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


# Creating dataset
data = {
 'ID':[1,2,3,4,5,6,7,8,9,10],
 'Name':['Paul','Ryan','Ron','Jenny','Mike','Jordan','Angela','Jessica','Harmon','Murphy'],
 'Maths':[88,76,99,56,76,9,66,98,12,24],
 'Science':[67,87,99,36,98,12,67,29,23,56],
 'Exam_Eligible':['Yes','Yes','Yes','Yes','Yes','No','Yes','Yes','No','Yes']
}


df = pd.DataFrame(data)


print(df)
```

```
   ID     Name  Maths  Science Exam_Eligible
0   1     Paul     88       67           Yes
1   2     Ryan     76       87           Yes
2   3      Ron     99       99           Yes
3   4    Jenny     56       36           Yes
4   5     Mike     76       98           Yes
5   6   Jordan      9       12            No
6   7   Angela     66       67           Yes
7   8  Jessica     98       29           Yes
8   9   Harmon     12       23            No
9  10   Murphy     24       56           Yes
```

```
#Scannig the dataset for any NULL value
df.isnull()
```

	ID	Name	Maths	Science	Exam_Eligible
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
ID	10.0	5.5	3.027650	1.0	3.25	5.5	7.75	10.0
Maths	10.0	60.4	34.176665	9.0	32.00	71.0	85.00	99.0
Science	10.0	57.4	31.535518	12.0	30.75	61.5	82.00	99.0
Exam_Eligible	10.0	0.8	0.421637	0.0	1.00	1.0	1.00	1.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   ID             10 non-null     int64
 1   Name           10 non-null     object
 2   Maths          10 non-null     int64
```

```
      3   Science        10 non-null     int64
      4   Exam_Eligible  10 non-null     uint8
     dtypes: int64(3), object(1), uint8(1)
     memory usage: 458.0+ bytes
```

```
#Scan all numeric variables for outliers.
#Consider Highest marks and lowest marks as outliers.
print(type(df['Maths'][0]))
print(type(df['Science'][0]))
```

```
     <class 'numpy.int64'>
     <class 'numpy.int64'>
```

```
#Searching for outliers in Maths
print('Max :', df['Maths'].max())
print('Min :', df['Maths'].min())
```

```
     Max : 99
     Min : 9
```

```
#Searching for outliers in Science
print('Max :', df['Science'].max())
print('Min :', df['Science'].min())
```

```
     Max : 99
     Min : 12
```

```
df['Maths'].value_counts()
```

```
     76    2
     88    1
     99    1
     56    1
     9     1
     66    1
     98    1
     12    1
     24    1
     Name: Maths, dtype: int64
```

```
df['Science'].value_counts()
```

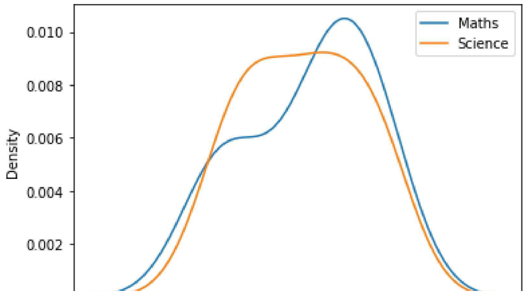
```
     67    2
     87    1
     99    1
     36    1
     98    1
     12    1
     29    1
     23    1
     56    1
     Name: Science, dtype: int64
```

```
#Finding student with highest and lowest marks
print('Highets Marks in Maths :')
print(df[df['Maths']==99])
print('Lowest Marks in Maths :')
print(df[df['Maths']==9])
```

```
     Highets Marks in Maths :
        ID Name  Maths  Science  Exam_Eligible
     2   3  Ron     99       99              1
     Lowest Marks in Maths :
        ID    Name  Maths  Science  Exam_Eligible
     5   6  Jordan      9       12              0
```

```
# Vizualizing data(marks) density
sns.kdeplot(data=df['Maths'],label='Maths')
sns.kdeplot(data=df['Science'],label='Science')
plt.xlabel('Marks Out of 100')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7faf41f7ba90>
```



```
#Transforming the data to make the data more usable
df.head()
```

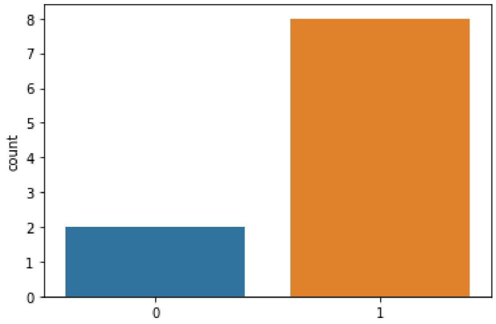
	ID	Name	Maths	Science	Exam_Eligible
0	1	Paul	88	67	1
1	2	Ryan	76	87	1
2	3	Ron	99	99	1
3	4	Jenny	56	36	1
4	5	Mike	76	98	1

```
# We can tranform column "Exam_Eligible" into Numerical Column
# because it is a Categorial Column.
df['Exam_Eligible'] = pd.get_dummies(df['Exam_Eligible'],drop_first=True)
df
```

	ID	Name	Maths	Science	Exam_Eligible
0	1	Paul	88	67	1
1	2	Ryan	76	87	1
2	3	Ron	99	99	1
3	4	Jenny	56	36	1
4	5	Mike	76	98	1
5	6	Jordan	9	12	0
6	7	Angela	66	67	1
7	8	Jessica	98	29	1
8	9	Harmon	12	23	0

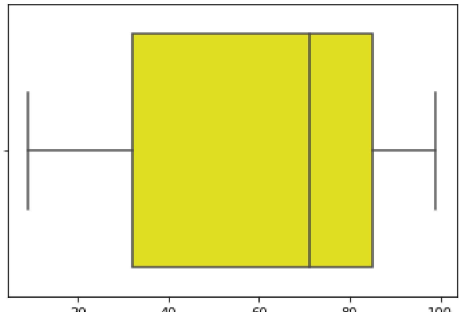
```
# 0 -> Not Eligible
# 1 -> Eligible
sns.countplot(data=df,x='Exam_Eligible')
```

```
<AxesSubplot:xlabel='Exam_Eligible', ylabel='count'>
```



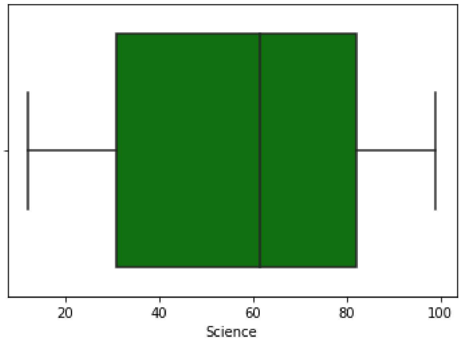
```
sns.boxplot(data=df,x='Maths',color='yellow')
```

```
<AxesSubplot:xlabel='Maths'>
```



```
sns.boxplot(data=df,x='Science',color='green')
```

```
<AxesSubplot:xlabel='Science'>
```



Colab paid products - Cancel contracts here

✓ 0s completed at 8:44 AM