```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('/content/StudentsPerformance (2).csv')
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 | View recommended plots | New interactive sheet

```
df.shape
```

```
(1000, 8)
```

Data Checks to perform

Check Missing values,

Check Duplicates,

Check data type,

Check the number of unique values of each column,

Check statistics of data set,

Check various categories present in the different categorical column.

```
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
```

```
df.duplicated().sum()
```

```
0
```

```
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
```

```
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.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

Insights or Observation

From the above description of numerical data,all means are very close to each other between 66 and 69 All the standard deviation are also close between 14.6- 15.19 While there is a minimum of 0 for maths,other are having 17 and 10 value

```
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

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

```
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

```
[feature for feature in df.columns if df[feature].dtype=='O']
```

```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']
```

```
df.columns
```

```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

```
#segrregate numerical and categorical features
num_features = [feature for feature in df.columns if df[feature].dtype!='O']
cat_features = [feature for feature in df.columns if df[feature].dtype=='O']
```

```
num_features
```

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

```
cat_features
```

```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']
```

```
df['gender'].unique()
```

```
array(['female', 'male'], dtype=object)
```

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

gender	count
female	518
male	482

dtype: int64

```
df['race/ethnicity'].value_counts()
```

race/ethnicity	count
group C	319
group D	262
group B	190
group E	140
group A	89

dtype: int64

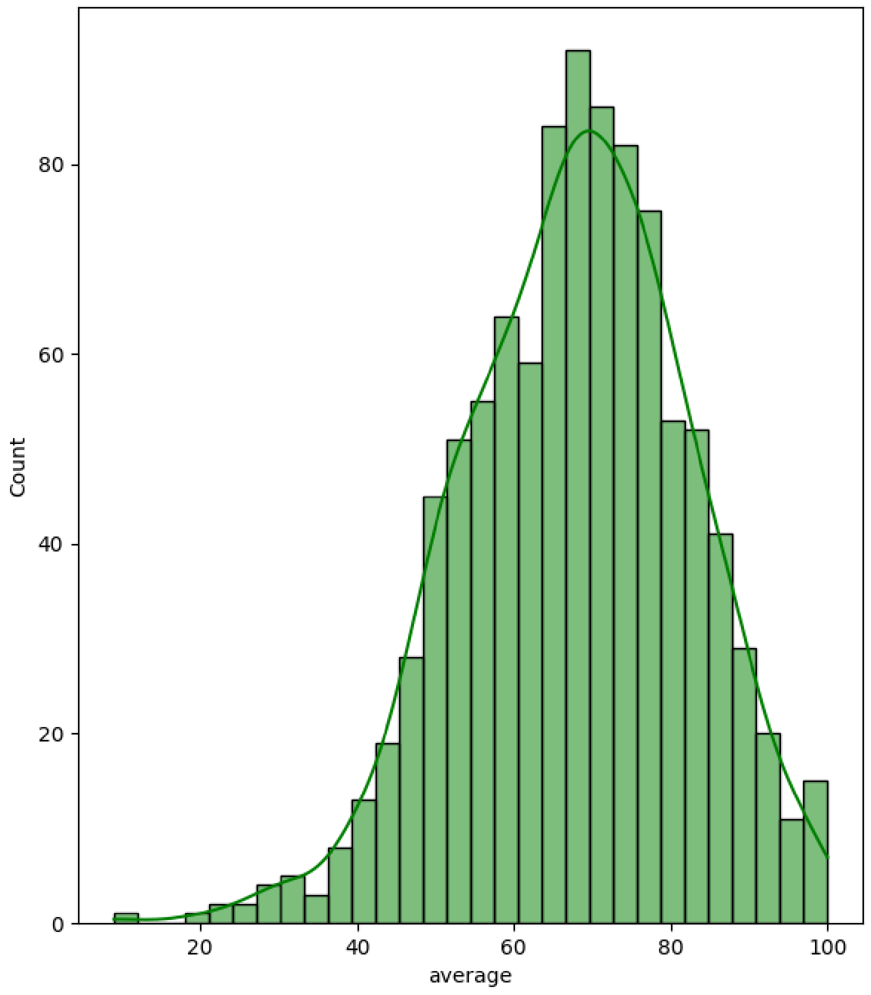
```
df['total_score'] = (df['math score'] + df['reading score'] + df['writing score'])
df['average'] = df['total_score'] /3
df.head()
```

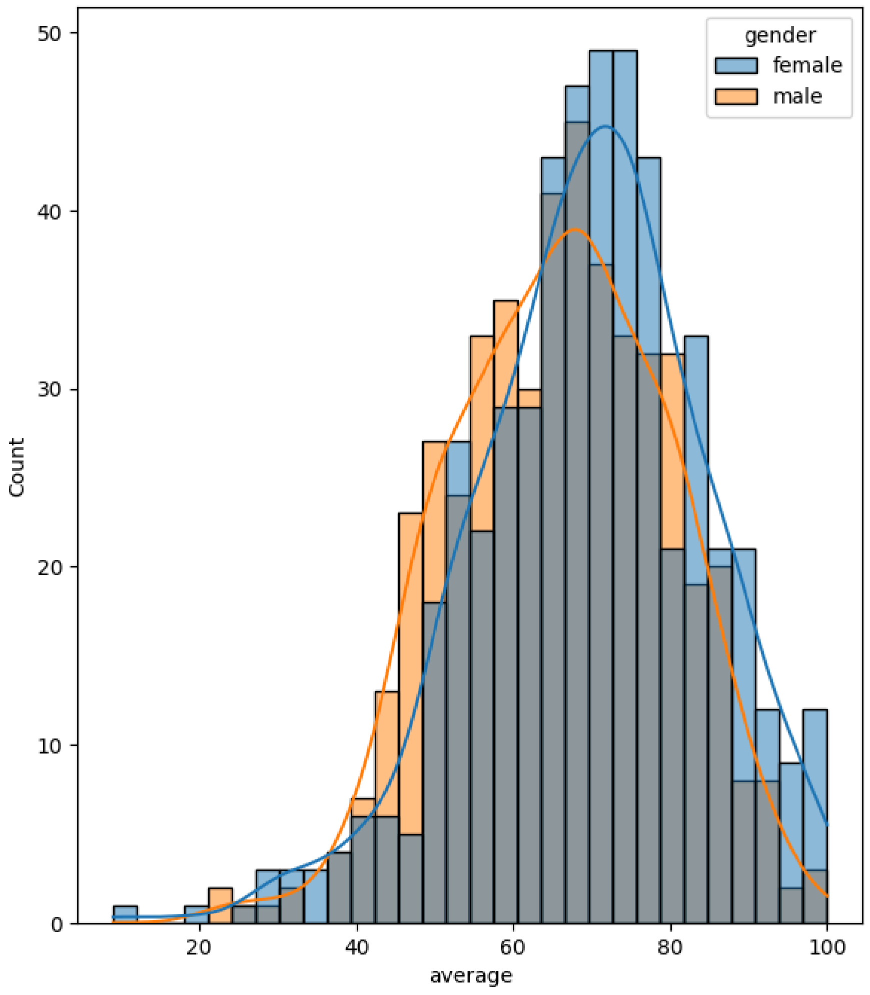
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total_score	average
0	female	group B	bachelor's degree	standard	none	72	72	74	218	72.666667
1	female	group C	some college	standard	completed	69	90	88	247	82.333333
2	female	group B	master's 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
4	male	group C	some	standard	none	76	78	75	229	76.333333

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

```
## Explore More Visulaization
fig,axis = plt.subplots(1,2,figsize=(15,8))
plt.subplot(121)
sns.histplot(data=df , x='average',bins=30 , kde=True,color='g')
plt.subplot(122)
sns.histplot(data=df, x='average',bins=30 , kde=True,hue='gender')
```

```
<Axes: xlabel='average', ylabel='Count'>
```



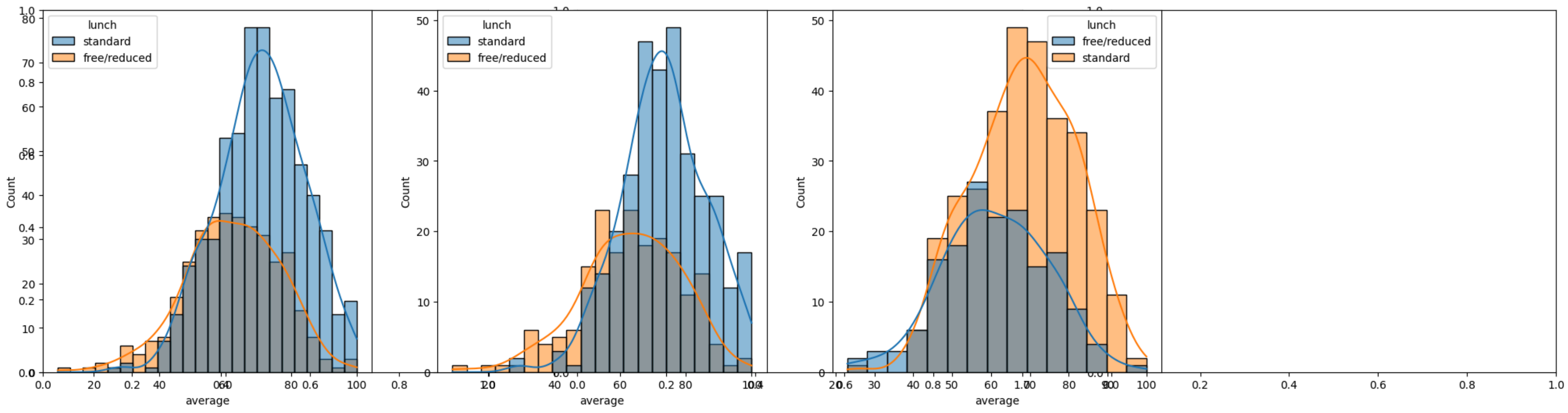


Insights

.Females students have performed well than male students

```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True , hue='lunch')
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'], x='average', kde=True, hue='lunch')
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'], x='average', kde=True, hue='lunch')
```

```
<Axes: xlabel='average', ylabel='Count'>
```



Insights

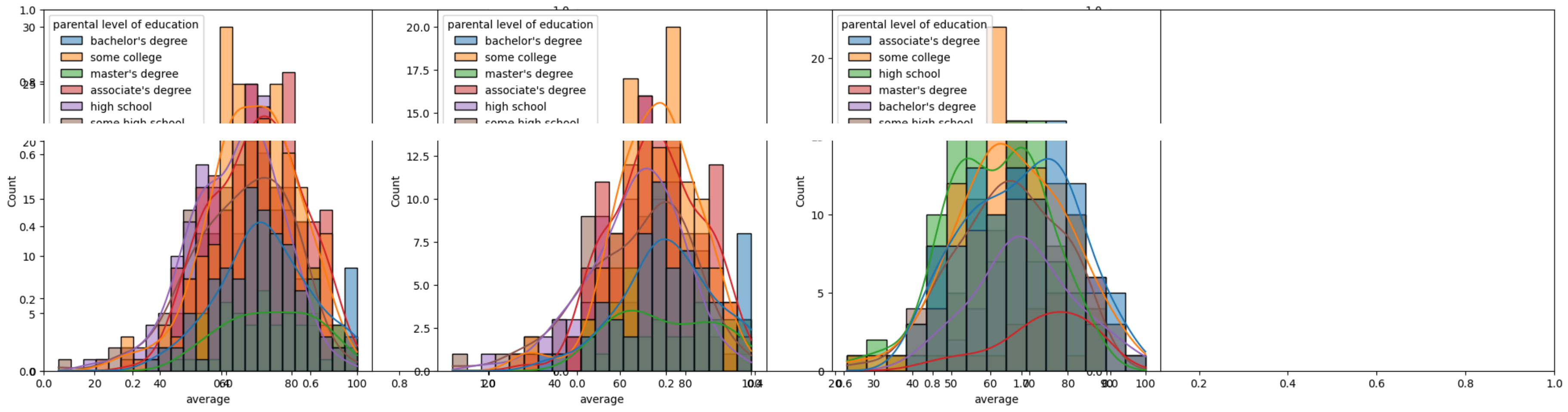
Standard Lunch help students perform well in exams

Standard Lunch helps perform well in exams both males and females

```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='parental level of education')
plt.subplot(142)
```

```
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='parental level of education'
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental level of education')
```

```
<Axes: xlabel='average', ylabel='Count'>
```



```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
ax =sns.histplot(data=df,x='average',kde=True,hue='race/ethnicity')
plt.subplot(142)
ax =sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='race/ethnicity')
plt.subplot(143)
ax =sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='race/ethnicity')
plt.show()
```

1.0
race/ethnicity
race/ethnicity
1.0
25
1.0
race/ethnicity