DHV LAB Sheet-8 using student dataset

```
#Import Python Libraries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Enable inline plotting
%matplotlib inline


import pandas as pd
df = pd.read_csv('CSG1STUDENTDATA.csv', encoding= 'unicode_escape')
```

df

	Roll No	Student Name	Gender	Python	Java	AI	CG	SE	DHV	Total
0	20201CSG0001	S NAMRATHA	F	99	97	96	71	61	83	507
1	20201CSG0003	NEHA R	F	75	67	71	95	62	79	449
2	20201CSG0004	HARSHITHA C	F	96	66	66	60	87	94	469
3	20201CSG0005	MAHANTH S	F	86	71	92	60	71	90	470
4	20201CSG0006	RAKSHITHA N	F	92	57	88	83	63	88	471
5	20201CSG0007	AKHIL R	M	65	85	96	73	95	75	489
6	20201CSG0008	RAFIYA NUSRATH D	F	66	97	75	84	85	91	498
7	20201CSG0009	ABHINEET GAUR	M	89	95	82	87	57	95	505
8	20201CSG0010	DEEKSHITHA S V	F	63	52	93	87	97	73	465
9	20201CSG0011	R CHARIS SUSANNA	M	76	91	66	72	61	76	442
10	20201CSG0016	NUTHANA D P	M	90	83	79	82	96	70	500
11	20201CSG0020	VIRENDAR S OZA	M	67	95	97	72	71	78	480
12	20201CSG0021	AFSA HOORIYA	F	83	67	72	62	75	79	438
13	20201CSG0022	CHAITHRA N Y	F	68	80	93	95	97	77	510
14	20201CSG0023	GAGAN GANAPATHY B B	M	60	81	92	77	100	91	501
15	20201CSG0025	JAYANTH S R	M	73	77	92	93	54	100	489
16	20201CSG0026	PRAJWAL P	M	93	53	97	90	98	70	501
17	20201CSG0027	PRAMOD N	M	91	77	74	75	76	93	486
18	20201CSG0029	SHARAJ O V	M	74	57	80	62	85	73	431
19	20201CSG0030	SURAJ KUMAR TRIPATHY	M	67	76	81	94	99	85	502
20	20201CSG0032	CHANDANA M S	F	92	75	83	70	65	82	467
21	20201CSG0033	ADIB MOHAMMED HAGALWADI	M	98	90	95	69	85	90	527
22	20201CSG0034	SAIMITHUN B	F	87	58	99	88	91	74	497
23	20201CSG0035	VAISHALI V	F	87	60	67	68	65	86	433
24	20201CSG0036	MEGHANA M	F	98	56	96	67	51	88	456
25	20201CSG0038	AMRUTHA UDAY	F	70	86	68	64	85	75	448

```
#Display a few first records
df.head(10)
```

	Roll No	Student Name	Gender	Python	Java	AI	CG	SE	DHV	Total
0	20201CSG0001	S NAMRATHA	F	99	97	96	71	61	83	507
1	20201CSG0003	NEHA R	F	75	67	71	95	62	79	449
2	20201CSG0004	HARSHITHA C	F	96	66	66	60	87	94	469

```
# Display structure of the data frame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Roll No       26 non-null     object
 1   Student Name  26 non-null     object
 2   Gender        26 non-null     object
 3   Python        26 non-null     int64
 4   Java          26 non-null     int64
 5   AI            26 non-null     int64
 6   CG            26 non-null     int64
 7   SE            26 non-null     int64
 8   DHV           26 non-null     int64
 9   Total         26 non-null     int64
dtypes: int64(7), object(3)
memory usage: 2.2+ KB
```

```
#Total number of elements in the Data Frame
df.size
```

```
260
```

```
#Number of rows and columns
df.shape
```

```
(26, 10)
```

```
#Output basic statistics for the numeric columns
df.describe()
```

|	Python	Java	AI	CG	SE	DHV	Total
count	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000
mean	80.961538	74.961538	84.230769	76.923077	78.153846	82.884615	478.115385
std	12.587234	14.649180	11.360661	11.644477	15.949150	8.561901	26.849696
min	60.000000	52.000000	66.000000	60.000000	51.000000	70.000000	431.000000
25%	68.500000	61.500000	74.250000	68.250000	63.500000	75.250000	458.250000
50%	84.500000	76.500000	85.500000	74.000000	80.500000	82.500000	483.000000
75%	91.750000	85.750000	94.500000	87.000000	94.000000	90.000000	500.750000
max	99.000000	97.000000	99.000000	95.000000	100.000000	100.000000	527.000000

```
#Show graphs withint Python notebook
%matplotlib inline
```

```
#Use matplotlib to draw a histogram of a salary data
plt.hist(df['Total'],bins=20, density=False)
```

```
(array([2., 1., 1., 2., 0., 1., 0., 3., 2., 0., 1., 1., 2., 2., 4., 2., 1.,
        0., 0., 1.])
```
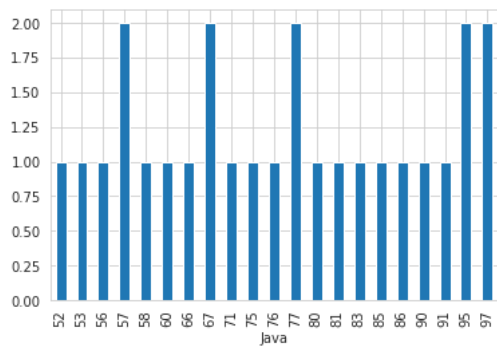
```
#Use seaborn package to draw a histogram
sns.distplot(df['Total']);
```

/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a de
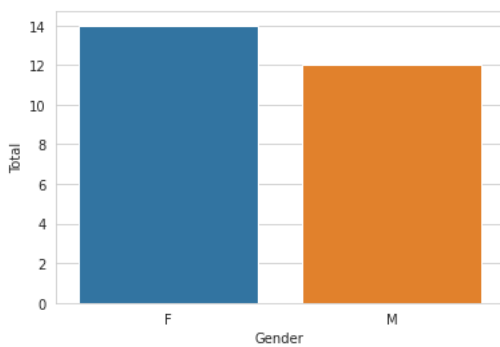 warnings.warn(msg, FutureWarning)
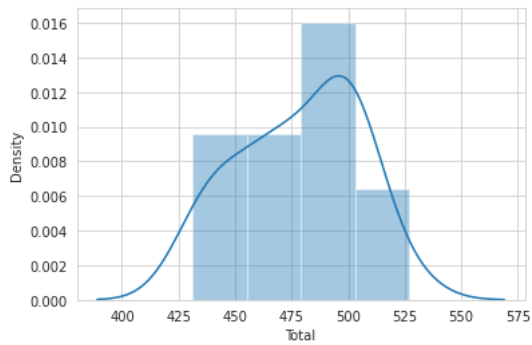

```
# Use regular matplotlib function to display a barplot
df.groupby(['Java'])['Total'].count().plot(kind='bar')
```
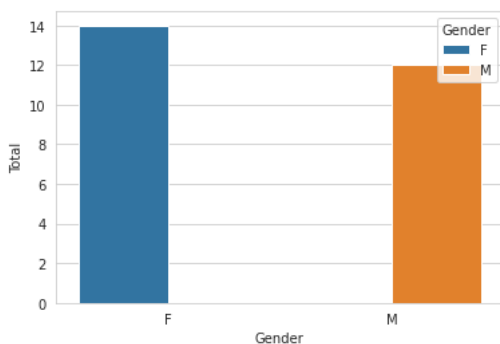
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa66996e160>
```



```
# Use seaborn package to display a barplot
sns.set_style("whitegrid")
```

```
ax = sns.barplot(x='Gender',y ='Total', data=df, estimator=len)
```



```
# Split into 2 groups:
ax = sns.barplot(x='Gender',y ='Total', hue='Gender', data=df, estimator=len)
```
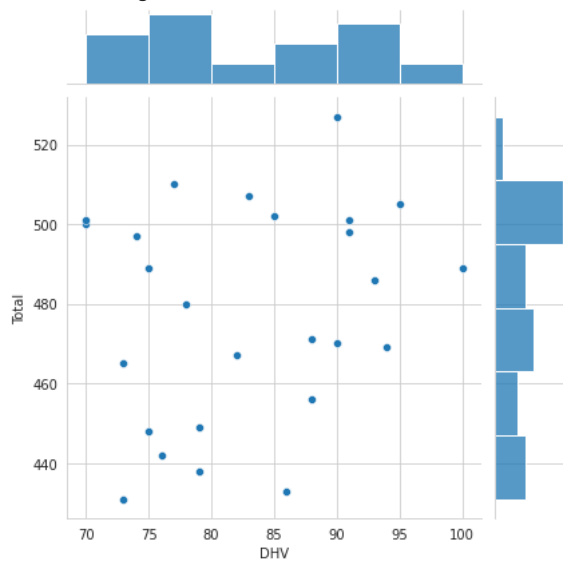
```
sns.scatterplot(x='Java', y='Total', data=df)
```

    <matplotlib.axes._subplots.AxesSubplot at 0x7fa669977c10>
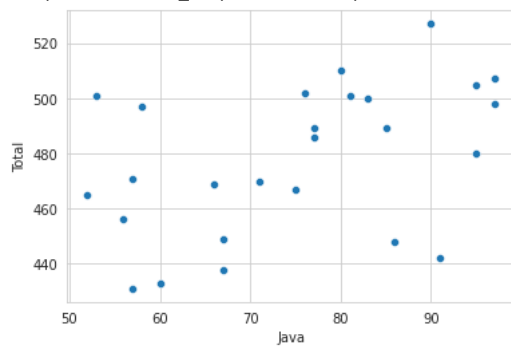

```
#Scatterplot in seaborn
sns.jointplot(x='DHV', y='Total', data=df)
```

    <seaborn.axisgrid.JointGrid at 0x7fa6697fdf10>

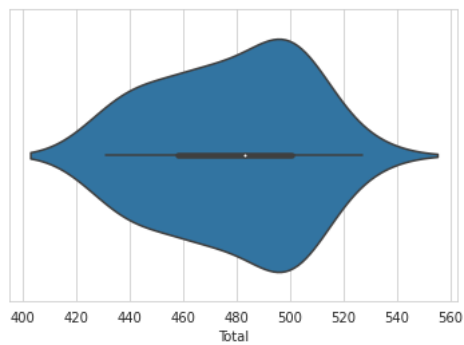
```
#Violinplot
sns.violinplot(x = "Total", data=df)
```

    <matplotlib.axes._subplots.AxesSubplot at 0x7fa669705100>


```
#If we are interested in linear regression plot for 2 numeric variables we can use regplot
sns.regplot(x='DHV', y='Total', data=df)
```
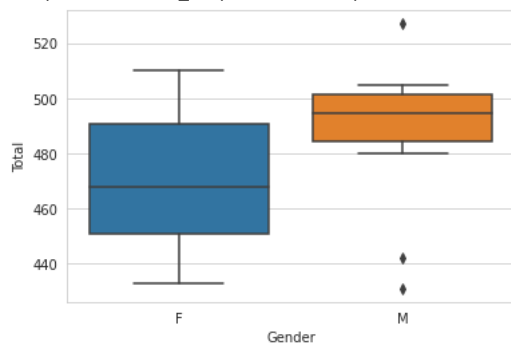
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa6696bbfd0>
```

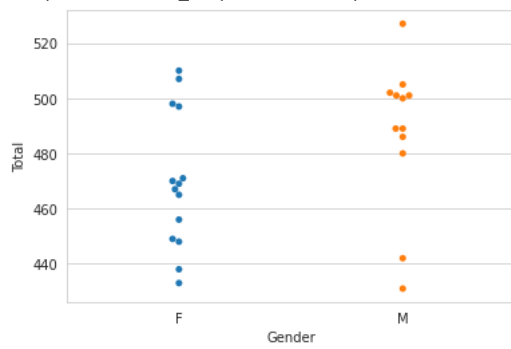# box plot
```
sns.boxplot(x='Gender',y='Total', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa6698d80d0>
```

swarm plot
```
sns.swarmplot(x='Gender',y='Total', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa6695fc9a0>
```

Pairplot
```
sns.pairplot(df)
```

<seaborn.axisgrid.PairGrid at 0x7fa66963cb50>