

Dataset <https://www.kaggle.com/datasets/spscientist/students-performance-in-exams> In Python (using pandas):

```
import pandas as pd
```

```
df = pd.read_csv('/content/StudentsPerformance.csv')
display(df.head())
```

	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

import dataset from drive <https://www.kaggle.com/datasets/spscientist/students-performance-in-exams>

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
display(df)
```

	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
...	...	...	...	...	...	...	...	...
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

1000 rows × 8 columns

Next steps:

[Generate code with df](#)

[New interactive sheet](#)

import libraries

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

read CSV file and give me the head

Start coding or [generate](#) with AI.

```
df = pd.read_csv('/content/StudentsPerformance.csv')
display(df.head())
```

	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

Start coding or [generate](#) with AI.

here the code for the type to know how we can work which type of data numerical or categorical

```
display(df.dtypes)
```

```

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
```

handling missing data of there is a missing (no missing data here = 0)

```
df.isnull().sum()
```

```

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
```

import Z-scores to Handle outliers

```

import pandas as pd
import numpy as np
from scipy import stats

df = pd.read_csv('StudentsPerformance.csv')

numeric_cols = ['math score', 'reading score', 'writing score']
```

```
z_scores = np.abs(stats.zscore(df[numeric_cols]))

df_no_outliers = df[(z_scores < 3).all(axis=1)]

print(f"Original data shape: {df.shape}")
print(f"After removing outliers: {df_no_outliers.shape}")

removed_rows = df.shape[0] - df_no_outliers.shape[0]
print(f"Number of outliers removed: {removed_rows}")
```

Original data shape: (1000, 8)
After removing outliers: (993, 8)
Number of outliers removed: 7

Start coding or [generate](#) with AI.

print dataset after Z-Scores

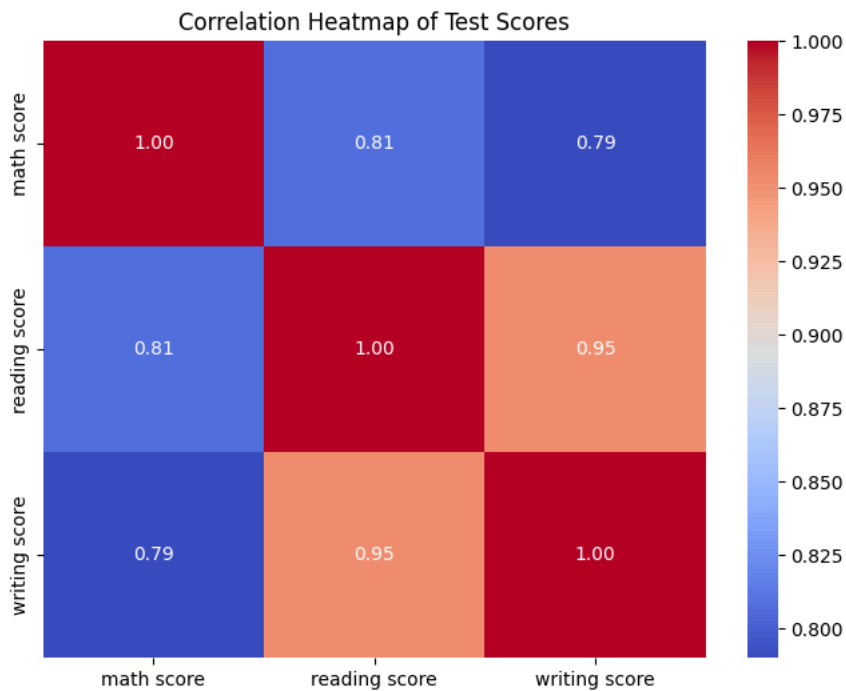
```
display(df_no_outliers)
```

	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
...	...	...	...	...	...	...	...	...
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
...	...	...	...	...	...	...	...	...

Next steps: [Generate code with df_no_outliers](#) [New interactive sheet](#)

Now here i Perform data visualization using Heatmap

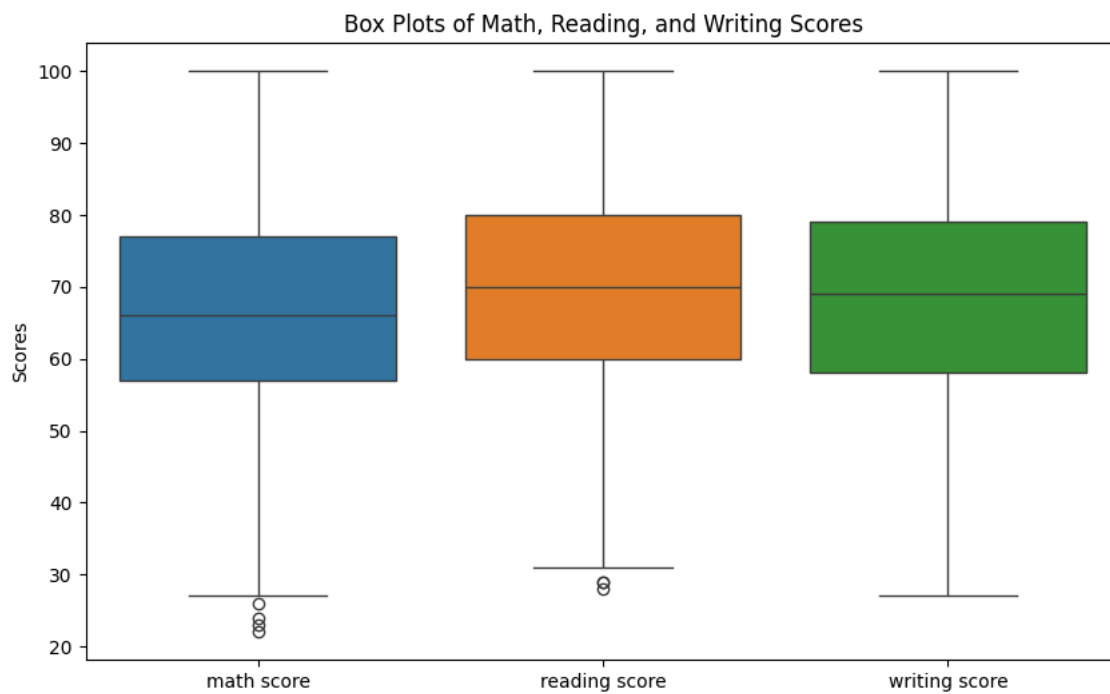
```
correlation_matrix = df_no_outliers.select_dtypes(include=np.number).corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of Test Scores')
plt.show()
```



Using boxplot

```
scores_df = df_no_outliers[['math score', 'reading score', 'writing score']]

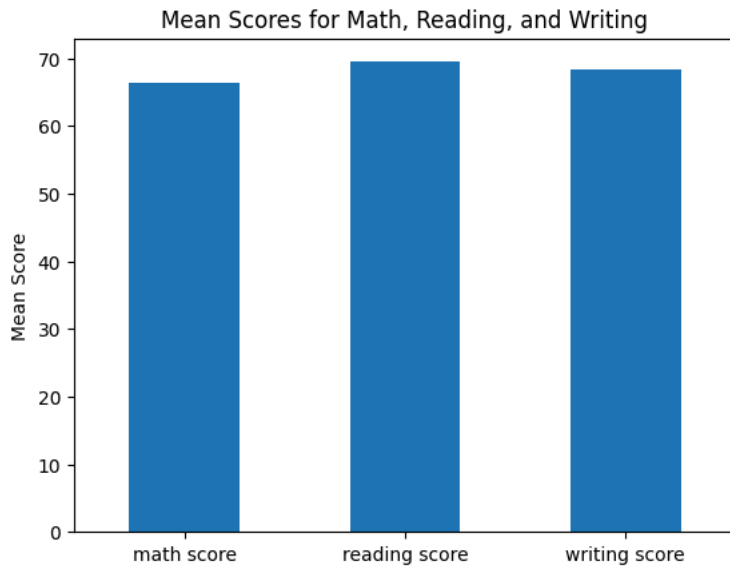
plt.figure(figsize=(10, 6))
sns.boxplot(data=scores_df)
plt.title('Box Plots of Math, Reading, and Writing Scores')
plt.ylabel('Scores')
plt.show()
```



Using lable

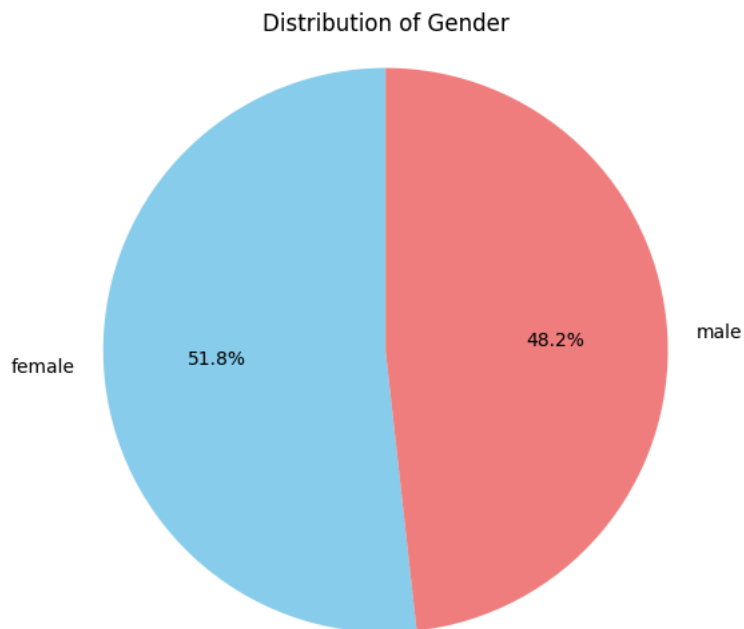
```
mean_scores.plot(kind='bar')
plt.title('Mean Scores for Math, Reading, and Writing')
plt.ylabel('Mean Score')
plt.xticks(rotation=0)
plt.show()
```

```
plt.show()
```



i use piechart to compare between gender

```
gender_counts = df_no_outliers['gender'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=90, colors=['skyblue', 'lightcoral'])
plt.title('Distribution of Gender')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



Thank you end of code