```
import pandas as pd
df = pd.read_csv('/content/StudentPerformanceFactors.csv')
df
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Pr
0	23	84	Low	High	No	7	
1	19	64	Low	Medium	No	8	
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
4	19	92	Medium	Medium	Yes	6	
...	...	...	...	...	...	...	
6602	25	69	High	Medium	No	7	
6603	23	76	High	Medium	No	8	
6604	20	90	Medium	Low	Yes	6	
6605	10	86	High	High	Yes	6	
6606	15	67	Medium	Low	Yes	9	

6607 rows × 20 columns

```
df.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previ
0	23	84	Low	High	No	7	
1	19	64	Low	Medium	No	8	
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
4	19	92	Medium	Medium	Yes	6	

```
df.tail()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Pr
6602	25	69	High	Medium	No	7	
6603	23	76	High	Medium	No	8	
6604	20	90	Medium	Low	Yes	6	
6605	10	86	High	High	Yes	6	
6606	15	67	Medium	Low	Yes	9	

```
df.shape
```

```
(6607, 20)
```

```
df.columns
```

```
Index(['Hours_Studied', 'Attendance', 'Parental_Involvement',
       'Access_to_Resources', 'Extracurricular_Activities', 'Sleep_Hours',
       'Previous_Scores', 'Motivation_Level', 'Internet_Access',
       'Tutoring_Sessions', 'Family_Income', 'Teacher_Quality', 'School_Type',
       'Peer_Influence', 'Physical_Activity', 'Learning_Disabilities',
       'Parental_Education_Level', 'Distance_from_Home', 'Gender',
       'Exam_Score'],
      dtype='object')
```

```
df.dtypes
```

	0
Hours_Studied	int64
Attendance	int64
Parental_Involvement	object
Access_to_Resources	object
Extracurricular_Activities	object
Sleep_Hours	int64
Previous_Scores	int64
Motivation_Level	object
Internet_Access	object
Tutoring_Sessions	int64
Family_Income	object
Teacher_Quality	object
School_Type	object
Peer_Influence	object
Physical_Activity	int64
Learning_Disabilities	object
Parental_Education_Level	object
Distance_from_Home	object
Gender	object
Exam_Score	int64

dtype: object

```
df[['Hours_Studied','Attendance']].describe(include='all')
```

	Hours_Studied	Attendance
count	6607.000000	6607.000000
mean	19.975329	79.977448
std	5.990594	11.547475
min	1.000000	60.000000
25%	16.000000	70.000000
50%	20.000000	80.000000
75%	24.000000	90.000000
max	44.000000	100.000000

```
score_columns = df.filter(like='Score')
score_columns.head()
```

	Previous_Scores	Exam_Score
0	73	67
1	59	61
2	91	74
3	98	71
4	65	70

```
high_scores = df[df['Exam_Score'] > 70]
high_scores.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Prev
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
5	19	88	Medium	Medium	Yes	8	
9	23	98	Medium	Medium	Yes	8	
11	17	97	Medium	High	Yes	6	

```
male_students = df[df['Gender'] == 'Male']
female_students = df[df['Gender'] == 'Female']
male_students.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previ
0	23	84	Low	High	No	7	
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
5	19	88	Medium	Medium	Yes	8	
6	29	84	Medium	Low	Yes	7	

```
female_students.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Prev
1	19	64	Low	Medium	No	8	
4	19	92	Medium	Medium	Yes	6	
15	17	68	Medium	Medium	No	8	
17	22	70	Low	Medium	Yes	6	
18	15	80	Medium	Medium	Yes	9	

```
male_highest = df[(df['Gender'] == 'Male') & (df['Exam_Score'] > 70)]
male_highest.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Prev
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
5	19	88	Medium	Medium	Yes	8	
9	23	98	Medium	Medium	Yes	8	
11	17	97	Medium	High	Yes	6	

```
female_highest = df[(df['Gender'] == 'Female') & (df['Exam_Score'] > 70)]
female_highest.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Pre
73	29	92	Low	Medium	No	4	
94	18	89	High	Medium	Yes	4	
113	35	99	High	High	Yes	7	
115	22	89	High	High	Yes	6	
118	27	97	Low	High	Yes	8	

```
df['Gender'].value_counts()
```

Gender	count
Male	3814
Female	2793

dtype: int64

```
df['Exam_Score'].value_counts()
```

```
             count
Exam_Score
```

```
import numpy as np
```

```
exam_score_array = df['Exam_Score'].to_numpy()
exam_score_array
```

```
array([67, 61, 74, ..., 68, 68, 64])
```

```
exam_score_array_2D = df['Exam_Score'].to_numpy().reshape(-1,1)
exam_score_array_2D
```

```
array([[67],
       [61],
       [74],
       ...,
       [68],
       [68],
       [64]])
```

```
exam_score_array_2D.shape
```

```
(6607, 1)
```

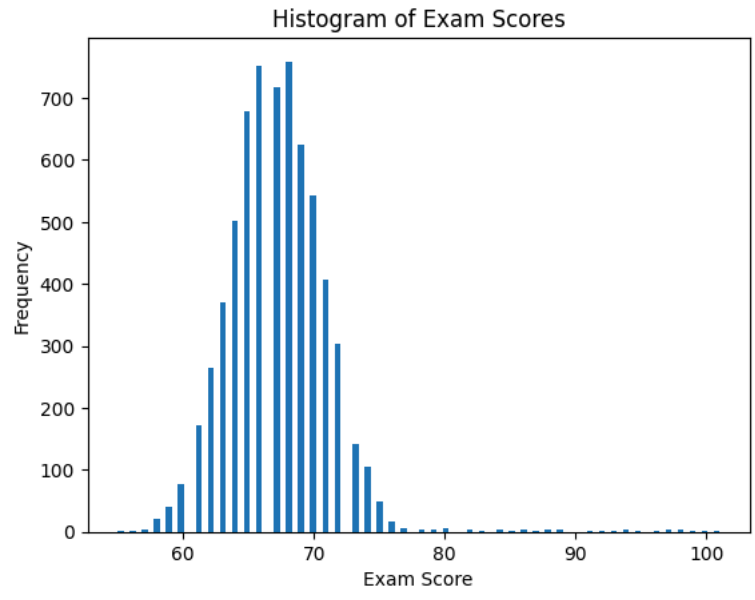
```
from numpy._core.defchararray import count
mean_score = np.mean(exam_score_array)
median_score = np.median(exam_score_array)
std_score = np.std(exam_score_array)
min_score = np.min(exam_score_array)
max_score = np.max(exam_score_array)
```

```
print('mean_score',mean_score)
print('median_score',median_score)
print('std_score',std_score)
```

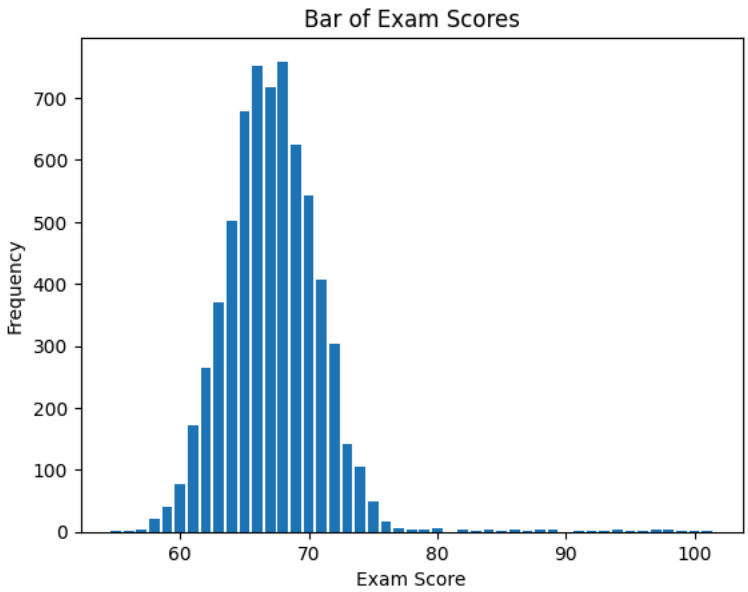
```
mean_score 67.23565914938702
median_score 67.0
std_score 3.8901613508847057
```

```
import matplotlib.pyplot as plt
```

```
plt.hist(exam_score_array,bins=100)
plt.xlabel('Exam Score')
plt.ylabel('Frequency')
plt.title('Histogram of Exam Scores')
plt.show()
```



```
scores, counts = np.unique(exam_score_array, return_counts=True)
plt.bar(scores, counts)
plt.xlabel('Exam Score')
plt.ylabel('Frequency')
plt.title('Bar of Exam Scores')
plt.show()
```




```
scores, counts = np.unique(exam_score_array, return_counts=True)
plt.pie(counts, labels=scores, autopct='%1.1f%%')
plt.title('Pie Chart of Exam Scores')
```

```
Text(0.5, 1.0, 'Pie Chart of Exam Scores')
```