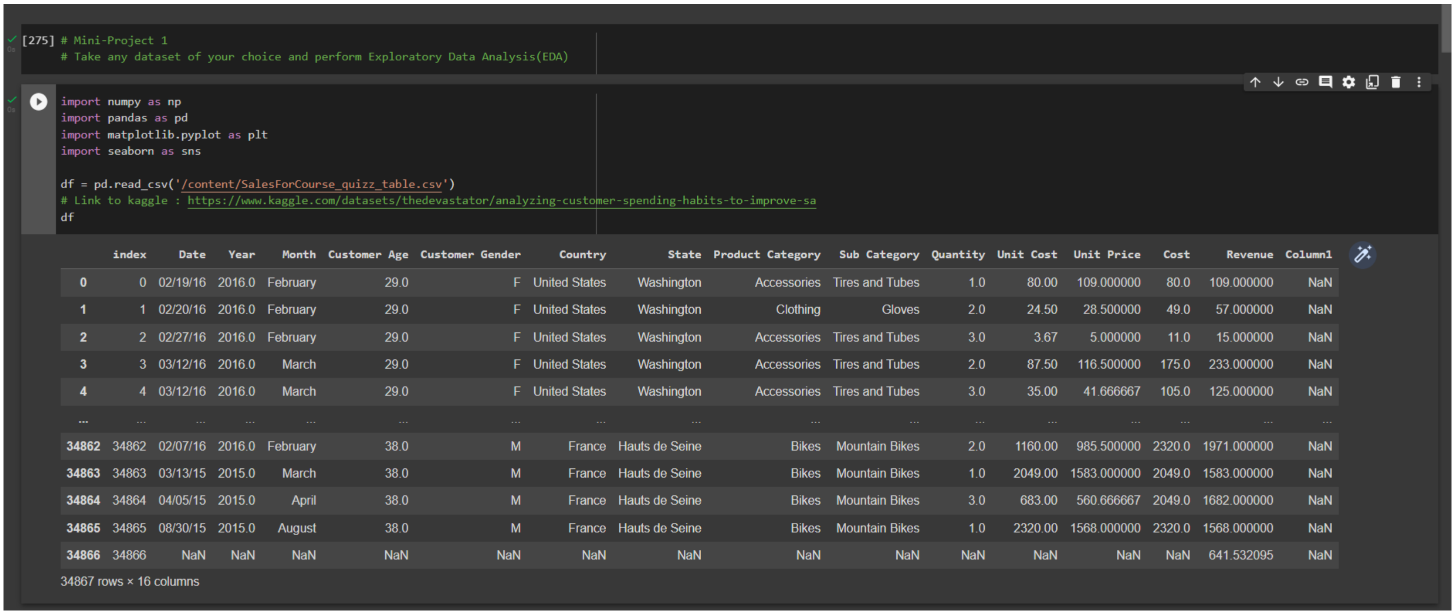
ARIN DEV
IIT BHUBANESWAR

Second Year, B.Tech (ECE)

Contact : arindev30@gmail.com
21ec01048@gmail.com

Mini Project 1

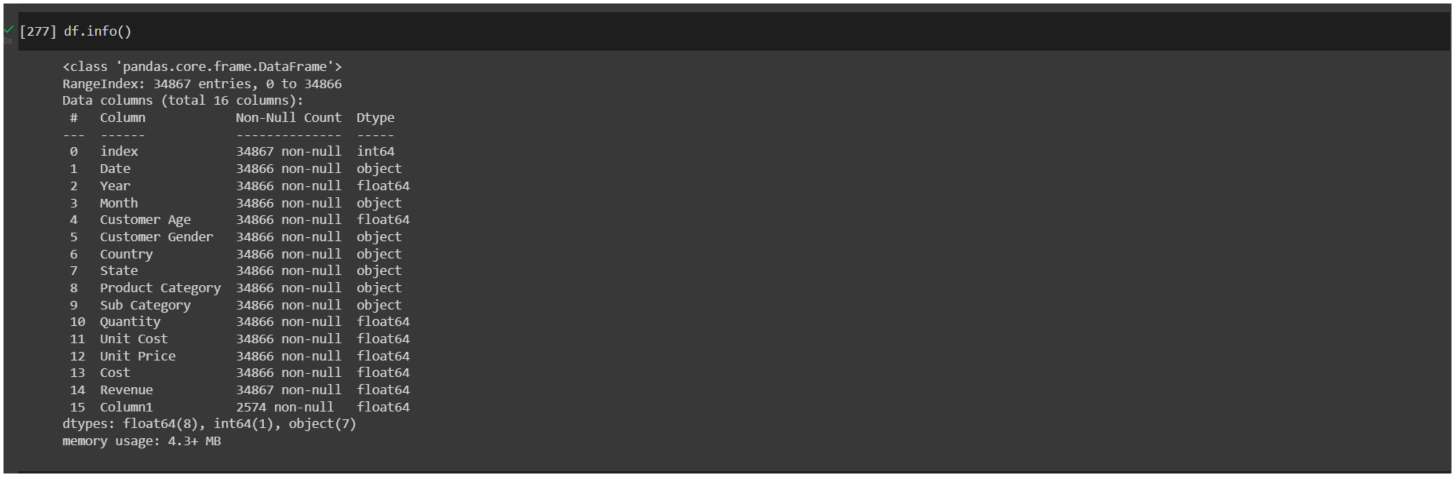


```
[275] # Mini-Project 1
      # Take any dataset of your choice and perform Exploratory Data Analysis(EDA)
```

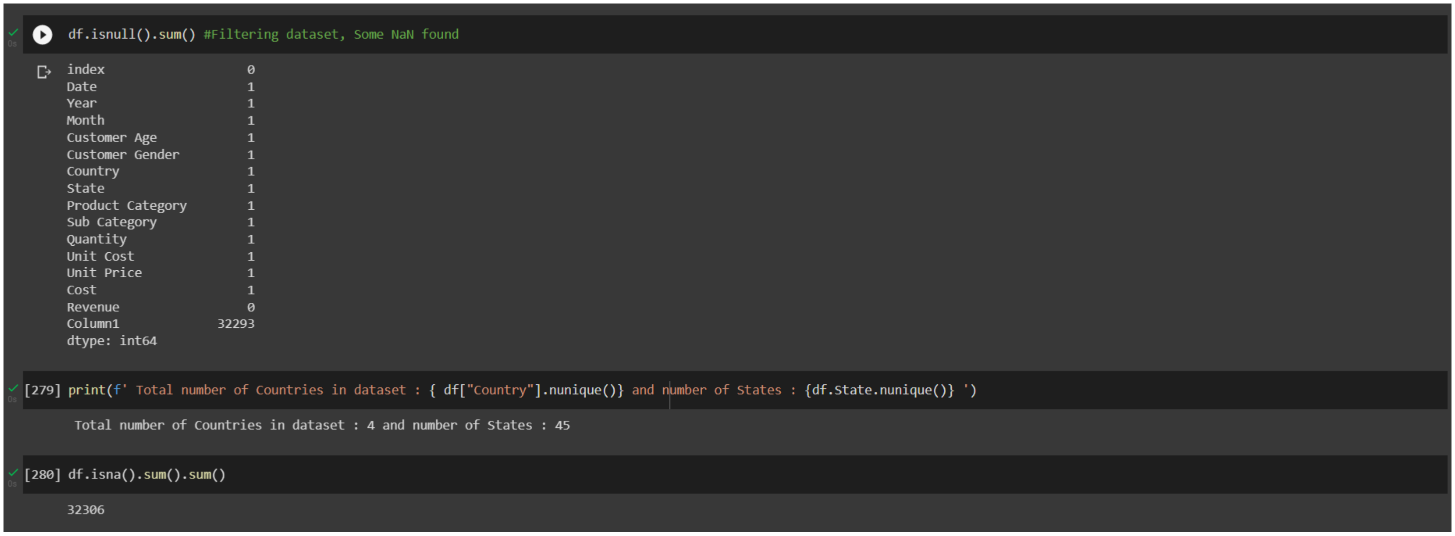
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/SalesForCourse_quizz_table.csv')
# Link to kaggle : https://www.kaggle.com/datasets/thedevastator/analyzing-customer-spending-habits-to-improve-sa
df
```

	index	Date	Year	Month	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Unit Price	Cost	Revenue	Column1
0	0	02/19/16	2016.0	February	29.0	F	United States	Washington	Accessories	Tires and Tubes	1.0	80.00	109.000000	80.0	109.000000	NaN
1	1	02/20/16	2016.0	February	29.0	F	United States	Washington	Clothing	Gloves	2.0	24.50	28.500000	49.0	57.000000	NaN
2	2	02/27/16	2016.0	February	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	3.67	5.000000	11.0	15.000000	NaN
3	3	03/12/16	2016.0	March	29.0	F	United States	Washington	Accessories	Tires and Tubes	2.0	87.50	116.500000	175.0	233.000000	NaN
4	4	03/12/16	2016.0	March	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	35.00	41.666667	105.0	125.000000	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34862	34862	02/07/16	2016.0	February	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	2.0	1160.00	985.500000	2320.0	1971.000000	NaN
34863	34863	03/13/15	2015.0	March	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2049.00	1583.000000	2049.0	1583.000000	NaN
34864	34864	04/05/15	2015.0	April	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	3.0	683.00	560.666667	2049.0	1682.000000	NaN
34865	34865	08/30/15	2015.0	August	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2320.00	1568.000000	2320.0	1568.000000	NaN
34866	34866	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	641.532095	NaN

34867 rows × 16 columns



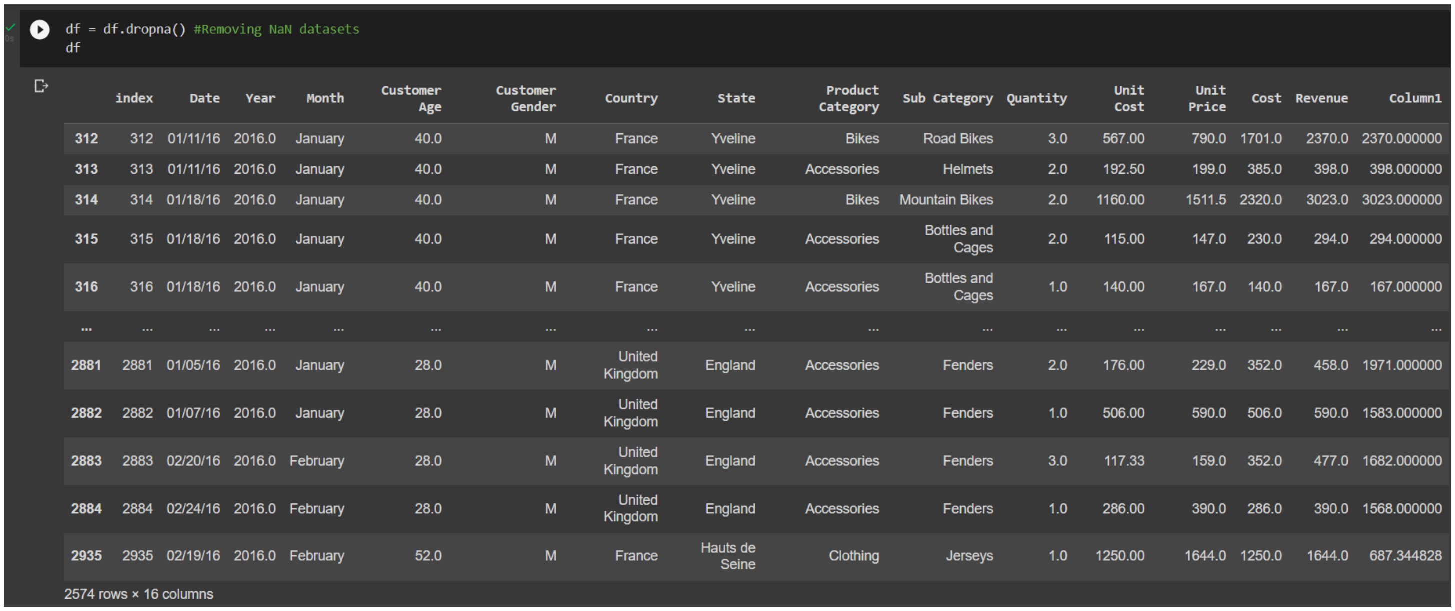
```
[277] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34867 entries, 0 to 34866
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   index             34867 non-null  int64
 1   Date              34866 non-null  object
 2   Year              34866 non-null  float64
 3   Month             34866 non-null  object
 4   Customer Age      34866 non-null  float64
 5   Customer Gender   34866 non-null  object
 6   Country           34866 non-null  object
 7   State             34866 non-null  object
 8   Product Category  34866 non-null  object
 9   Sub Category      34866 non-null  object
 10  Quantity          34866 non-null  float64
 11  Unit Cost         34866 non-null  float64
 12  Unit Price        34866 non-null  float64
 13  Cost              34866 non-null  float64
 14  Revenue           34867 non-null  float64
 15  Column1           2574 non-null   float64
dtypes: float64(8), int64(1), object(7)
memory usage: 4.3+ MB
```



```
df.isnull().sum() #Filtering dataset, Some NaN found

index                   0
Date                    1
Year                    1
Month                   1
Customer Age            1
Customer Gender         1
Country                 1
State                   1
Product Category        1
Sub Category            1
Quantity                1
Unit Cost               1
Unit Price              1
Cost                    1
Revenue                 0
Column1             32293
dtype: int64
```

```
[279] print(f' Total number of Countries in dataset : { df["Country"].nunique()} and number of States : {df.State.nunique()} ')

 Total number of Countries in dataset : 4 and number of States : 45
```

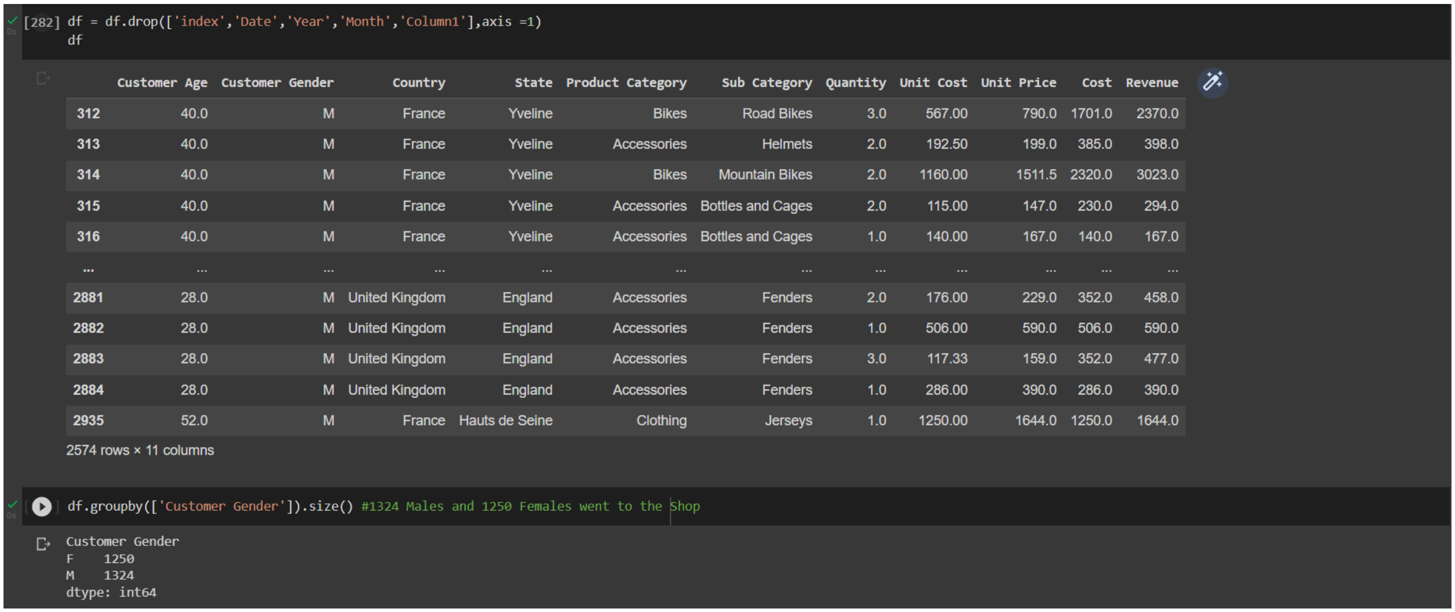
```
[280] df.isna().sum().sum()

32306
```



```
df = df.dropna() #Removing NaN datasets
df
```

	index	Date	Year	Month	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Unit Price	Cost	Revenue	Column1
312	312	01/11/16	2016.0	January	40.0	M	France	Yveline	Bikes	Road Bikes	3.0	567.00	790.0	1701.0	2370.0	2370.000000
313	313	01/11/16	2016.0	January	40.0	M	France	Yveline	Accessories	Helmets	2.0	192.50	199.0	385.0	398.0	398.000000
314	314	01/18/16	2016.0	January	40.0	M	France	Yveline	Bikes	Mountain Bikes	2.0	1160.00	1511.5	2320.0	3023.0	3023.000000
315	315	01/18/16	2016.0	January	40.0	M	France	Yveline	Accessories	Bottles and Cages	2.0	115.00	147.0	230.0	294.0	294.000000
316	316	01/18/16	2016.0	January	40.0	M	France	Yveline	Accessories	Bottles and Cages	1.0	140.00	167.0	140.0	167.0	167.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2881	2881	01/05/16	2016.0	January	28.0	M	United Kingdom	England	Accessories	Fenders	2.0	176.00	229.0	352.0	458.0	1971.000000
2882	2882	01/07/16	2016.0	January	28.0	M	United Kingdom	England	Accessories	Fenders	1.0	506.00	590.0	506.0	590.0	1583.000000
2883	2883	02/20/16	2016.0	February	28.0	M	United Kingdom	England	Accessories	Fenders	3.0	117.33	159.0	352.0	477.0	1682.000000
2884	2884	02/24/16	2016.0	February	28.0	M	United Kingdom	England	Accessories	Fenders	1.0	286.00	390.0	286.0	390.0	1568.000000
2935	2935	02/19/16	2016.0	February	52.0	M	France	Hauts de Seine	Clothing	Jerseys	1.0	1250.00	1644.0	1250.0	1644.0	687.344828

2574 rows × 16 columns



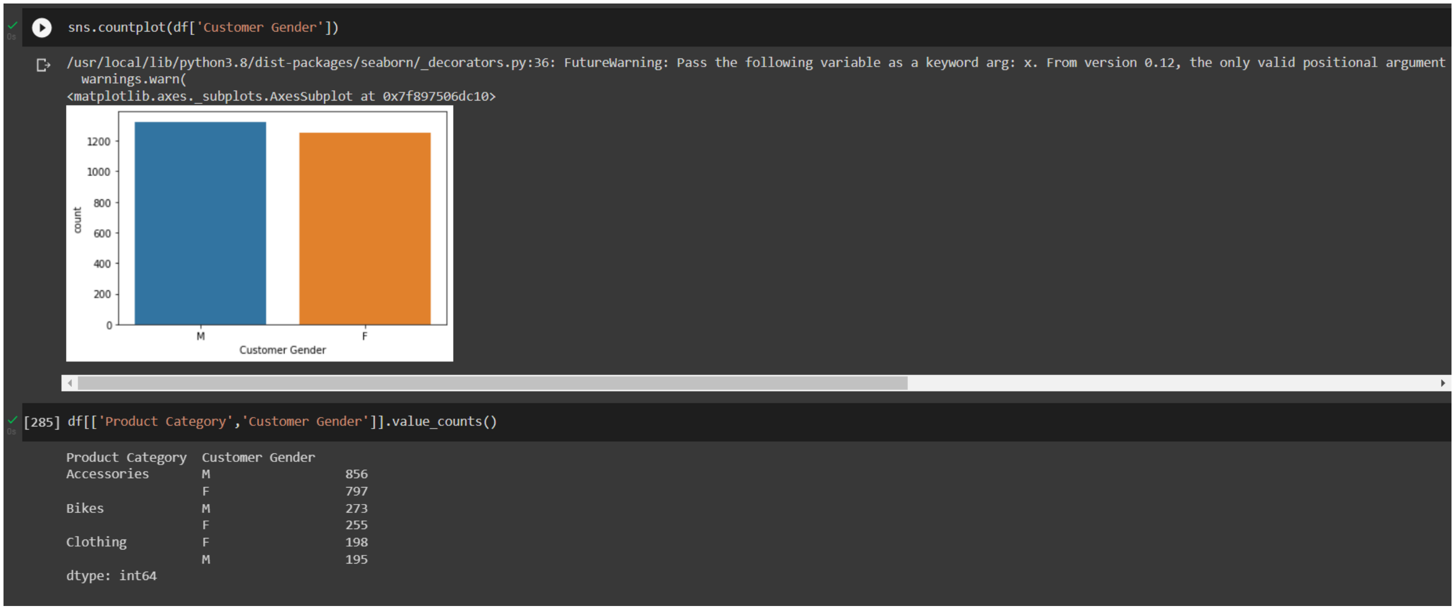
```
[282] df = df.drop(['index','Date','Year','Month','Column1'],axis =1)
      df
```

	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Unit Price	Cost	Revenue
312	40.0	M	France	Yveline	Bikes	Road Bikes	3.0	567.00	790.0	1701.0	2370.0
313	40.0	M	France	Yveline	Accessories	Helmets	2.0	192.50	199.0	385.0	398.0
314	40.0	M	France	Yveline	Bikes	Mountain Bikes	2.0	1160.00	1511.5	2320.0	3023.0
315	40.0	M	France	Yveline	Accessories	Bottles and Cages	2.0	115.00	147.0	230.0	294.0
316	40.0	M	France	Yveline	Accessories	Bottles and Cages	1.0	140.00	167.0	140.0	167.0
...	...	...	...	...	...	...	...	...	...	...	...
2881	28.0	M	United Kingdom	England	Accessories	Fenders	2.0	176.00	229.0	352.0	458.0
2882	28.0	M	United Kingdom	England	Accessories	Fenders	1.0	506.00	590.0	506.0	590.0
2883	28.0	M	United Kingdom	England	Accessories	Fenders	3.0	117.33	159.0	352.0	477.0
2884	28.0	M	United Kingdom	England	Accessories	Fenders	1.0	286.00	390.0	286.0	390.0
2935	52.0	M	France	Hauts de Seine	Clothing	Jerseys	1.0	1250.00	1644.0	1250.0	1644.0

2574 rows × 11 columns

```
df.groupby(['Customer Gender']).size() #1324 Males and 1250 Females went to the Shop
```

```
Customer Gender
F    1250
M    1324
dtype: int64
```

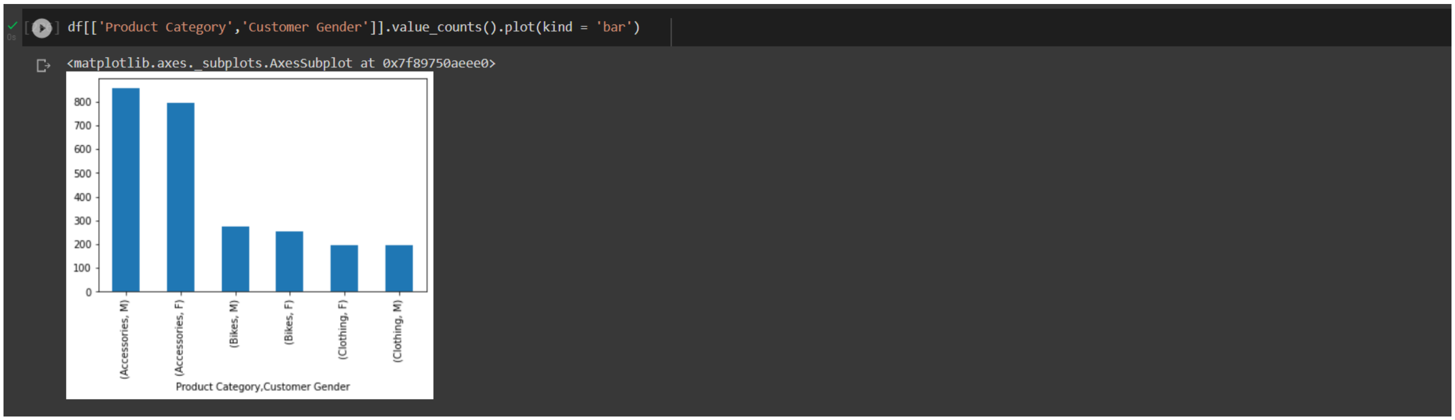


```
sns.countplot(df['Customer Gender'])
```

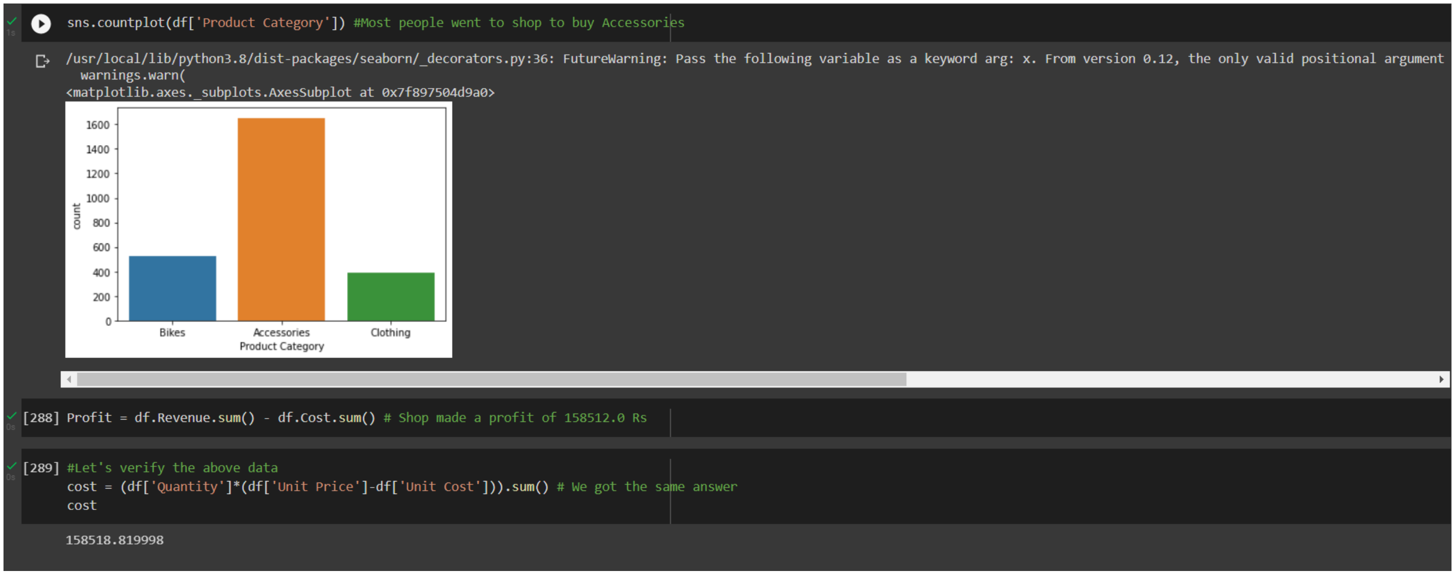
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument
  warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f897506dc10>
```

```
[285] df[['Product Category','Customer Gender']].value_counts()
```

```
Product Category  Customer Gender
Accessories       M                  856
                  F                  797
Bikes             M                  273
                  F                  255
Clothing          F                  198
                  M                  195
dtype: int64
```

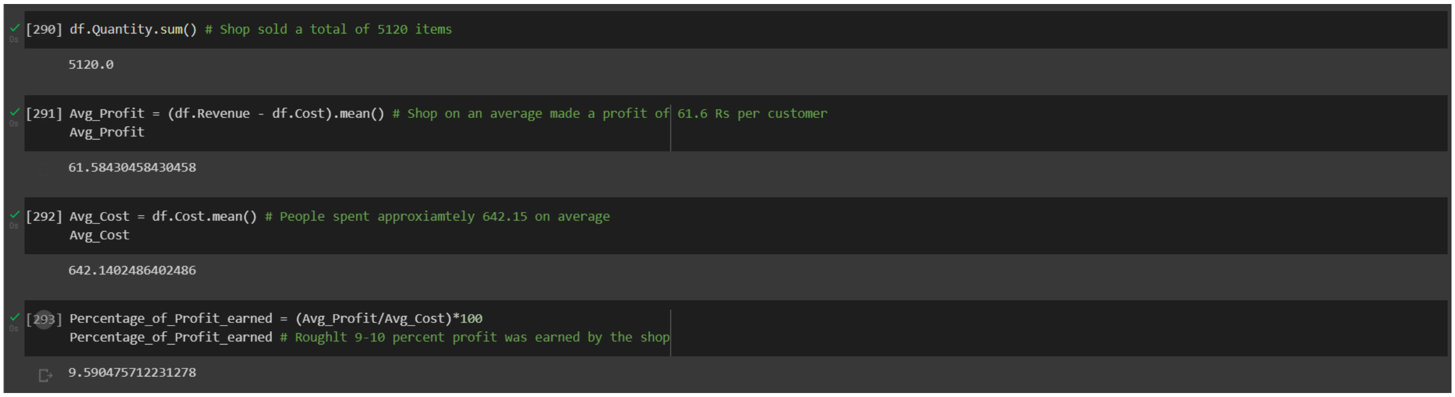


```
[ ] df[['Product Category','Customer Gender']].value_counts().plot(kind = 'bar')

<matplotlib.axes._subplots.AxesSubplot at 0x7f89750aeee0>
```

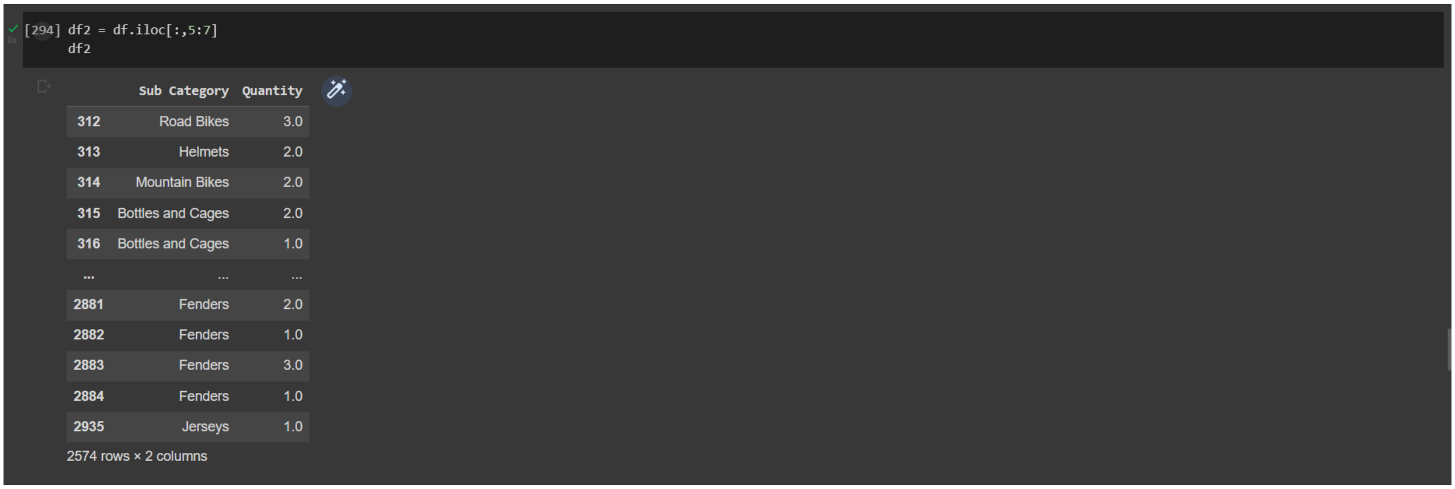


```
sns.countplot(df['Product Category']) #Most people went to shop to buy Accessories

/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument
  warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f897504d9a0>
```

```
[288] Profit = df.Revenue.sum() - df.Cost.sum() # Shop made a profit of 158512.0 Rs

[289] #Let's verify the above data
      cost = (df['Quantity']*(df['Unit Price']-df['Unit Cost'])).sum() # We got the same answer
      cost

      158518.819998
```



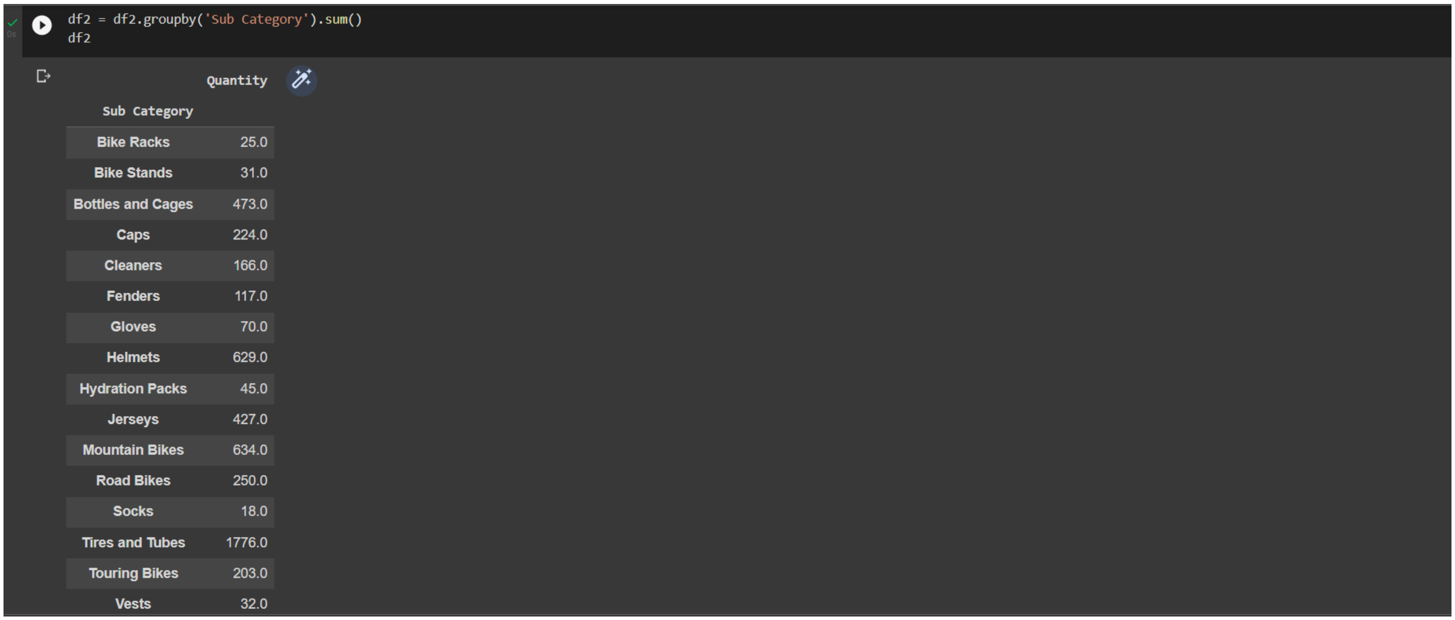
```
[290] df.Quantity.sum() # Shop sold a total of 5120 items

      5120.0

[291] Avg_Profit = (df.Revenue - df.Cost).mean() # Shop on an average made a profit of 61.6 Rs per customer
      Avg_Profit

      61.58430458430458

[292] Avg_Cost = df.Cost.mean() # People spent approxiamtely 642.15 on average
      Avg_Cost

      642.1402486402486

[293] Percentage_of_Profit_earned = (Avg_Profit/Avg_Cost)*100
      Percentage_of_Profit_earned # Roughlt 9-10 percent profit was earned by the shop

      9.590475712231278
```



```
[294] df2 = df.iloc[:,5:7]
      df2
```

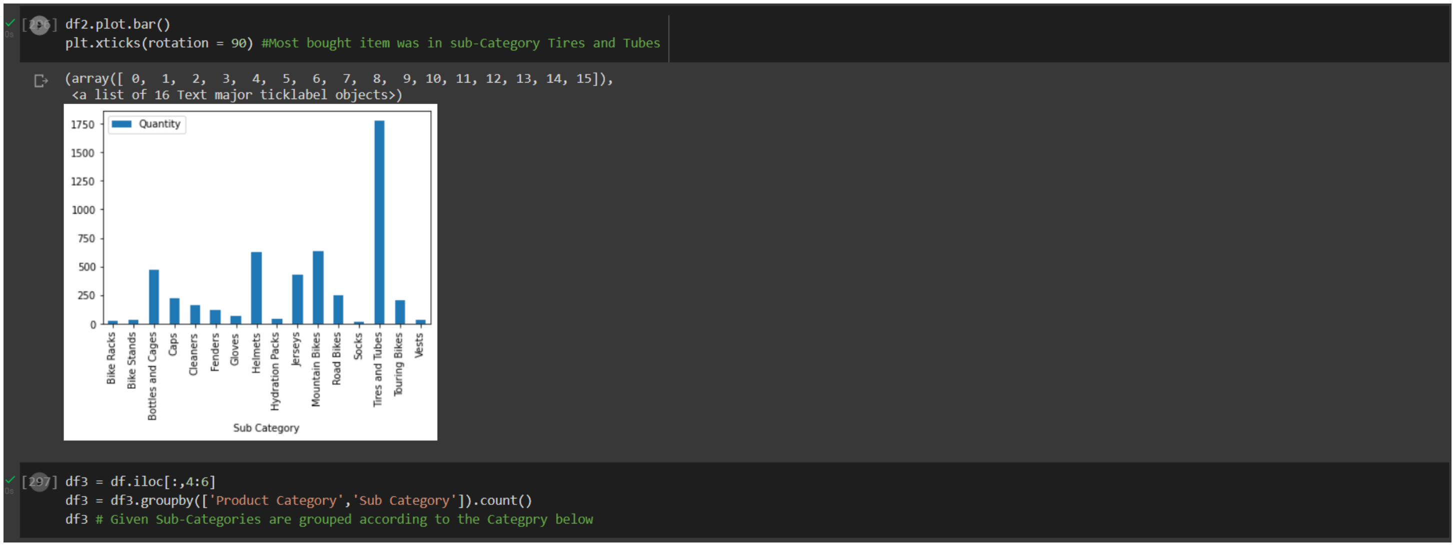
	Sub Category	Quantity
312	Road Bikes	3.0
313	Helmets	2.0
314	Mountain Bikes	2.0
315	Bottles and Cages	2.0
316	Bottles and Cages	1.0
...	...	...
2881	Fenders	2.0
2882	Fenders	1.0
2883	Fenders	3.0
2884	Fenders	1.0
2935	Jerseys	1.0

2574 rows × 2 columns



```
df2 = df2.groupby('Sub Category').sum()
df2
```

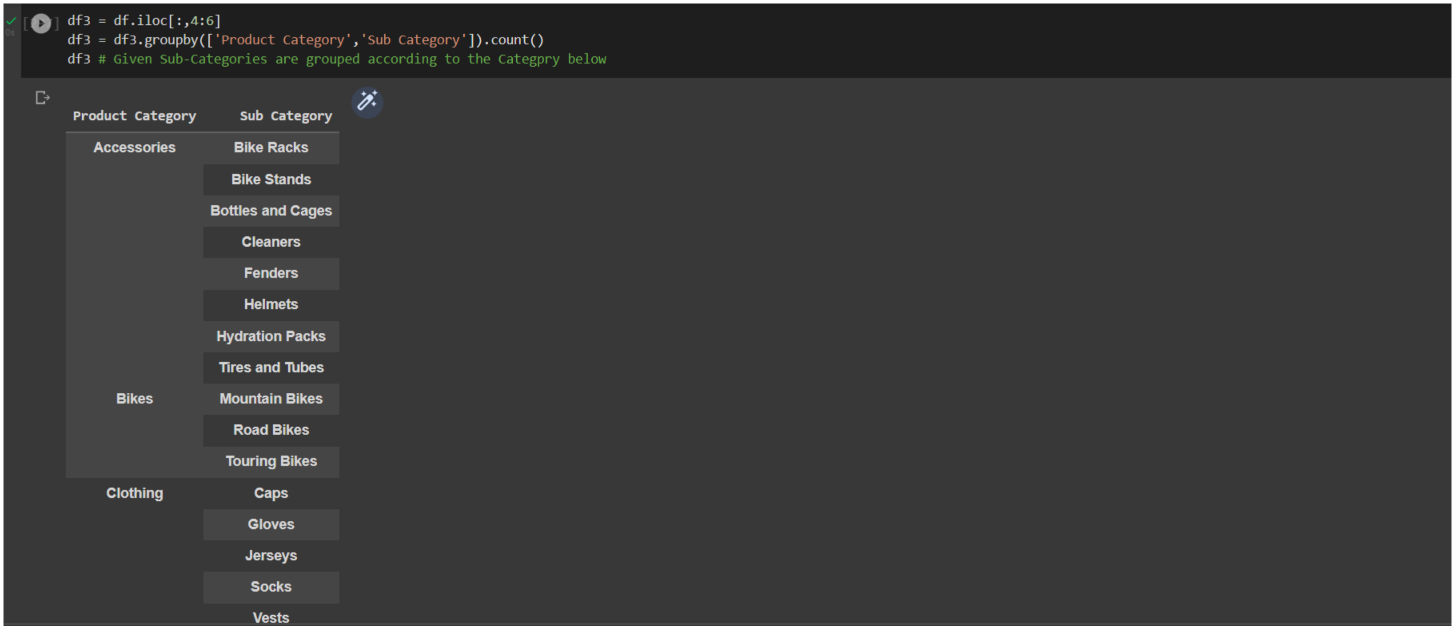
	Quantity
Sub Category	
Bike Racks	25.0
Bike Stands	31.0
Bottles and Cages	473.0
Caps	224.0
Cleaners	166.0
Fenders	117.0
Gloves	70.0
Helmets	629.0
Hydration Packs	45.0
Jerseys	427.0
Mountain Bikes	634.0
Road Bikes	250.0
Socks	18.0
Tires and Tubes	1776.0
Touring Bikes	203.0
Vests	32.0



```
df2.plot.bar()
plt.xticks(rotation = 90) #Most bought item was in sub-Category Tires and Tubes
```

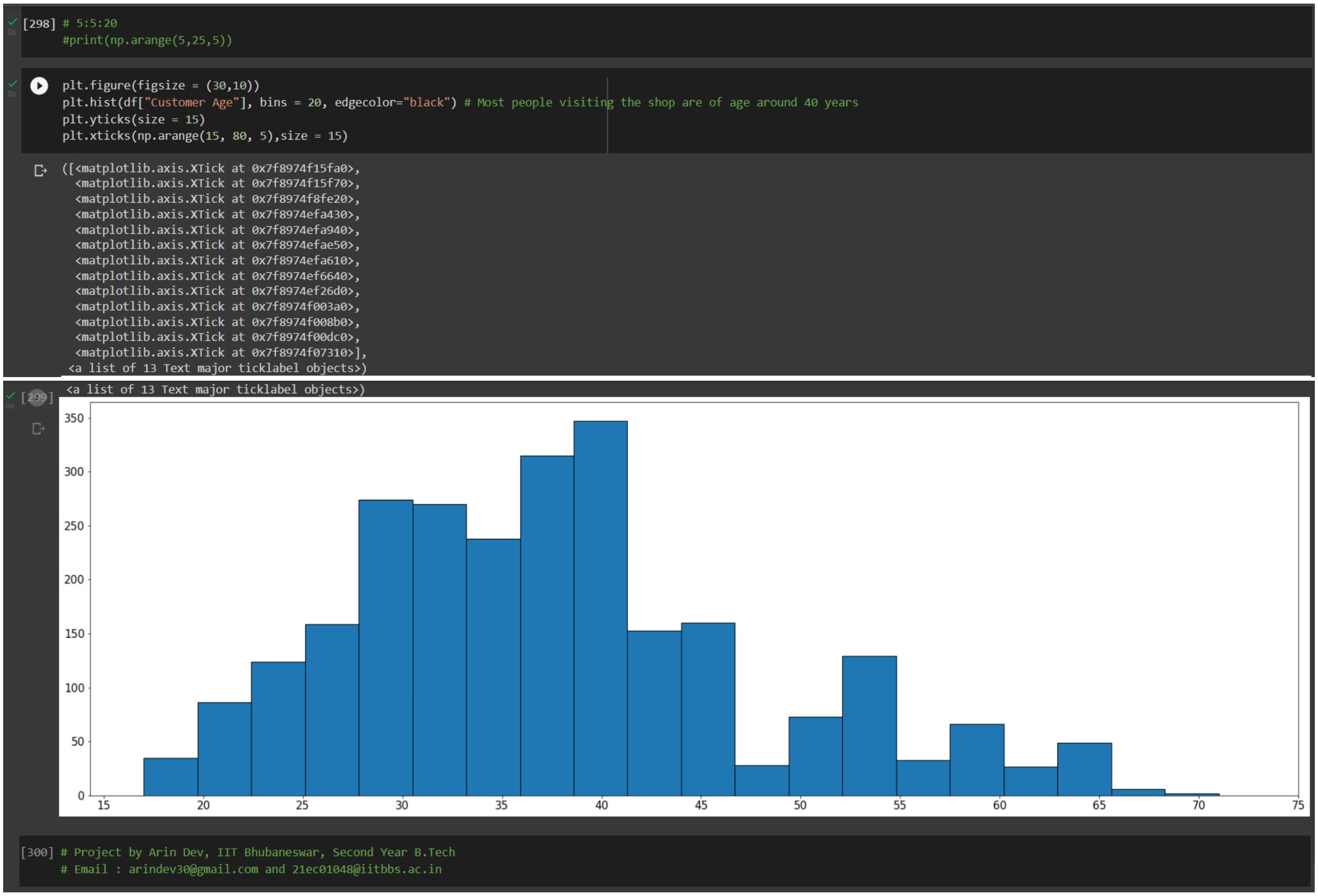
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),
 <a list of 16 Text major ticklabel objects>)
```

```
[297] df3 = df.iloc[:,4:6]
      df3 = df3.groupby(['Product Category','Sub Category']).count()
      df3 # Given Sub-Categories are grouped according to the Categpry below
```



```
df3 = df.iloc[:,4:6]
df3 = df3.groupby(['Product Category','Sub Category']).count()
df3 # Given Sub-Categories are grouped according to the Categpry below
```

Product Category	Sub Category
Accessories	Bike Racks
	Bike Stands
	Bottles and Cages
	Cleaners
	Fenders
	Helmets
	Hydration Packs
	Tires and Tubes
Bikes	Mountain Bikes
	Road Bikes
	Touring Bikes
Clothing	Caps
	Gloves
	Jerseys
	Socks
	Vests



```
[298] # 5:5:20
      #print(np.arange(5,25,5))
```

```
plt.figure(figsize = (30,10))
plt.hist(df["Customer Age"], bins = 20, edgecolor="black") # Most people visiting the shop are of age around 40 years
plt.yticks(size = 15)
plt.xticks(np.arange(15, 80, 5),size = 15)
```

```
([<matplotlib.axis.XTick at 0x7f8974f15fa0>,
  <matplotlib.axis.XTick at 0x7f8974f15f70>,
  <matplotlib.axis.XTick at 0x7f8974f8fe20>,
  <matplotlib.axis.XTick at 0x7f8974efa430>,
  <matplotlib.axis.XTick at 0x7f8974efa940>,
  <matplotlib.axis.XTick at 0x7f8974efae50>,
  <matplotlib.axis.XTick at 0x7f8974efa610>,
  <matplotlib.axis.XTick at 0x7f8974ef6640>,
  <matplotlib.axis.XTick at 0x7f8974ef26d0>,
  <matplotlib.axis.XTick at 0x7f8974f003a0>,
  <matplotlib.axis.XTick at 0x7f8974f008b0>,
  <matplotlib.axis.XTick at 0x7f8974f00dc0>,
  <matplotlib.axis.XTick at 0x7f8974f07310>],
 <a list of 13 Text major ticklabel objects>)
```

[299] <a list of 13 Text major ticklabel objects>)

```
[300] # Project by Arin Dev, IIT Bhubaneswar, Second Year B.Tech
      # Email : arindev30@gmail.com and 21ec01048@iitbbs.ac.in
```

Project by : Arin Dev (IIT Bhubaneswar, Second Year B.Tech student)
Email : arindev30@gmail.com and 21ec01048@iitbbs.ac.in

Link to this Notebook :
https://colab.research.google.com/drive/1D5zsIMZpglDyw_cxxAWwZP8c828KO65f?usp=sharing

Link to my Colab Notebooks :
https://drive.google.com/drive/folders/1WyxyVfRdAYuGWbqHI9O3rFCyMN0RSo1t?usp=sharing

Link to my GitHub : https://github.com/arin-dev/Rinex