

PROJECT REPORT ON

STUDENTS PERFORMANCE

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud Computing

Submitted by:

Rahila

Tushar



Year 2023(Batch 2022-24)

NATIONAL SKILL TRAINING INSTITUTE FOR WOMEN

Sector-1, Block-D, Noida- 201301

**(Affiliated to Ministry of Skill Development and
Entrepreneurship Government of India.)**

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to the Almighty for blessing us with his grace and guiding us to successfully complete our endeavour.

Our sincere thanks go to our Edunet trainers, Mrs. Deepika Singh, Mrs. Mala Mishra, and Ms. Ankita Shukla. Their invaluable guidance and timely advice played a pivotal role in leading us towards the right path.

We extend our appreciation to you for your unwavering support and valuable contributions. Special thanks to our course coordinator, Mr. D.A. Guruvulu, for his insightful suggestions and guidance.

We are grateful to our respected Head of the division, Smt. Shashi Mathur [JD NSTI(W) NOIDA], for granting us access to the facilities and resources that were crucial for our project's success.

Our heartfelt thanks also go to the other esteemed faculty members who have been instrumental in our journey.

Lastly, we would like to express our deep appreciation to our friends and family for their unwavering support and encouragement throughout our project. Their presence has been a source of our strength and motivation.

TEAM COMPOSITION AND WORKLOAD DIVISION

Team Members	Workload Division
Rahila (Team Leader) Tushar	✓ have worked together equally

TABLE OF CONTENTS

S. NO.	TOPICS	PAGE NO.
1.	Introduction to Problem statement	3
2.	Requirement	3
3.	Appendix A Project Code and Screenshot	3 - 13
4.	Appendix B abbreviation	13
5.	References	13

1. Introduction to Problem Statement

The aim of this project is to analyse the academic performance of students and predict their overall results based on individual subject scores. The dataset includes information on students' scores in math, reading, and writing, as well as their pass/fail status in each subject.

2. Requirement

Hardware -Desktop /laptop, Internet connectivity

Software- Jupyter Notebook, Goggle Colab, Dataset, Excel


3. Appendix A Project Code and Screenshot

jupyter Peojectfile Last Checkpoint: 13 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)

CORE MODULE-5 FINAL PROJECT

Data Science Project - Student Performance Analysis with Machine Learning



Import The Libraries

```
In [133]: 1 import pandas as pd
          2 import numpy as np
          3 import matplotlib.pyplot as plt
          4 import seaborn as sns
          5 %matplotlib inline
          6 sns.set_style("whitegrid")
          7 plt.style.use("fivethirtyeight")
          8
```

Read Dataset

```
In [134]: 1 data = pd.read_csv("StudentsPerformance.csv")
          2
```

```
In [135]: 1 data.head()
```

```
Out[135]:
```

	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

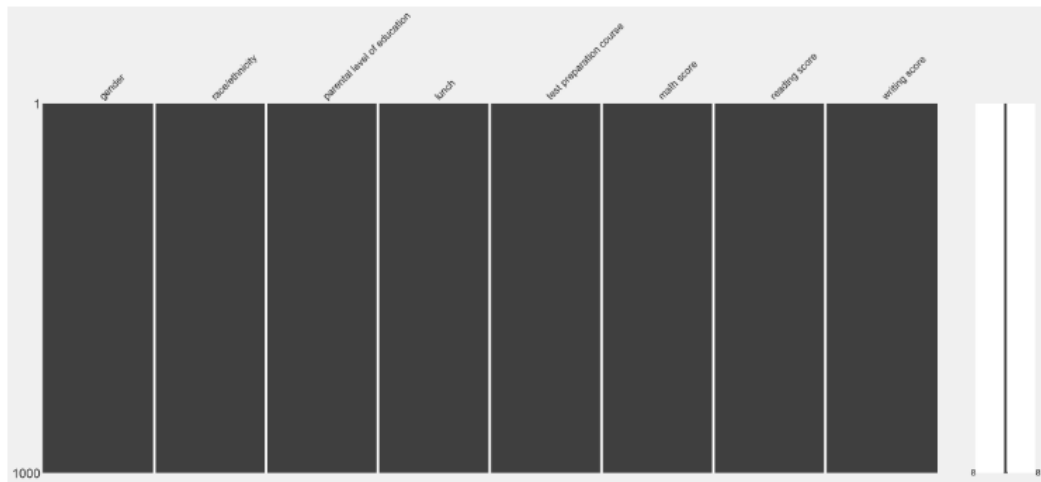
```
In [136]: 1 data.info()

<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 [139]: 1 import missingno as msno
          2
```

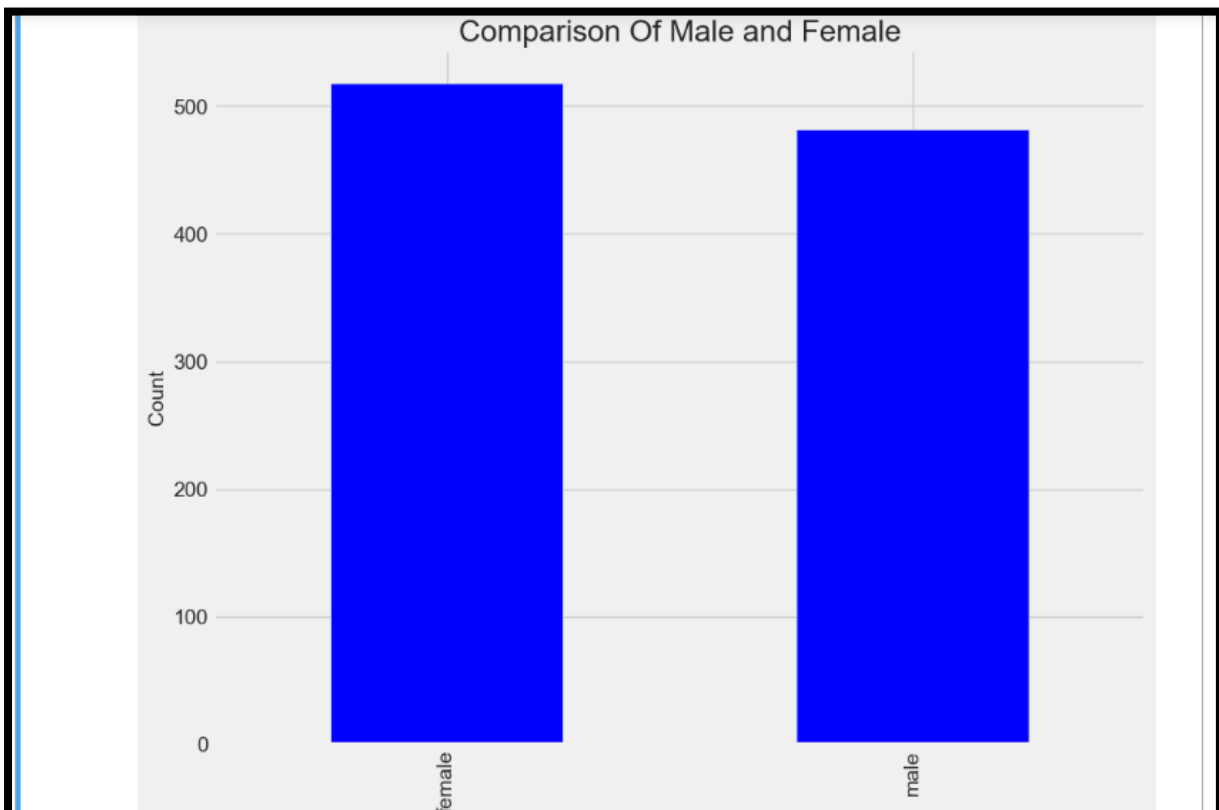
```
In [140]: 1 msno.matrix(data)
```

```
Out[140]: <Axes: >
```



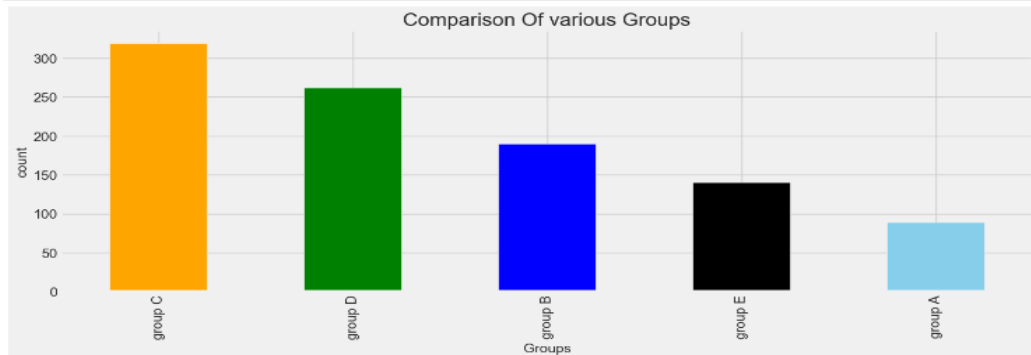
Comparison Of Male & Female in bar chart

```
] 1 plt.subplots(figsize=(10,8))
   2 data['gender'].value_counts(normalize=True)
   3 data['gender'].value_counts(dropna=True).plot.bar(color="blue")
   4 plt.title('Comparison Of Male and Female')
   5 plt.xlabel('Gender')
   6 plt.ylabel('Count')
   7 plt.show()
```



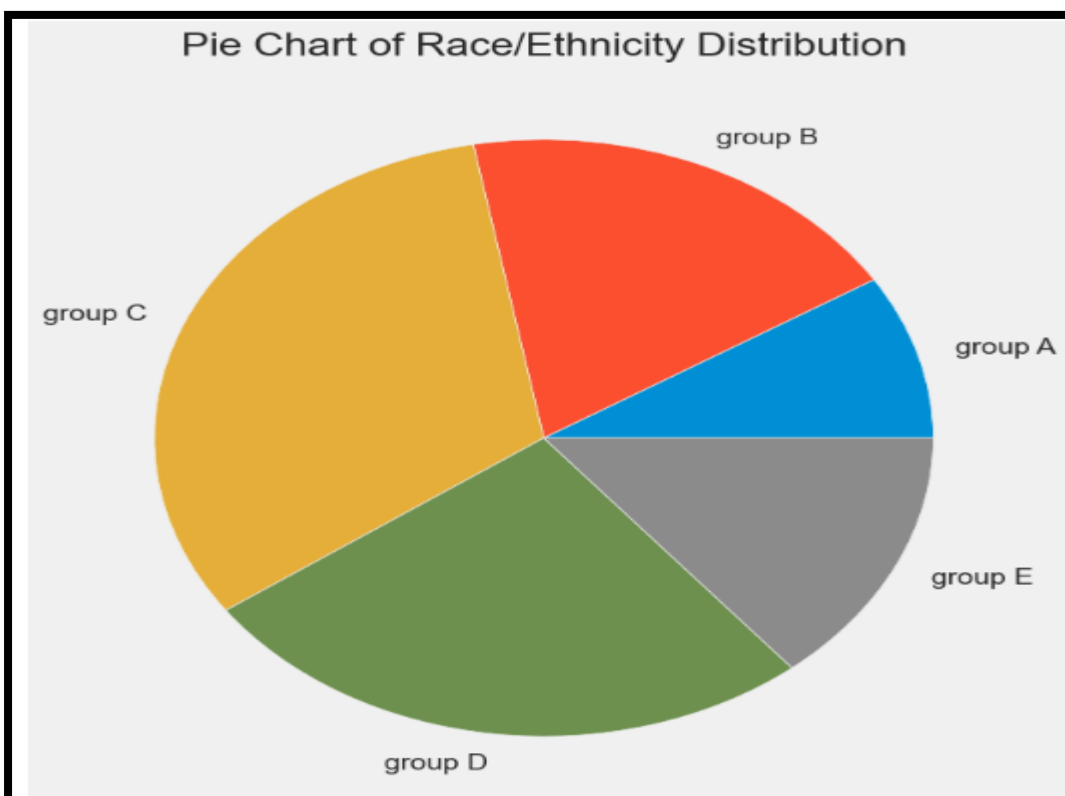
Visualization the different Groups in the Dataset

```
In [143]: 1 plt.subplots(figsize=(15,5))
2 data['race/ethnicity'].value_counts(normalize=True)
3 data['race/ethnicity'].value_counts(dropna=False).plot.bar(color=['orange', 'green', 'blue', 'black', 'skyblue'])
4 plt.title('Comparison Of various Groups')
5 plt.xlabel('Groups')
6 plt.ylabel('count')
7 plt.show()
```



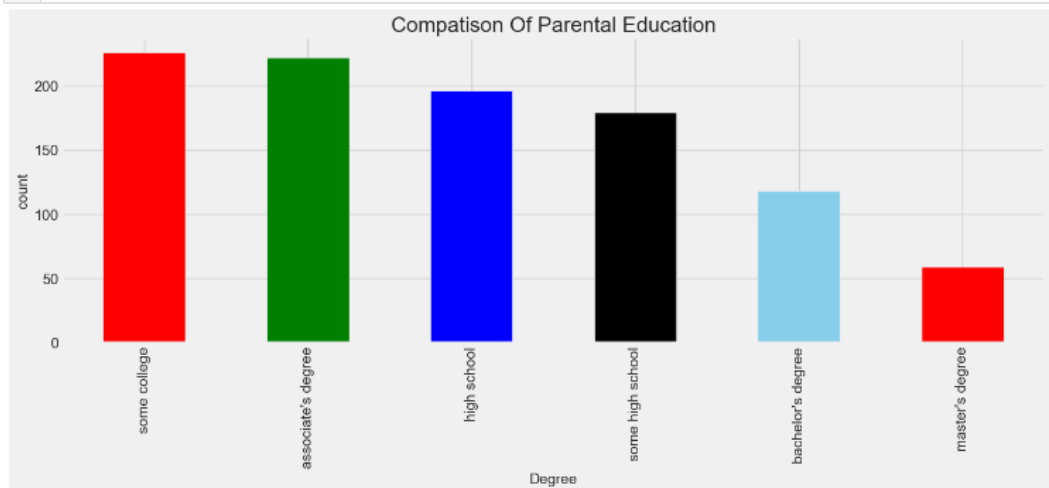
prepare Data

```
4]: 1 df = data.groupby('race/ethnicity').size()
2
3 # make plot
4 df.plot(kind='pie', subplots=True, figsize=(15,8))
5 plt.title('Pie Chart of Race/Ethnicity Distribution')
6 plt.ylabel('')
7 plt.show()
8
```



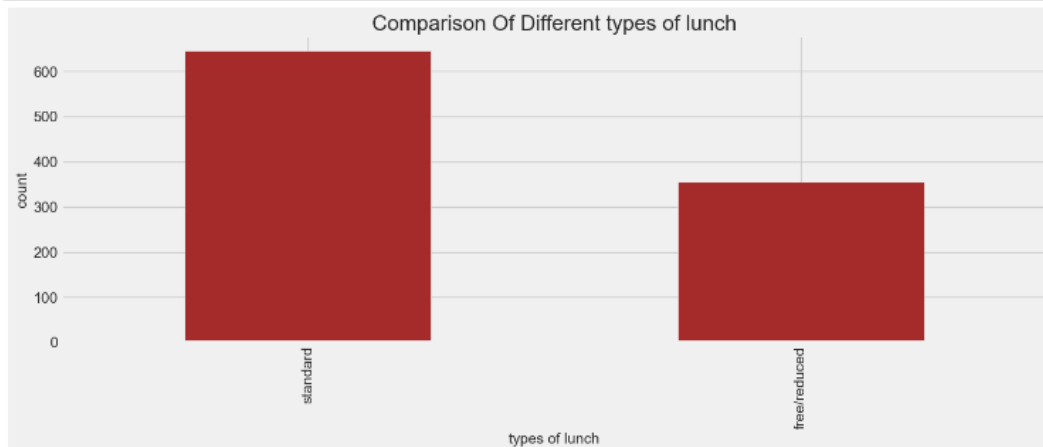
visualizing the Different parental education levels

```
In [146]: 1 plt.subplots(figsize=(15,5))
2 data['parental level of education'].value_counts(normalize=True)
3 data['parental level of education'].value_counts(dropna=False).plot.bar(color= ['red', 'green', 'blue', 'black', 'skyblue'])
4 plt.title('Compatison Of Parental Education')
5 plt.xlabel('Degree')
6 plt.ylabel('count')
7 plt.show()
```



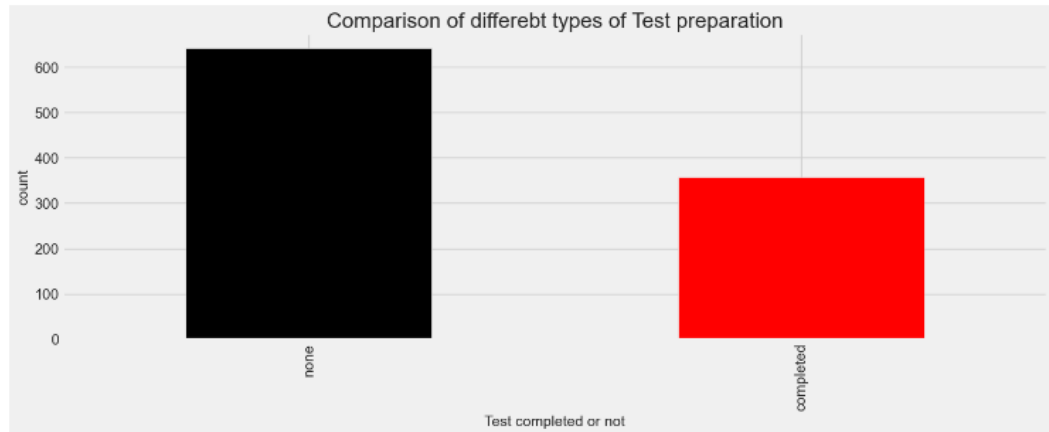
visualizing the Different Types of lunch

```
In [147]: 1 plt.subplots(figsize=(15,5))
2 data['lunch'].value_counts(normalize=True)
3 data['lunch'].value_counts(dropna=False).plot.bar(color='brown')
4 plt.title('Comparison Of Different types of lunch')
5 plt.xlabel('types of lunch')
6 plt.ylabel('count')
7 plt.show()
```



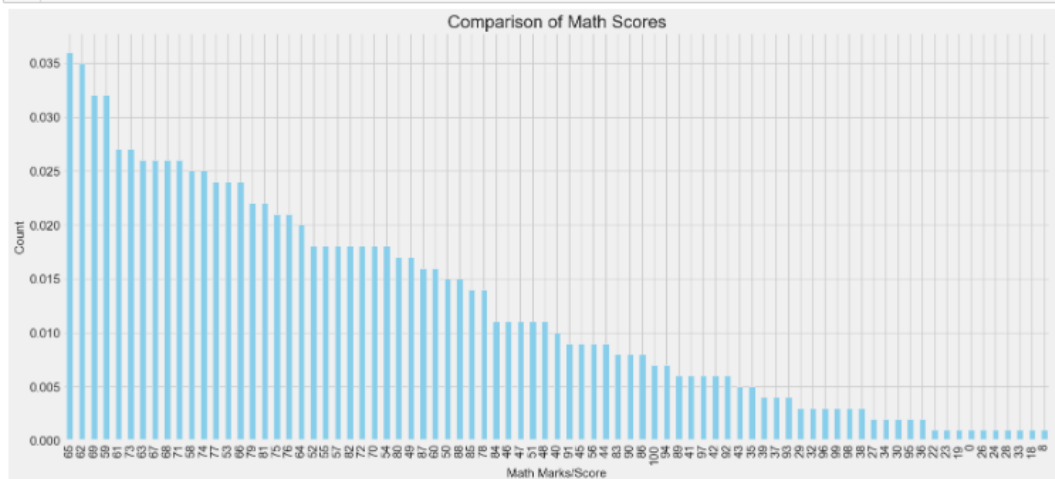
Comparison of Differebt Types of Test Praparetion Like Test Completed or Not

```
In [148]: 1 plt.subplots(figsize=(15,5))
2 data['test preparation course'].value_counts(normalize=True)
3 data['test preparation course'].value_counts(dropna=False).plot.bar(color=['black', 'red'])
4 plt.title('Comparison of differebt types of Test preparation')
5 plt.xlabel('Test completed or not')
6 plt.ylabel('count')
7 plt.show()
8
```



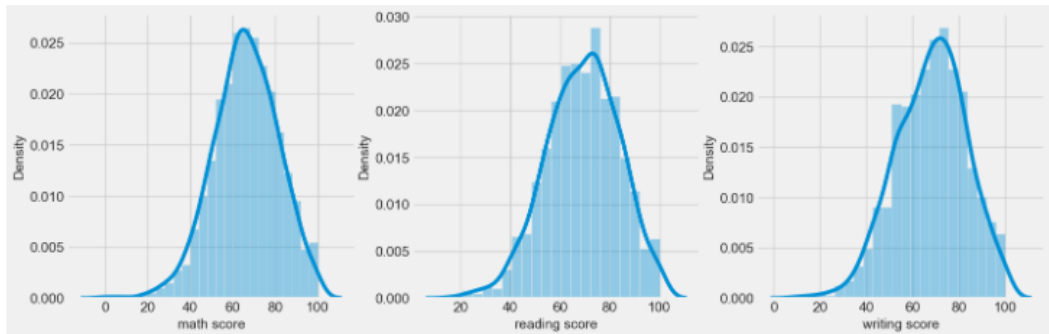
Visualizing The Maths Score

```
In [149]: 1 plt.figure(figsize=(15, 5))
2
3
4 data['math score'].value_counts(normalize=True).plot.bar(figsize=(18, 8), color='skyblue')
5
6 plt.title('Comparison of Math Scores')
7 plt.xlabel('Math Marks/Score')
8 plt.ylabel('Count')
9 plt.show()
10
```



Outliers

```
In [152]: 1 plt.figure(figsize=(16, 5))
2
3 plt.subplot(1, 3, 1)
4 sns.distplot(data['math score'])
5
6 plt.subplot(1, 3, 2)
7 sns.distplot(data['reading score'])
8
9 plt.subplot(1, 3, 3)
10 sns.distplot(data['writing score'])
11
12 plt.show()
13
14 sns.distplot(data['writing score'])
```



lets see the Mean of sample and population

Maths Score

```
In [153]: 1 #seed randome values
2 np.random.seed(6)
3
4 #Take 100 sample values from the dataset of 1000 values
5 sample_math_marks = np.random.choice(a=data['math score'], size=100)
6
7 # sample mean
8 print('Sample mean for math Score:', sample_math_marks.mean())
9
10 #Population mean
11 print('Population mean for Math Score', data['math score'].mean())
```

Sample mean for math Score: 63.12
Population mean for Math Score 66.089

Reading Score

```
In [154]: 1 #Take 100 sample values from the dataset of 1000 values
          2 sample_reading_marks = np.random.choice(a=data['reading score'], size=100)
          3
          4
          5 #Sample mean
          6 print('\nSample mean for Reading Score:', sample_reading_marks.mean())
          7
          8 #Population mean
          9 print('Population mean for reading Score', data['reading score'].mean())
         10
```

Sample mean for Reading Score: 68.5
Population mean for reading Score 69.169

Writing Score

```
In [155]: 1 #Take 100 sample values from the dataset of 1000 values
          2 sample_writing_marks = np.random.choice(a=data['writing score'], size=100)
          3
          4 #Sample mean
          5 print('\nsample mean for Writing Score:', sample_writing_marks.mean())
          6
          7 #Population mean
          8 print('Population mean for Writing Score', data['writing score'].mean())
          9
```

Sample mean for Writing Score: 71.46
Population mean for Writing Score 68.054

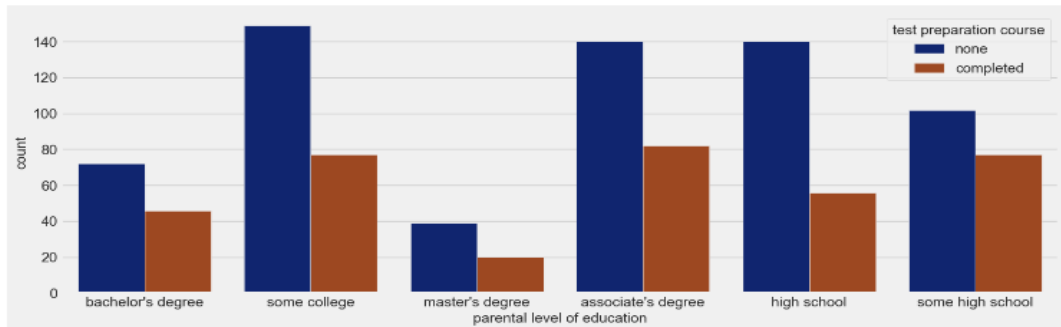
Let check the Confidence Interval for math score

```
In [156]: 1 # Impoer the scipy library
          2 import scipy.stats as stats
          3 import math
          4
          5 # Seed random values
          6 np.random.seed(10)
          7
          8 #Take sample size
          9 sample_size=1000
         10 sample = np.random.choice(a= data['math score'], size = sample_size)
         11 sample_mean = sample.mean()
         12
         13 # Get the z-critical values
         14 z_critical = stats.norm.ppf(q = 0.95)
         15
         16 #check the z-critical value
         17 print('z-critical value', z_critical)
         18
         19 #Get the Population standard deviation
         20 pop_stdev = data['math score'].std()
         21
         22 #check the margin of error
         23 margin_of_error = z_critical * (pop_stdev/math.sqrt(sample_size))
         24
         25 #defining our confidence interval
         26 confidence_interval = (sample_mean - margin_of_error,
         27                        sample_mean +margin_of_error)
         28 #lets print the result
         29 print('Confidence interval:', end=" ")
         30 print(confidence_interval)
         31 print('True mean: {}'.format(data['math score'].mean()))
```

z-critical value 1.6448536269514722
Confidence interval: (64.82729483328328, 66.40470516671672)
True mean: 66.089

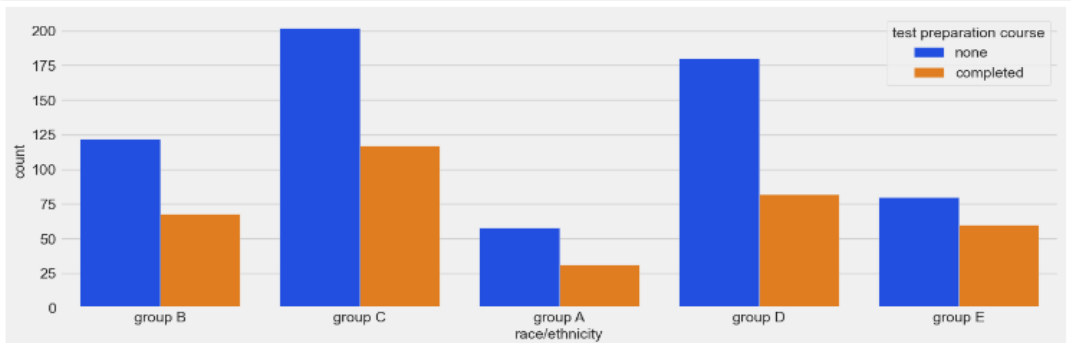
Comparison of Parental degree and test course

```
In [159]: 1 plt.subplots(figsize=(15,5))
2 sns.countplot(x = 'parental level of education', data = data, hue = 'test preparation course', palette = 'dark')
3 plt.show()
```



Comparison of race/ethnicity and test preparation course

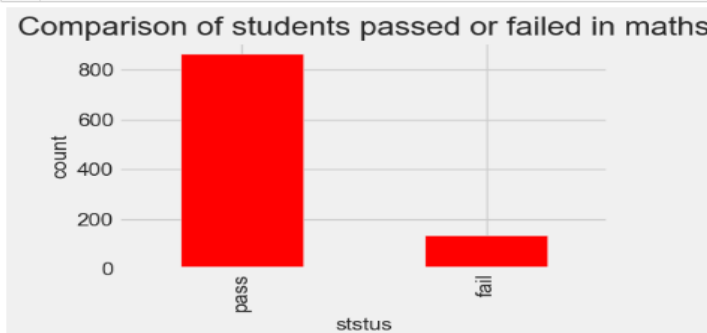
```
In [160]: 1 plt.subplots(figsize = (15,5))
2 sns.countplot(x = 'race/ethnicity', data = data, hue = 'test preparation course', palette = 'bright')
3 plt.show()
```



setting a passing mark for the students to pass on the whether the students are pass or fail

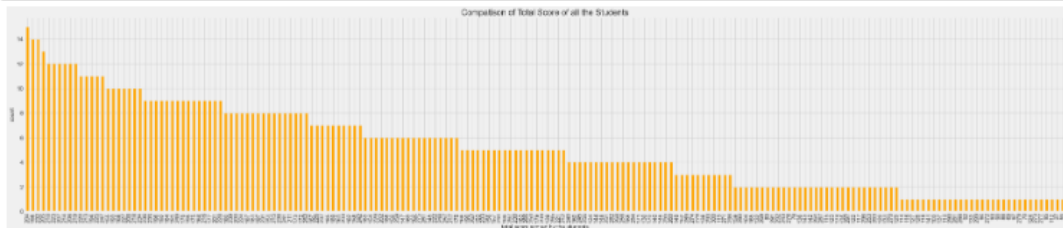
In Math Score

```
In [161]: 1 plt.subplots(figsize=(15,8))
2 passmarks = 50
3
4 data['pass_math'] = np.where(data['math score'] < passmarks, 'fail', 'pass')
5 data['pass_math'].value_counts(dropna = False).plot.bar(color = 'red', figsize = (5, 3))
6
7 plt.title('Comparison of students passed or failed in maths')
8 plt.xlabel('ststus')
9 plt.ylabel('count')
10 plt.show()
```



Comparison the total Score for each student

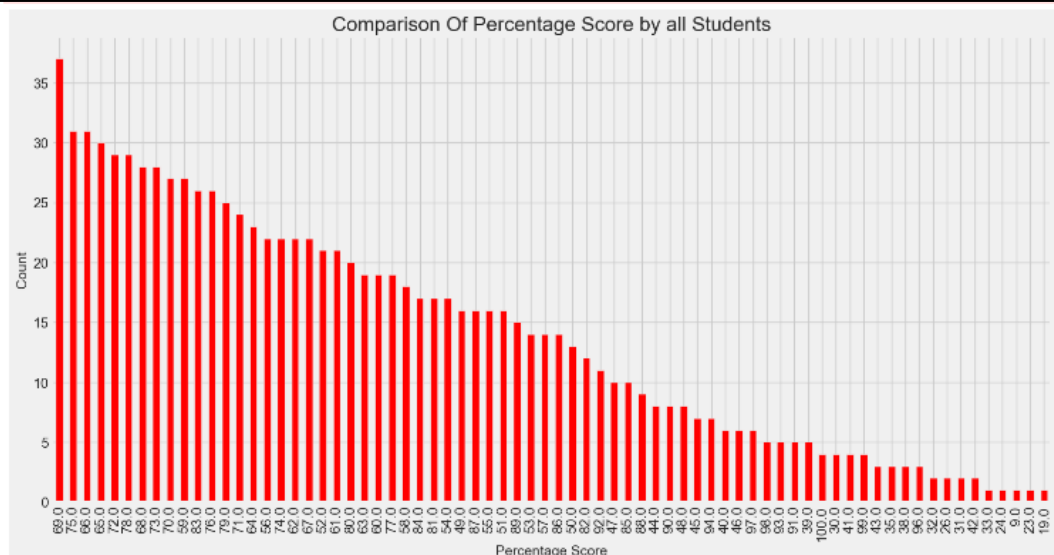
```
In [164]: 1 plt.subplots(figsize=(15,8))
2 data['total_score'] = data['math score'] + data['reading score'] + data['writing score']
3
4 data['total_score'].value_counts(normalize = True)
5 data['total_score'].value_counts(dropna = True).plot.bar(color = 'orange', figsize = (40, 8))
6
7 plt.title('Compatison of Total Score of all the Students')
8 plt.xlabel('total score scored by the students')
9 plt.ylabel('count')
10 plt.show()
```



Comparison Percentage for each of the students

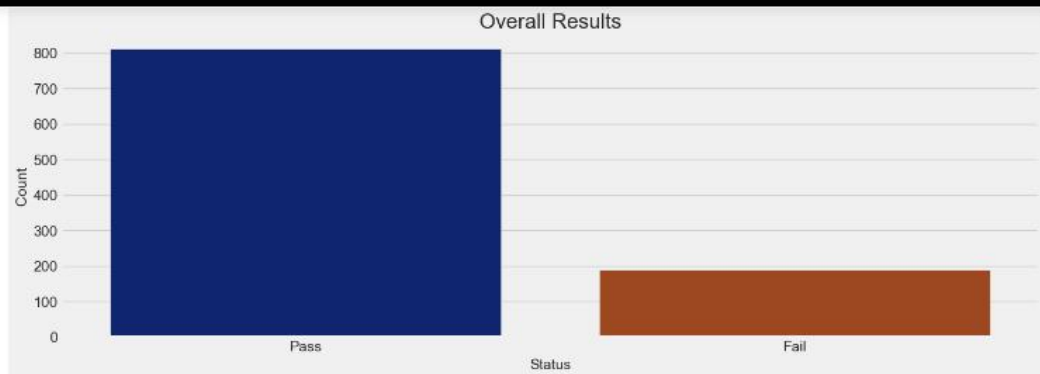
Importing the math library to use for ceil

```
In [165]: 1 from math import *
2 data['percentage'] = data['total_score'] / 3
3
4 for i in range(0, 1000):
5     data['percentage'][i] = ceil(data['percentage'][i])
6
7 data['percentage'].value_counts(normalize=True)
8 data['percentage'].value_counts(dropna=False).plot.bar(figsize=(16, 8), color='red')
9
10 plt.title('Comparison Of Percentage Score by all Students')
11 plt.xlabel('Percentage Score')
12 plt.ylabel('Count')
13 plt.show()
```



Checking with student is Fail & Pass Overall

```
In [166]: 1 import seaborn as sns
2
3 data['status'] = data.apply(lambda x: 'Fail' if x['pass_math'] != 'pass' or
4                             x['pass_reading'] != 'pass' or x['pass_writing'] != 'pass'
5                             else 'Pass', axis=1)
6
7 # Plotting with seaborn
8 plt.figure(figsize=(15, 5))
9 sns.countplot(x='status', data=data, palette='dark')
10 plt.title('Overall Results')
11 plt.xlabel('Status')
12 plt.ylabel('Count')
13 plt.show()
14
```



Submitted By:- Rahila & Tushar

4. Appendix B abbreviation

Abbreviation

Python - High level language

Pandas - Python's library

ML- Machin learning

Kaggle - subsidiary of Google

5. References

[Kaggle: Your Machine Learning and Data Science Community](#)