

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

In [2]: df = pd.read_csv('StudentsPerformance.csv')

In [3]: df.head()

Out[3]:
```

	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

```


In [4]: df.shape

Out[4]: (1000, 8)

In [5]: df.isnull().sum()

Out[5]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
gender	0	2	0	0	0	0	0	0
race/ethnicity	0	0	0	0	0	0	0	0
parental level of education	0	0	0	0	0	0	0	0
lunch	0	0	0	0	0	0	0	0
test preparation course	0	0	0	0	0	0	0	0
math score	0	0	0	0	0	0	0	0
reading score	0	0	0	0	0	0	0	0
writing score	0	0	0	0	0	0	0	0
dtype:	int64(3)	object(5)	int64(3)	object(3)	object(5)	int64(3)	int64(3)	int64(3)

```


In [6]: df.duplicated().sum()

Out[6]: np.int64(0)

In [7]: 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

In [8]: df.nunique()

Out[8]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
gender	2	5	6	2	2	81	72	77
race/ethnicity	5	6	2	2	2	81	72	77
parental level of education	6	2	2	2	2	81	72	77
lunch	2	2	2	2	2	81	72	77
test preparation course	2	2	2	2	2	81	72	77
math score	81	72	77	81	72	77	81	72
reading score	72	77	81	72	77	81	72	77
writing score	77	81	72	77	81	72	77	77
dtype:	int64	int64	int64	int64	int64	int64	int64	int64

```


In [9]: df.describe()

Out[9]:
```

	math score	reading score	writing score
count	1000.000000	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

```


In [10]: df.head()

Out[10]:
```

	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

```


In [11]: num_cols=[features for features in df.columns if df[features].dtype != "object"]
print("these are all the numerical columns:",num_cols)

these are all the numerical columns: ['math score', 'reading score']

In [12]: cat_cols=[features for features in df.columns if df[features].dtype == "object"]
print("these are all the categorical columns:",cat_cols)

these are all the categorical columns: ['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test preparation course']

In [13]: df["Total Score"] = df["math score"] + df["reading score"] + df["writing score"]
df["Average Score"] = df["Total Score"] / 3

In [14]: df


Out[14]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Total Score	Average Score
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 college	standard	none	76	78	75	229	76.333333
...	...	...	...	...	...	...	...	...	...	...
993	female	group E	master's degree	standard	completed	88	99	95	282	94.000000
996	male	group C	high school	free/reduced	none	62	55	55	172	57.333333
997	female	group C	high school	free/reduced	completed	59	71	65	195	65.000000
998	female	group D	some college	standard	completed	68	78	77	223	74.333333
999	female	group D	some college	free/reduced	none	77	86	86	249	83.000000

```
1000 rows * 10 columns

In [15]: fig,axs=plt.subplots(1,2,figsize=(15,7))
plt.subplot(121)
sns.histplot(data=df,x="Average Score",bins=30,kde=True,color='g',hue="gender")
plt.subplot(122)
sns.histplot(data=df,x="Average Score",bins=30,kde=True,color='g',)

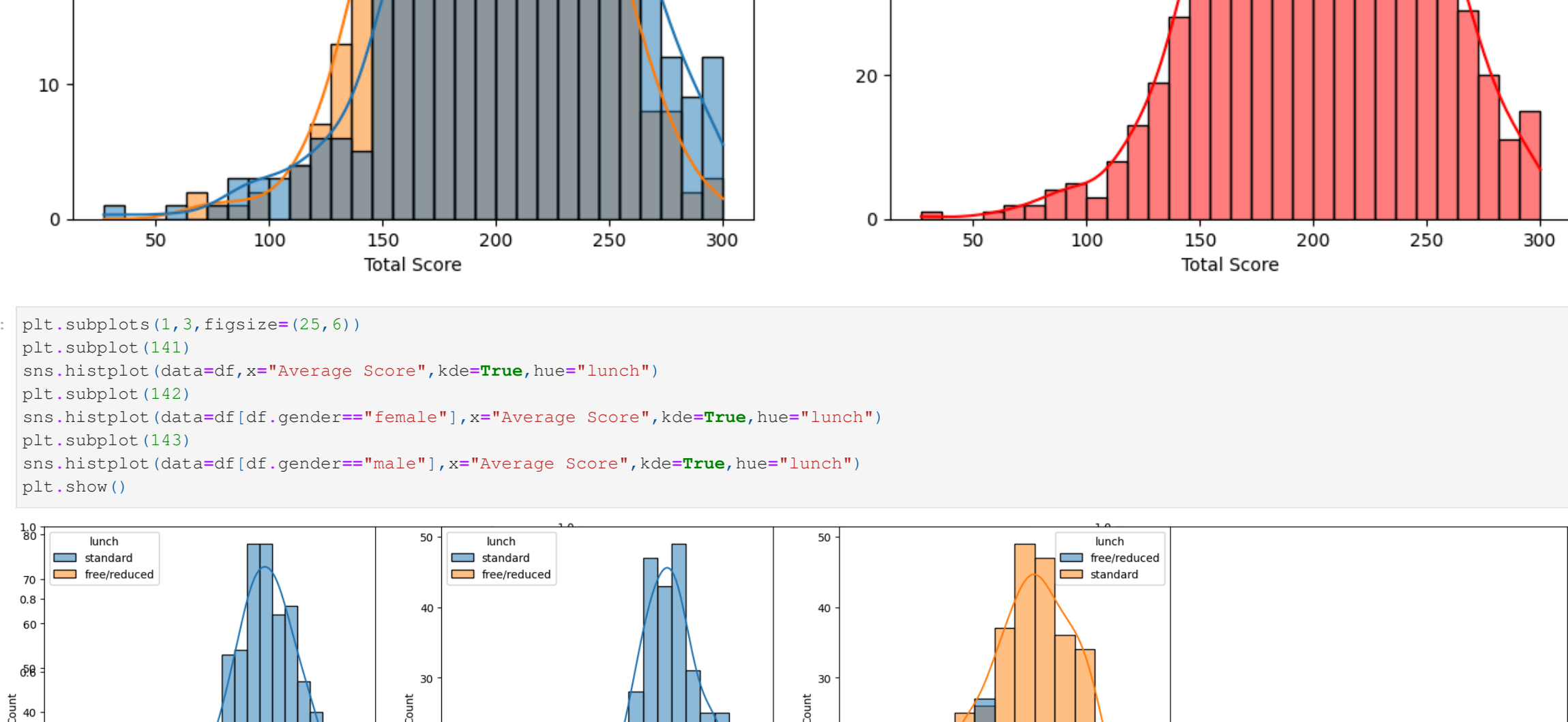
Out[15]: <Axes: xlabel='Average Score', ylabel='Count'>
```



```

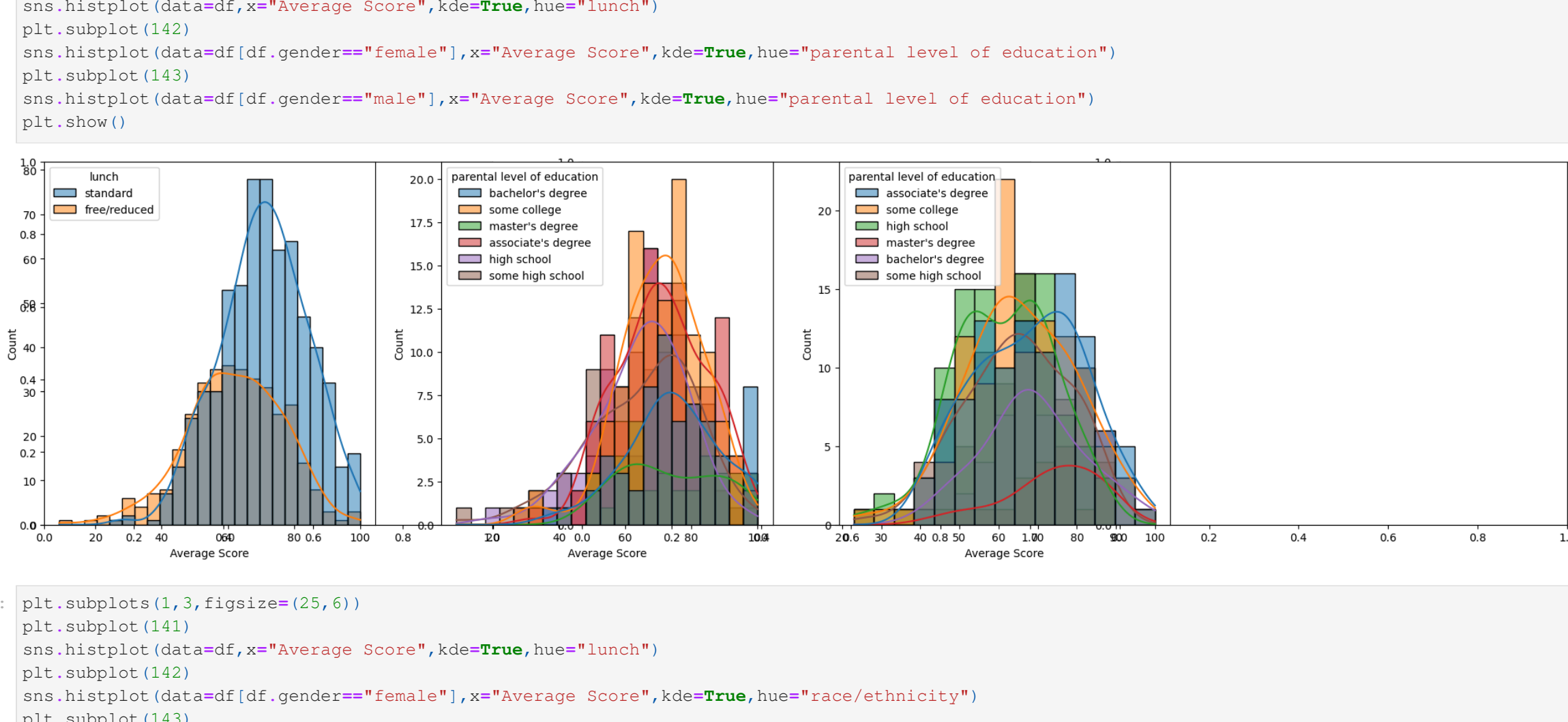

In [16]: fig,axs=plt.subplots(1,2,figsize=(15,7))
plt.subplot(121)
sns.histplot(data=df,x="Total Score",bins=30,kde=True,color='r',hue="gender")
plt.subplot(122)
sns.histplot(data=df,x="Total Score",bins=30,kde=True,color='r',)

Out[16]: <Axes: xlabel='Total Score', ylabel='Count'>
```



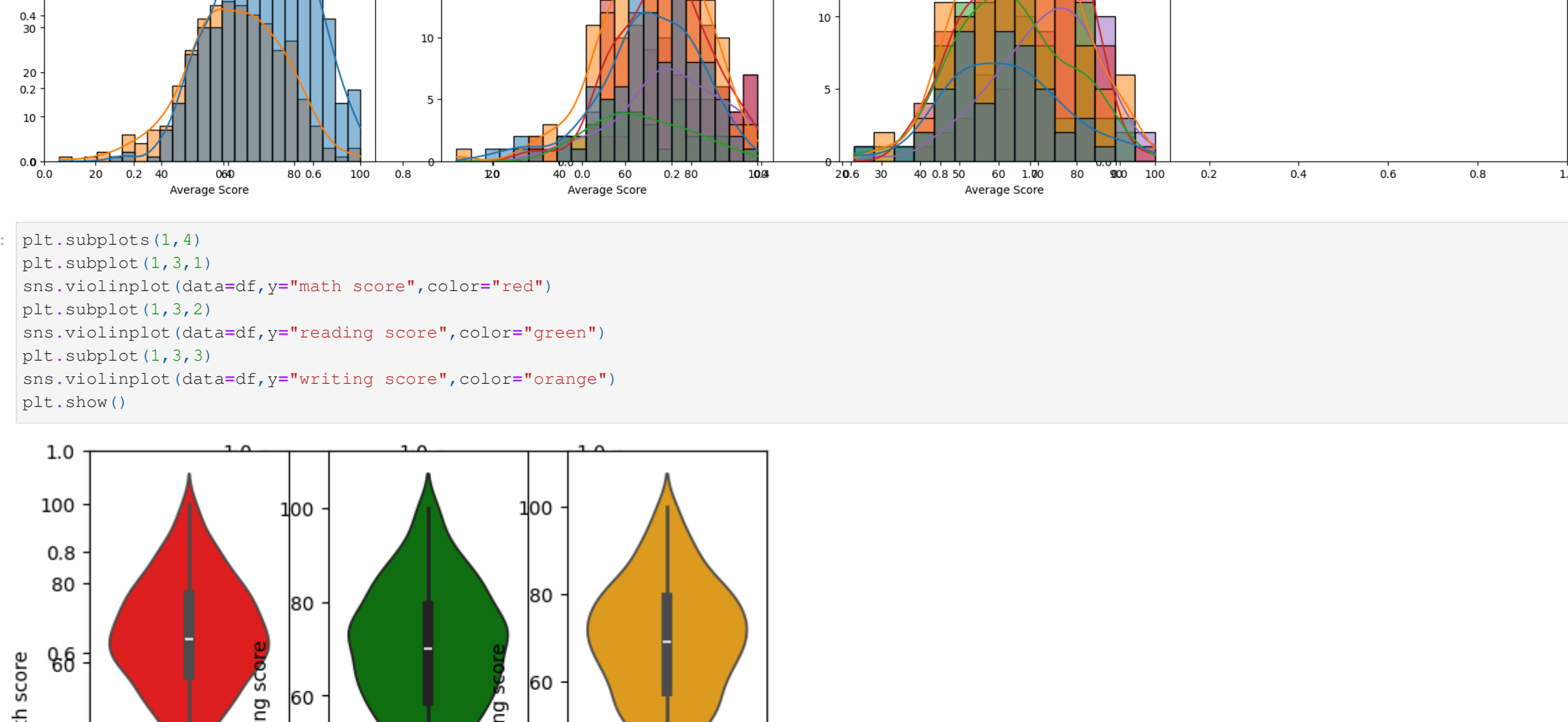
```


In [17]: plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
sns.histplot(data=df,x="Average Score",kde=True,hue="lunch")
plt.subplot(132)
sns.histplot(data=df[df.gender=="female"],x="Average Score",kde=True,hue="lunch")
plt.subplot(133)
sns.histplot(data=df[df.gender=="male"],x="Average Score",kde=True,hue="lunch")
plt.show()
```



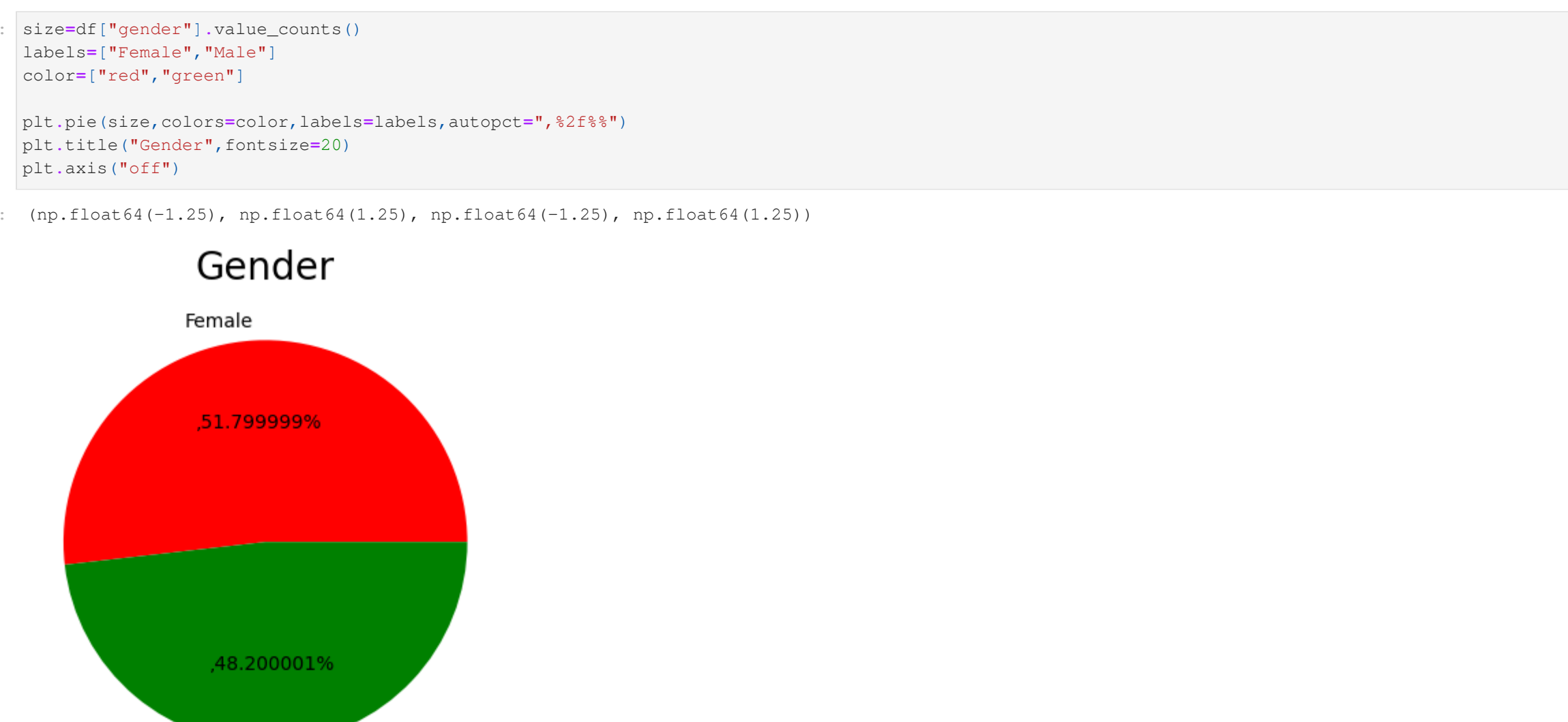
```


In [18]: plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
sns.histplot(data=df,x="Average Score",kde=True,hue="lunch")
plt.subplot(132)
sns.histplot(data=df[df.gender=="female"],x="Average Score",kde=True,hue="parental level of education")
plt.subplot(133)
sns.histplot(data=df[df.gender=="male"],x="Average Score",kde=True,hue="parental level of education")
plt.show()
```



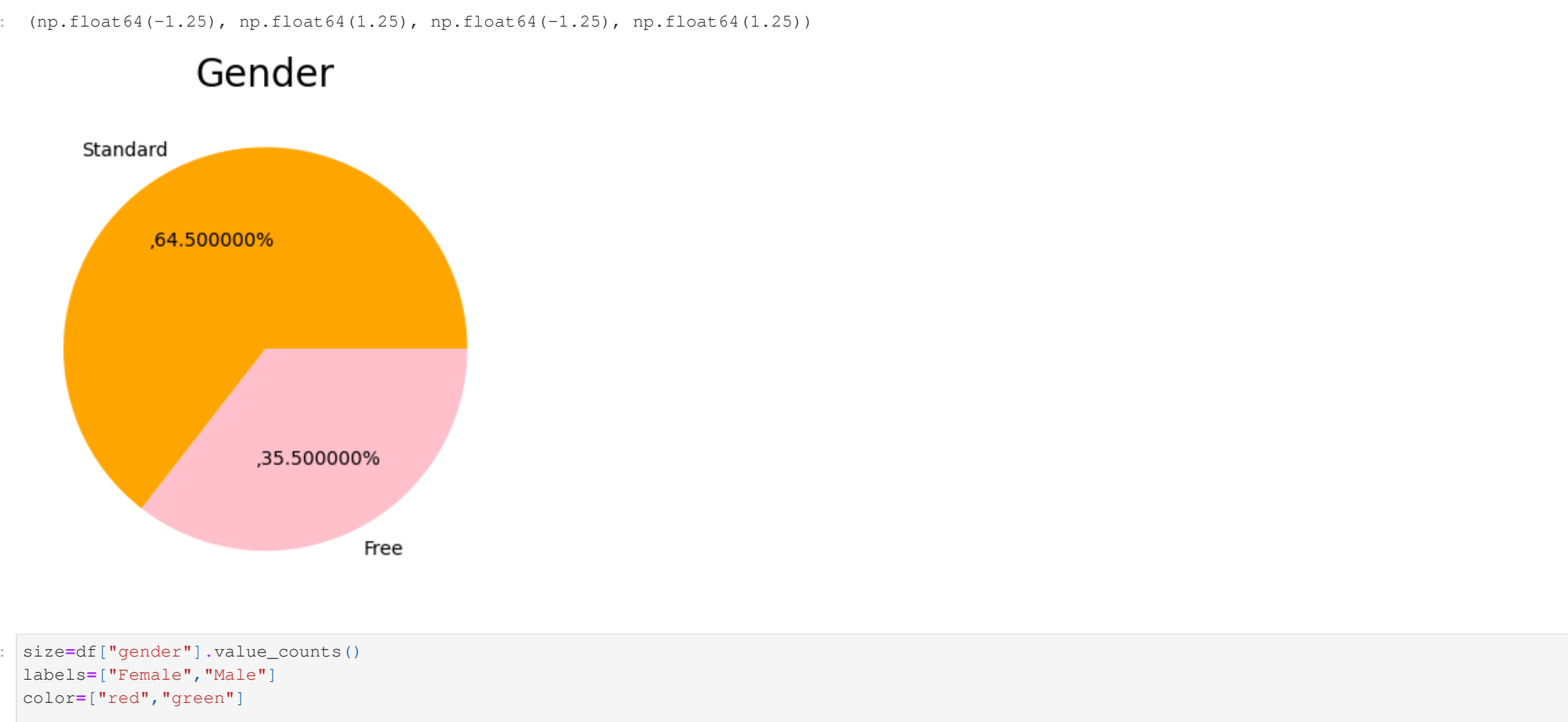
```


In [19]: plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
sns.histplot(data=df,x="Average Score",kde=True,hue="lunch")
plt.subplot(132)
sns.histplot(data=df[df.gender=="female"],x="Average Score",kde=True,hue="race/ethnicity")
plt.subplot(133)
sns.histplot(data=df[df.gender=="male"],x="Average Score",kde=True,hue="race/ethnicity")
plt.show()
```



```


In [20]: plt.subplots(1,4,figsize=(25,6))
plt.subplot(141)
sns.violinplot(data=df,y="math score",color="red")
plt.subplot(142)
sns.violinplot(data=df,y="reading score",color="green")
plt.subplot(143)
sns.violinplot(data=df,y="writing score",color="orange")
plt.show()
```

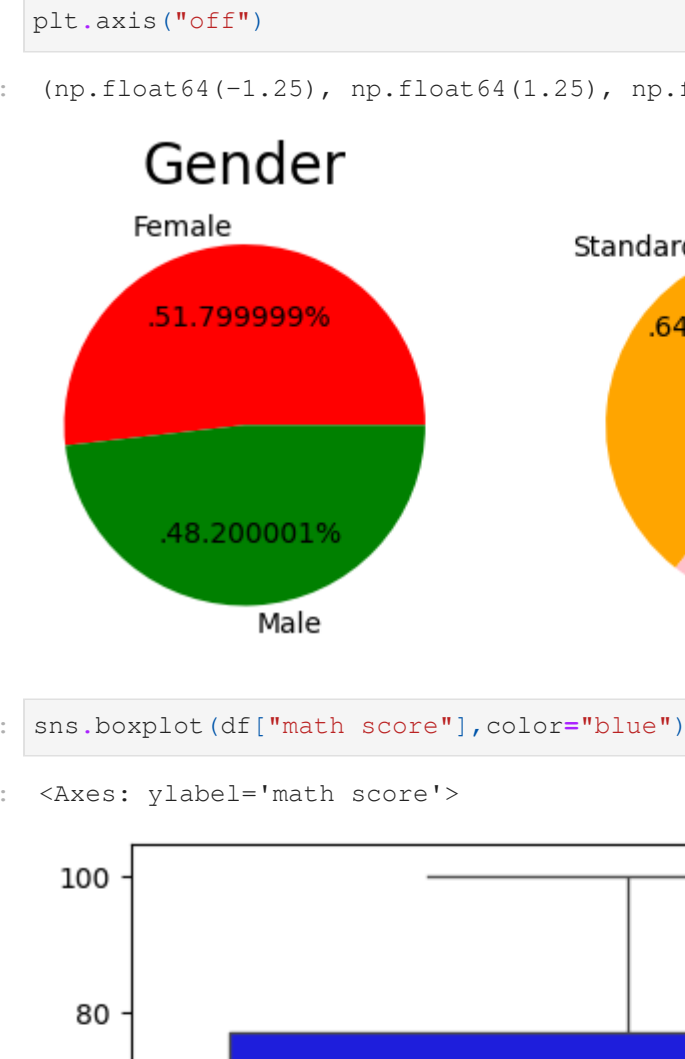


```


In [21]: size=df["gender"].value_counts()
labels=["Female","Male"]
color=["red","green"]

plt.pie(size,color=color,labels=labels,autopct='%2E%%')
plt.title("Gender",fontsize=20)
plt.axis("off")

Out[21]: (np.float64(-1.25), np.float64(1.25), np.float64(-1.25), np.float64(1.25))
```

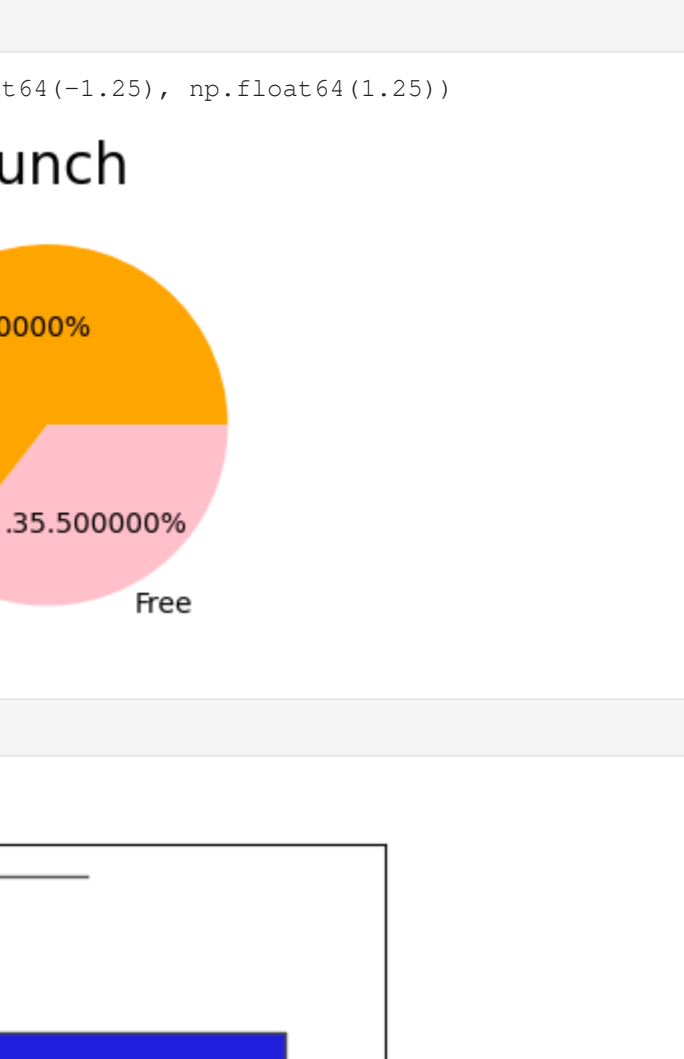


```


In [22]: size=df["lunch"].value_counts()
labels=["Standard","Free"]
color=["orange","pink"]

plt.pie(size,color=color,labels=labels,autopct='%2E%%')
plt.title("Lunch",fontsize=20)
plt.axis("off")

Out[22]: (np.float64(-1.25), np.float64(1.25), np.float64(-1.25), np.float64(1.25))
```



```



In [23]: size=df["gender"].value_counts()
labels=["Female","Male"]
color=["red","green"]

plt.subplot(1,2,1)
plt.pie(size,color=color,labels=labels,autopct='%2E%%')
plt.title("Gender",fontsize=20)
plt.axis("off")

plt.subplot(1,2,2)
size=df["lunch"].value_counts()
labels=["Standard","Free"]
color=["orange","pink"]

plt.pie(size,color=color,labels=labels,autopct='%2E%%')
plt.title("Lunch",fontsize=20)
plt.axis("off")

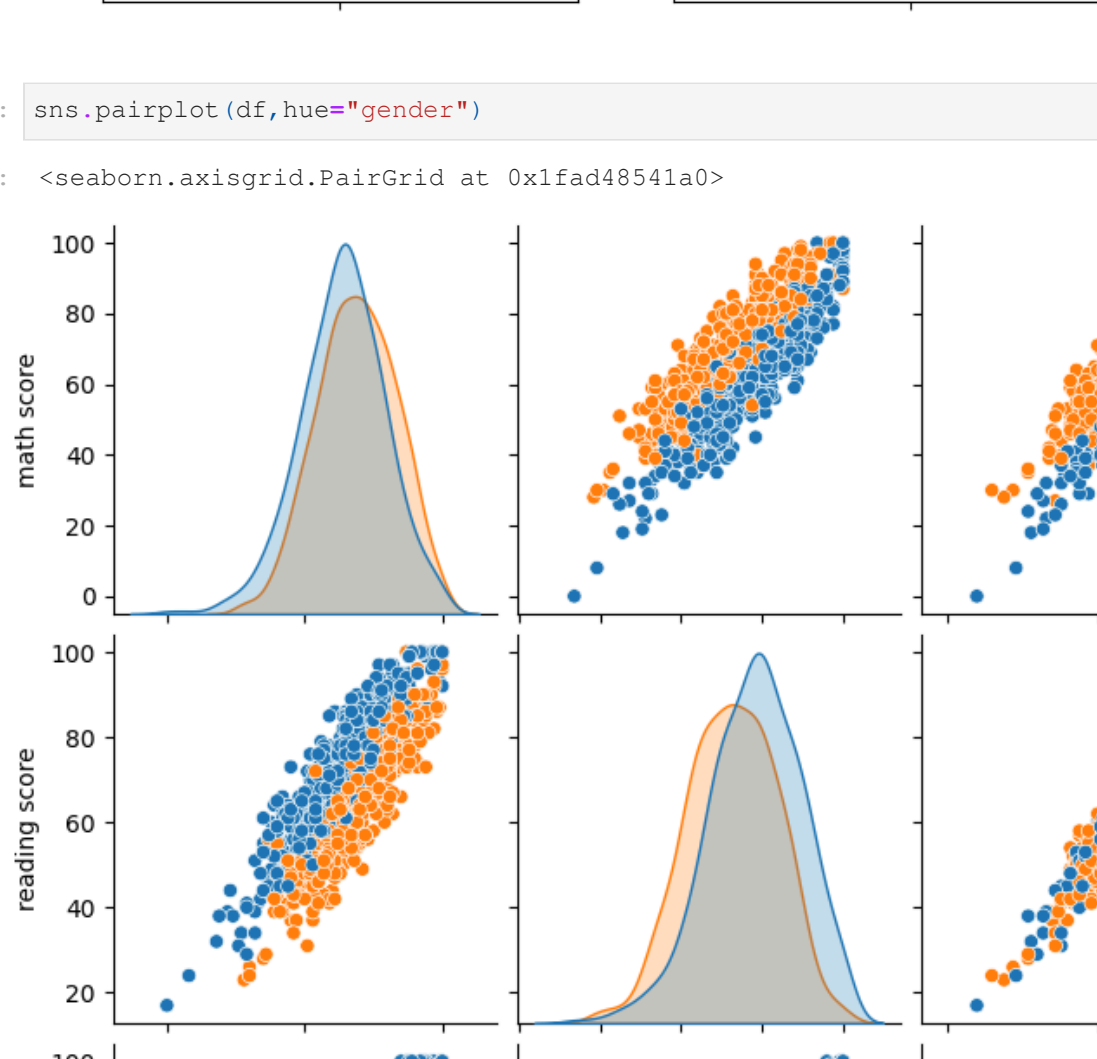
Out[23]: (np.float64(-1.25), np.float64(1.25), np.float64(-1.25), np.float64(1.25))
```



```


In [24]: sns.boxplot(df["math score"],color="blue")

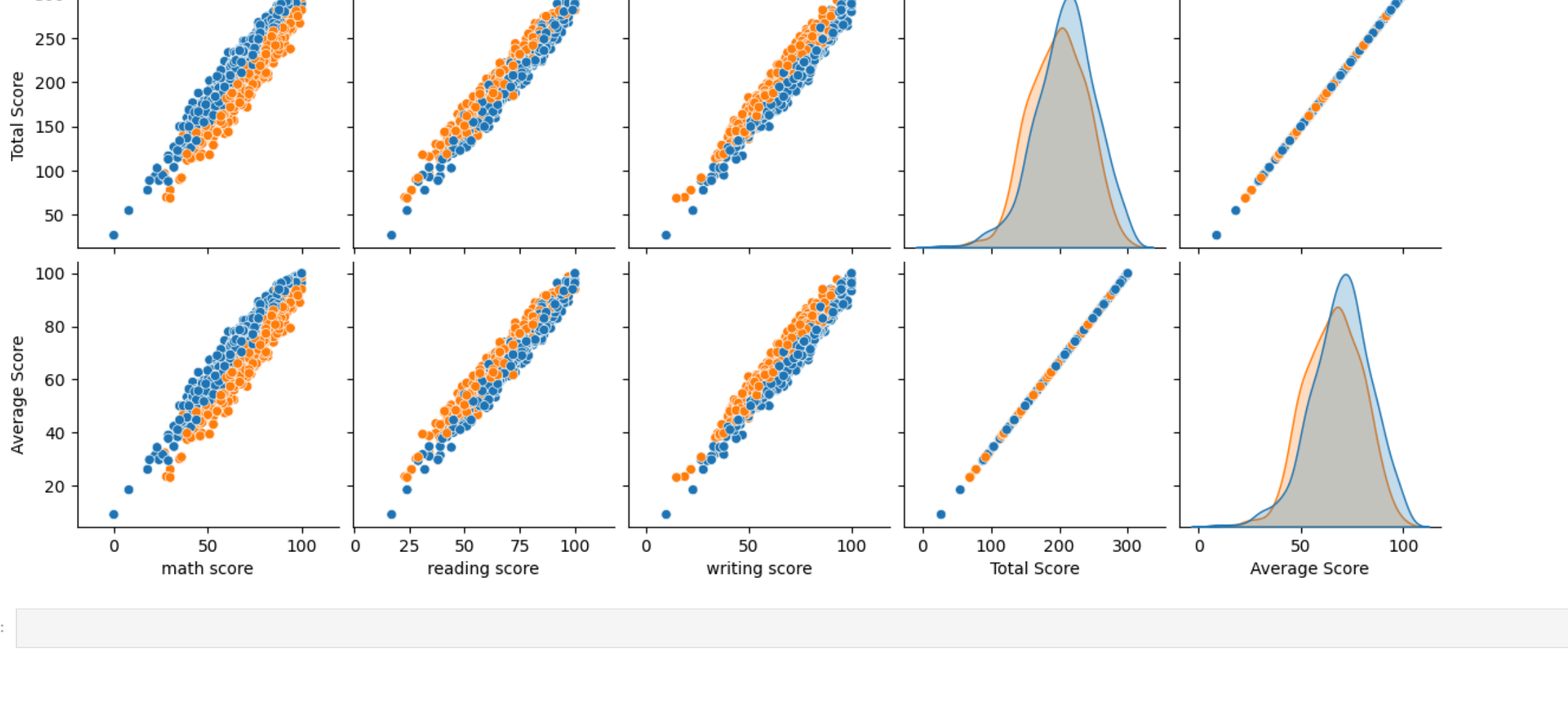
Out[24]: <Axes: ylabel='math score'>
```



```


In [25]: plt.subplots(1,4,figsize=(18,8))
plt.subplot(141)
sns.boxplot(df["math score"],color="skyblue")
plt.subplot(142)
sns.boxplot(df["reading score"],color="hotpink")
plt.subplot(143)
sns.boxplot(df["writing score"],color="yellow")
plt.show()

Out[25]: <Axes: ylabel='writing score'>
```



```


In [26]: sns.pairplot(df,hue="gender")

Out[26]: <seaborn.axisgrid.PairGrid at 0x1ad48541a0>
```

