Analyzing student performance in exams using Python

"Analyzing Student Performance in Exams with Python" is a Jupyter Notebook that explores and visualizes student exam data using Python. This notebook provides a comprehensive analysis of student scores, including statistical insights, data visualization, and potentially machine learning techniques to identify patterns and trends in student performance. It serves as a valuable tool for educators, researchers, and anyone interested in understanding and improving educational outcomes.

Import Library

In [1]:
```
import pandas as pd
```

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy versi
on >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Uploading Csv fle

In [3]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Downloads\Compressed\Top 50 Accounts\StudentsPerformance.csv
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

In [4]:
```
df.head()
```

Out[4]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

.tail()

tail is used to show rows by Descending order

In [5]:
```
df.tail()
```
Out[5]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

.shape

It show the total no of rows & Column in the dataset

In [6]:
```
df.shape
```
Out[6]:
```
(1000, 8)
```

.Columns

It show the no of each Column

In [7]:
```
df.columns
```
Out[7]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

.dtypes

This Attribute show the data type of each column

In [8]:
```
df.dtypes
```
Out[8]:
```
gender                         object
race/ethnicity                 object
parental level of education    object
lunch                          object
test preparation course        object
math score                      int64
reading score                   int64
writing score                   int64
dtype: object
```

.unique()

In a column, It show the unique value of specific column.

```
In [9]: df["lunch"].unique()
```

```
Out[9]: array(['standard', 'free/reduced'], dtype=object)
```

.nuique()

It will show the total no of unque value from whole data frame

```
In [10]: df.nunique()
```

```
Out[10]: 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
```

.describe()

It show the Count, mean , median etc

```
In [11]: df.describe()
```

Out[11]:

	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

.value_counts

It Shows all the unique values with their count

```
In [12]: df["gender"].value_counts()
```

```
Out[12]: female    518
         male      482
         Name: gender, dtype: int64
```

.isnull()

It shows the how many null values

In [13]:
```
df.isnull()
```

Out[13]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
995	False	False	False	False	False	False	False	False
996	False	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False

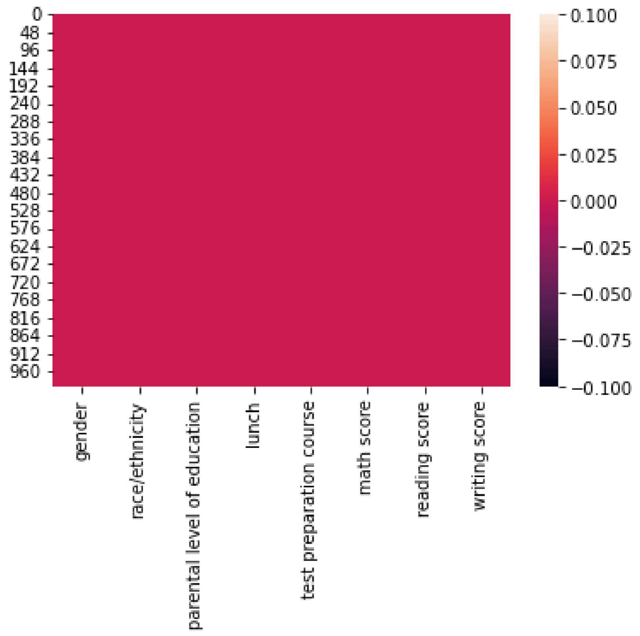
1000 rows × 8 columns

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
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
```

How many Null value present show all the null values in Heatmap

In [15]:
```
sns.heatmap(df.isnull())
plt.show()
```



Data Visualization

Performance Overview

In [16]:
```
preparation =(len(df[df['test preparation course'] == 'completed']) / len(df)) * 100
preparation
```

Out[16]:
```
35.8
```

Calculate average math Score

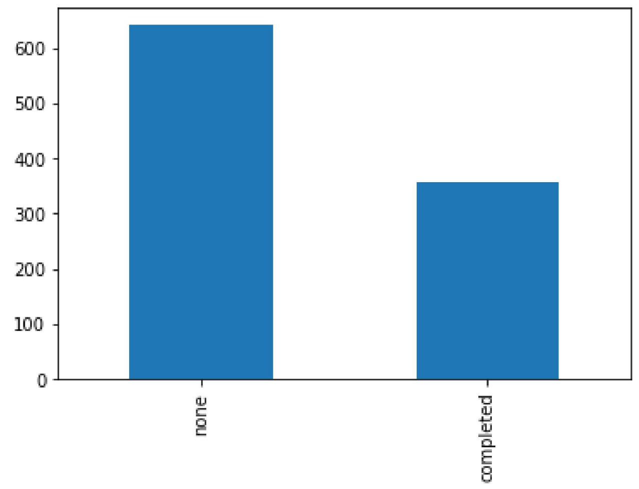
In [17]:
```
average_score = df['math score'].mean()
average_score
```

Out[17]:
```
66.089
```

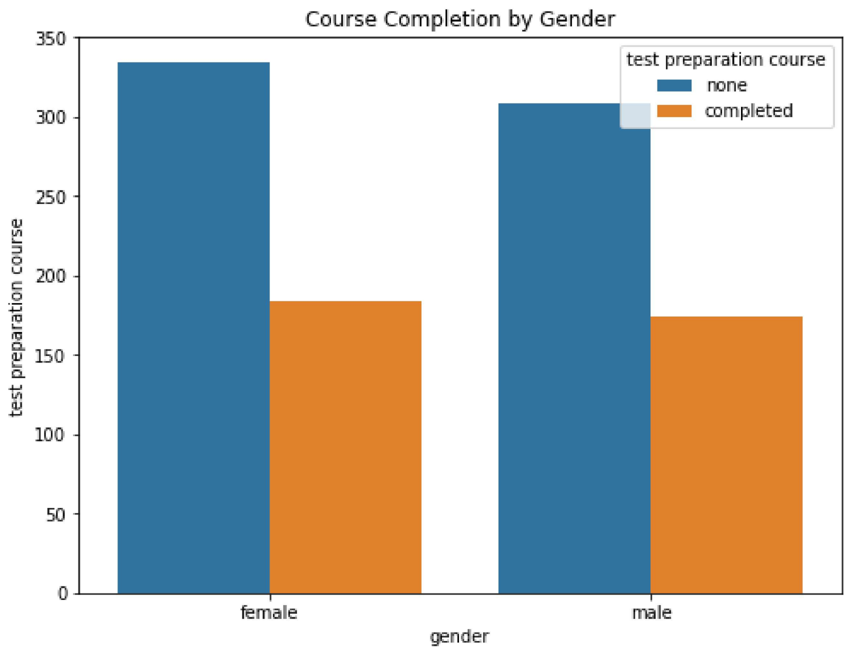
Calculate the preparation course & Show in Barplot

In [18]:
```
df['test preparation course'].value_counts().plot(kind = "bar")
```

Out[18]:
```
<AxesSubplot:>
```



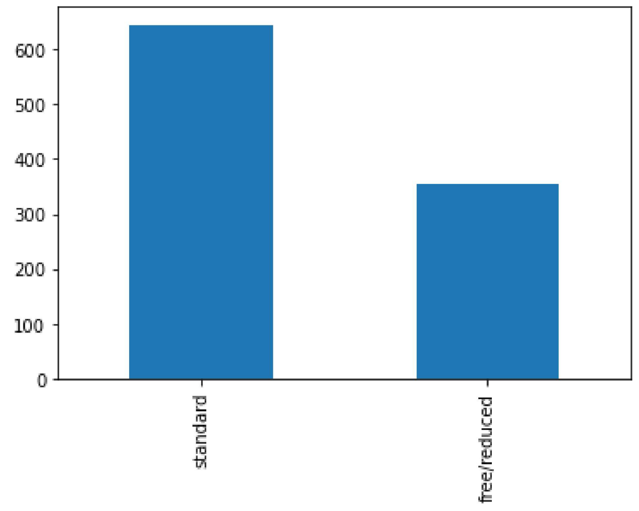
In [19]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='gender', hue='test preparation course')
plt.xlabel('gender')
plt.ylabel('test preparation course')
plt.title('Course Completion by Gender')
plt.show()
```



Calculate the lunch type and show the values using Borplot

```
In [20]: df['lunch'].value_counts().plot(kind = "bar")
```

```
Out[20]: <AxesSubplot:>
```



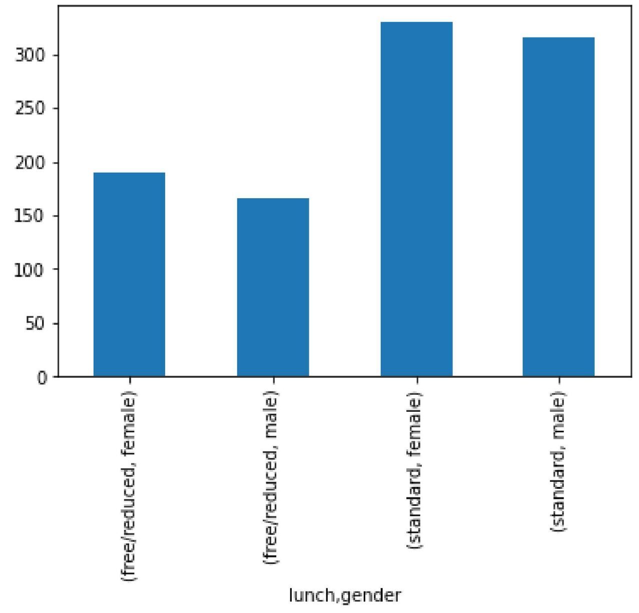
```
In [21]: df['lunch'].value_counts()
```

```
Out[21]: standard        645
         free/reduced    355
         Name: lunch, dtype: int64
```

Show the lunch type by gender wise

```
In [22]: lunch_avg_gender = df.groupby('lunch')['gender'].value_counts().plot(kind = "bar")
         lunch_avg_gender
```

```
Out[22]: <AxesSubplot:xlabel='lunch,gender'>
```



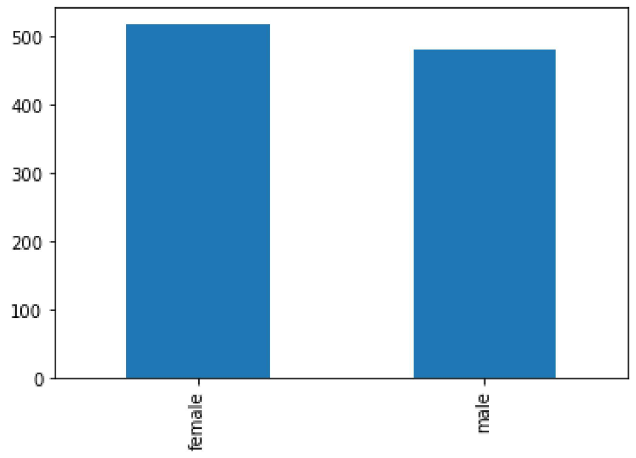
Calculate average scores by gender in reading Score

In [23]:
```
gender_avg_scores = df.groupby('gender')['reading score'].mean()
gender_avg_scores
```

Out[23]:
```
gender
female    72.608108
male      65.473029
Name: reading score, dtype: float64
```

In [24]:
```
df.gender.value_counts().plot(kind = "bar")
```

Out[24]:
```
<AxesSubplot:>
```



Calculate average by gender in level of education Score

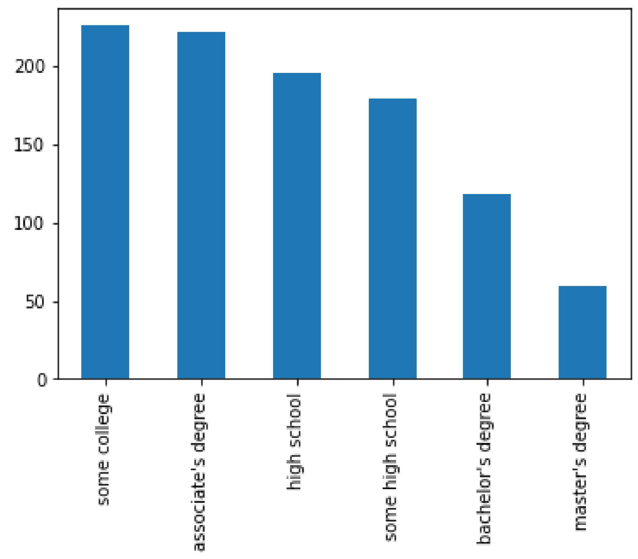
In [25]:
```
level_edu = df['parental level of education'].value_counts()
level_edu
```

Out[25]:
```
some college          226
associate's degree    222
high school           196
some high school      179
bachelor's degree     118
master's degree        59
Name: parental level of education, dtype: int64
```

Show the level of education Using Barplot

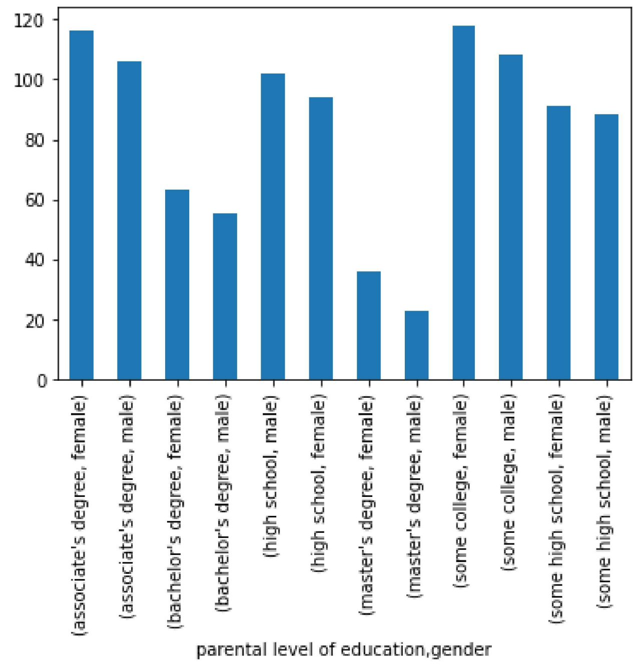
In [26]:
```
df['parental level of education'].value_counts().plot(kind = "bar")
```

Out[26]:
```
<AxesSubplot:>
```

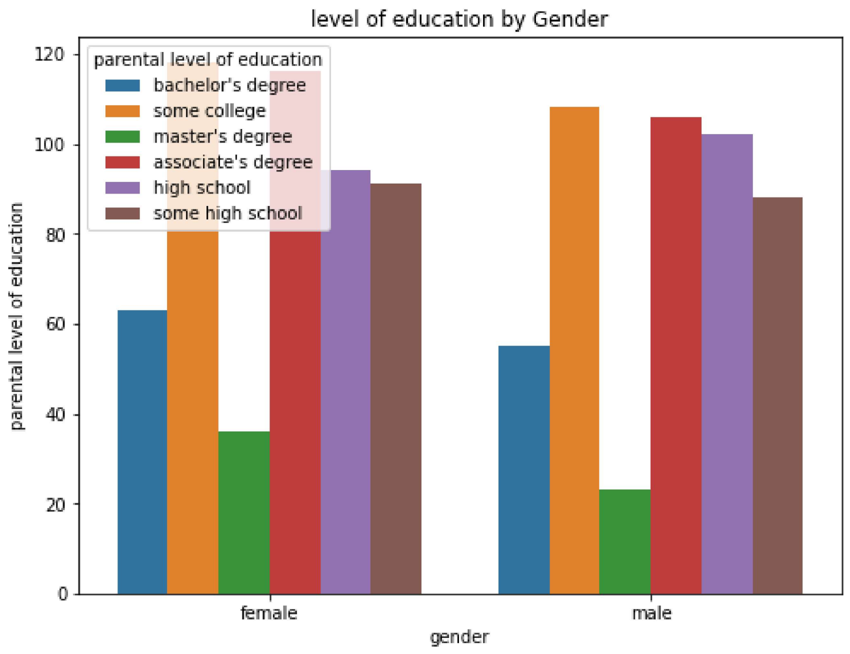


In [27]:
```
level_gender = df.groupby('parental level of education')['gender'].value_counts().plot(kind = "ba
level_gender
```

Out[27]:
```
<AxesSubplot:xlabel='parental level of education,gender'>
```



In [30]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='gender', hue='parental level of education')
plt.xlabel('gender')
plt.ylabel('parental level of education')
plt.title('level of education by Gender')
plt.show()
```



Test Preparation Course

Calculate students who completed the test preparation course

In [28]:
```
df[df['test preparation course'] == "completed"]
```

Out[28]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group C	some college	standard	completed	69	90	88
6	female	group B	some college	standard	completed	88	95	92
8	male	group D	high school	free/reduced	completed	64	64	67
13	male	group A	some college	standard	completed	78	72	70
18	male	group C	master's degree	free/reduced	completed	46	42	46
...	...	...	...	...	...	...	...	...
990	male	group E	high school	free/reduced	completed	86	81	75
991	female	group B	some high school	standard	completed	65	82	78
995	female	group E	master's degree	standard	completed	88	99	95
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77

358 rows × 8 columns

In [ ]: