

```
In [28]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
matplotlib inline
```

```
In [29]: df = pd.read_csv('Students.csv')
```

```
In [30]: 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

```
In [31]: df.shape
```

```
Out [31]: (1090, 8)
```

```
In [32]: 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

```
In [33]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1090 entries, 0 to 999
Data columns (total 8 columns):
#   Column              Non-Null Count  Dtype
--  --
0   gender              1090 non-null  object
1   race/ethnicity      1090 non-null  object
2   parental level of education 1090 non-null  object
3   lunch               1090 non-null  object
4   test preparation course 1090 non-null  object
5   math score          1090 non-null  int64
6   reading score       1090 non-null  int64
7   writing score        1090 non-null  int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

Statistics Analysis

```
In [34]: df.describe()
```

	math score	reading score	writing score
count	1090.00000	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 Cleaning

```
In [35]: df.nunique()
```

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

```
In [36]: df['race/ethnicity'].unique()
```

```
Out [36]: array(['group B', 'group C', 'group A', 'group D', 'group E'],
      dtype=object)
```

```
In [37]: df['parental level of education'].unique()
```

```
Out [37]: array(['bachelor's degree', 'some college', 'master's degree',
      'associate's degree', 'high school', 'some high school'],
      dtype=object)
```

```
In [38]: df['test preparation course'].value_counts()
```

none	642
completed	358
Name: test preparation course, dtype: int64	

```
In [39]: df['race/ethnicity'].value_counts()
```

group C	319
group D	262
group B	190
group E	148
group A	89
Name: race/ethnicity, dtype: int64	

```
In [40]: student_df = df.drop(columns=['race/ethnicity','parental level of education','test preparation course'],axis=1)
```

```
In [41]: student_df.head()
```

	gender	lunch	math score	reading score	writing score
0	female	standard	72	72	74
1	female	standard	69	90	88
2	female	standard	90	95	93
3	male	free/reduced	47	57	44
4	male	standard	76	78	75

Looking missing value

```
In [42]: df.isnull().sum()
```

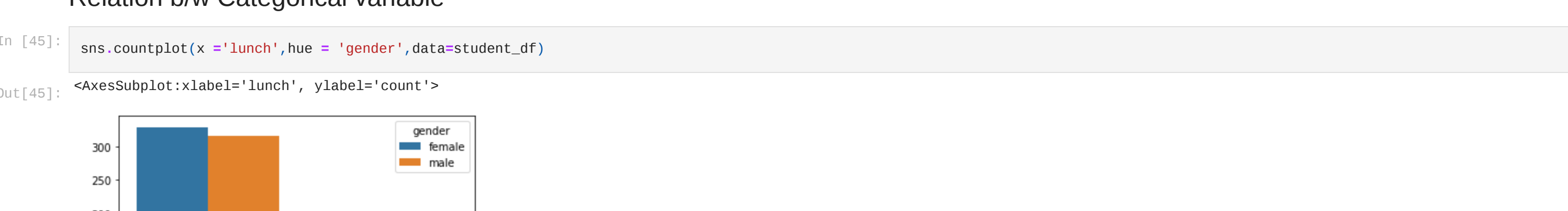
gender	0
race/ethnicity	0
parental level of education	0
lunch	0
test preparation course	0
math score	0
reading score	0
writing score	0
dtype:	int64

correlation

```
In [43]: student_df.corr()
```

	math score	reading score	writing score
math score	1.000000	0.817580	0.802642
reading score	0.817580	1.000000	0.954598
writing score	0.802642	0.954598	1.000000

```
In [44]: sns.heatmap(student_df.corr(),annot=True)
```



Relation b/w Categorical variable

```
In [45]: sns.countplot(x='lunch',hue='gender',data=student_df)
```



Numerical data analysis

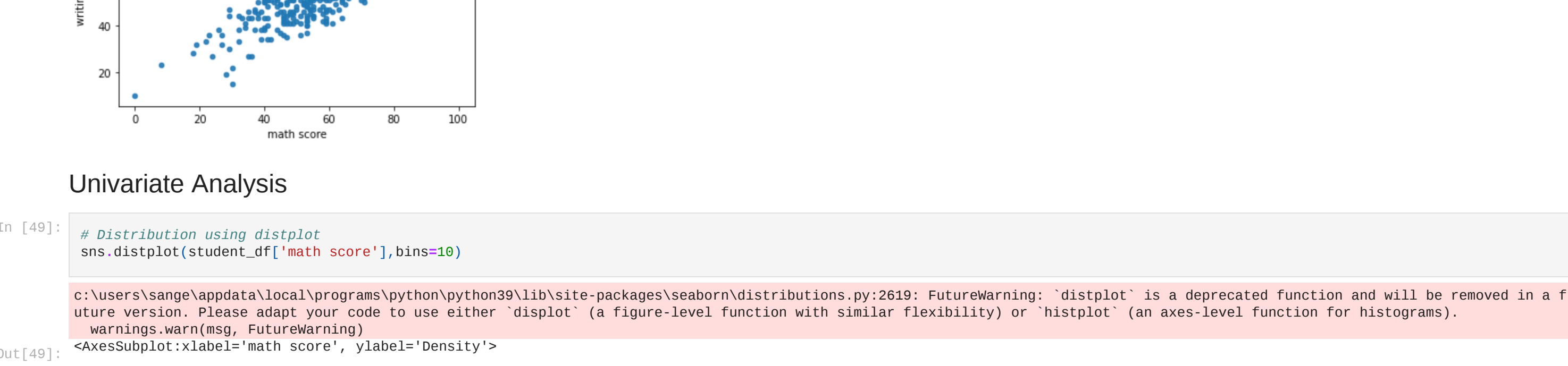
```
In [46]: # Relation b/w two variable
sns.relplot(x='math score',y='reading score',hue='gender',data=student_df)

Out [46]: <seaborn.axisgrid.FacetGrid at 0x2847b96940>
```

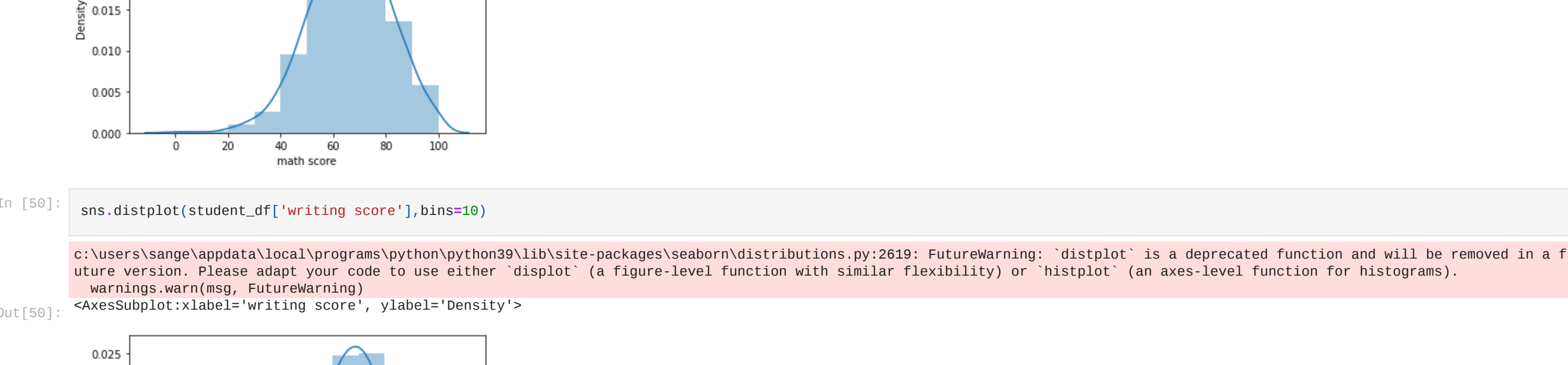


```
In [47]: sns.relplot(x='math score',y='reading score',hue='lunch',data=student_df)

Out [47]: <seaborn.axisgrid.FacetGrid at 0x2847bf18cd9>
```



```
In [48]: student_df.plot(kind='scatter',x='math score',y='writing score',alpha=0.9)
```

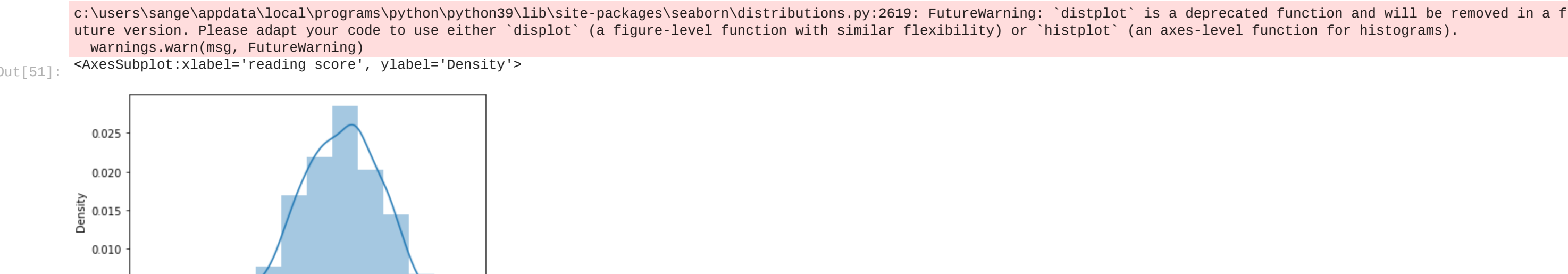


Univariate Analysis

```
In [49]: # Distribution using distplot
sns.distplot(student_df['math score'],bins=10)

C:\Users\ange\appdata\local\programs\python\python39\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

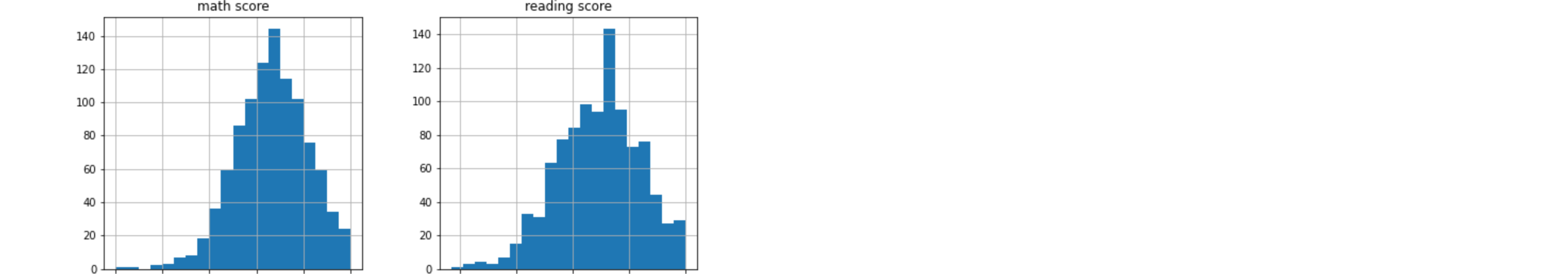
Out [49]: <AxesSubplot:xlabel='math score', ylabel='Density'>
```



```
In [50]: sns.distplot(student_df['writing score'],bins=10)

C:\Users\ange\appdata\local\programs\python\python39\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

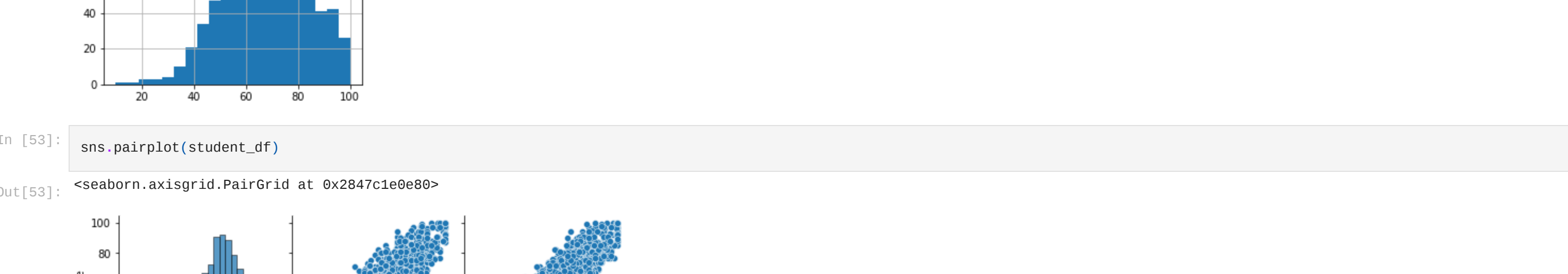
Out [50]: <AxesSubplot:xlabel='writing score', ylabel='Density'>
```



```
In [51]: sns.distplot(student_df['reading score'],bins=10)

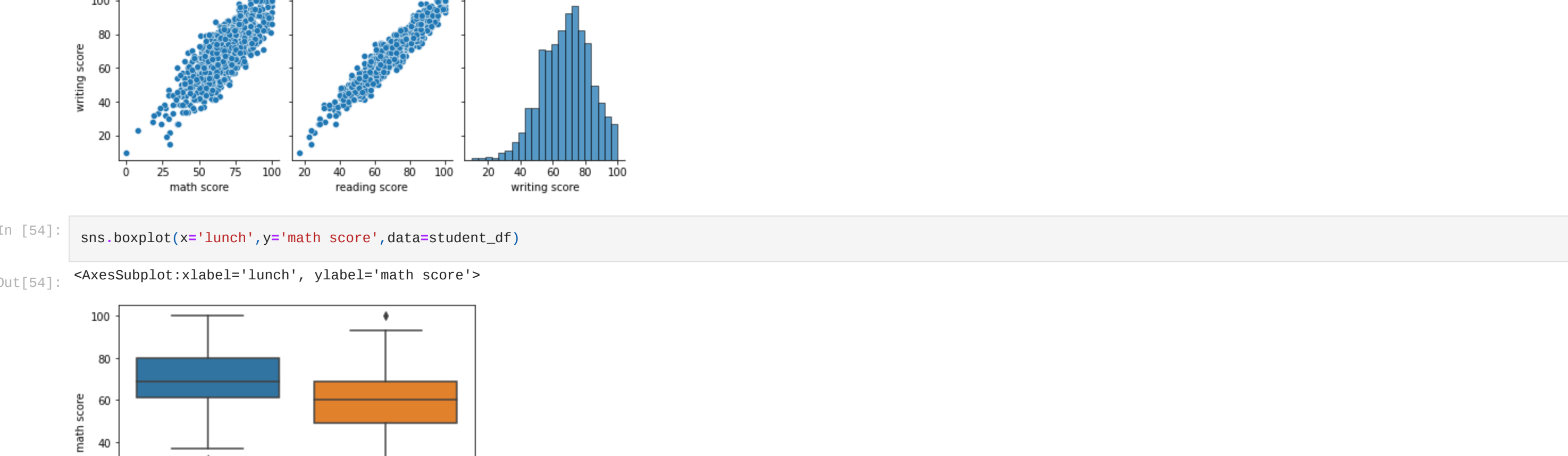
C:\Users\ange\appdata\local\programs\python\python39\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

Out [51]: <AxesSubplot:xlabel='reading score', ylabel='Density'>
```



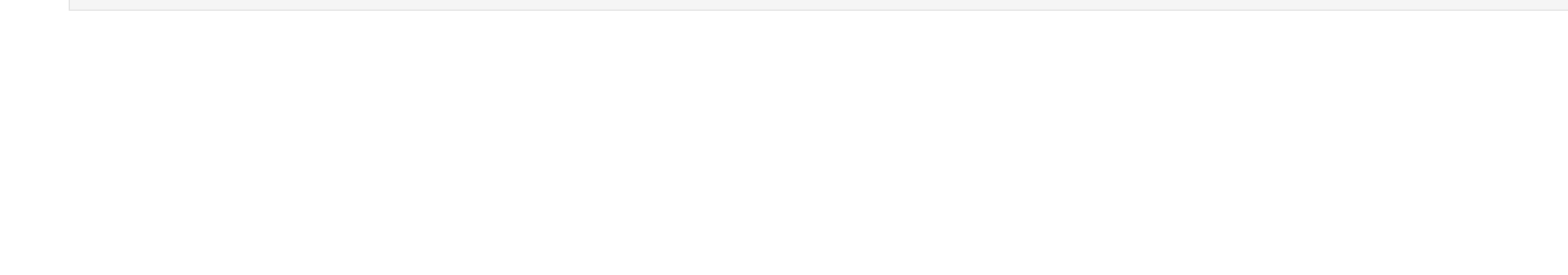
```
In [52]: student_df.hist(bins=20,figsize=(10,10))
```

```
Out [52]: array([[<AxesSubplot:title='center': 'math score'>],
      [<AxesSubplot:title='center': 'reading score'>]],
      [<AxesSubplot:title='center': 'writing score'>]],
      dtype=object)
```



```
In [53]: sns.pairplot(student_df)
```

```
Out [53]: <seaborn.axisgrid.PairGrid at 0x2847c1e0e80>
```



```
In [54]: sns.boxplot(x='lunch',y='math score',data=student_df)
```

```
Out [54]: <AxesSubplot:xlabel='lunch', ylabel='math score'>
```

