
```
chart_promo
```

Out[71]:
```
           Promo  Sales       Year
Customers
        0  11203      0  348129468
        3      1      0       2014
        5      1      0       2013
        8      0     46       2014
       13      0    124       2015
      ...    ...    ...        ...
     5297      0  37403       2014
     5387      0  34692       2014
     5458      1  38484       2015
     5494      1  35702       2014
     7388      1  27190       2013
```

4086 rows × 3 columns

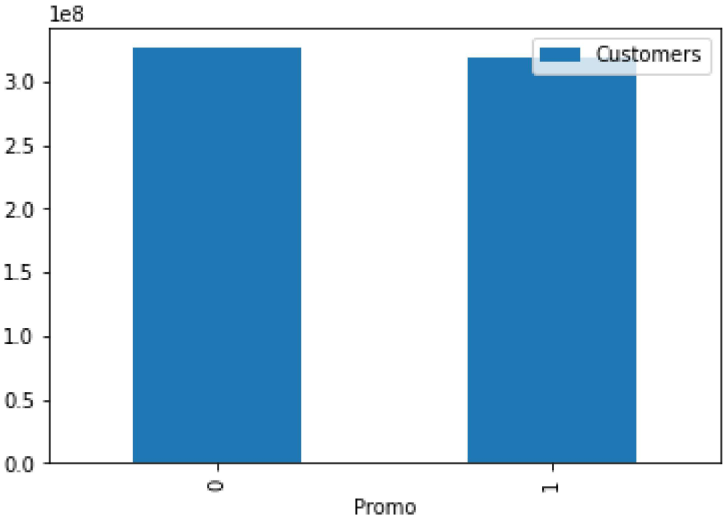
In [36]:
```
# make a pivot table, add an aggregate function and select some columns
chart_promo_customers = train.pivot_table(index="Promo", values=["Customers"], ag
```

In [37]:
```
chart_promo_customers
```

Out[37]:
```
        Customers
Promo
    0   325777807
    1   318263948
```

In [40]:
```
# plot a chart with chart using x=promo, y=customers
# 0 = no_promo, 1 = promo
chart_promo_customers.plot(kind="bar", linewidth=2.0)
```

Out[40]:
```
<AxesSubplot:xlabel='Promo'>
```



In [41]:
```
# make a pivot table, add an aggregate function and select some columns
chart_promo_sales = train.pivot_table(index="Promo", values=["Sales"], aggfunc="s
```

In [42]:
```
chart_promo_sales
```

Out[42]:

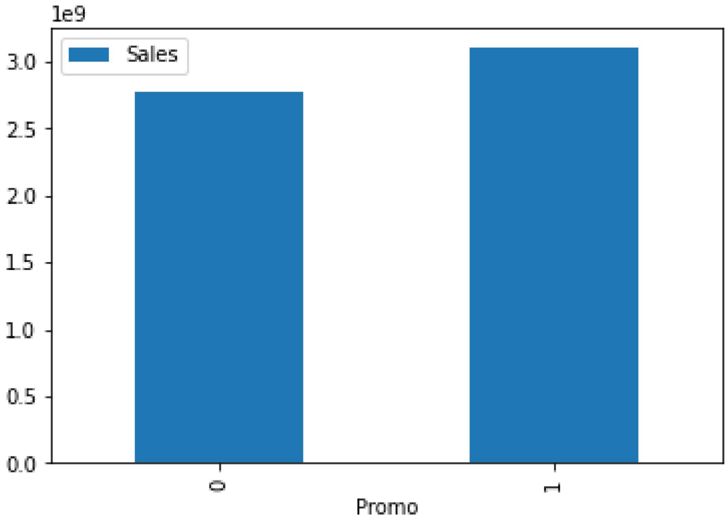
	Sales
Promo	
0	2771974337
1	3101206286

In [43]:

```
# plot a chart with chart using x=promo, y=sales
# 0 = no_promo, 1 = promo
chart_promo_sales.plot(kind="bar", linewidth=2.0)
```

Out[43]:

```
<AxesSubplot:xlabel='Promo'>
```



In [58]:

```
# make a pivot table, add an aggregate function and select some columns
chart = train.pivot_table(index="Year", values=["Promo"], aggfunc="sum"). round(
```

In [59]:

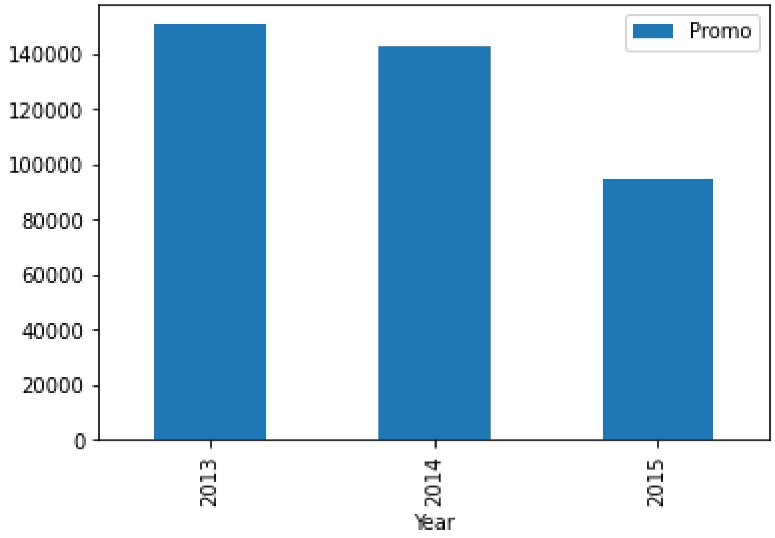
```
chart
```

Out[59]:

	Promo
Year	
2013	150525
2014	142780
2015	94775

In [60]:
```
# plot a chart with chart using x=year, y=promo
chart.plot(kind="bar", linewidth=2.0)
```

Out[60]:
```
<AxesSubplot:xlabel='Year'>
```



In [8]:
```
# make a pivot table, add an aggregate function and select some columns
chartA = train.pivot_table(index="Year", values=["Promo", "Sales"], aggfunc="sum'
```

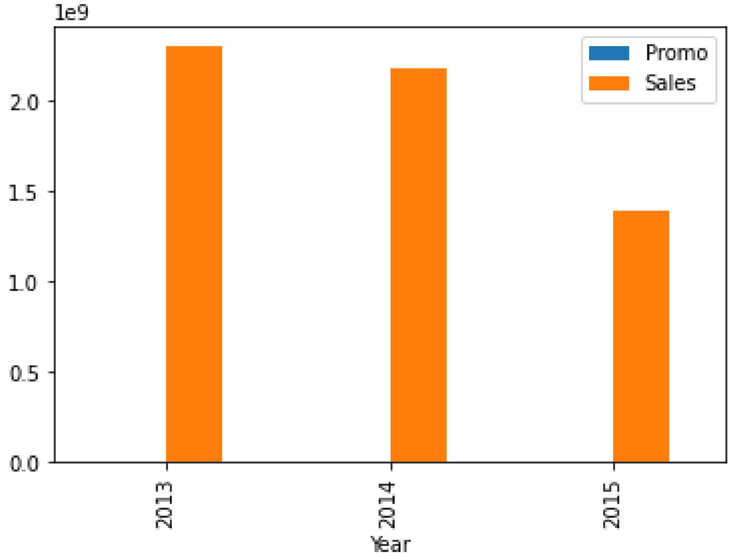
In [9]:
```
chartA
```

Out[9]:

	Promo	Sales
Year		
2013	150525	2302876084
2014	142780	2180804896
2015	94775	1389499643

In [15]:
```
# plot a chart with chartA using x=year, y=sales z=promo
chartA.plot(kind="bar", linewidth=2.0)
```

Out[15]:
```
<AxesSubplot:xlabel='Year'>
```



In [11]:
```
# make a pivot table, add an aggregate function and select some columns
chartB = train.pivot_table(index="Year", values=["Promo", "Customers"], aggfunc='
```

In [12]:
```
chartB
```

Out[12]:

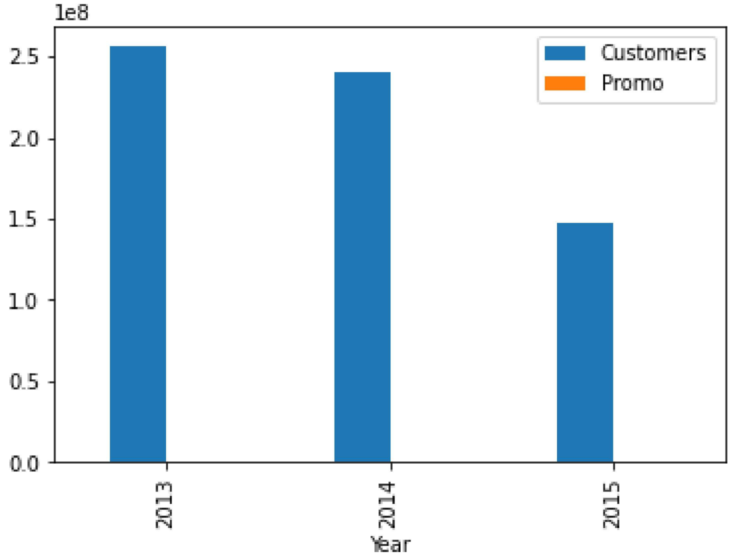
	Customers	Promo
Year		
2013	256004425	150525
2014	240488971	142780
2015	147548359	94775

In [13]:

```
# plot a chart with chartB using x=year, y=customers and promo
chartB.plot(kind="bar", linewidth=2.0)
```

Out[13]:

```
<AxesSubplot:xlabel='Year'>
```


1e8
Customers
Promo
Year