```
import pandas as pd
```

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

```
df
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previou
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	Previous_S
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	Previou
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
```

```
(25000, 16)
```

```
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.describe(include='all')
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Prev
count	6607.000000	6607.000000	6607	6607	6607	6607.00000	
unique	NaN	NaN	3	3	2	NaN	
top	NaN	NaN	Medium	Medium	Yes	NaN	
freq	NaN	NaN	3362	3319	3938	NaN	
mean	19.975329	79.977448	NaN	NaN	NaN	7.02906	
std	5.990594	11.547475	NaN	NaN	NaN	1.46812	
min	1.000000	60.000000	NaN	NaN	NaN	4.00000	
25%	16.000000	70.000000	NaN	NaN	NaN	6.00000	
50%	20.000000	80.000000	NaN	NaN	NaN	7.00000	
75%	24.000000	90.000000	NaN	NaN	NaN	8.00000	
max	44.000000	100.000000	NaN	NaN	NaN	10.00000	

```
df['math_score'].describe()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'math_score'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                                          2 frames
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'math_score'
```

```
scores=df.filter(like='score')
scores
```

0
1
2
3
4
...
6602
6603
6604
6605
6606

6607 rows × 0 columns

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

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_
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	

```
gender=df.filter(like='Gender')
gender
```

	Gender
0	Male
1	Female
2	Male
3	Male
4	Female
...	...
6602	Female
6603	Female
6604	Female
6605	Female
6606	Male

6607 rows × 1 columns

```
male_candidates=df[df['Gender']=='Male']
female_candidates=df[df['Gender']=='Female']
```

```
male_candidates
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previou
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	
...	...	...	...	...	...	...	
6591	13	74	Medium	High	Yes	8	
6592	29	100	Medium	Low	Yes	8	
6594	9	90	High	High	Yes	7	
6596	17	92	Medium	Medium	No	7	
6606	15	67	Medium	Low	Yes	9	

3814 rows × 20 columns

```
female_candidates
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previou
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	
...	...	...	...	...	...	...	
6601	20	83	Medium	Low	No	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	

2793 rows × 20 columns

```
male_high_score=df[(df['Gender']=='Male')&(df['Exam_Score']>70)]
male_high_score
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previou
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	
...	...	...	...	...	...	...	
6531	23	96	Medium	High	No	6	
6537	24	78	High	High	Yes	8	
6565	24	89	Medium	Low	No	4	
6566	29	96	High	Medium	No	8	
6592	29	100	Medium	Low	Yes	8	

635 rows × 20 columns

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

Gender	count
Male	3814
Female	2793

dtype: int64

```
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]])
```

```
mean=np.mean(exam_score_array)
mean
```

```
np.float64(67.23565914938702)
```

```
median=np.median(exam_score_array)
median
```

```
np.float64(67.0)
```

```
max=np.max(exam_score_array)
max
```

```
np.int64(101)
```

```
min=np.min(exam_score_array)
min
```

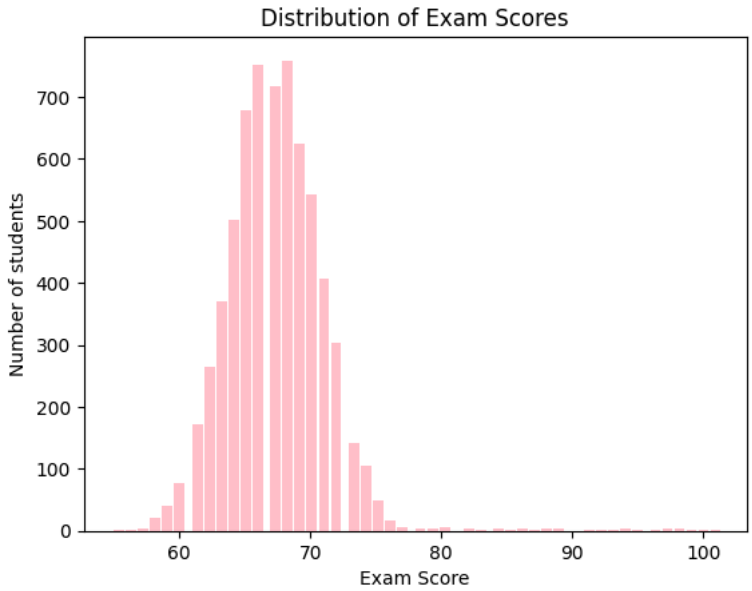
```
np.int64(55)
```

```
std=np.std(exam_score_array)
std
```

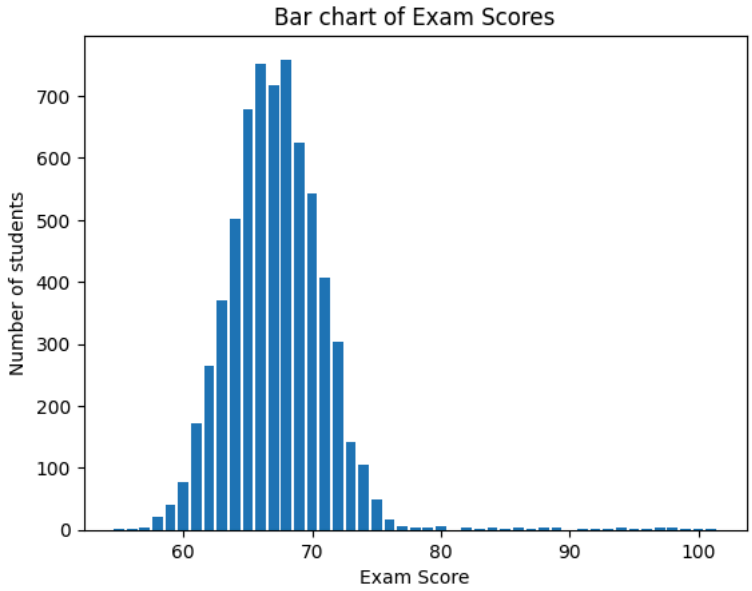
```
np.float64(3.8901613508847057)
```

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

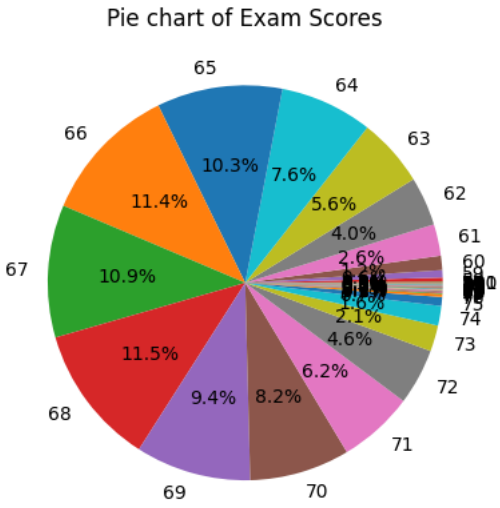
```
plt.hist(exam_score_array,bins=100,width=0.8,color='pink')
plt.xlabel('Exam Score')
plt.ylabel('Number of students')
plt.title('Distribution of Exam Scores')
plt.show()
```




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




```
hourly_studied_array=df['Hours_Studied'].to_numpy()
```

```
plt.scatter(hourly_studied_array, exam_score_array,s=1)
plt.xlabel('Hours Studied')
plt.ylabel('Exam Score')
plt.title('Hourly studied vs Exam score')
plt.show()
```