Data Analysis on Walmart Dataset :

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

In [2]:
```
df = pd.read_excel('Walmart.xlsx')
df
```

Out[2]:

	Order ID	Order Date	Ship Date	Customer Name	Country	City	State	Category	Product Name	Sales	Quantity	Profit
0	CA-2013-138688	2013-06-13	2013-06-17	Darrin Van Huff	United States	Los Angeles	California	Labels	Self-Adhesive Address Labels for Typewriters b...	14.620	2	6.8714
1	CA-2011-115812	2011-06-09	2011-06-14	Brosina Hoffman	United States	Los Angeles	California	Furnishings	Eldon Expressions Wood and Plastic Desk Access...	48.860	7	14.1694
2	CA-2011-115812	2011-06-09	2011-06-14	Brosina Hoffman	United States	Los Angeles	California	Art	Newell 322	7.280	4	1.9656
3	CA-2011-115812	2011-06-09	2011-06-14	Brosina Hoffman	United States	Los Angeles	California	Phones	Mitel 5320 IP Phone VoIP phone	907.152	6	90.7152
4	CA-2011-115812	2011-06-09	2011-06-14	Brosina Hoffman	United States	Los Angeles	California	Binders	DXL Angle-View Binders with Locking Rings by S...	18.504	3	5.7825
...	...	...	...	...	...	...	...	...	...	...	...	...
3198	CA-2013-125794	2013-09-30	2013-10-04	Maris Laiblaw	United States	Los Angeles	California	Accessories	Memorex Mini Travel Drive 64 GB USB 2.0 Flash ...	36.240	1	15.2208
3199	CA-2014-121258	2014-02-27	2014-03-04	Dave Brooks	United States	Costa Mesa	California	Furnishings	Tenex B1-RE Series Chair Mats for Low Pile Car...	91.960	2	15.6332
3200	CA-2014-121258	2014-02-27	2014-03-04	Dave Brooks	United States	Costa Mesa	California	Phones	Aastra 57i VoIP phone	258.576	2	19.3932
3201	CA-2014-121258	2014-02-27	2014-03-04	Dave Brooks	United States	Costa Mesa	California	Paper	It's Hot Message Books with Stickers, 2 3/4" x 5"	29.600	4	13.3200
3202	CA-2014-118914	2014-05-05	2014-05-10	Chris Cortes	United States	Westminster	California	Appliances	Acco 7-Outlet Masterpiece Power Center, Wihtou...	243.160	2	72.9480

3203 rows × 12 columns

In [3]:
```
df.describe()
```

Out[3]:

	Order Date	Ship Date	Sales	Quantity	Profit
count	3203	3203	3203.000000	3203.000000	3203.000000
mean	2013-05-10 03:06:07.530644192	2013-05-14 01:25:25.195129600	226.493233	3.829910	33.849032
min	2011-01-07 00:00:00	2011-01-09 00:00:00	0.990000	1.000000	-3399.980000
25%	2012-05-22 00:00:00	2012-05-26 00:00:00	19.440000	2.000000	3.852000
50%	2013-07-22 00:00:00	2013-07-25 00:00:00	60.840000	3.000000	11.166400
75%	2014-05-23 00:00:00	2014-05-27 00:00:00	215.809000	5.000000	33.006400
max	2014-12-31 00:00:00	2015-01-06 00:00:00	13999.960000	14.000000	6719.980800
std	NaN	NaN	524.876877	2.260947	174.109081

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3203 entries, 0 to 3202
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   Order ID       3203 non-null   object
 1   Order Date     3203 non-null   datetime64[ns]
 2   Ship Date      3203 non-null   datetime64[ns]
 3   Customer Name  3203 non-null   object
 4   Country        3203 non-null   object
 5   City           3203 non-null   object
 6   State          3203 non-null   object
 7   Category       3203 non-null   object
 8   Product Name   3203 non-null   object
 9   Sales          3203 non-null   float64
 10  Quantity       3203 non-null   int64
 11  Profit         3203 non-null   float64
dtypes: datetime64[ns](2), float64(2), int64(1), object(7)
memory usage: 300.4+ KB
```

In [5]:
```
df.loc[:, ['City','Sales']]
```

Out[5]:

	City	Sales
0	Los Angeles	14.620
1	Los Angeles	48.860
2	Los Angeles	7.280
3	Los Angeles	907.152
4	Los Angeles	18.504
...	...	...
3198	Los Angeles	36.240
3199	Costa Mesa	91.960
3200	Costa Mesa	258.576
3201	Costa Mesa	29.600
3202	Westminster	243.160

3203 rows × 2 columns

In [6]:
```
df.index
```

Out[6]:
```
RangeIndex(start=0, stop=3203, step=1)
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Order ID', 'Order Date', 'Ship Date', 'Customer Name', 'Country',
       'City', 'State', 'Category', 'Product Name', 'Sales', 'Quantity',
       'Profit'],
      dtype='object')
```

In [8]:
```
df.count()
```

Out[8]:
```
Order ID         3203
Order Date       3203
Ship Date        3203
Customer Name    3203
Country          3203
City             3203
State            3203
Category         3203
Product Name     3203
Sales            3203
Quantity         3203
Profit           3203
dtype: int64
```

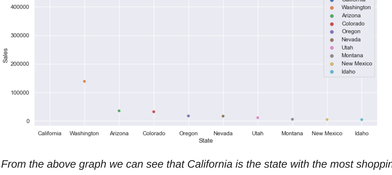
In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Order ID         0
Order Date       0
Ship Date        0
Customer Name    0
Country          0
City             0
State            0
Category         0
Product Name     0
Sales            0
Quantity         0
Profit           0
dtype: int64
```

Exploratory Data Analysis :

In [10]:
```
sales_state = df.groupby('State').as_index=False)['Sales'].sum().sort_values(by='Sales', ascending=False).head(10)
sns.set(rc={'figure.figsize':(12,5)})
sns.scatterplot(x='State', y='Sales', data=sales_state, hue='State').set(title='Sales vs State graphical statement'))
```

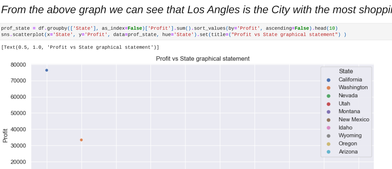
Out[10]:
```
[Text(0.5, 1.0, 'Sales vs State graphical statement')]
```



From the above graph we can see that California is the state with the most shopping.

In [11]:
```
sales_city = df.groupby(['City'], as_index=False)['Sales'].sum().sort_values(by='Sales', ascending=False).head(10)
sns.scatterplot(x='City', y='Sales', data=sales_city, hue='City').set(title='Sales vs City graphical statement'))
```

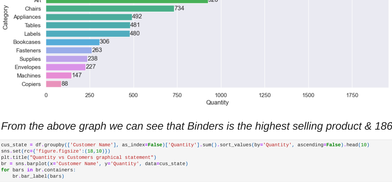
Out[11]:
```
[Text(0.5, 1.0, 'Sales vs City graphical statement')]
```



From the above graph we can see that Los Angles is the City with the most shopping.

In [12]:
```
prof_state = df.groupby(['State'], as_index=False)['Profit'].sum().sort_values(by='Profit', ascending=False).head(10)
sns.scatterplot(x='State', y='Profit', data=prof_state, hue='State').set(title='Profit vs State graphical statement') )
```

Out[12]:
```
[Text(0.5, 1.0, 'Profit vs State graphical statement')]
```



From the above graph we can see that highest profit gained from the state California.

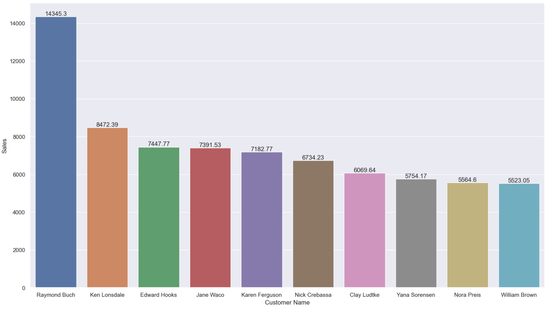
In [26]:
```
cus_state = df.groupby(['Category'], as_index=False)['Quantity'].sum().sort_values(by='Quantity', ascending=False)
sns.set(rc={'figure.figsize':(12,5)})
plt.title('Quantity vs Customers graphical statement")
br = sns.barplot(y='Category', x='Quantity', data=cus_state)
for bars in br.containers:
    br.bar_label(bars)
```

From the above graph we can see that Binders is the highest selling product & 1868 unit products was sold.

In [22]:
```
cus_state = df.groupby(['Customer Name'], as_index=False)['Quantity'].sum().sort_values(by='Quantity', ascending=False).head(10)
sns.set(rc={'figure.figsize':(10,10)})
plt.title('Quantity vs Customers graphical statement")
br = sns.barplot(x='Customer Name', y='Quantity', data=cus_state)
for bars in br.containers:
    br.bar_label(bars)
```


From the above graph we can see that William Brown is the customer with highest number of order.

In [15]:
```
cus_state = df.groupby(['Customer Name'], as_index=False)['Sales'].sum().sort_values(by='Sales', ascending=False).head(10)
sns.set(rc={'figure.figsize':(10,10)})
in = sns.barplot(x='Customer Name', y='Sales', data=cus_state)
for bar in in.containers:
    in.bar_label(bar)
```



From the above graph we can see that Raymond Buch is the Customer with the highest amount of order.