Weee!_Take-home(Inventory_Analyst)_Test

Import Data from excel file

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib as plt
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.offline as pyo
pyo.init_notebook_mode()
```

In [2]:

```
data_1 = pd.read_excel('./Weee!_Take-home(Inventory_Analyst)_Test.xlsx',
                      sheet_name="Tab1",
                      header = 0)

data_2 = pd.read_excel('./Weee!_Take-home(Inventory_Analyst)_Test.xlsx',
                      sheet_name="Tab2",
                      header = 0)

data_3 = pd.read_excel('./Weee!_Take-home(Inventory_Analyst)_Test.xlsx',
                      sheet_name="Tab3",
                      header = 0)
```

In [3]:

```
data_1.shape
```

Out[3]:

```
(450, 2)
```

In [4]:

```
data_2.shape
```

Out[4]:

```
(950, 2)
```

In [5]:

```
data_3.head()
```

Out[5]:

	Month	Salesman	Region	Product	No. Customers	Net Sales	Profit / Loss	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Ur
0	2019-01-01	Joseph	North	FastCar	8.0	1592.0	563.0	NaN	NaN	NaN	...	
1	2019-01-01	Joseph	North	RapidZoo	8.0	1088.0	397.0	NaN	NaN	NaN	...	
2	2019-01-01	Joseph	West	SuperGlue	8.0	1680.0	753.0	NaN	NaN	NaN	...	
3	2019-01-01	Joseph	West	FastCar	9.0	2133.0	923.0	NaN	NaN	NaN	...	
4	2019-01-01	Joseph	West	RapidZoo	10.0	1610.0	579.0	NaN	NaN	NaN	...	

5 rows × 26 columns

Question 1

1. Find the common keys in Tab 1 and Tab 2, then list the keys and their corresponding values in your result table
2. If a key only appears in one of the sheets, do not list it.
3. If a common key appears multiple times in either of the sheets, only list the value on top of the list.

Join two table on the *Key* and remove duplicated record

In [6]:

```
q1 = data_1.merge(data_2, on ='Key',how = 'left')
```

In [7]:

```
q1.head()
```

Out[7]:

	Key	Value_x	Value_y
0	9671	10.0	10.0
1	9671	10.0	10.0
2	9671	10.0	10.0
3	9671	10.0	10.0
4	9671	10.0	10.0

Remove other record for duplicated Key

In [8]:

```
# Question 1 Answer
q1 = q1.dropna(axis=0)
q1 = q1.drop_duplicates(subset=['Key'])
q1
```

Out[8]:

	Key	Value_x	Value_y
0	9671	10.0	10.0
5	95272	10.0	10.0
11	18899	15.0	15.0
14	45782	10.0	10.0
18	11479	10.0	10.0
...	...	...	...
716	96935	17.0	17.0
718	51010	10.0	10.0
743	32022	31.0	13.0
746	97930	10.0	10.0
750	95637	10.0	10.0

270 rows × 3 columns

In [9]:

```
q1.shape
```

Out[9]:

```
(270, 3)
```

Question 2

Use a pivot table to show: For each sales person, how many customers and how much total sales they made in 2019.

Only count those lines that the profit/loss is larger than 300.

Data are in Tab3

Step 1: Clean Tab3 data, remove *Unnamed* columns and *NaN* rows

In [10]:

```
data_3 = data_3.loc[ : , ~data_3.columns.str.contains("^Unnamed")]
```

In [11]:

```
data_3 = data_3.dropna(axis=0)
```

In [12]:

```
data_3
```

Out[12]:

	Month	Salesman	Region	Product	No. Customers	Net Sales	Profit / Loss
0	2019-01-01	Joseph	North	FastCar	8.0	1592.0	563.0
1	2019-01-01	Joseph	North	RapidZoo	8.0	1088.0	397.0
2	2019-01-01	Joseph	West	SuperGlue	8.0	1680.0	753.0
3	2019-01-01	Joseph	West	FastCar	9.0	2133.0	923.0
4	2019-01-01	Joseph	West	RapidZoo	10.0	1610.0	579.0
...	...	...	...	...	...	...	...
461	2020-01-01	Matt	West	FastCar	10.0	2170.0	832.0
462	2020-01-01	Matt	West	RapidZoo	9.0	2610.0	1090.0
463	2020-01-01	Matt	Middle	SuperGlue	8.0	2312.0	1000.0
464	2020-01-01	Matt	Middle	FastCar	6.0	1020.0	308.0
465	2020-01-01	Matt	Middle	RapidZoo	8.0	872.0	331.0

466 rows × 7 columns

Step 2: Filter all the sale made in *2019* and *Profit / Loss* is larger than 300

In [13]:

```
q2 = data_3[(data_3['Month'].dt.year == 2019) & (data_3['Profit / Loss'] > 300)]
```

Step 3: Creat a pivot table by *groupby* and aggrate *No. Customers* and *Net Sales*

In [14]:

```
# Question2 Answer
q2[['Salesman','No. Customers','Net Sales']].groupby(['Salesman']).sum()
```

Out[14]:

	No. Customers	Net Sales
Salesman		
Joseph	850.0	172089.0
Lawrence	810.0	172726.0

	No. Customers	Net Sales
Salesman		
Maria	795.0	164886.0
Matt	821.0	167810.0

Question 3

Step 1: Create a pivot table to indicate how much the sales revenue is made by each salesman every month

In [15]:

```
q3 = data_3.pivot_table(index='Month',columns='Salesman',values='Net Sales',aggfunc=np.sur
q3
```

Out[15]:

Salesman	Joseph	Lawrence	Maria	Matt
Month				
2019-01-01	12758.0	13363.0	15199.0	15566.0
2019-02-01	14098.0	13384.0	13909.0	16493.0
2019-03-01	12692.0	14649.0	12733.0	11547.0
2019-04-01	13847.0	17050.0	15288.0	13596.0
2019-05-01	17063.0	16401.0	14182.0	14847.0
2019-06-01	16188.0	15969.0	16725.0	13475.0
2019-07-01	13376.0	15811.0	12048.0	14814.0
2019-08-01	15339.0	13650.0	14834.0	14596.0
2019-09-01	15715.0	13732.0	12138.0	15643.0
2019-10-01	15318.0	16518.0	13304.0	14367.0
2019-11-01	14610.0	14853.0	15225.0	13682.0
2019-12-01	13225.0	14531.0	16903.0	14735.0
2020-01-01	12758.0	13363.0	15199.0	15566.0

Step 2 Create the line chart

In [16]:

```
# Question 3 Answer

q3=q3.reset_index()

trace1 = go.Scatter(
                    x = q3.Month,
                    y = q3.Joseph,
                    mode = "lines+markers",
                    name = "Joseph",
                    marker = dict(color = 'red'),
                   )

trace2 = go.Scatter(
                    x = q3.Month,
                    y = q3.Lawrence,
                    mode = "lines+markers",
                    name = "Lawrence",
                    marker = dict(color = 'green'),
                   )
```

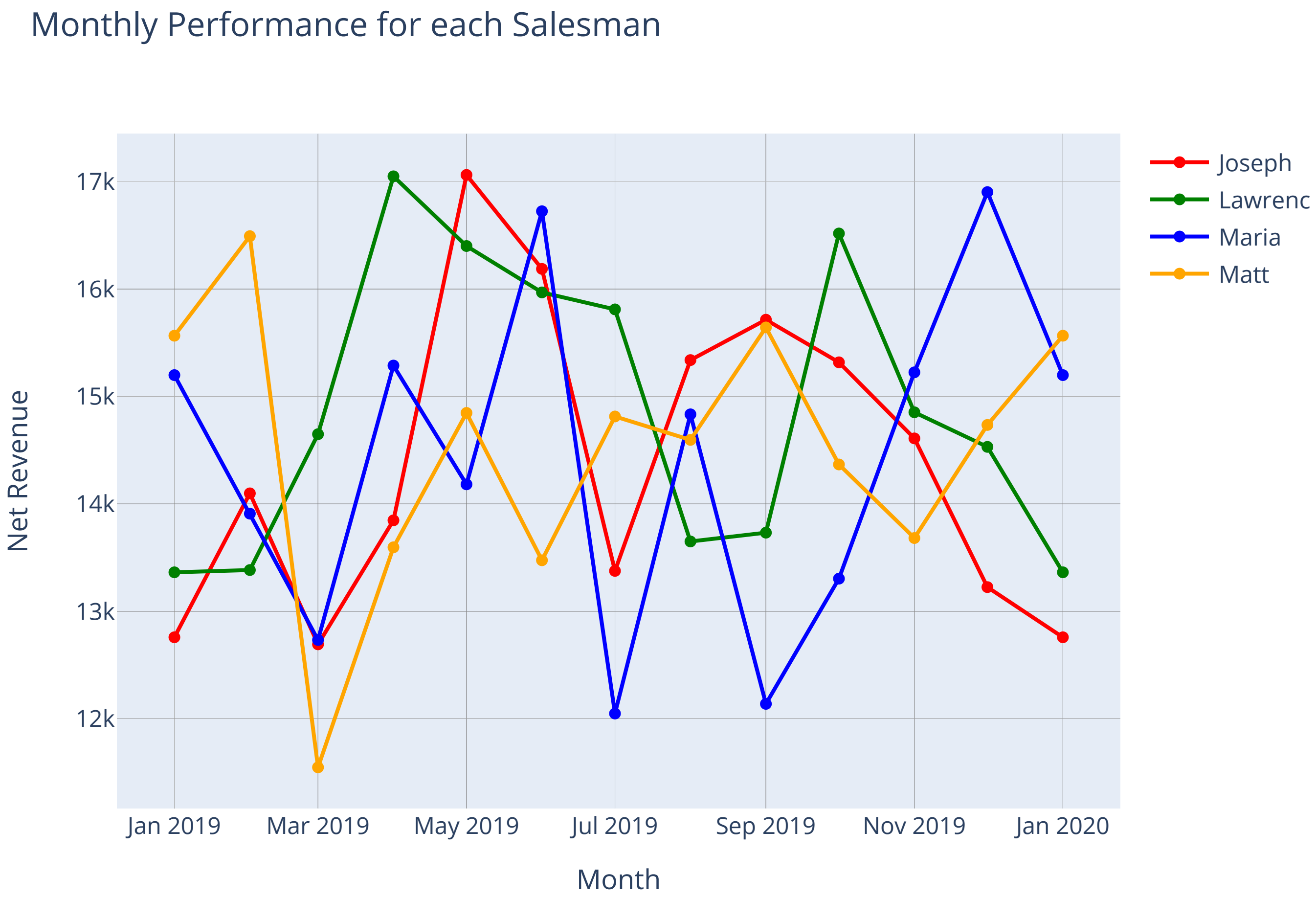
```
trace3 = go.Scatter(
                    x = q3.Month,
                    y = q3.Maria,
                    mode = "lines+markers",
                    name = "Maria",
                    marker = dict(color = 'blue'),
                   )

trace4 = go.Scatter(
                    x = q3.Month,
                    y = q3.Matt,
                    mode = "lines+markers",
                    name = "Matt",
                    marker = dict(color = 'orange'),
                   )

data = [trace1, trace2, trace3, trace4]

layout = dict(title = 'Monthly Performance for each Salesman',
              xaxis= dict(title= 'Month',ticklen= 5,zeroline= False),
              yaxis= dict(title= 'Net Revenue',ticklen= 5,zeroline= False)
             )

fig = dict(data = data, layout = layout)
pyo.iplot(fig)
```



```
In [ ]:
```