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

```
#Load the data
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
df = pd.read_csv('/content/drive/MyDrive/StudentsPerformance.csv')
```

```
print(df.head())
```

```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

```
df.tail()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
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

```
#Column data types
df.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
```

```
#Numerical features
print(df.describe())
```

```
       math score  reading score  writing score
count  1000.00000    1000.000000    1000.000000
mean     66.08900      69.169000      68.054000
std      15.16308      14.600192      15.195657
min       0.00000      17.000000      10.000000
25%      57.00000      59.000000      57.750000
50%      66.00000      70.000000      69.000000
75%      77.00000      79.000000      79.000000
max     100.00000     100.000000     100.000000
```

```
#data shape
df.shape
```

```
(1000, 8)
```

```
#missing values
df.isnull().sum()
```

	0
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

```
#Categorical features
df.value_counts()
```

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	count
female	**group A**	**associate's degree**	**free/reduced**	**none**	**37**	**57**	**56**	1
male	**group C**	**associate's degree**	**standard**	**completed**	**57**	**54**	**56**	1
			free/reduced	**completed**	**60**	**51**	**56**	1
					65	**67**	**65**	1
						73	**68**	1
...	...	...	...	...	...	...	...	...
female	**group D**	**associate's degree**	**standard**	**none**	**71**	**71**	**74**	1
					74	**81**	**83**	1
					76	**74**	**73**	1
					77	**77**	**73**	1
male	**group E**	**some high school**	**standard**	**none**	**94**	**88**	**78**	1

1000 rows × 1 columns

```
df.nunique()
```

	0
gender	2
race/ethnicity	5
parental level of education	6
lunch	2
test preparation course	2
math score	81
reading score	72
writing score	77

dtype: int64

```
df['Total score']=df['math score']+df['reading score']+df['writing score']
df['Average']=df['Total score']/3
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total score	Average
0	female	group B	bachelor's degree	standard	none	72	72	74	218	72.666667
1	female	group C	some college	standard	completed	69	90	88	247	82.333333
2	female	group B	master's degree	standard	none	90	95	93	278	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	148	49.333333

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Converting score from int --> float
df['math score']=pd.to_numeric(df['math score'],downcast='float')
```

```
print("Average math score is    : {}".format(np.mean(df['math score'])))
print("Average reading score is : {}".format(np.mean(df['reading score'])))
print("Average writing score is : {}".format(np.mean(df['writing score'])))
print("Average total score is   : {}".format(np.mean(df['Total score'])/3))
```

```
Average math score is    : 66.08899688720703
Average reading score is : 69.169
Average writing score is : 68.054
Average total score is   : 67.77066666666667
```

```
Students = df.drop(['race/ethnicity','parental level of education'],axis=1)
Students.head()
```

	gender	lunch	test preparation course	math score	reading score	writing score	Total score	Average
0	female	standard	none	72.0	72	74	218	72.666667
1	female	standard	completed	69.0	90	88	247	82.333333
2	female	standard	none	90.0	95	93	278	92.666667
3	male	free/reduced	none	47.0	57	44	148	49.333333
4	male	standard	none	76.0	78	75	229	76.333333

Next steps: Generate code with Students | View recommended plots | New interactive sheet

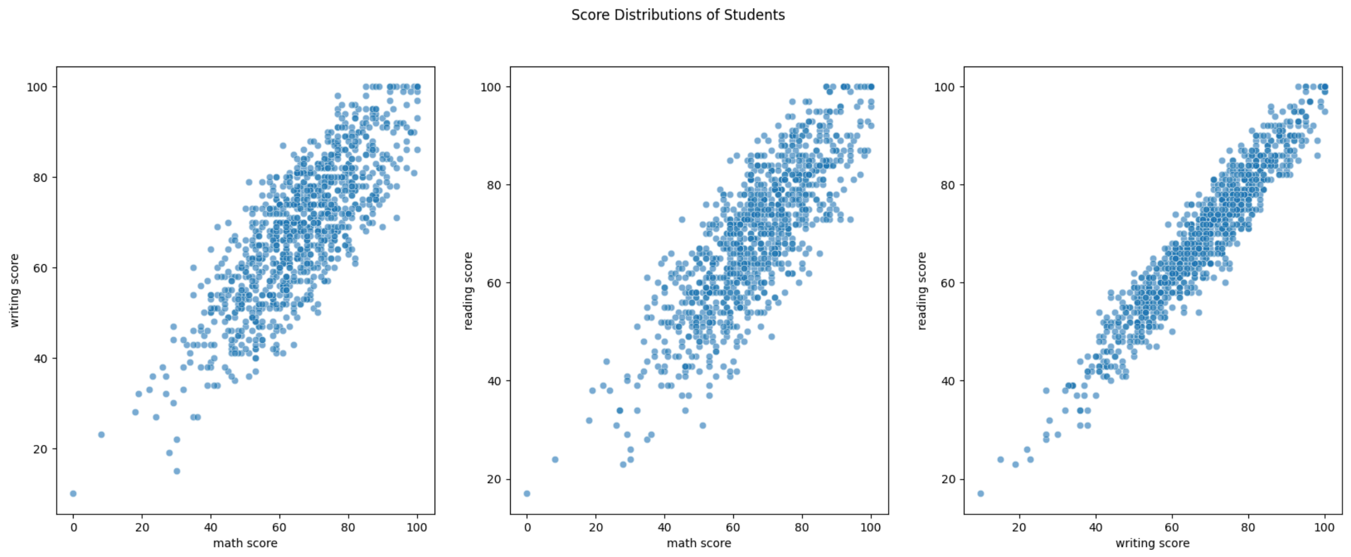
```
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total score	Average
0	female	group B	bachelor's degree	standard	none	72.0	72	74	218	72.666667
1	female	group C	some college	standard	completed	69.0	90	88	247	82.333333
2	female	group B	master's degree	standard	none	90.0	95	93	278	92.666667
3	male	group A	associate's degree	free/reduced	none	47.0	57	44	148	49.333333

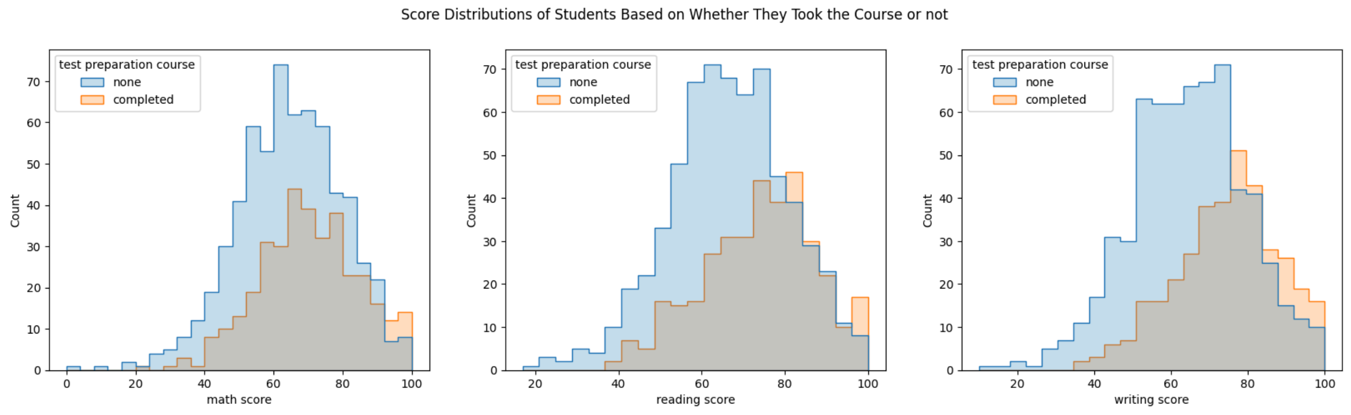
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
fig, ax = plt.subplots(ncols=3, figsize=(20,7))

fig.suptitle('Score Distributions of Students')

sns.scatterplot(data=df, x='math score', y='writing score', ax=ax[0], alpha=0.6)
sns.scatterplot(data=df, x='math score', y='reading score', ax=ax[1], alpha=0.6)
sns.scatterplot(data=df, x='writing score', y='reading score', ax=ax[2], alpha=0.6)
```

```
<Axes: xlabel='writing score', ylabel='reading score'>
```

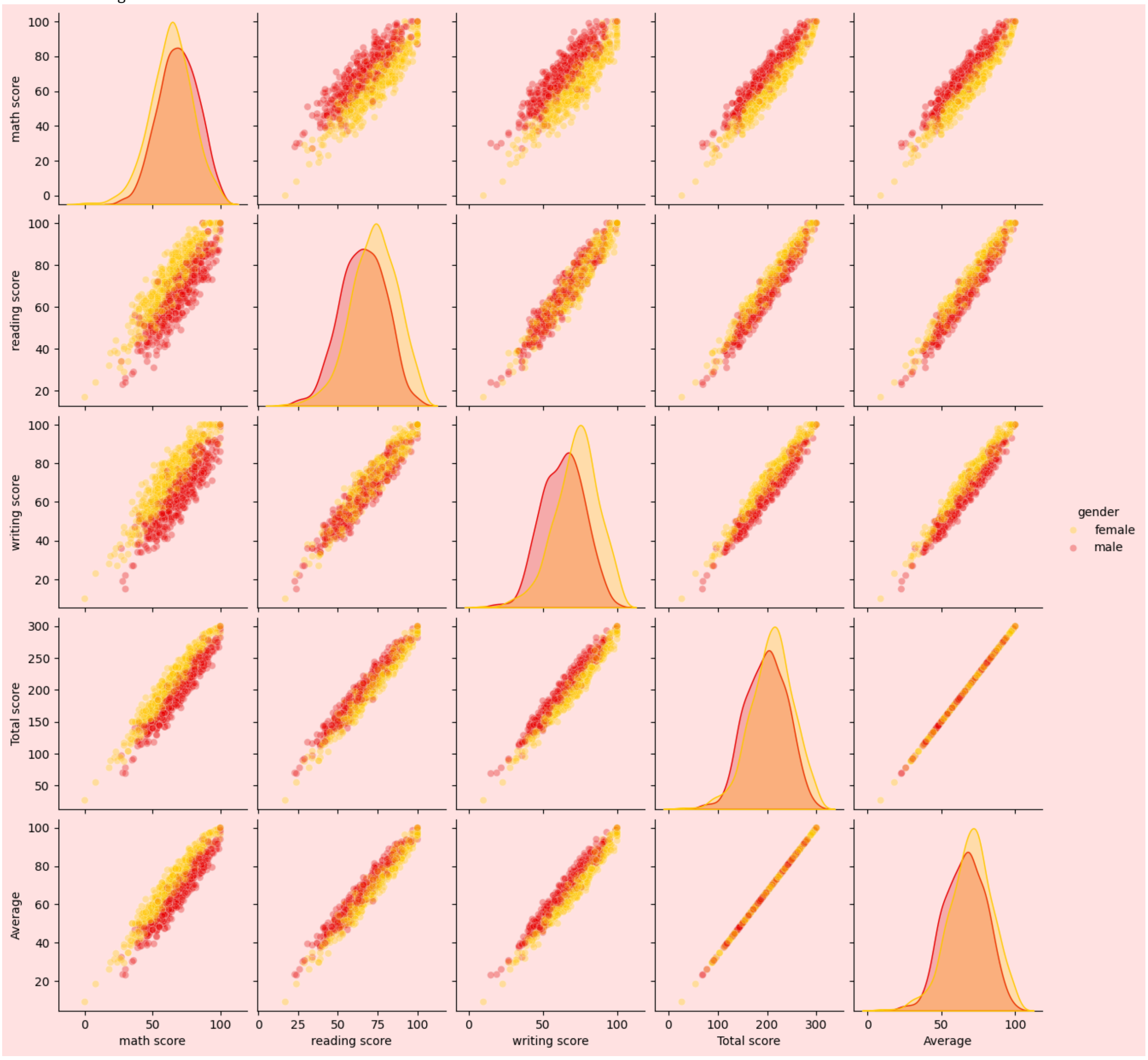


```
fig, ax = plt.subplots(ncols=3, figsize=(20,5))

fig.suptitle('Score Distributions of Students Based on Whether They Took the Course or not')

a= sns.histplot(df, x='math score', ax=ax[0], hue='test preparation course', element='step')
b= sns.histplot(df, x='reading score', ax=ax[1], hue='test preparation course', element='step')
c= sns.histplot(df, x='writing score', ax=ax[2], hue='test preparation course', element='step')

sns.move_legend(a, "upper left", bbox_to_anchor=(0, 1))
sns.move_legend(b, "upper left", bbox_to_anchor=(0, 1))
sns.move_legend(c, "upper left", bbox_to_anchor=(0, 1))
```



```
plt.rcParams['axes.facecolor'] = "#ffe5e5"
plt.rcParams['figure.facecolor'] = "#ffe5e5"
sns.pairplot(data=df,hue='gender',plot_kws={'alpha':0.3},palette='hot_r')
```

```
<seaborn.axisgrid.PairGrid at 0x7eb221cf7400>
```


math score
reading score
writing score
Total score
Average
gender
female
male