

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as pltly
```

```
from google.colab import files
```

```
uploaded = files.upload()
```

SuperMart_P...tTable.xlsx

- **SuperMart_PivotTable.xlsx**(n/a) - 533594 bytes, last modified: 11/19/2020 - 100% done
Saving SuperMart_PivotTable.xlsx to SuperMart_PivotTable.xlsx

```
import io
df = pd.read_excel(io.BytesIO(uploaded['SuperMart_PivotTable.xlsx']))
```

```
print(df)
```

	Customer Name	Segment	...	Sales_in_M\$	Qty
0	Claire Gute	Consumer	...	261.9600	452
1	Claire Gute	Consumer	...	731.9400	19356
2	Darrin Van Huff	Corporate	...	14.6200	6986
3	Sean O'Donnell	Consumer	...	957.5775	7554
4	Sean O'Donnell	Consumer	...	22.3680	18992
...
9795	Sally Hughsby	Corporate	...	3.7980	3271
9796	Cindy Schnelling	Corporate	...	10.3680	18586
9797	Cindy Schnelling	Corporate	...	235.1880	1067
9798	Cindy Schnelling	Corporate	...	26.3760	14447
9799	Cindy Schnelling	Corporate	...	10.3840	10310

```
[9800 rows x 8 columns]
```

STEP 1: DATA CLEANING

```
# creating bool series True for NaN values
bool_series = pd.isnull(df["Customer Name"])
```

```
# filtering data
# displaying data only with team = NaN
df[bool_series]
```

```
# creating bool series True for NaN values
bool_series = pd.isnull(df["Segment"])
```

```
# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["City"])

# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["State"])

# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["Category"])

# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["Sub-Category"])

# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["Sales_in_M$"])

# filtering data
# displayind data only with team = NaN
df[bool_series]

# creating bool series True for NaN values
bool_series = pd.isnull(df["Qty"])

# filtering data
# displayind data only with team = NaN
df[bool_series]
```

Customer Name	Segment	City	State	Category	Sub-Category	Sales_in_M\$	Qty
---------------	---------	------	-------	----------	--------------	--------------	-----

We infer that, There are no null values for any rows and columns

Step 2: Data Processing

Creating A Pivot Table

```
pd.pivot_table(df, index=["Customer Name"])
```

All the numerical data with correspondance to the index, Customer Name is printed.

```
r=pd.pivot_table(df, index=["Customer Name", "Segment"])
r
```

```
r.loc[r['Sales_in_M$'].idxmax()]
```

```
Qty          6133.333333
Sales_in_M$   1751.292000
Name: (Mitch Willingham, Corporate), dtype: float64
```

```
a=pd.pivot_table(df, index=["Category"], values=["Sales_in_M$"], aggfunc=np.sum)
a
```

```
a.loc[a['Sales_in_M$'].idxmax()]
```

```
Sales_in_M$    827455.873
Name: Technology, dtype: float64
```

```
b=pd.pivot_table(df, index=["Sub-Category", "Category"], values=["Sales_in_M$"], aggfunc=np.sum)
b
```

```
b.loc[b['Sales_in_M$'].idxmax()]
```

```
Sales_in_M$    327782.448
Name: (Phones, Technology), dtype: float64
```

```
c=pd.pivot_table(df, index=["Customer Name", "Segment"], values=["Sales_in_M$"], columns=["Sub-Category"])
c
```

```
c.loc[c['Sales_in_M$'].idxmax()]
```

```
c.plot(kind='bar',figsize=(20,10))
```

```
d = pd.pivot_table(df,index=["Segment","State"],columns=["Category"],values=["Sales_in_M$"],aggfunc='sum')
d
```

```
d1=d.query('Segment == ["Consumer"]')
d1
```

```
d1.loc[d1['Sales_in_M$'].idxmax()]
```

```
e=df.pivot_table(index=["State","Category"], values="Qty", aggfunc='sum')
e
```

```
e.loc[e['Qty'].idxmax()]
```

```
f=df.pivot_table(index="Segment", columns = ['Category'] , values="Sales_in_M$", aggfunc='sum')
f
```

	Category	Furniture	Office Supplies	Technology
Segment				
Consumer		354.708379	116.976760	428.431266
Corporate		350.830735	125.704170	450.261692
Home Office		337.928896	115.691831	544.484690

```
f.plot(kind='bar',figsize=(20,10));
```

```
g=pd.pivot_table(df,index="State", columns = ['Sub-Category'] , values=["Sales_in_M$"],aggfunc='sum')
g
```

```
g.loc[g['Sales_in_M$'].idxmax()]
```

```
g.plot(kind='bar',figsize=(20,10));
```

```
h=pd.pivot_table(df,index="City", columns = ['Category'] , values=["Sales_in_M$"],aggfunc='sum')
h
```

```
h.loc[h['Sales_in_M$'].idxmin()]
```

```
i=pd.pivot_table(df,index="Segment" , values=["Qty"], aggfunc='mean')
i

i.loc[i['Qty'].idxmax()]

j=pd.pivot_table(df,index=["Sub-Category","Segment"] ,values=["Qty"], aggfunc='sum'
j

j.loc[t7['Qty'].idxmax()]

k=pd.pivot_table(df,index=["Segment"] ,values=["City"], aggfunc=len, fill_value=0)
k

k.plot(kind='bar',figsize=(20,10));

l=pd.pivot_table(df,index=["Category","City"] ,values=["Qty"], aggfunc=len, fill_va
l

l.loc[t10['Qty'].idxmax()]
```

