In [581]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
sd=pd.read_excel('StudentsPerformance1.xlsx')
sd
```

Out[581]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	Column	Column.1	test preparation course	math score	reading score	writing score
0	1	female	group B	bachelor's degree	standard	NaN	NaN	none	72.0	72.0	74.0
1	2	female	group C	Diploma	standard	NaN	NaN	completed	69.0	90.0	88.0
2	3	female	group B	master's degree	standard	NaN	NaN	none	90.0	95.0	93.0
3	4	male	group A	associate's degree	free/reduced	NaN	NaN	none	47.0	57.0	44.0
4	5	male	group C	Diploma	standard	NaN	NaN	none	76.0	78.0	75.0
...	...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	NaN	NaN	completed	88.0	99.0	95.0
1001	1002	male	group C	high school	free/reduced	NaN	NaN	none	62.0	55.0	55.0
1002	1003	female	group C	high school	free/reduced	NaN	NaN	completed	59.0	71.0	65.0
1003	1004	female	group D	Diploma	standard	NaN	NaN	completed	68.0	78.0	77.0
1004	1005	female	group D	Diploma	free/reduced	NaN	NaN	none	77.0	86.0	86.0

1005 rows × 11 columns

In [582]:

```
sd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Student ID                   1005 non-null   int64  
 1   gender                       1005 non-null   object 
 2   race/ethnicity               1005 non-null   object 
 3   parental level of education  1005 non-null   object 
 4   lunch                        1005 non-null   object 
 5   Column                       0 non-null      float64
 6   Column.1                     0 non-null      float64
 7   test preparation course      1005 non-null   object 
 8   math score                   1000 non-null   float64
 9   reading score                1000 non-null   float64
 10  writing score                1000 non-null   float64
dtypes: float64(5), int64(1), object(5)
memory usage: 86.5+ KB
```

In [583]:

```
sd1=sd.copy()
```

In [584]:

```
sd1.isna().sum()
```

Out[584]:

```
Student ID                     0
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
Column                      1005
Column.1                    1005
test preparation course        0
math score                     5
reading score                  5
writing score                  5
dtype: int64
```

In [585]:

```
sd1['Total Score']=sd1['math score']+sd1['reading score']+sd1['writing score']
sd1
```

Out[585]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	Column	Column.1	test preparation course	math score	reading score	writing score
0	1	female	group B	bachelor's degree	standard	NaN	NaN	none	72.0	72.0	74.0
1	2	female	group C	Diploma	standard	NaN	NaN	completed	69.0	90.0	88.0
2	3	female	group B	master's degree	standard	NaN	NaN	none	90.0	95.0	93.0
3	4	male	group A	associate's degree	free/reduced	NaN	NaN	none	47.0	57.0	44.0
4	5	male	group C	Diploma	standard	NaN	NaN	none	76.0	78.0	75.0
...	...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	NaN	NaN	completed	88.0	99.0	95.0
1001	1002	male	group C	high school	free/reduced	NaN	NaN	none	62.0	55.0	55.0
1002	1003	female	group C	high school	free/reduced	NaN	NaN	completed	59.0	71.0	65.0
1003	1004	female	group D	Diploma	standard	NaN	NaN	completed	68.0	78.0	77.0
1004	1005	female	group D	Diploma	free/reduced	NaN	NaN	none	77.0	86.0	86.0

1005 rows × 12 columns

In [ ]:

In [586]:

```
sd1.drop(['Column','Column.1'],axis=1,inplace=True)
sd1
```

Out[586]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0
...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	completed	88.0	99.0	95.0	282.0
1001	1002	male	group C	high school	free/reduced	none	62.0	55.0	55.0	172.0
1002	1003	female	group C	high school	free/reduced	completed	59.0	71.0	65.0	195.0
1003	1004	female	group D	Diploma	standard	completed	68.0	78.0	77.0	223.0
1004	1005	female	group D	Diploma	free/reduced	none	77.0	86.0	86.0	249.0

1005 rows × 10 columns

In [587]:

```
# 1).check whether null values are present in the given data.
sd1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Student ID                   1005 non-null   int64  
 1   gender                       1005 non-null   object 
 2   race/ethnicity               1005 non-null   object 
 3   parental level of education  1005 non-null   object 
 4   lunch                        1005 non-null   object 
 5   test preparation course      1005 non-null   object 
 6   math score                   1000 non-null   float64
 7   reading score                1000 non-null   float64
 8   writing score                1000 non-null   float64
 9   Total Score                  1000 non-null   float64
dtypes: float64(4), int64(1), object(5)
memory usage: 78.6+ KB
```

In [588]:

```
#sd.isna().sum()
sd1.median()
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\2039403970.py:2: FutureWarning: The default va
lue of numeric_only in DataFrame.median is deprecated. In a future version, it will default to F
alse. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or sp
ecify the value of numeric_only to silence this warning.
  sd1.median()
```

Out[588]:

```
Student ID       503.0
math score        66.0
reading score     70.0
writing score     69.0
Total Score      205.0
dtype: float64
```

In [589]:

```
#sd=sd.fillna().mean() # replace with median
#sd.head(3)
sd1.median()
sd2=sd1.fillna(sd.median())
sd2
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\885568831.py:3: FutureWarning: The default val
ue of numeric_only in DataFrame.median is deprecated. In a future version, it will default to Fa
lse. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or spe
cify the value of numeric_only to silence this warning.
  sd1.median()
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\885568831.py:4: FutureWarning: The default val
ue of numeric_only in DataFrame.median is deprecated. In a future version, it will default to Fa
lse. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or spe
cify the value of numeric_only to silence this warning.
  sd2=sd1.fillna(sd.median())
```

Out[589]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0
...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	completed	88.0	99.0	95.0	282.0
1001	1002	male	group C	high school	free/reduced	none	62.0	55.0	55.0	172.0
1002	1003	female	group C	high school	free/reduced	completed	59.0	71.0	65.0	195.0
1003	1004	female	group D	Diploma	standard	completed	68.0	78.0	77.0	223.0
1004	1005	female	group D	Diploma	free/reduced	none	77.0	86.0	86.0	249.0

1005 rows × 10 columns

In [ ]:

```

```

In [590]:

```
sd1.isna().sum()
```

Out[590]:

```
Student ID                     0
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     5
reading score                  5
writing score                  5
Total Score                    5
dtype: int64
```

In [591]:

```
#2.find the total number of students.
sd1['Student ID'].value_counts().sum()
```

Out[591]:

```
1005
```

In [592]:

```
#3.find the average for reading score.
sd1['reading score'].mean()
```

Out[592]:

```
69.169
```

In [593]:

```
#4.check the descriptive statistics for writing score.
sd1.describe()['writing score']
```

Out[593]:

```
count    1000.000000
mean       68.054000
std        15.195657
min        10.000000
25%        57.750000
50%        69.000000
75%        79.000000
max       100.000000
Name: writing score, dtype: float64
```

In [594]:

```
#5.find how many students had completed the test preparation.
sd1['test preparation course'].value_counts()
```

Out[594]:

```
none         644
completed    361
Name: test preparation course, dtype: int64
```

In [595]:

```
#6.find the min score in math.
sd1['math score'].min()
```

Out[595]:

```
0.0
```

In [596]:

```
#7.find the max score in writing.
sd1['writing score'].max()
```

Out[596]:

```
100.0
```

In [597]:

```
#8.find the number of students by lunch.
sd1.groupby('lunch').count()['Student ID']
```

Out[597]:

```
lunch
free/reduced    356
standard        649
Name: Student ID, dtype: int64
```

In [598]:

```
sd1['math score'].sum()
```

Out[598]:

```
66089.0
```

In [599]:

```
sd1['reading score'].sum()
```

Out[599]:

```
69169.0
```

In [600]:

```
sd1['writing score'].sum()
```

Out[600]:

```
68054.0
```

In [601]:

```
#9.find the average score for each by race/ethnicity.
#average_list=[66089.0,69169.0,68054.0]
#sd1.groupby('race/ethnicity').mean(average_list)
#sd.drop('Column',axis=1,inplace=True)

#sd
```

In [602]:

```
#9.find the average score for each by race/ethnicity.
sd1.groupby('race/ethnicity').mean()['Total Score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\2399675795.py:2: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  sd1.groupby('race/ethnicity').mean()['Total Score']
```

Out[602]:

```
race/ethnicity
group A    188.977528
group B    196.405263
group C    201.394984
group D    207.538168
group E    218.257143
Name: Total Score, dtype: float64
```

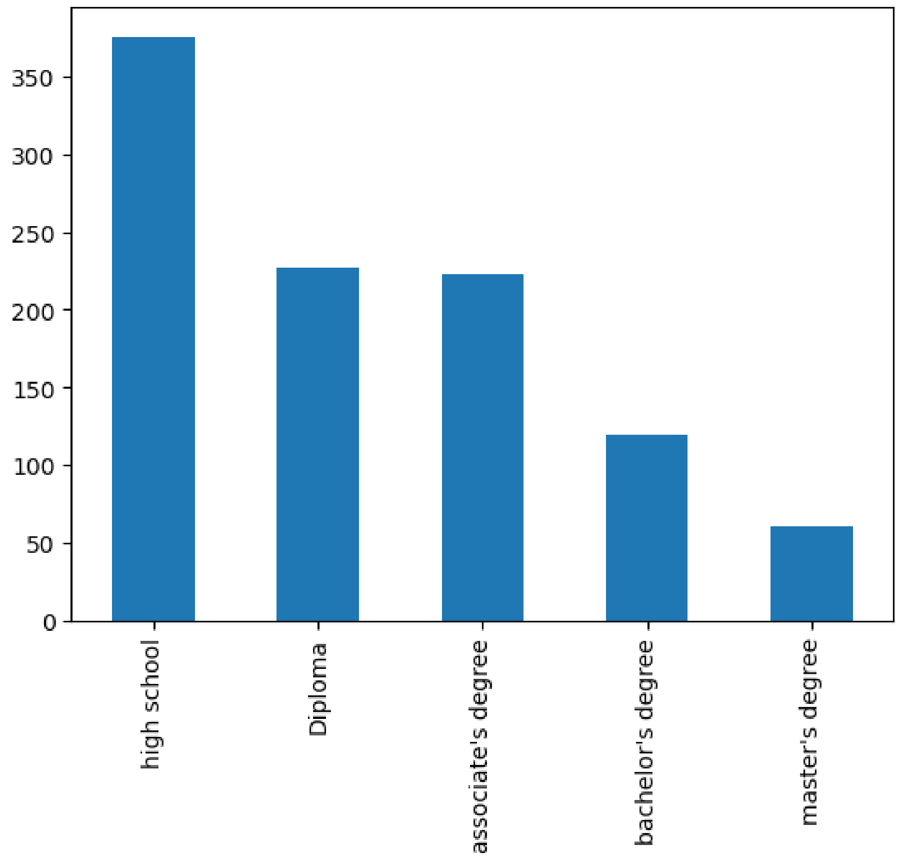
In [603]:

```
#10.create a count plot for parental level of education.
#sns.countplot(x='parental level of education',sd1=sd1)
#sd1
sd1['parental level of education'].value_counts().plot(kind='bar')
sd1
```

Out[603]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0
...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	completed	88.0	99.0	95.0	282.0
1001	1002	male	group C	high school	free/reduced	none	62.0	55.0	55.0	172.0
1002	1003	female	group C	high school	free/reduced	completed	59.0	71.0	65.0	195.0
1003	1004	female	group D	Diploma	standard	completed	68.0	78.0	77.0	223.0
1004	1005	female	group D	Diploma	free/reduced	none	77.0	86.0	86.0	249.0

1005 rows × 10 columns



In [604]:

```
#11. Find the total score of each student
#total_list=[66089.0,69169.0,68054.0]
#sd.groupby('Student ID').sum(total_list)
#sd.drop('Column',axis=1,inplace=True)
#sd
sd.head(5)
```

Out[604]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	Column	Column.1	test preparation course	math score	reading score	writing score
0	1	female	group B	bachelor's degree	standard	NaN	NaN	none	72.0	72.0	74.0
1	2	female	group C	Diploma	standard	NaN	NaN	completed	69.0	90.0	88.0
2	3	female	group B	master's degree	standard	NaN	NaN	none	90.0	95.0	93.0
3	4	male	group A	associate's degree	free/reduced	NaN	NaN	none	47.0	57.0	44.0
4	5	male	group C	Diploma	standard	NaN	NaN	none	76.0	78.0	75.0

In [605]:

```
#11. Find the total score of each student
sd1.groupby('Student ID').sum()['Total Score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\478177445.py:2: FutureWarning: The default val
ue of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will
default to False. Either specify numeric_only or select only columns which should be valid for t
he function.
  sd1.groupby('Student ID').sum()['Total Score']
```

Out[605]:

```
Student ID
1       218.0
2       247.0
3       278.0
4       148.0
5       229.0
        ...
1001    282.0
1002    172.0
1003    195.0
1004    223.0
1005    249.0
Name: Total Score, Length: 1005, dtype: float64
```

In [606]:

```
#12.What is the average of total score for male and female students?
total1_list=[66089.0,69169.0,68054.0]

sd.groupby('gender').sum(total1_list)
#sd.drop('Column.1',axis=1,inplace=True)
#sd
```

Out[606]:

	Student ID	Column	Column.1	math score	reading score	writing score
gender						
female	268503	0.0	0.0	32962.0	37611.0	37538.0
male	237012	0.0	0.0	33127.0	31558.0	30516.0

In [607]:

```
#12.What is the average of total score for male and female students?
sd1.groupby('gender').mean()['Total Score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\2031592804.py:2: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  sd1.groupby('gender').mean()['Total Score']
```

Out[607]:

```
gender
female    208.708494
male      197.512448
Name: Total Score, dtype: float64
```

In [608]:

```
#What is the average of total score by gender and parental education level?
#total2_list=[66089.0,69169.0,68054.0]
sd1.groupby(['gender','parental level of education']).sum()['Total Score']
#sd.drop('Column',axis=1,inplace=True)
#sd
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\2622692308.py:3: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only wil
l default to False. Either specify numeric_only or select only columns which should be valid for
the function.
  sd1.groupby(['gender','parental level of education']).sum()['Total Score']
```

Out[608]:

```
gender  parental level of education
female  Diploma                        25135.0
        associate's degree             24751.0
        bachelor's degree              14113.0
        high school                    36158.0
        master's degree                 7954.0
male    Diploma                        21292.0
        associate's degree             21582.0
        bachelor's degree              11348.0
        high school                    35906.0
        master's degree                 5073.0
Name: Total Score, dtype: float64
```

In [609]:

```
#13.What is the average of total score by gender and parental education level?
sd1.groupby(['gender','parental level of education']).mean()['Total Score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\2899788497.py:2: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  sd1.groupby(['gender','parental level of education']).mean()['Total Score']
```

Out[609]:

```
gender  parental level of education
female  Diploma                        213.008475
        associate's degree             213.370690
        bachelor's degree              224.015873
        high school                    195.448649
        master's degree                220.944444
male    Diploma                        197.148148
        associate's degree             203.603774
        bachelor's degree              206.327273
        high school                    188.978947
        master's degree                220.565217
Name: Total Score, dtype: float64
```

In [610]:

```
#14.What is the average math score by race/ethnicity?
sd1.groupby('race/ethnicity').mean()['math score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\4051283016.py:2: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  sd1.groupby('race/ethnicity').mean()['math score']
```

Out[610]:

```
race/ethnicity
group A    61.629213
group B    63.452632
group C    64.463950
group D    67.362595
group E    73.821429
Name: math score, dtype: float64
```

In [611]:

```
#15.Find the number of Students by test preparation course
sd1.groupby('parental level of education').count()['Student ID']
```

Out[611]:

```
parental level of education
Diploma               227
associate's degree    223
bachelor's degree     119
high school           376
master's degree        60
Name: Student ID, dtype: int64
```

In [612]:

```
#16. Find the average total score by test preparation course
sd1.groupby('test preparation course').mean()['Total Score']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_20916\1182840673.py:2: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  sd1.groupby('test preparation course').mean()['Total Score']
```

Out[612]:

```
test preparation course
completed    218.008380
none         195.116822
Name: Total Score, dtype: float64
```

In [613]:

```
#17. create a new variable as, 'Score', and the values in the column should be the sum of all the three papers
```

In [614]:

```
sd1['Score']=(sd1['math score']+sd1['reading score']+sd1['writing score'])/300*3
sd1
```

Out[614]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score	Score
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0	2.18
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0	2.47
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0	2.78
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0	1.48
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0	2.29
...	...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	completed	88.0	99.0	95.0	282.0	2.82
1001	1002	male	group C	high school	free/reduced	none	62.0	55.0	55.0	172.0	1.72
1002	1003	female	group C	high school	free/reduced	completed	59.0	71.0	65.0	195.0	1.95
1003	1004	female	group D	Diploma	standard	completed	68.0	78.0	77.0	223.0	2.23
1004	1005	female	group D	Diploma	free/reduced	none	77.0	86.0	86.0	249.0	2.49

1005 rows × 11 columns

In [615]:

```
#18. create a new variable as, " Percentage", and find the percentage for the score

sd1['Percentage']=(sd1['math score']+sd1['reading score']+sd1['writing score'])/300*(100)
sd1
```

Out[615]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score	Score	Percentage
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0	2.18	72.666667
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0	2.47	82.333333
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0	2.78	92.666667
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0	1.48	49.333333
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0	2.29	76.333333
...	...	...	...	...	...	...	...	...	...	...	...	...
1000	1001	female	group E	master's degree	standard	completed	88.0	99.0	95.0	282.0	2.82	94.000000

In [616]:

```
#from math import *

#sd1['Percentage'] = sd1['Total Score']/3

#for i in range(0, 1000):

 #sd1['Percentage'][i] = ceil(sd1['Percentage'][i])

#sd1['Percentage'].value_counts(normalize = True)
#sd1['Percentage'].value_counts(dropna = False).plot.bar(figsize = (16, 8), color = 'red')

#plt.title('Comparison of percentage scored by all the students')
#plt.xlabel('percentage score')
#plt.ylabel('count')
#plt.show()
```

In [627]:

```
#19. create a new variable called, 'status'. Inside this column, find the status as Pass \ Fail, those who scor
passmark=120
sd1['Status']=np.where(sd1['Total Score']>passmark,'pass','fail')
#sd1['reading status']=np.where(sd1['reading score']>passmark,'pass', 'fail')
#sd1['writing status']=np.where(sd1['writing score']>passmark,'pass','fail')
sd1
sd1.head(20)
```

Out[627]:

	Student ID	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score	Score	Percenta
0	1	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0	218.0	2.18	72.666
1	2	female	group C	Diploma	standard	completed	69.0	90.0	88.0	247.0	2.47	82.333
2	3	female	group B	master's degree	standard	none	90.0	95.0	93.0	278.0	2.78	92.666
3	4	male	group A	associate's degree	free/reduced	none	47.0	57.0	44.0	148.0	1.48	49.333
4	5	male	group C	Diploma	standard	none	76.0	78.0	75.0	229.0	2.29	76.333
5	6	female	group B	associate's degree	standard	none	71.0	83.0	78.0	232.0	2.32	77.333
6	7	female	group B	Diploma	standard	completed	88.0	95.0	92.0	275.0	2.75	91.666
7	8	male	group B	Diploma	free/reduced	none	40.0	43.0	39.0	122.0	1.22	40.666
8	9	male	group D	high school	free/reduced	completed	64.0	64.0	67.0	195.0	1.95	65.000
9	10	female	group B	high school	free/reduced	none	38.0	60.0	50.0	148.0	1.48	49.333
10	11	male	group C	associate's degree	standard	none	58.0	54.0	52.0	164.0	1.64	54.666
11	12	male	group D	associate's degree	standard	none	40.0	52.0	43.0	135.0	1.35	45.000
12	13	female	group B	high school	standard	none	65.0	81.0	73.0	219.0	2.19	73.000
13	14	male	group A	Diploma	standard	completed	78.0	72.0	70.0	220.0	2.20	73.333
14	15	female	group A	master's degree	standard	none	50.0	53.0	58.0	161.0	1.61	53.666
15	16	female	group C	high school	standard	none	69.0	75.0	78.0	222.0	2.22	74.000
16	17	male	group C	high school	standard	none	88.0	89.0	86.0	263.0	2.63	87.666
17	18	female	group B	high school	free/reduced	none	18.0	32.0	28.0	78.0	0.78	26.000
18	19	male	group C	master's degree	free/reduced	completed	46.0	42.0	46.0	134.0	1.34	44.666
19	20	female	group C	associate's degree	free/reduced	none	54.0	58.0	61.0	173.0	1.73	57.666

In [620]:

```
#20. create a pie chart using the status column
sd1['Status'].plot(kind='bar',autopct='%1.2f%%')
sd1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[620], line 2
      1 #20. create a pie chart using the status column
----> 2 sd1['Status'].plot(kind='bar',autopct='%1.2f%%')
      3 sd1

File ~\anaconda3\lib\site-packages\pandas\plotting\_core.py:1000, in PlotAccessor.__call__(self,
*args, **kwargs)
    997             label_name = label_kw or data.columns
    998             data.columns = label_name
-> 1000 return plot_backend.plot(data, kind=kind, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\__init__.py:71, in plot(data, kin
d, **kwargs)
     69         kwargs["ax"] = getattr(ax, "left_ax", ax)
     70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71 plot_obj.generate()
     72 plot_obj.draw()
     73 return plot_obj.result

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:450, in MPLPlot.generate
(self)
    448 def generate(self) -> None:
    449     self._args_adjust()
--> 450     self._compute_plot_data()
    451     self._setup_subplots()
    452     self._make_plot()

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:635, in MPLPlot._compute_
plot_data(self)
    633 # no non-numeric frames or series allowed
    634 if is_empty:
--> 635     raise TypeError("no numeric data to plot")
    637 self.data = numeric_data.apply(self._convert_to_ndarray)

TypeError: no numeric data to plot
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```