In [23]:
```
import pandas as pd
import numpy as np
import matplotlib as mt
import seaborn as sns
import math
```

In [24]:
```
rno = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 ]
name=["avi","kushi","reena","rivi","rushi","krushi","sia","ria","pia","ruhi"
marks=[20,40,20,30,np.nan,55,65,np.nan,np.nan,0,np.nan,12,33,79,55]
grade=["F","P","F","F","P","P","P","P","P","F","P","F","P",np.nan,np.nan]
```

In [25]:
```
df=pd.DataFrame({"rno":rno,"name":name,"marks":marks,"grade":grade})
```

In [26]:
```
df
# count This shows the number of non-null values in each numerical column
#mean
#std ,
#25%= Q1  value below which 25% of the data falls.
# 50% median
#75%= Q3   value below which 75% of the data falls.
#max and min val in that column
```

Out[26]:

	rno	name	marks	grade
0	1	avi	20.0	F
1	2	kushi	40.0	P
2	3	reena	20.0	F
3	4	rivi	30.0	F
4	5	rushi	NaN	P
5	6	krushi	55.0	P
6	7	sia	65.0	P
7	8	ria	NaN	P
8	9	pia	NaN	P
9	10	ruhi	0.0	F
10	11	anya	NaN	P
11	12	ananya	12.0	F
12	13	rina	33.0	P
13	14	rinku	79.0	NaN
14	15	dinki	55.0	NaN

In [27]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   rno     15 non-null     int64
 1   name    15 non-null     object
 2   marks   11 non-null     float64
 3   grade   13 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 612.0+ bytes
```

In [28]:
```
df.describe()
```

Out[28]:

	rno	marks
count	15.000000	11.000000
mean	8.000000	37.181818
std	4.472136	24.169477
min	1.000000	0.000000
25%	4.500000	20.000000
50%	8.000000	33.000000
75%	11.500000	55.000000
max	15.000000	79.000000

In [29]:
```
df.isnull()
```

Out[29]:

	rno	name	marks	grade
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	True	False
5	False	False	False	False
6	False	False	False	False
7	False	False	True	False
8	False	False	True	False
9	False	False	False	False
10	False	False	True	False
11	False	False	False	False
12	False	False	False	False
13	False	False	False	True
14	False	False	False	True

In [30]:
```
df.isnull().sum()
```

Out[30]:
```
rno      0
name     0
marks    4
grade    2
dtype: int64
```

In [31]:
```
df.shape
```

Out[31]:
```
(15, 4)
```

In [32]:
```
df.dtypes
```

Out[32]:
```
rno        int64
name      object
marks    float64
grade     object
dtype: object
```

In [33]:
```
df.columns
```

Out[33]:
```
Index(['rno', 'name', 'marks', 'grade'], dtype='object')
```

In [34]:
```
df.isna().sum()
```

Out[34]:
```
rno      0
name     0
marks    4
grade    2
dtype: int64
```

In [35]:
```
df.to_csv("academic_performance.csv")
```

In [36]:
```
df["marks"]=df["marks"].fillna(df["marks"].mean())
df
```

Out[36]:

	rno	name	marks	grade
0	1	avi	20.000000	F
1	2	kushi	40.000000	P
2	3	reena	20.000000	F
3	4	rivi	30.000000	F
4	5	rushi	37.181818	P
5	6	krushi	55.000000	P
6	7	sia	65.000000	P
7	8	ria	37.181818	P
8	9	pia	37.181818	P
9	10	ruhi	0.000000	F
10	11	anya	37.181818	P
11	12	ananya	12.000000	F
12	13	rina	33.000000	P
13	14	rinku	79.000000	NaN
14	15	dinki	55.000000	NaN

In [37]:
```
def fun1(value):
    return int(math.floor(value))
```

In [38]:
```
df["marks"]=df["marks"].apply(fun1)
df
```

Out[38]:

	rno	name	marks	grade
0	1	avi	20	F
1	2	kushi	40	P
2	3	reena	20	F
3	4	rivi	30	F
4	5	rushi	37	P
5	6	krushi	55	P
6	7	sia	65	P
7	8	ria	37	P
8	9	pia	37	P
9	10	ruhi	0	F
10	11	anya	37	P
11	12	ananya	12	F
12	13	rina	33	P
13	14	rinku	79	NaN
14	15	dinki	55	NaN

In [39]:
```
for index,row in df.iterrows():
    if(row['marks']>40):
        df.loc[index,'grade']='P'
    else:
        df.loc[index,'grade']='F'
```

In [40]:
```
df
```

Out[40]:

	rno	name	marks	grade
0	1	avi	20	F
1	2	kushi	40	F
2	3	reena	20	F
3	4	rivi	30	F
4	5	rushi	37	F
5	6	krushi	55	P
6	7	sia	65	P
7	8	ria	37	F
8	9	pia	37	F
9	10	ruhi	0	F
10	11	anya	37	F
11	12	ananya	12	F
12	13	rina	33	F
13	14	rinku	79	P
14	15	dinki	55	P

In [41]:
```
outlier_one=[16,'nalin',290,'P']
outlier_two=[17,'nitin',-100,'F']
outlier_three=[18,'nrinn',500,'P']
outlier_four=[19,'niki',-120,'F']
```

In [42]:
```
df.loc[16]=outlier_one
df.loc[17]=outlier_two
df.loc[18]=outlier_three
df.loc[19]=outlier_four
```

In [43]:
```
df
```

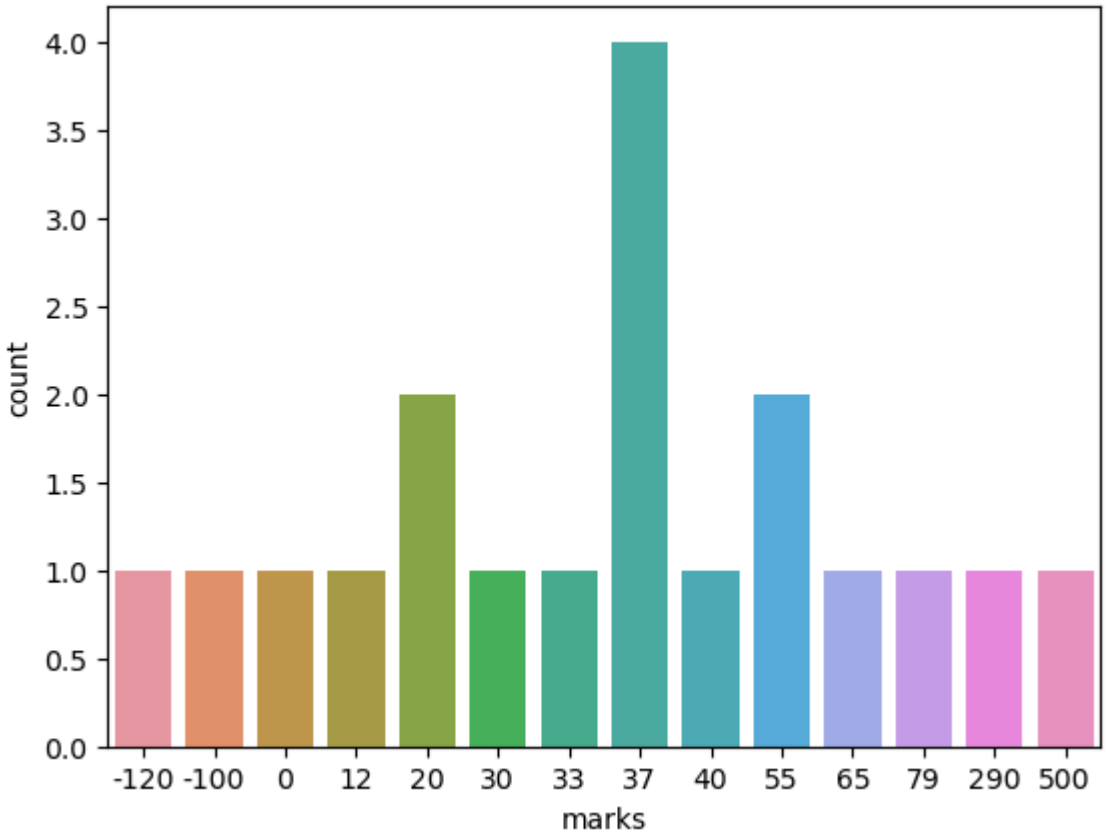
Out[43]:

	rno	name	marks	grade
0	1	avi	20	F
1	2	kushi	40	F
2	3	reena	20	F
3	4	rivi	30	F
4	5	rushi	37	F
5	6	krushi	55	P
6	7	sia	65	P
7	8	ria	37	F
8	9	pia	37	F
9	10	ruhi	0	F
10	11	anya	37	F
11	12	ananya	12	F
12	13	rina	33	F
13	14	rinku	79	P
14	15	dinki	55	P
16	16	nalin	290	P
17	17	nitin	-100	F
18	18	nrinn	500	P
19	19	niki	-120	F

In [44]:
```
print('Outliers:')
for index,row in df.iterrows():
    if(row['marks']>100):
        print(row['marks'])
    elif(row['marks']<0):
         print(row['marks'])
```

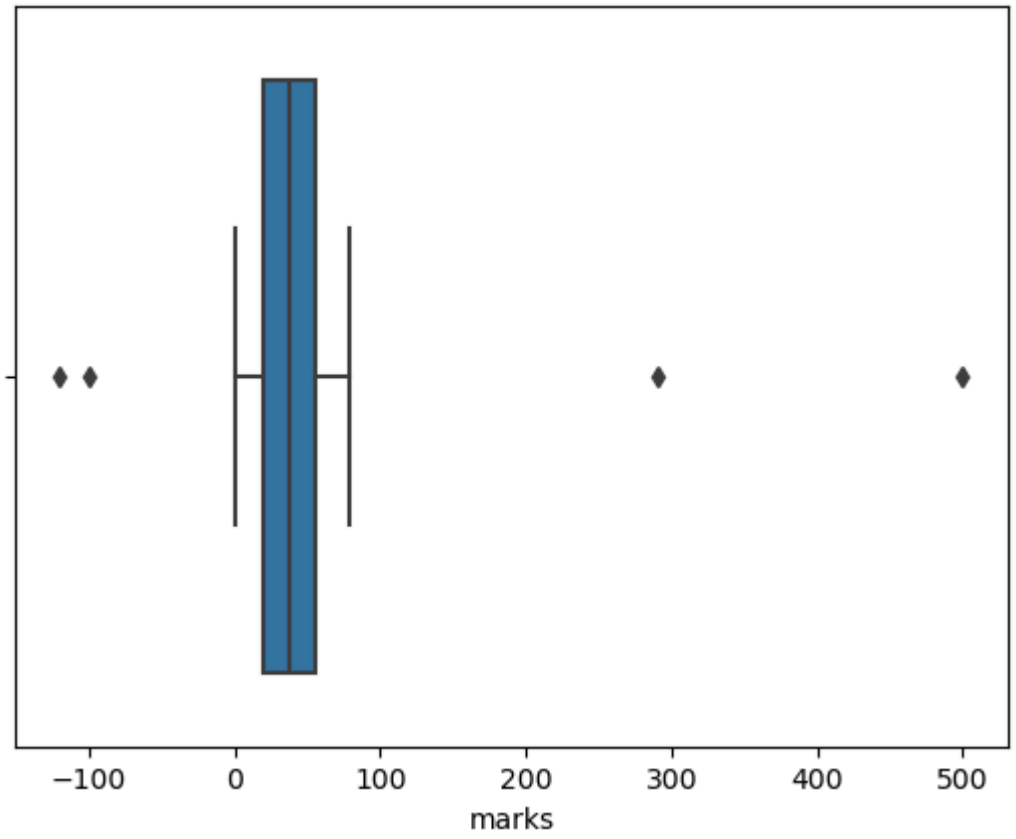
```
Outliers:
290
-100
500
-120
```

In [45]:
```
sns.countplot(data=df,x=df['marks']);
```



In [46]:
```
sns.boxplot(data=df, x='marks')
```

Out[46]: <Axes: xlabel='marks'>



In [47]:
```
df=df.drop([16,17,18,19],axis=0)
df
```

Out[47]:

	rno	name	marks	grade
0	1	avi	20	F
1	2	kushi	40	F
2	3	reena	20	F
3	4	rivi	30	F
4	5	rushi	37	F
5	6	krushi	55	P
6	7	sia	65	P
7	8	ria	37	F
8	9	pia	37	F
9	10	ruhi	0	F
10	11	anya	37	F
11	12	ananya	12	F
12	13	rina	33	F
13	14	rinku	79	P
14	15	dinki	55	P

In [48]:
```
#log-tranform
df['log_marks']=np.log1p(df['marks'])
df
```

Out[48]:

	rno	name	marks	grade	log_marks
0	1	avi	20	F	3.044522
1	2	kushi	40	F	3.713572
2	3	reena	20	F	3.044522
3	4	rivi	30	F	3.433987
4	5	rushi	37	F	3.637586
5	6	krushi	55	P	4.025352
6	7	sia	65	P	4.189655
7	8	ria	37	F	3.637586
8	9	pia	37	F	3.637586
9	10	ruhi	0	F	0.000000
10	11	anya	37	F	3.637586
11	12	ananya	12	F	2.564949
12	13	rina	33	F	3.526361
13	14	rinku	79	P	4.382027
14	15	dinki	55	P	4.025352

In [49]:
```
def fun1(value):
    floor_v=math.floor(value)
    decim_v=value-floor_v
    res=floor_v+round(decim_v,2)
    return res
```

In [50]:
```
df["log_marks"]=df["log_marks"].apply(fun1)
```

In [51]:
```
df
```

Out[51]:

	rno	name	marks	grade	log_marks
0	1	avi	20	F	3.04
1	2	kushi	40	F	3.71
2	3	reena	20	F	3.04
3	4	rivi	30	F	3.43
4	5	rushi	37	F	3.64
5	6	krushi	55	P	4.03
6	7	sia	65	P	4.19
7	8	ria	37	F	3.64
8	9	pia	37	F	3.64
9	10	ruhi	0	F	0.00
10	11	anya	37	F	3.64
11	12	ananya	12	F	2.56
12	13	rina	33	F	3.53
13	14	rinku	79	P	4.38
14	15	dinki	55	P	4.03