```
import pandas as pd
df=pd.read_csv('Student_Performance.csv')
df.head()
```

	student_id	age	gender	school_type	parent_education	study_hours	attendance_percentage	internet_access	travel_time
0	1	14	male	public	post graduate	3.1	84.3	yes	<15 min
1	2	18	female	public	graduate	3.7	87.8	yes	>60 min
2	3	17	female	private	post graduate	7.9	65.5	no	<15 min
3	4	16	other	public	high school	1.1	58.1	no	15-30 min
4	5	16	female	public	high school	1.3	61.0	yes	30-60 min

Next steps: Generate code with df New interactive sheet

```
from google.colab import drive
drive.mount('/content/drive')
```

```
df.head()
```

	student_id	age	gender	school_type	parent_education	study_hours	attendance_percentage	internet_access	travel_time
0	1	14	male	public	post graduate	3.1	84.3	yes	<15 min
1	2	18	female	public	graduate	3.7	87.8	yes	>60 min
2	3	17	female	private	post graduate	7.9	65.5	no	<15 min
3	4	16	other	public	high school	1.1	58.1	no	15-30 min
4	5	16	female	public	high school	1.3	61.0	yes	30-60 min

Next steps: Generate code with df New interactive sheet

```
df.tail()
```

	student_id	age	gender	school_type	parent_education	study_hours	attendance_percentage	internet_access	travel_t
24995	12047	17	female	public	phd	1.8	55.2	yes	15-30
24996	1102	16	female	private	diploma	2.7	97.1	yes	<15
24997	4422	19	other	private	post graduate	1.0	63.0	yes	<15
24998	7858	14	male	private	diploma	1.0	69.4	yes	15-30
24999	11621	18	other	public	no formal	0.7	60.3	yes	30-60

```
df.shape
```

```
(25000, 16)
```

```
df.columns
```

```
Index(['student_id', 'age', 'gender', 'school_type', 'parent_education',
       'study_hours', 'attendance_percentage', 'internet_access',
       'travel_time', 'extra_activities', 'study_method', 'math_score',
       'science_score', 'english_score', 'overall_score', 'final_grade'],
      dtype='object')
```

```
df.dtypes
```

	0
student_id	int64
age	int64
gender	object
school_type	object
parent_education	object
study_hours	float64
attendance_percentage	float64
internet_access	object
travel_time	object
extra_activities	object
study_method	object
science_score	float64
overall_score	float64

dtype: object

```
df.describe()
```

	student_id	age	study_hours	attendance_percentage	math_score	science_score	english_score	overall_sc
count	25000.00000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.0000
mean	7493.04680	16.482760	4.253224	75.084084	63.785944	63.745320	63.681948	64.0061
std	4323.56215	1.703895	2.167541	14.373171	20.875262	20.970529	20.792693	18.9320
min	1.00000	14.000000	0.500000	50.000000	0.000000	0.000000	0.000000	14.5000
25%	3743.75000	15.000000	2.400000	62.800000	48.300000	48.200000	48.300000	49.0000
50%	7461.50000	16.000000	4.300000	75.100000	64.100000	64.100000	64.200000	64.2000
75%	11252.00000	18.000000	6.100000	87.500000	80.000000	80.000000	80.000000	79.0000
max	15000.00000	19.000000	8.000000	100.000000	100.000000	100.000000	100.000000	100.0000

```
scores_columns =['math_score','science_score','overall_score']
scores_df = df[scores_columns]
print(scores_df.head())
```

```
   math_score  science_score  overall_score
0        42.7           55.4           53.1
1        57.6           68.8           61.3
2        84.8           95.0           89.6
3        44.4           27.5           41.6
4         8.9           32.7           25.4
```

```
math_above_70 = df[df['math_score']>70]
print(math_above_70.head())
```

```
    student_id  age  gender school_type parent_education  study_hours  \
2            3   17  female     private    post graduate          7.9   
9           10   14  female      public          diploma          6.8   
10          11   17  female     private         graduate          6.1   
12          13   18  female     private      high school          6.8   
14          15   18   other      public      high school          4.9   

    attendance_percentage internet_access travel_time extra_activities  \
2                    65.5              no     <15 min               no   
9                    62.4             yes     >60 min               no   
10                   90.5             yes   15-30 min               no   
12                   58.2             yes     >60 min               no   
14                   85.3             yes     <15 min               no   

   study_method  math_score  science_score  english_score  overall_score  \
2         notes        84.8           95.0           79.2           89.6   
9         mixed        71.9           70.4           81.3           69.6   
10        notes        75.4           82.7           70.8           77.5   
12        mixed        75.1           80.5           56.2           73.6   
14     coaching        88.1           66.1           70.3           69.3   

   final_grade  
2            b  
9            d  
10           c  
12           c  
14           d  
```

```
male_students = df[df['gender']=='male']
female_students = df[df['gender']=='female']
```

```
gender_counts=df['gender'].value_counts()
print(gender_counts)
```

```
gender
other     8463
female    8290
male      8247
Name: count, dtype: int64
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Load the dataset
df = pd.read_csv('/content/Student_Performance.csv')

# Display first 5 rows
df.head()
```

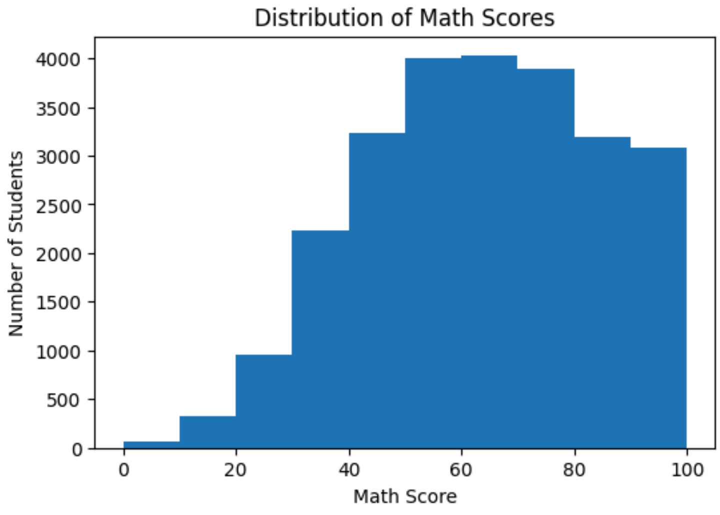
	student_id	age	gender	school_type	parent_education	study_hours	attendance_percentage	internet_access	travel_time
0	1	14	male	public	post graduate	3.1	84.3	yes	<15 min
1	2	18	female	public	graduate	3.7	87.8	yes	>60 min
2	3	17	female	private	post graduate	7.9	65.5	no	<15 min
3	4	16	other	public	high school	1.1	58.1	no	15-30 min
4	5	16	female	public	high school	1.3	61.0	yes	30-60 min

Next steps: Generate code with df New interactive sheet

```
df.columns
```

```
Index(['student_id', 'age', 'gender', 'school_type', 'parent_education',
       'study_hours', 'attendance_percentage', 'internet_access',
       'travel_time', 'extra_activities', 'study_method', 'math_score',
       'science_score', 'english_score', 'overall_score', 'final_grade'],
      dtype='object')
```

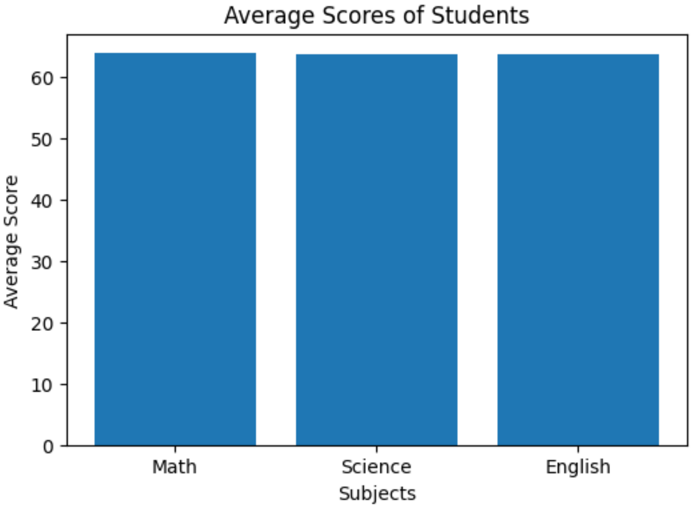
```
plt.figure(figsize=(6,4))
plt.hist(df['math_score'], bins=10)
plt.xlabel('Math Score')
plt.ylabel('Number of Students')
plt.title('Distribution of Math Scores')
plt.show()
```



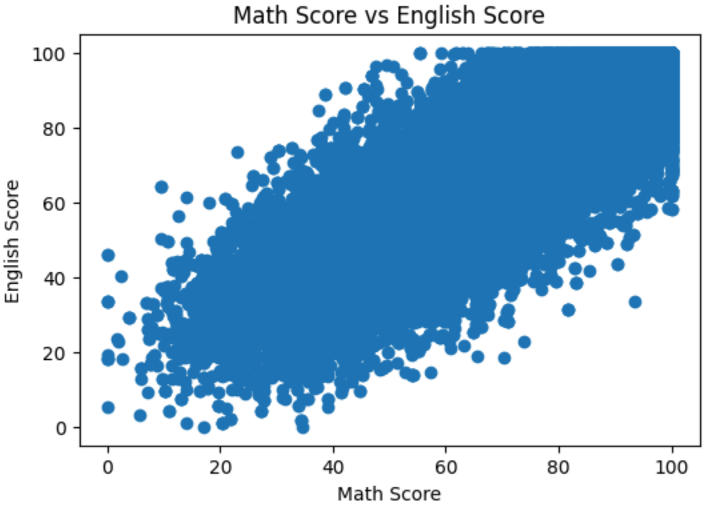
```
# Calculate average scores
average_scores = [
    df['math_score'].mean(),
    df['science_score'].mean(),
    df['english_score'].mean()
]

subjects = ['Math', 'Science', 'English']
```

```
plt.figure(figsize=(6,4))
plt.bar(subjects, average_scores)
plt.xlabel('Subjects')
plt.ylabel('Average Score')
plt.title('Average Scores of Students')
plt.show()
```



```
plt.figure(figsize=(6,4))
plt.scatter(df['math_score'], df['english_score'])
plt.xlabel('Math Score')
plt.ylabel('English Score')
plt.title('Math Score vs English Score')
plt.show()
```



```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Load dataset
df = pd.read_csv('/content/Student_Performance.csv')

# Display first few rows
df.head()
```

```
   student_id  age  gender  school_type  parent_education  study_hours  attendance_percentage  internet_access  travel_time
0           1   14    male       public     post graduate          3.1                   84.3              yes      <15 min
1           2   18  female       public          graduate          3.7                   87.8              yes      >60 min
2           3   17  female      private     post graduate          7.9                   65.5               no      <15 min
3           4   16   other       public       high school          1.1                   58.1               no    15-30 min
4           5   16  female       public       high school          1.3                   61.0              yes    30-60 min
```

Next steps: Generate code with df New interactive sheet

```
df.columns
```

```
Index(['student_id', 'age', 'gender', 'school_type', 'parent_education',
       'study_hours', 'attendance_percentage', 'internet_access',
       'travel_time', 'extra_activities', 'study_method', 'math_score',
       'science_score', 'english_score', 'overall_score', 'final_grade'],
      dtype='object')
```

```
gender_group = df.groupby('gender')[['math_score', 'science_score', 'english_score']].mean()

gender_group
```

```
         math_score  science_score  english_score
gender
female    64.045428      63.965862      63.893317
male      63.872754      63.852104      63.661150
other     63.447170      63.425227      63.495167
```

Next steps: Generate code with gender_group New interactive sheet

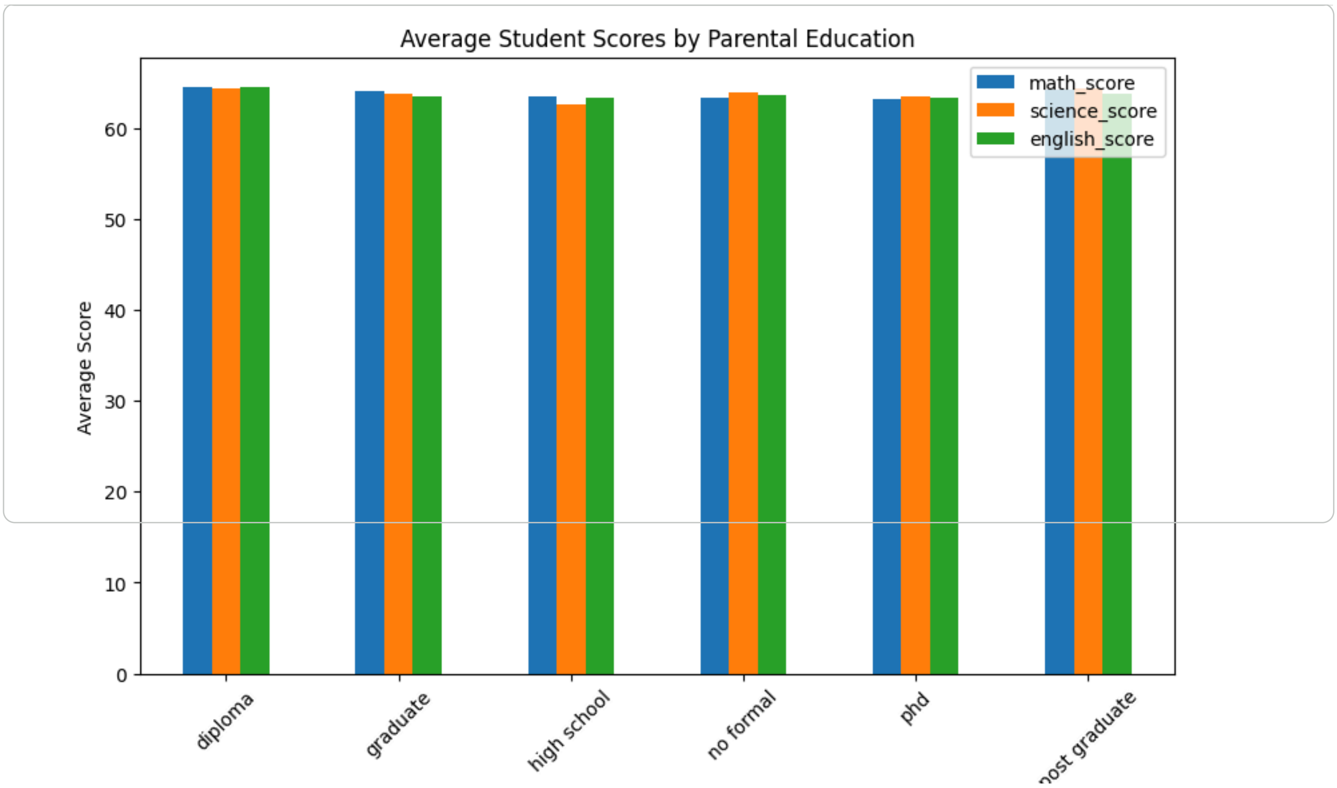
```
parent_edu_group = df.groupby('parent_education')[['math_score', 'science_score', 'english_score']].mean()

parent_edu_group
```

```
                  math_score  science_score  english_score
parent_education
diploma            64.546847      64.349374      64.539569
graduate           64.040974      63.794766      63.468839
high school        63.413127      62.622568      63.358502
no formal          63.384531      63.930326      63.594680
phd                63.158813      63.443859      63.304633
post graduate      64.126287      64.314013      63.785582
```

Next steps: Generate code with parent_edu_group New interactive sheet

```
parent_edu_group.plot(kind='bar', figsize=(10,6))
plt.xlabel('Parental Level of Education')
plt.ylabel('Average Score')
plt.title('Average Student Scores by Parental Education')
plt.xticks(rotation=45)
plt.show()
```


Average Student Scores by Parental Education
Average Score
0
10
20
30
40
50
60
math_score
science_score
english_score
diploma
graduate
high school
no formal
phd
post graduate