


```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [2]: study_performance = pd.read_csv(r"D:\786\Documents\study_performance.csv")
study_performance
```

```
Out[2]:
```

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	m
0	female	group B	bachelor's degree	standard	none	
1	female	group C	some college	standard	completed	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C	some college	standard	none	
...
995	female	group E	master's degree	standard	completed	
996	male	group C	high school	free/reduced	none	
997	female	group C	high school	free/reduced	completed	
998	female	group D	some college	standard	completed	
999	female	group D	some college	free/reduced	none	

1000 rows × 8 columns



```
In [3]: study_performance.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   gender                                1000 non-null   object
1   race_ethnicity                        1000 non-null   object
2   parental_level_of_education           1000 non-null   object
3   lunch                                  1000 non-null   object
4   test_preparation_course               1000 non-null   object
5   math_score                            1000 non-null   int64
6   reading_score                         1000 non-null   int64
7   writing_score                          1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

```
In [4]: study_performance.index
```

```
Out[4]: RangeIndex(start=0, stop=1000, step=1)
```

```
In [5]: study_performance['gender'].unique()
```

```
Out[5]: array(['female', 'male'], dtype=object)
```

```
In [6]: study_performance['race_ethnicity'].unique()
```

```
Out[6]: array(['group B', 'group C', 'group A', 'group D', 'group E'],  
             dtype=object)
```

```
In [7]: study_performance.nunique()
```

```
Out[7]: gender                2  
       race_ethnicity         5  
       parental_level_of_education  6  
       lunch                  2  
       test_preparation_course  2  
       math_score             81  
       reading_score          72  
       writing_score           77  
       dtype: int64
```

```
In [8]: study_performance['math_score'].nunique()
```

```
Out[8]: 81
```

```
In [9]: study_performance.count()
```

```
Out[9]: gender                1000  
       race_ethnicity         1000  
       parental_level_of_education  1000  
       lunch                  1000  
       test_preparation_course  1000  
       math_score             1000  
       reading_score          1000  
       writing_score           1000  
       dtype: int64
```

```
In [10]: study_performance['writing_score'].count()
```

```
Out[10]: 1000
```

```
In [11]: study_performance.value_counts
```

```
Out[11]: <bound method DataFrame.value_counts of          gender race_ethnicity parenta
1_level_of_education      lunch  \
0    female      group B      bachelor's degree      standard
1    female      group C      some college      standard
2    female      group B      master's degree      standard
3     male      group A      associate's degree  free/reduced
4     male      group C      some college      standard
..     ...      ...      ...      ...
995  female      group E      master's degree      standard
996   male      group C      high school  free/reduced
997  female      group C      high school  free/reduced
998  female      group D      some college      standard
999  female      group D      some college  free/reduced

      test_preparation_course  math_score  reading_score  writing_score
0                none          72          72          74
1            completed          69          90          88
2                none          90          95          93
3                none          47          57          44
4                none          76          78          75
..                ...      ...      ...      ...
995            completed          88          99          95
996                none          62          55          55
997            completed          59          71          65
998            completed          68          78          77
999                none          77          86          86

[1000 rows x 8 columns]>
```

```
In [12]: study_performance['reading_score'].value_counts()
```

```
Out[12]: 72    34
74    33
64    32
67    30
73    30
..
28     1
26     1
17     1
32     1
40     1
Name: reading_score, Length: 72, dtype: int64
```

```
In [13]: study_performance.describe()
```

```
Out[13]:
```

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

```
In [14]: study_performance.shape
```

```
Out[14]: (1000, 8)
```

```
In [15]: study_performance.rename(columns={'math_score': 'Math'})
```

```
Out[15]:
```

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	M
0	female	group B	bachelor's degree	standard	none	
1	female	group C	some college	standard	completed	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C	some college	standard	none	
...	
995	female	group E	master's degree	standard	completed	
996	male	group C	high school	free/reduced	none	
997	female	group C	high school	free/reduced	completed	
998	female	group D	some college	standard	completed	
999	female	group D	some college	free/reduced	none	

1000 rows × 8 columns



```
In [16]: study_performance.isnull()
```

```
Out[16]:
```

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...
995	False	False	False	False	False	False
996	False	False	False	False	False	False
997	False	False	False	False	False	False
998	False	False	False	False	False	False
999	False	False	False	False	False	False

1000 rows × 8 columns



```
In [17]: study_performance.isnull().sum()
```

```
Out[17]: gender                0
race_ethnicity                0
parental_level_of_education    0
lunch                        0
test_preparation_course        0
math_score                    0
reading_score                  0
writing_score                  0
dtype: int64
```

```
In [18]: study_performance.head()
```

```
Out[18]:
```

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score
0	female	group B	bachelor's degree	standard	none	80
1	female	group C	some college	standard	completed	81
2	female	group B	master's degree	standard	none	82
3	male	group A	associate's degree	free/reduced	none	83
4	male	group C	some college	standard	none	84



```
In [19]: study_performance.tail()
```

Out[19]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	m
995	female	group E	master's degree	standard	completed	
996	male	group C	high school	free/reduced	none	
997	female	group C	high school	free/reduced	completed	
998	female	group D	some college	standard	completed	
999	female	group D	some college	free/reduced	none	

```
In [20]: study_performance.drop(['lunch'],axis=1,inplace=True)
```

```
In [21]: study_performance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   gender                                1000 non-null   object
1   race_ethnicity                        1000 non-null   object
2   parental_level_of_education           1000 non-null   object
3   test_preparation_course               1000 non-null   object
4   math_score                            1000 non-null   int64
5   reading_score                         1000 non-null   int64
6   writing_score                         1000 non-null   int64
dtypes: int64(3), object(4)
memory usage: 54.8+ KB
```

```
In [22]: study_performance[['math_score']].describe()
```

Out[22]:

	math_score
count	1000.00000
mean	66.08900
std	15.16308
min	0.00000
25%	57.00000
50%	66.00000
75%	77.00000
max	100.00000

```
In [23]: study_performance[['math_score', 'writing_score']].describe()
```

```
Out[23]:
```

	math_score	writing_score
count	1000.00000	1000.000000
mean	66.08900	68.054000
std	15.16308	15.195657
min	0.00000	10.000000
25%	57.00000	57.750000
50%	66.00000	69.000000
75%	77.00000	79.000000
max	100.00000	100.000000

```
In [24]: study_performance.describe()
```

```
Out[24]:
```

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

```
In [25]: study_performance['math_score'].dtype
```

```
Out[25]: dtype('int64')
```

```
In [26]: study_performance.columns
```

```
Out[26]: Index(['gender', 'race_ethnicity', 'parental_level_of_education',  
               'test_preparation_course', 'math_score', 'reading_score',  
               'writing_score'],  
              dtype='object')
```

```
In [27]: study_performance['writing_score'].dtype
```

```
Out[27]: dtype('int64')
```

```
In [28]: study_performance['reading_score'].dtype
```

```
Out[28]: dtype('int64')
```

```
In [29]: study_performance
```

```
Out[29]:
```

	gender	race_ethnicity	parental_level_of_education	test_preparation_course	math_score	re
0	female	group B	bachelor's degree	none	72	
1	female	group C	some college	completed	69	
2	female	group B	master's degree	none	90	
3	male	group A	associate's degree	none	47	
4	male	group C	some college	none	76	
...
995	female	group E	master's degree	completed	88	
996	male	group C	high school	none	62	
997	female	group C	high school	completed	59	
998	female	group D	some college	completed	68	
999	female	group D	some college	none	77	

1000 rows × 7 columns



```
In [30]: study_performance.describe(include=object)
```

```
Out[30]:
```

	gender	race_ethnicity	parental_level_of_education	test_preparation_course
count	1000	1000	1000	1000
unique	2	5	6	2
top	female	group C	some college	none
freq	518	319	226	642

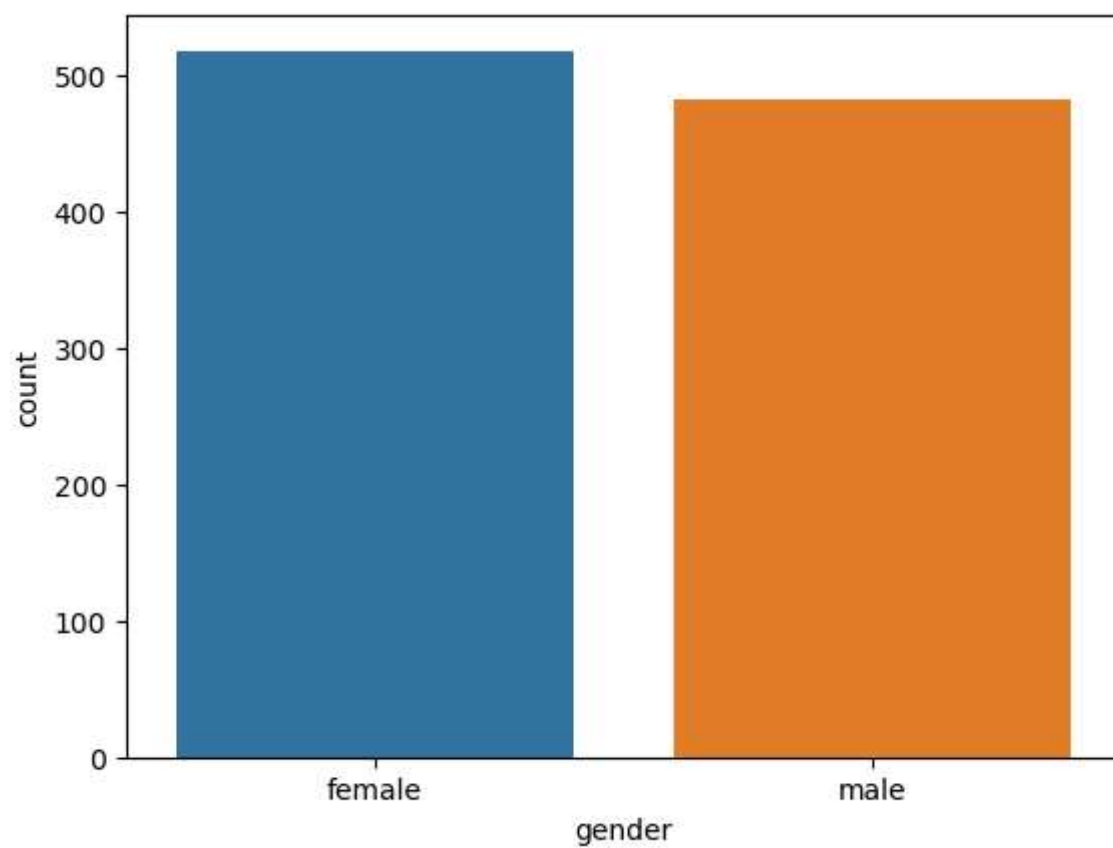
Exploratory Data Analysis

```
In [31]: study_performance.columns
```

```
Out[31]: Index(['gender', 'race_ethnicity', 'parental_level_of_education',  
               'test_preparation_course', 'math_score', 'reading_score',  
               'writing_score'],  
              dtype='object')
```

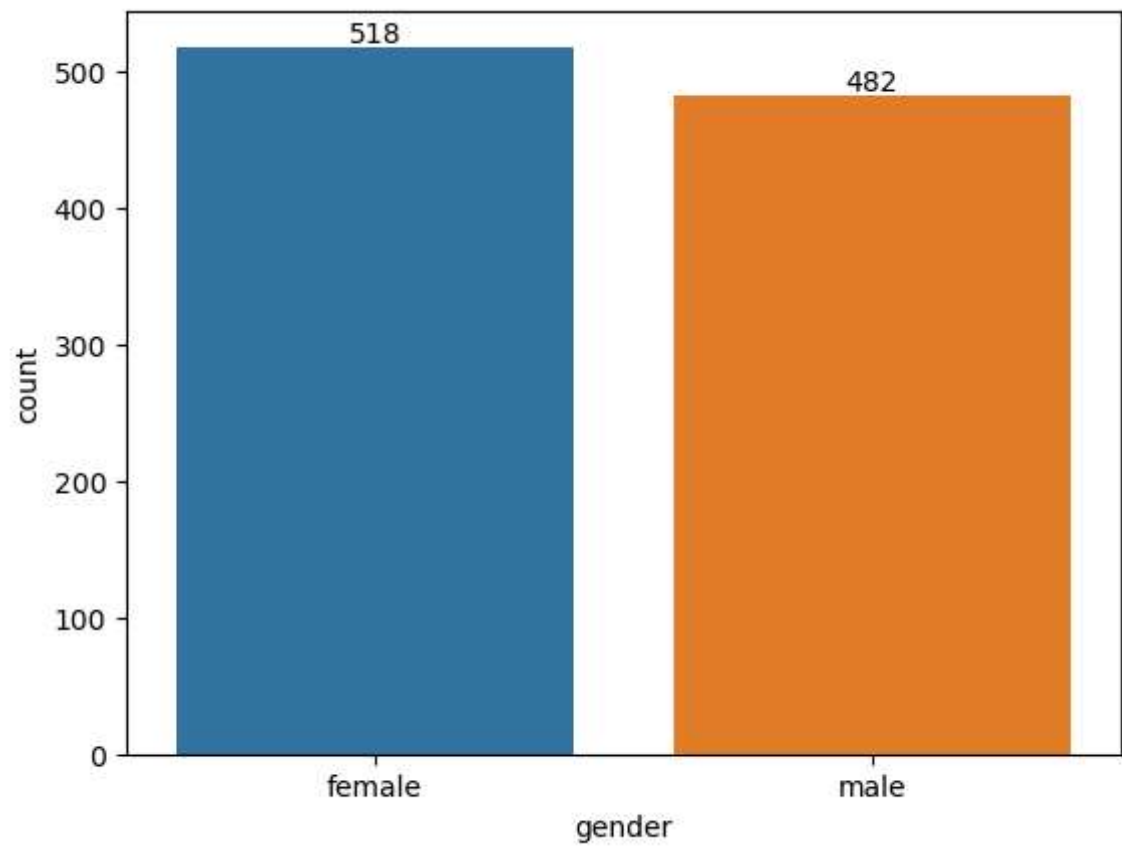


```
In [32]: ax=sns.countplot(x='gender',data=study_performance)
```

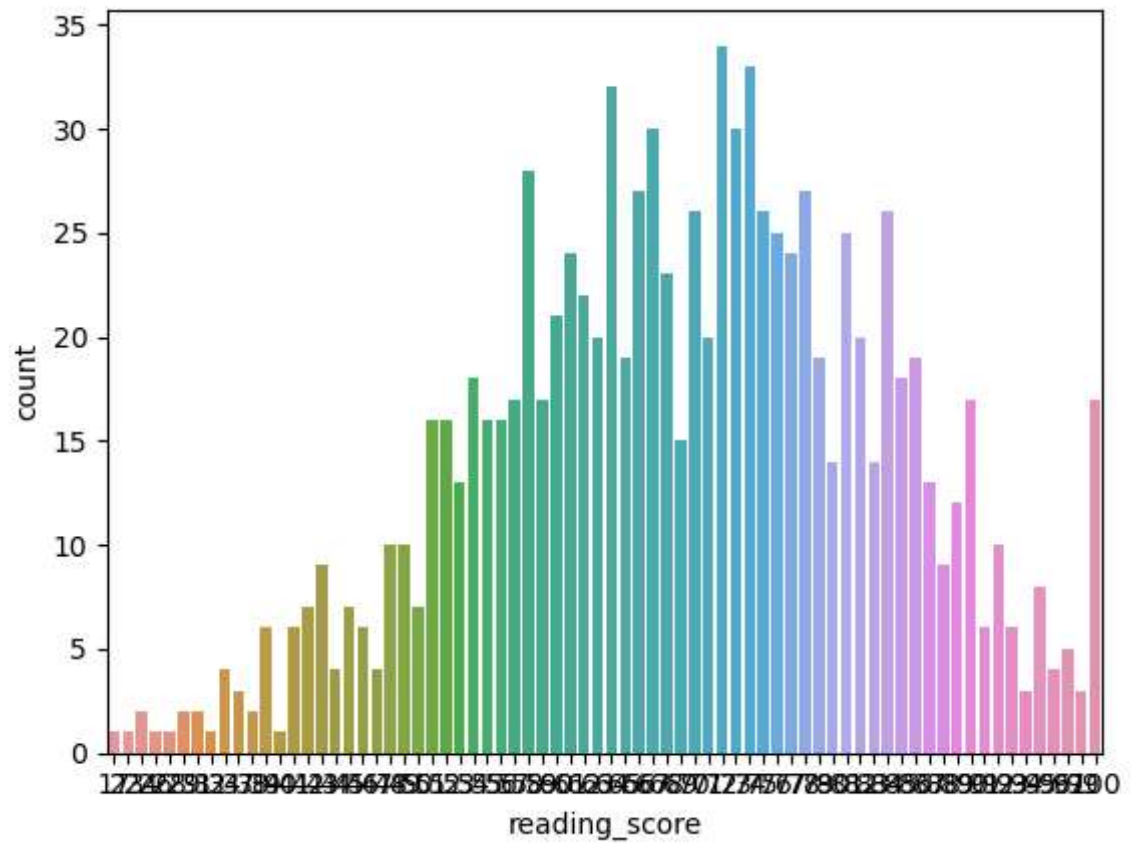


```
In [33]: ax=sns.countplot(x='gender',data=study_performance)
```

```
for bars in ax.containers:  
    ax.bar_label(bars)
```

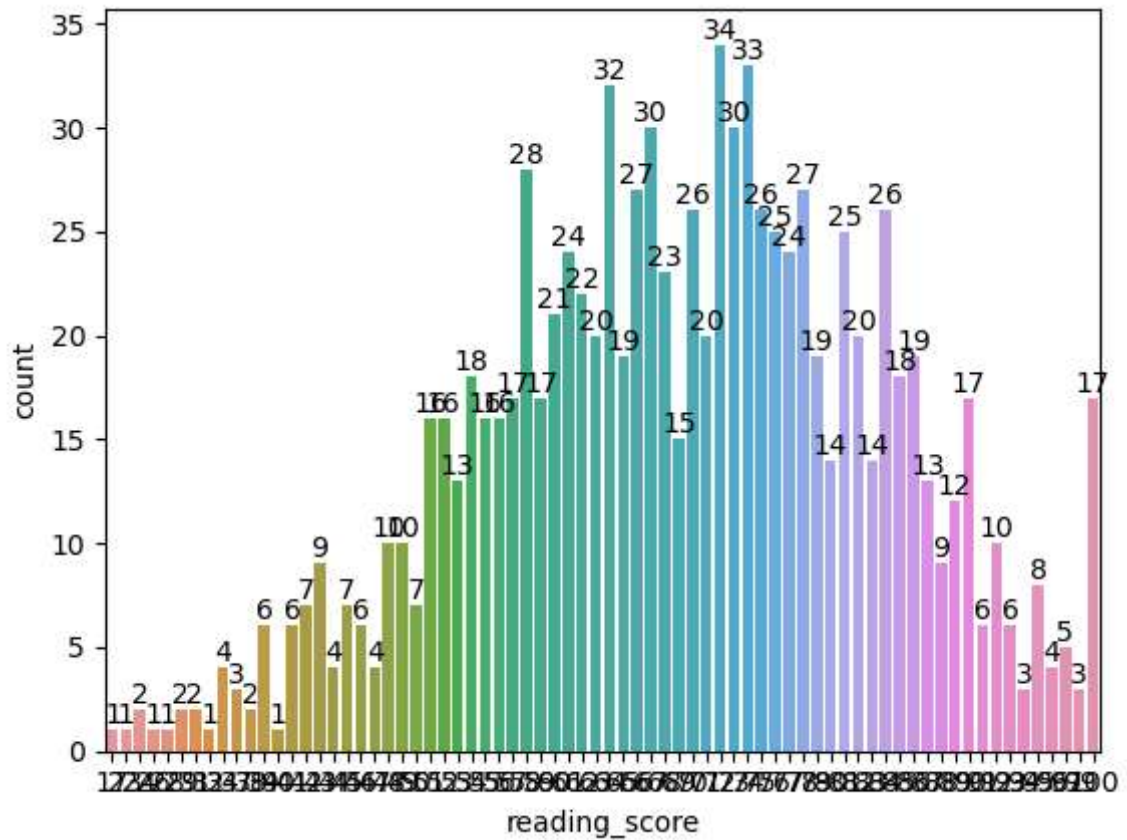


```
In [34]: ax=sns.countplot(x='reading_score',data=study_performance)
```



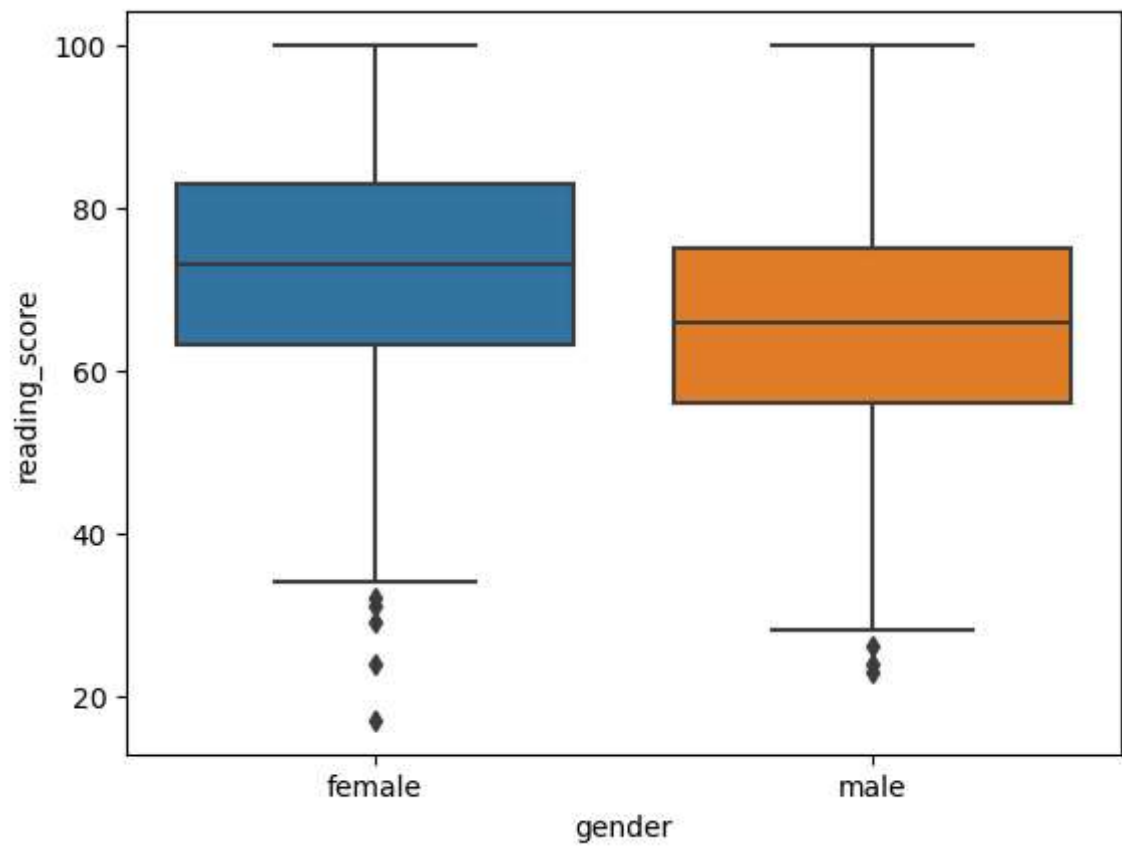
```
In [35]: ax=sns.countplot(x="reading_score",data=study_performance)
```

```
for bars in ax.containers:  
    ax.bar_label(bars)
```



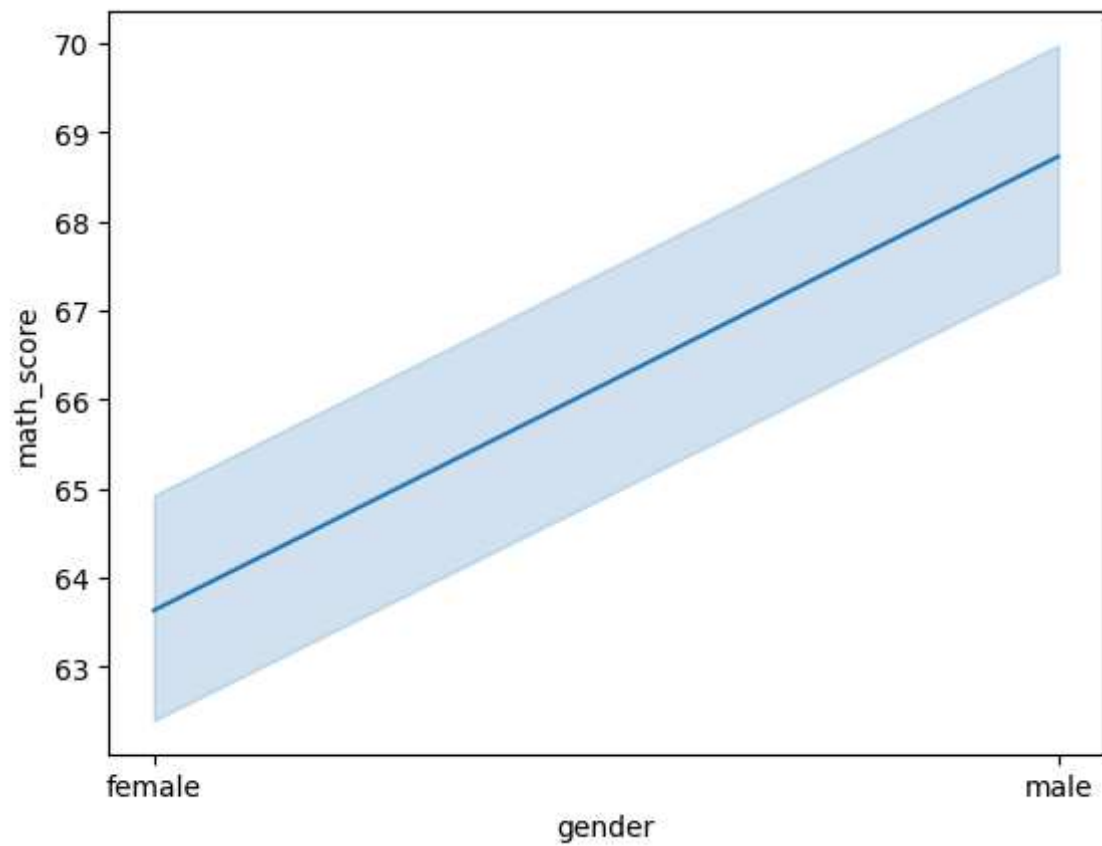
```
In [36]: sns.boxplot(x='gender',y='reading_score',data=study_performance)
```

```
Out[36]: <AxesSubplot:xlabel='gender', ylabel='reading_score'>
```



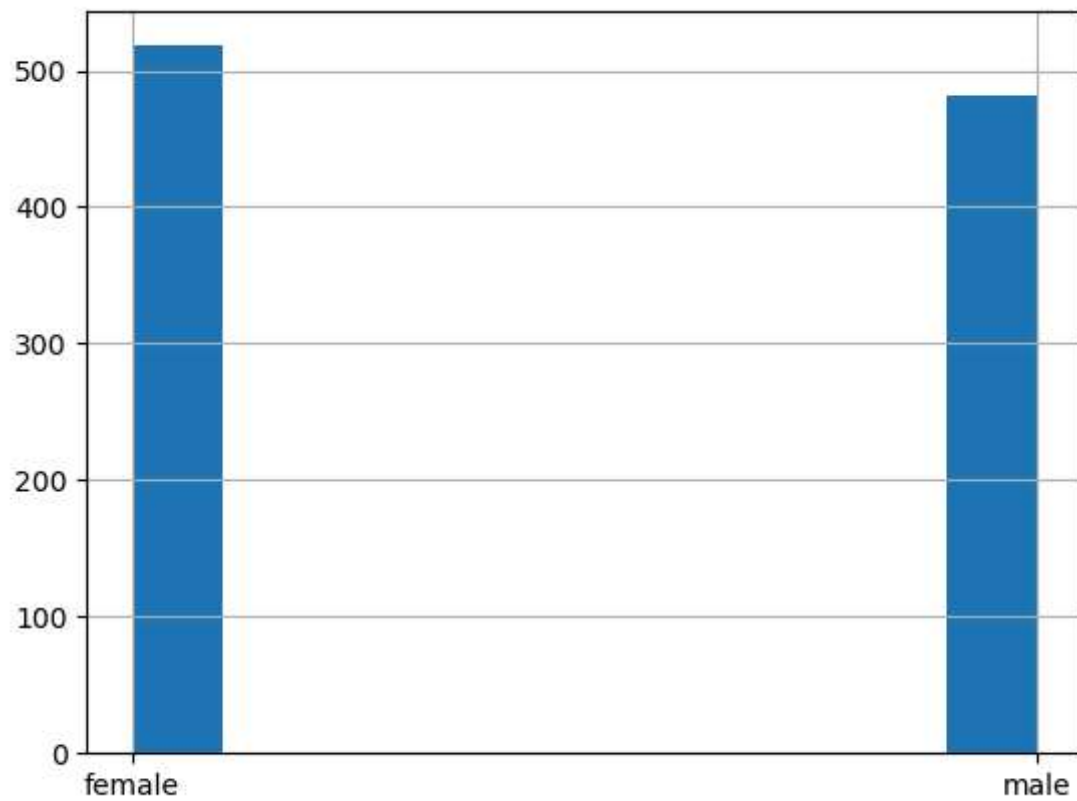
```
In [37]: sns.lineplot(x='gender',y='math_score',data=study_performance)
```

```
Out[37]: <AxesSubplot:xlabel='gender', ylabel='math_score'>
```



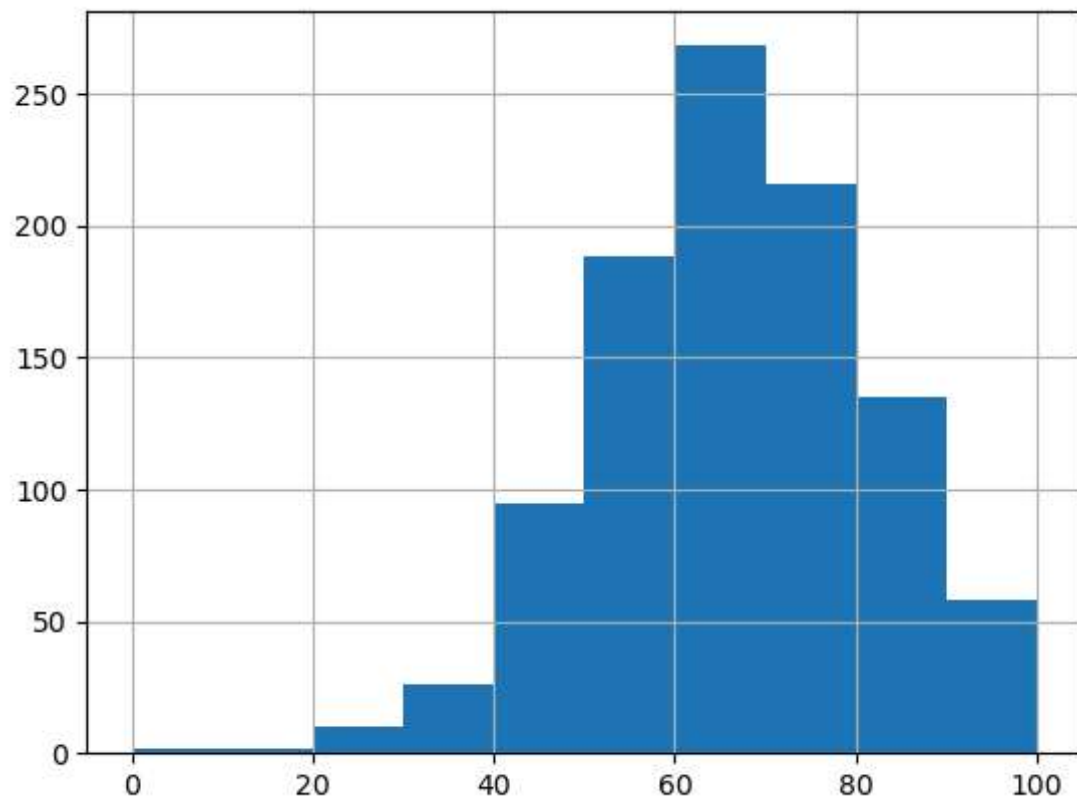
```
In [38]: study_performance['gender'].hist()
```

```
Out[38]: <AxesSubplot:>
```

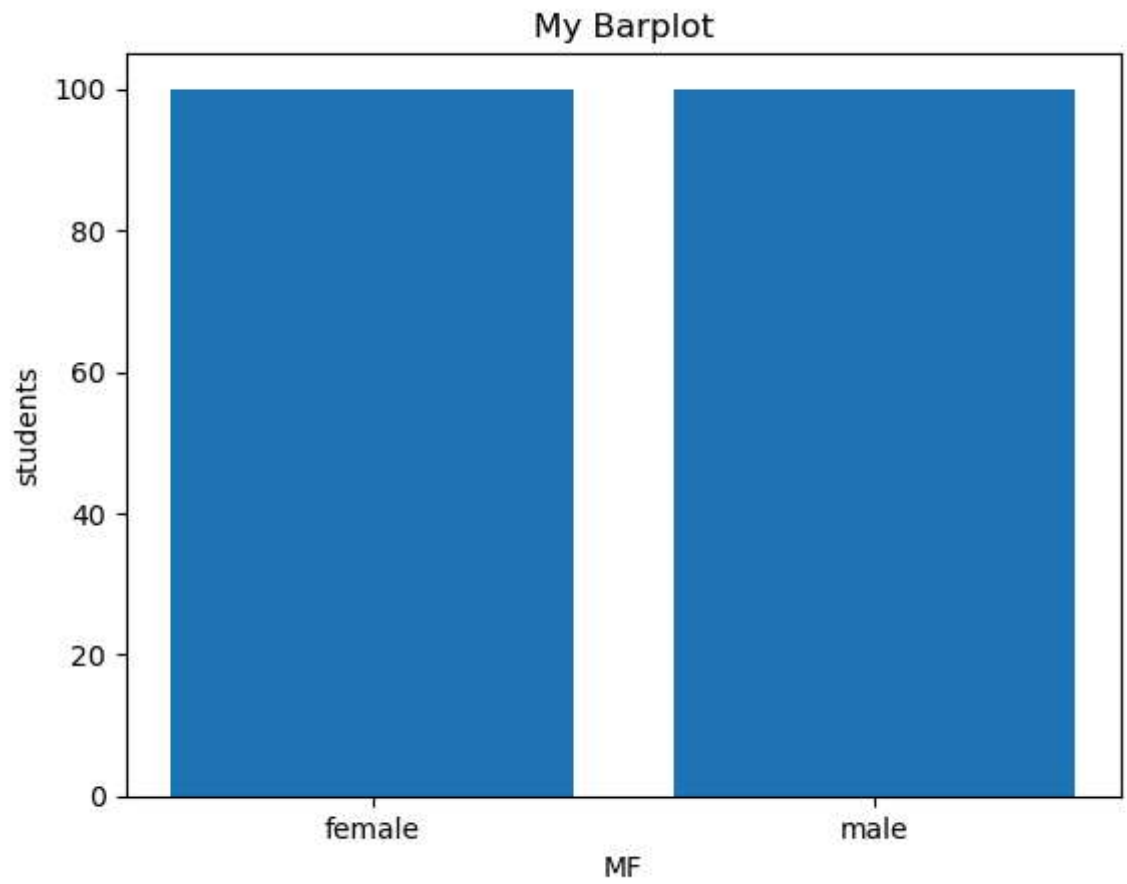


```
In [39]: study_performance['math_score'].hist()
```

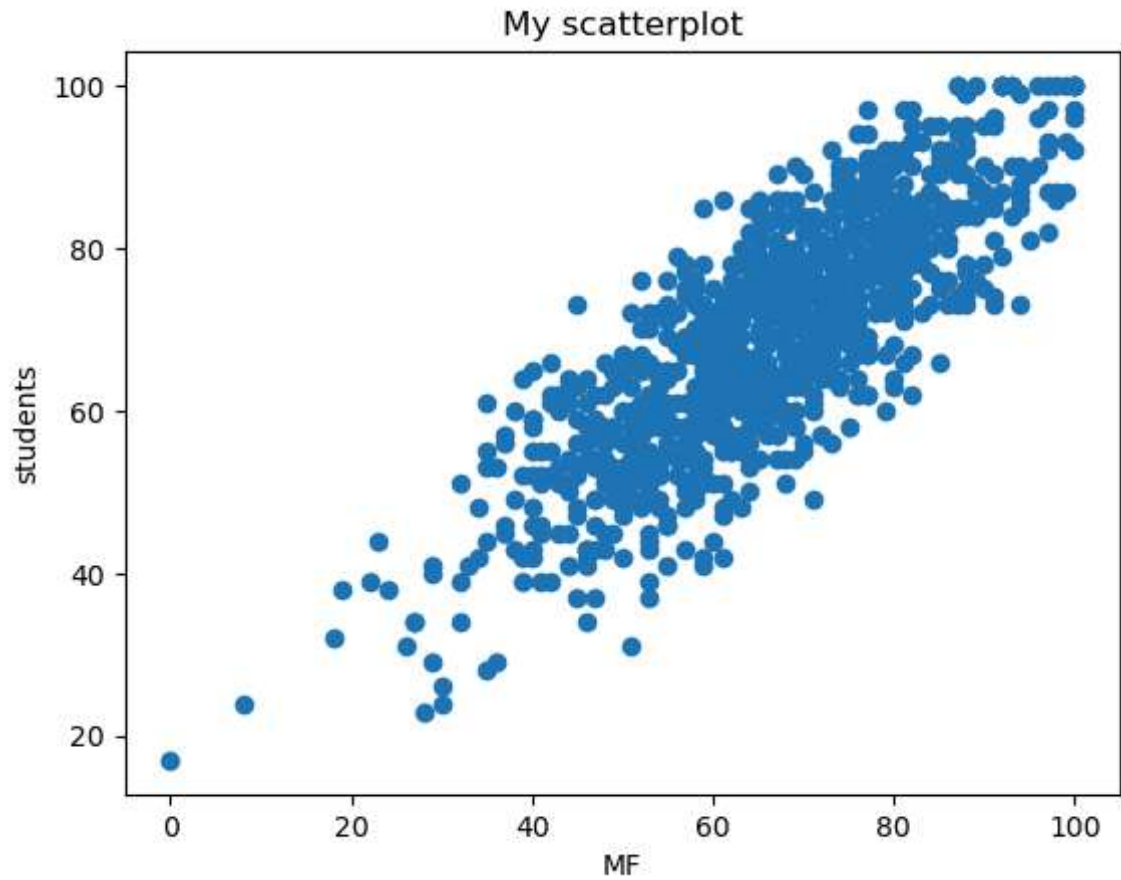
```
Out[39]: <AxesSubplot:>
```



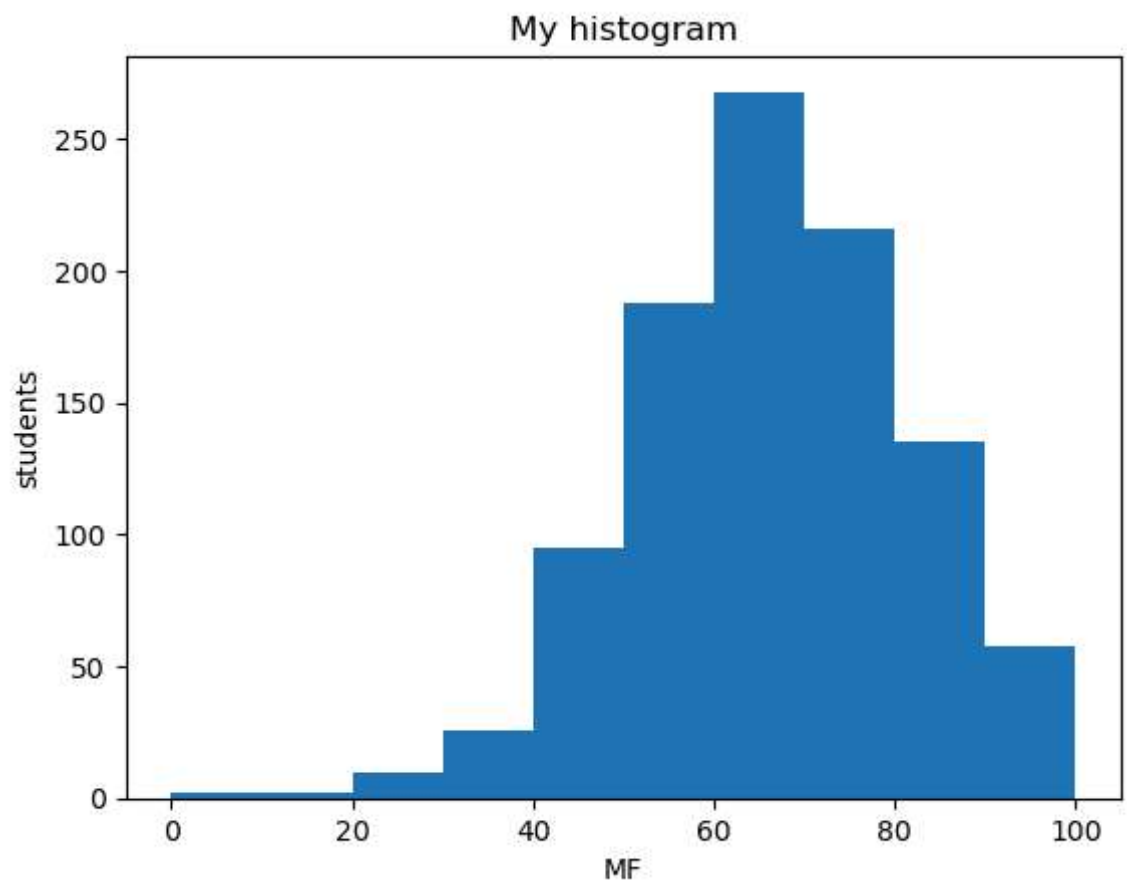

```
In [40]: x = study_performance['gender']  
y = study_performance['math_score']  
plt.xlabel('MF')  
plt.ylabel('students')  
plt.title('My Barplot')  
plt.bar(x,y)  
plt.show()
```



```
In [41]: x = study_performance['math_score']  
y = study_performance['reading_score']  
plt.xlabel('MF')  
plt.ylabel('students')  
plt.title('My scatterplot')  
plt.scatter(x,y)  
plt.show()
```



```
In [42]: x = study_performance['math_score']  
plt.xlabel('MF')  
plt.ylabel('students')  
plt.title('My histogram')  
plt.hist(x)  
plt.show()
```



```
In [43]: x=study_performance['writing_score']  
y=study_performance['reading_score']  
plt.pie(x,labels=y)  
plt.show()
```

