

```
[202]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

1 Creating the dataset

```
[203]: rollno = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
name = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", np.nan, np.nan, "k", "l", "m"]
marks = [40, 23, 50, 78, 48, 89, 90, 67, 84, 96, 76, np.nan, 97, np.nan, 65]
grade = ["F", "F", "P", "P", "P", "P", "P", "P", "P", "P", "P", "P", "F", "P", np.nan, np.nan]
```

```
[204]: df = pd.DataFrame({"rollno" : rollno, "name" : name, "marks" : marks, "grade" : grade})
```

```
[205]: df
```

```
[205]:
```

	rollno	name	marks	grade
0	1	a	40.0	F
1	2	b	23.0	F
2	3	c	50.0	P
3	4	d	78.0	P
4	5	e	48.0	P
5	6	f	89.0	P
6	7	g	90.0	P
7	8	h	67.0	P
8	9	i	84.0	P
9	10	j	96.0	P
10	11	NaN	76.0	P
11	12	NaN	NaN	F

12	13	k	97.0	P
13	14	l	NaN	NaN
14	15	m	65.0	NaN

2 Dataset Statistics

```
[206]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0  rollno   15 non-null      int64
 1  name     13 non-null      object
 2  marks    13 non-null      float64
 3  grade    13 non-null      object
dtypes: float64(1), int64(1), object(2)
memory usage: 608.0+ bytes
```

```
[207]: df.describe()
```

```
[207]:
```

	rollno	marks
count	15.000000	13.000000
mean	8.000000	69.461538
std	4.472136	23.247277
min	1.000000	23.000000
25%	4.500000	50.000000
50%	8.000000	76.000000
75%	11.500000	89.000000
max	15.000000	97.000000

```
[208]: df.dtypes
```

```
[208]: rollno      int64
name         object
marks        float64
grade        object
dtype: object
```

```
[209]: df.columns
```

```
[209]: Index(['rollno', 'name', 'marks', 'grade'], dtype='object')
```

```
[210]: df.isna().sum()
```

```
[210]: rollno    0
      name      2
      marks     2
      grade     2
      dtype: int64
```

```
[211]: df.to_csv("academic_performance.csv")
```

3 Null values

```
[212]: df.isna().sum()
```

```
[212]: rollno    0
      name      2
      marks     2
      grade     2
      dtype: int64
```

```
[213]: df["marks"] = df["marks"].fillna(df["marks"].mean())
```

```
[214]: df
```

```
[214]:
```

	rollno	name	marks	grade
0	1	a	40.000000	F
1	2	b	23.000000	F
2	3	c	50.000000	P
3	4	d	78.000000	P
4	5	e	48.000000	P
5	6	f	89.000000	P
6	7	g	90.000000	P
7	8	h	67.000000	P
8	9	i	84.000000	P
9	10	j	96.000000	P
10	11	NaN	76.000000	P
11	12	NaN	69.461538	F
12	13	k	97.000000	P
13	14	l	69.461538	NaN
14	15	m	65.000000	NaN

```
[215]: def fun1(value):
      return int(math.floor(value))
```

```
[216]: df["marks"] = df["marks"].apply(fun1)
```

```
[217]: df
```

```
[217]:
```

	rollno	name	marks	grade
0	1	a	40	F
1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
10	11	NaN	76	P
11	12	NaN	69	F
12	13	k	97	P
13	14	l	69	NaN
14	15	m	65	NaN

```
[218]: df = df[df['name'].notna()]
```

```
[219]: df
```

```
[219]:
```

	rollno	name	marks	grade
0	1	a	40	F
1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
12	13	k	97	P
13	14	l	69	NaN
14	15	m	65	NaN

```
[220]: for index, row in df.iterrows():
        # print(row['marks'], row['grade'])
        if (row['marks'] > 40):
            df.loc[index, 'grade'] = 'P'
        else:
            df.loc[index, 'grade'] = 'F'
```

```
[221]: df
```

```
[221]:
```

	rollno	name	marks	grade
0	1	a	40	F

1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
12	13	k	97	P
13	14	l	69	P
14	15	m	65	P

4 Outliers

```
[222]: first_outlier = [16, 'n', 200, 'P']
       second_outlier = [17, 'o', -100, 'F']
```

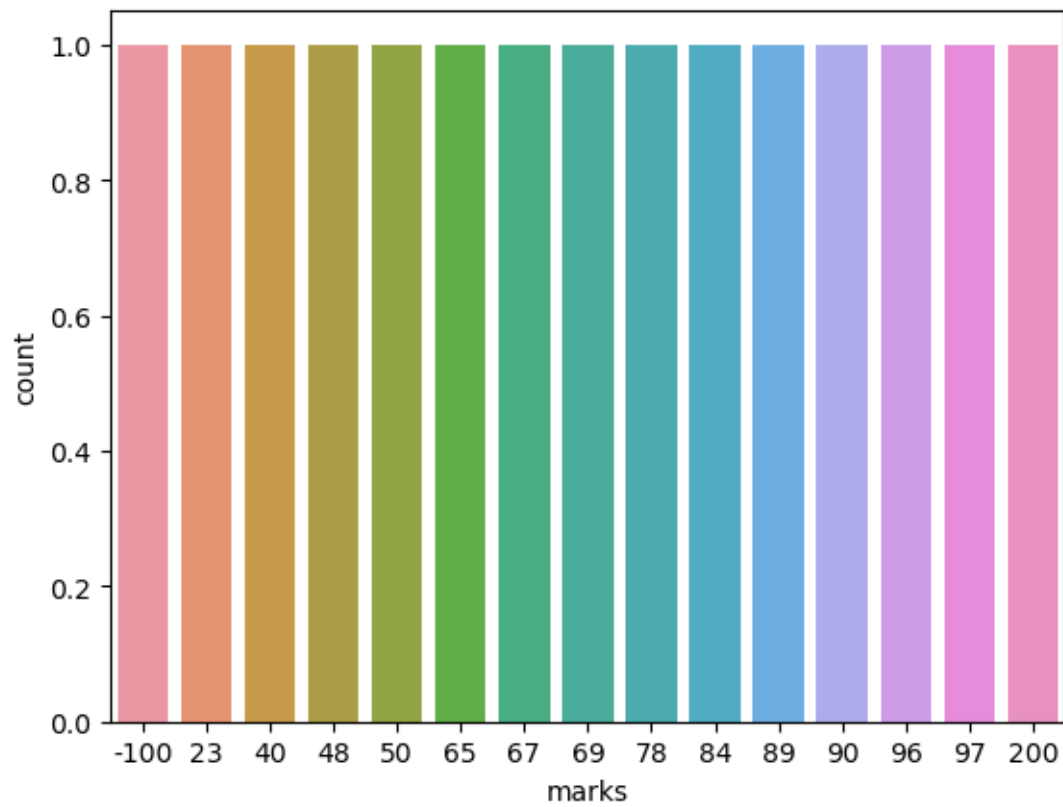
```
[223]: df.loc[15] = first_outlier
       df.loc[16] = second_outlier
```

```
[224]: df
```

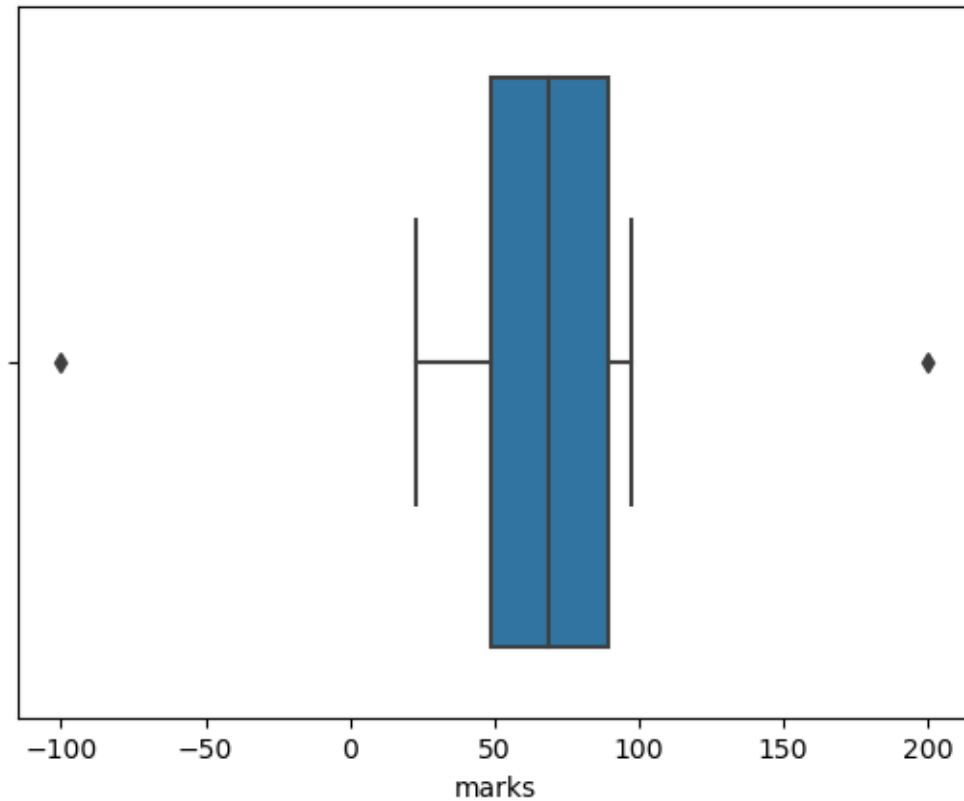
```
[224]:
```

	rollno	name	marks	grade
0	1	a	40	F
1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
12	13	k	97	P
13	14	l	69	P
14	15	m	65	P
15	16	n	200	P
16	17	o	-100	F

```
[225]: sns.countplot(data=df, x=df['marks']);
```



```
[226]: sns.boxplot(data=df, x='marks');
```



```
[227]: from matplotlib.cbook import boxplot_stats
outliers = boxplot_stats(df['marks']).pop(0)['fliers']
outliers
```

```
[227]: array([-100, 200], dtype=int64)
```

```
[228]: df
```

```
[228]:
```

	rollno	name	marks	grade
0	1	a	40	F
1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
12	13	k	97	P
13	14	l	69	P

14	15	m	65	P
15	16	n	200	P
16	17	o	-100	F

```
[229]: df = df.drop([15,16], axis=0)
```

```
[230]: df
```

```
[230]:
```

	rollno	name	marks	grade
0	1	a	40	F
1	2	b	23	F
2	3	c	50	P
3	4	d	78	P
4	5	e	48	P
5	6	f	89	P
6	7	g	90	P
7	8	h	67	P
8	9	i	84	P
9	10	j	96	P
12	13	k	97	P
13	14	l	69	P
14	15	m	65	P

5 Scaling the marks column

```
[231]: from sklearn.preprocessing import MinMaxScaler
```

```
[232]: scaler = MinMaxScaler()
```

```
[234]: df[['marks']] = scaler.fit_transform(df[['marks']])
```

```
[235]: df
```

```
[235]:
```

	rollno	name	marks	grade
0	1	a	0.229730	F
1	2	b	0.000000	F
2	3	c	0.364865	P
3	4	d	0.743243	P
4	5	e	0.337838	P
5	6	f	0.891892	P
6	7	g	0.905405	P
7	8	h	0.594595	P
8	9	i	0.824324	P
9	10	j	0.986486	P
12	13	k	1.000000	P
13	14	l	0.621622	P

14 15 m 0.567568 P