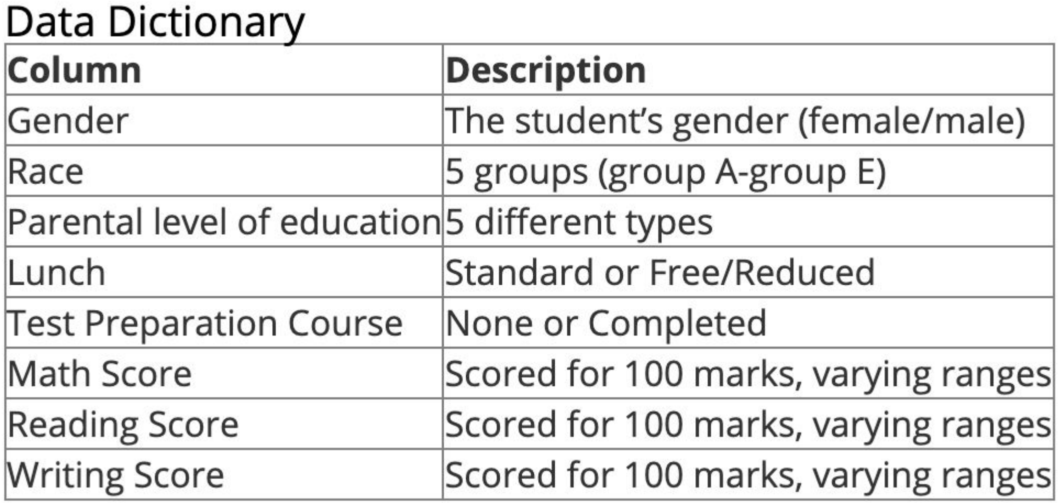
Statistics for Data Science Hackathon Assignment

Mahika Gupta
PES1UG20CS243
Student_44

About the Dataset:

The dataset consists of information about students and the marks obtained by them in 3 different subjects.

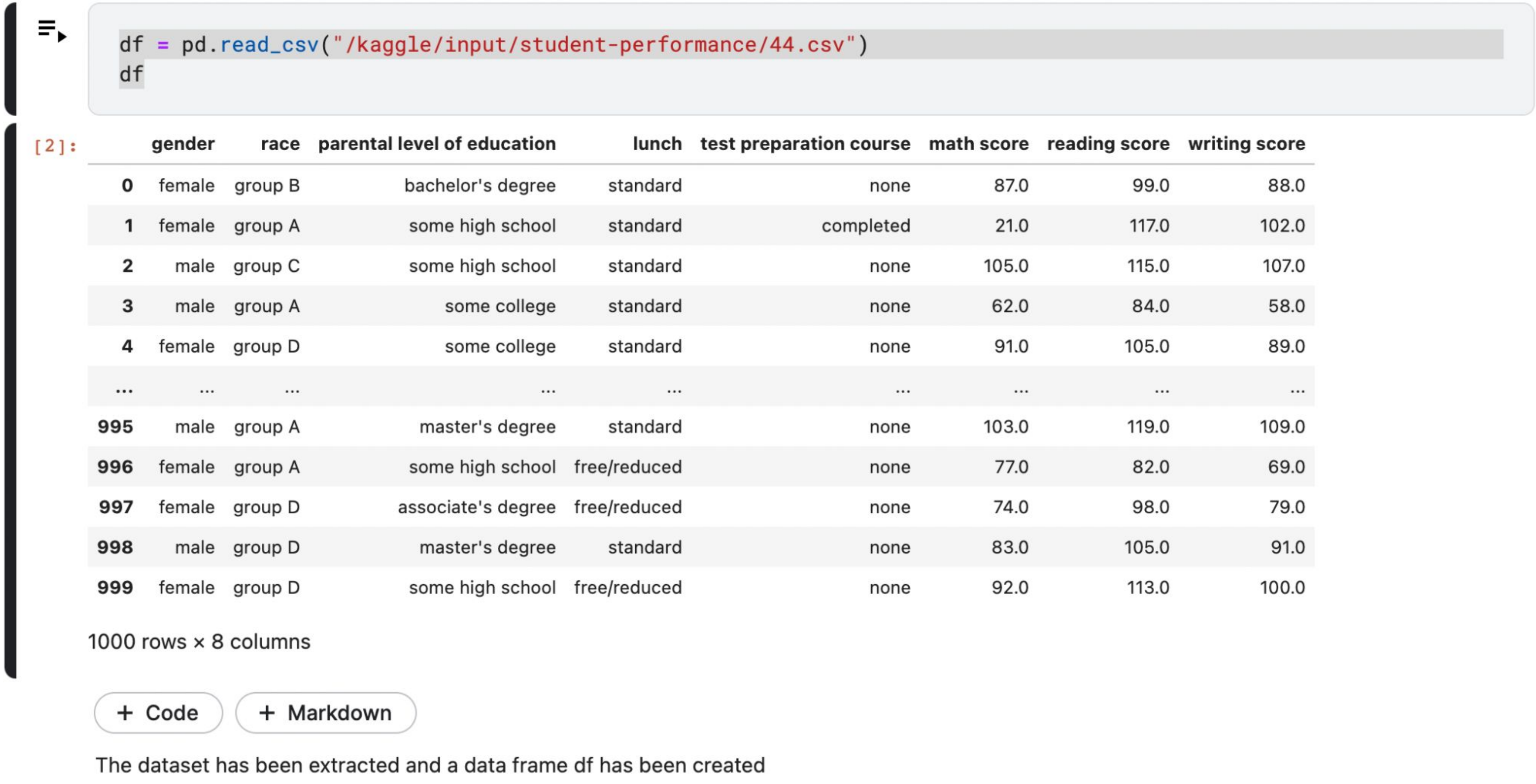


Data Dictionary

Column	Description
Gender	The student's gender (female/male)
Race	5 groups (group A-group E)
Parental level of education	5 different types
Lunch	Standard or Free/Reduced
Test Preparation Course	None or Completed
Math Score	Scored for 100 marks, varying ranges
Reading Score	Scored for 100 marks, varying ranges
Writing Score	Scored for 100 marks, varying ranges

Size: (1000,8)

Extracted dataset:

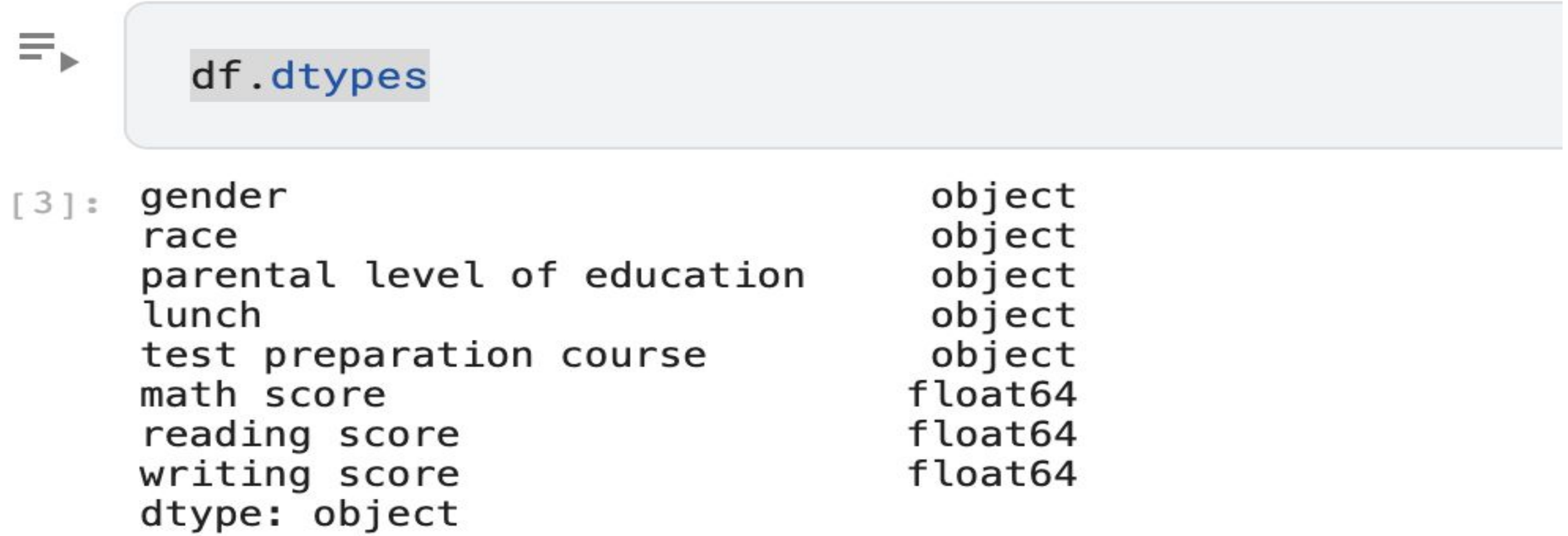


```
df = pd.read_csv("/kaggle/input/student-performance/44.csv")
df
```

	gender	race	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	87.0	99.0	88.0
1	female	group A	some high school	standard	completed	21.0	117.0	102.0
2	male	group C	some high school	standard	none	105.0	115.0	107.0
3	male	group A	some college	standard	none	62.0	84.0	58.0
4	female	group D	some college	standard	none	91.0	105.0	89.0
...	...	...	...	...	...	...	...	...
995	male	group A	master's degree	standard	none	103.0	119.0	109.0
996	female	group A	some high school	free/reduced	none	77.0	82.0	69.0
997	female	group D	associate's degree	free/reduced	none	74.0	98.0	79.0
998	male	group D	master's degree	standard	none	83.0	105.0	91.0
999	female	group D	some high school	free/reduced	none	92.0	113.0	100.0

1000 rows × 8 columns

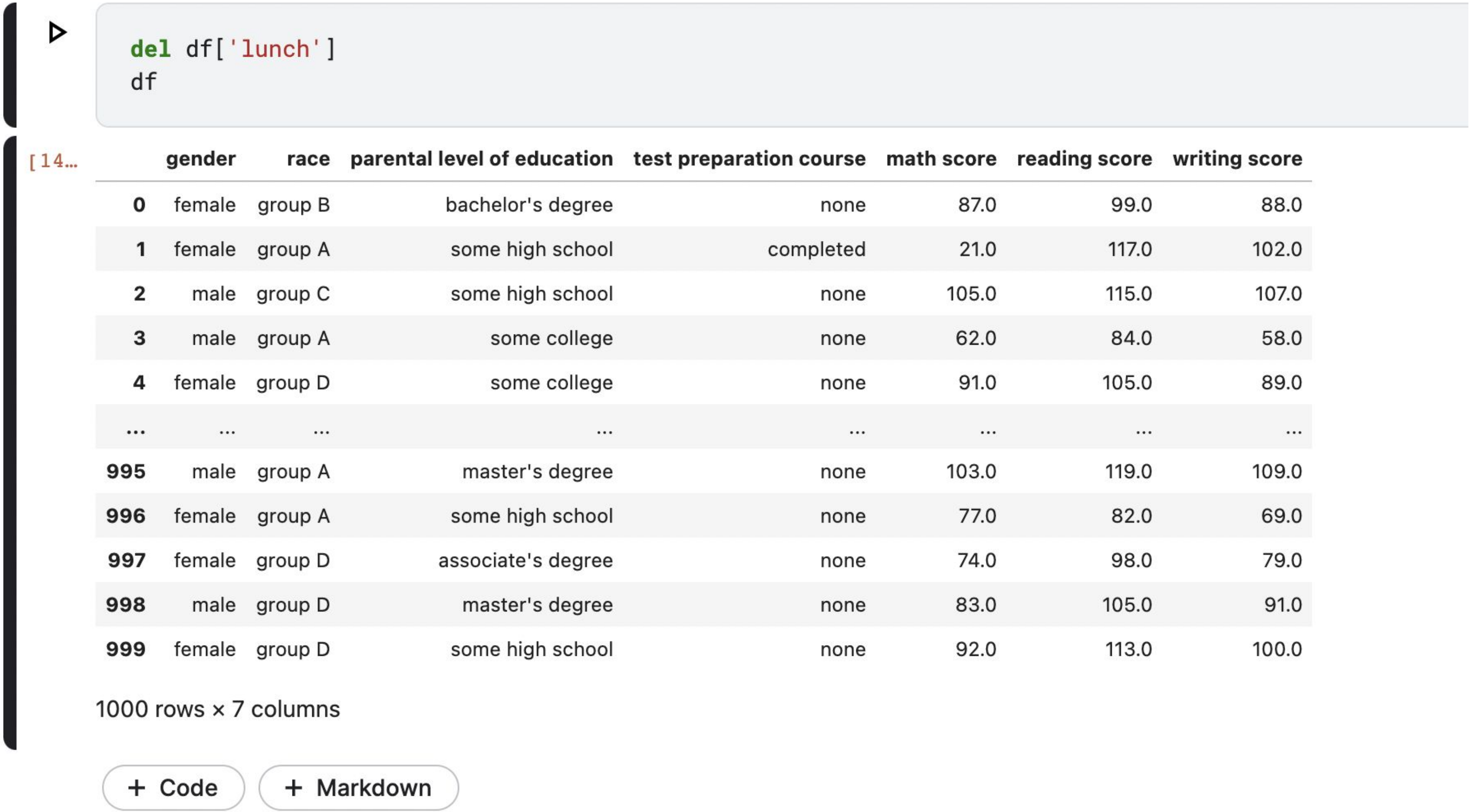
The dataset has been extracted and a data frame df has been created



```
df.dtypes
```

```
[3]: gender                             object
     race                               object
     parental level of education        object
     lunch                              object
     test preparation course            object
     math score                        float64
     reading score                     float64
     writing score                     float64
     dtype: object
```

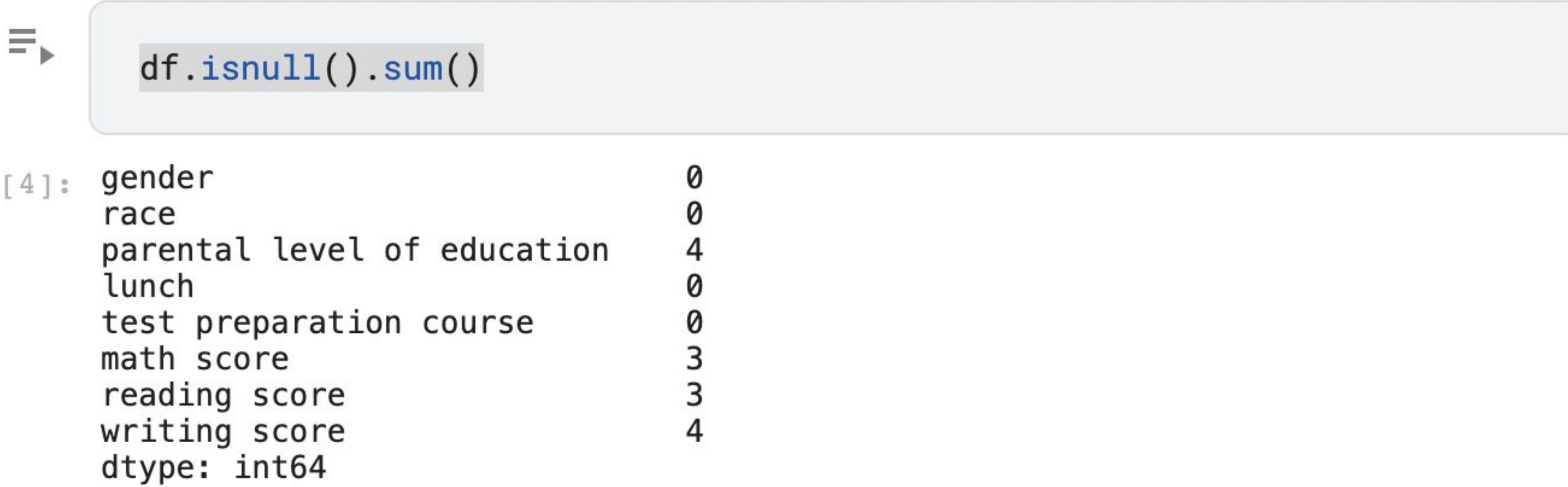
This checks the datatype of all the attributes



```python
del df['lunch']
df
```

	gender	race	parental level of education	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	none	87.0	99.0	88.0
1	female	group A	some high school	completed	21.0	117.0	102.0
2	male	group C	some high school	none	105.0	115.0	107.0
3	male	group A	some college	none	62.0	84.0	58.0
4	female	group D	some college	none	91.0	105.0	89.0
...	...	...	...	...	...	...	...
995	male	group A	master's degree	none	103.0	119.0	109.0
996	female	group A	some high school	none	77.0	82.0	69.0
997	female	group D	associate's degree	none	74.0	98.0	79.0
998	male	group D	master's degree	none	83.0	105.0	91.0
999	female	group D	some high school	none	92.0	113.0	100.0

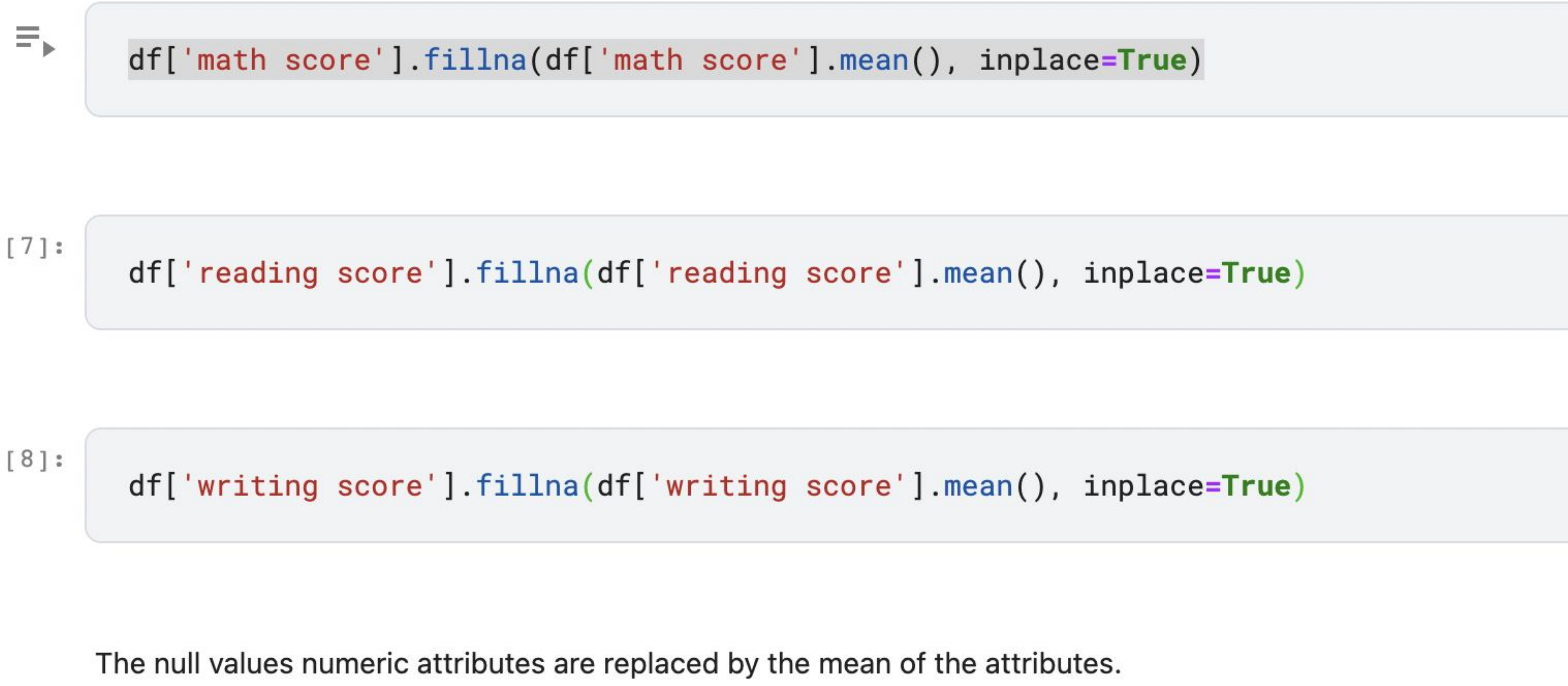
1000 rows × 7 columns



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

```
[4]: gender                         0
     race                           0
     parental level of education    4
     lunch                          0
     test preparation course        0
     math score                     3
     reading score                  3
     writing score                  4
     dtype: int64
```

To check the number of null values in the attributes

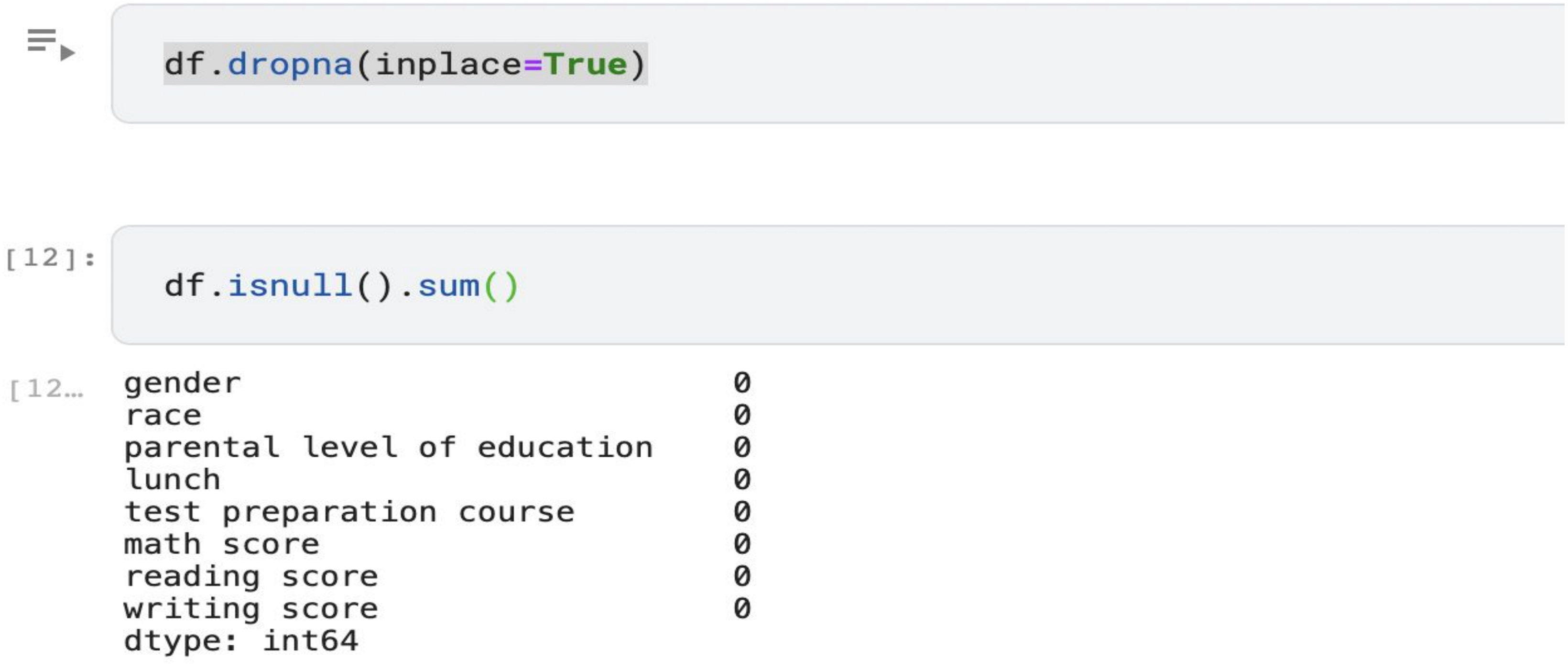


```python
df['math score'].fillna(df['math score'].mean(), inplace=True)
```

```python
df['reading score'].fillna(df['reading score'].mean(), inplace=True)
```

```python
df['writing score'].fillna(df['writing score'].mean(), inplace=True)
```

The null values numeric attributes are replaced by the mean of the attributes.

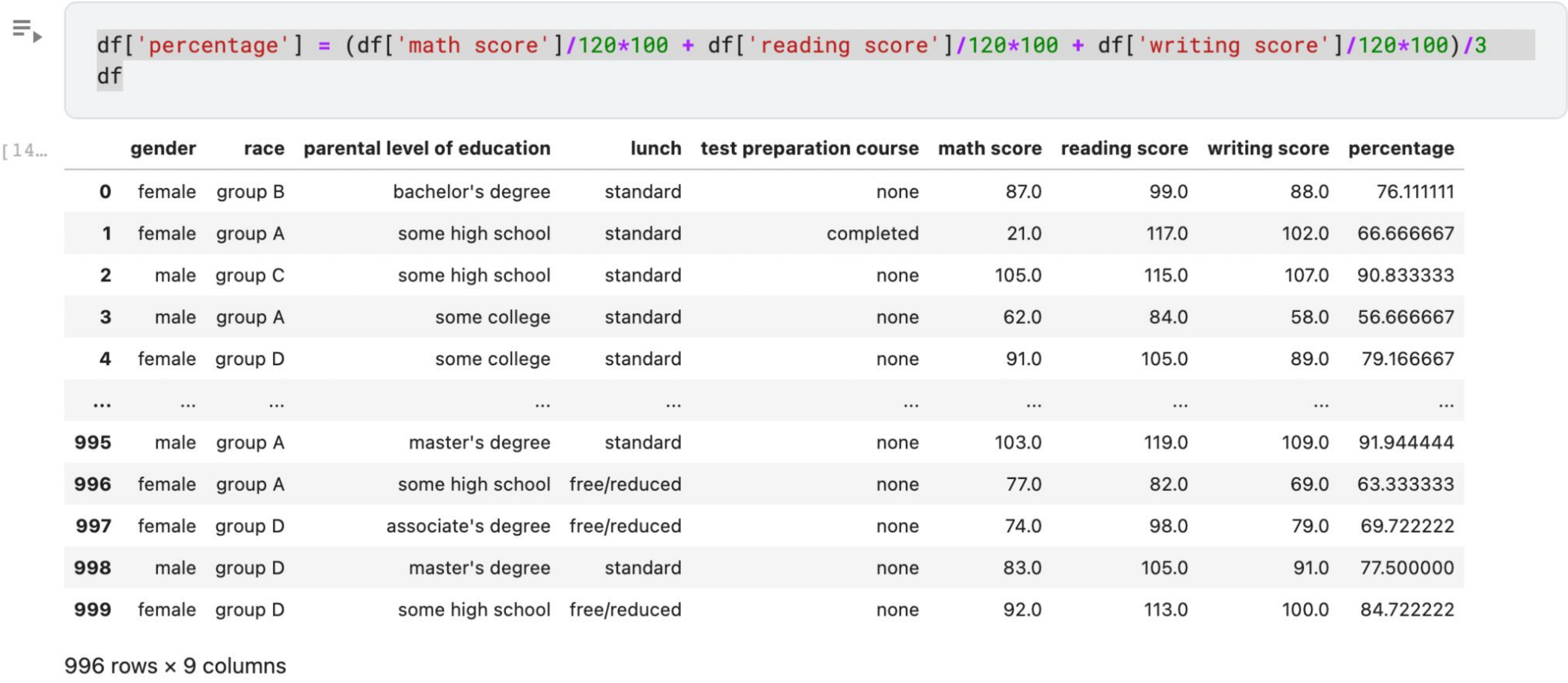


```
df.dropna(inplace=True)
```

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

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

Categorical null values have been dropped and the data contains no null values

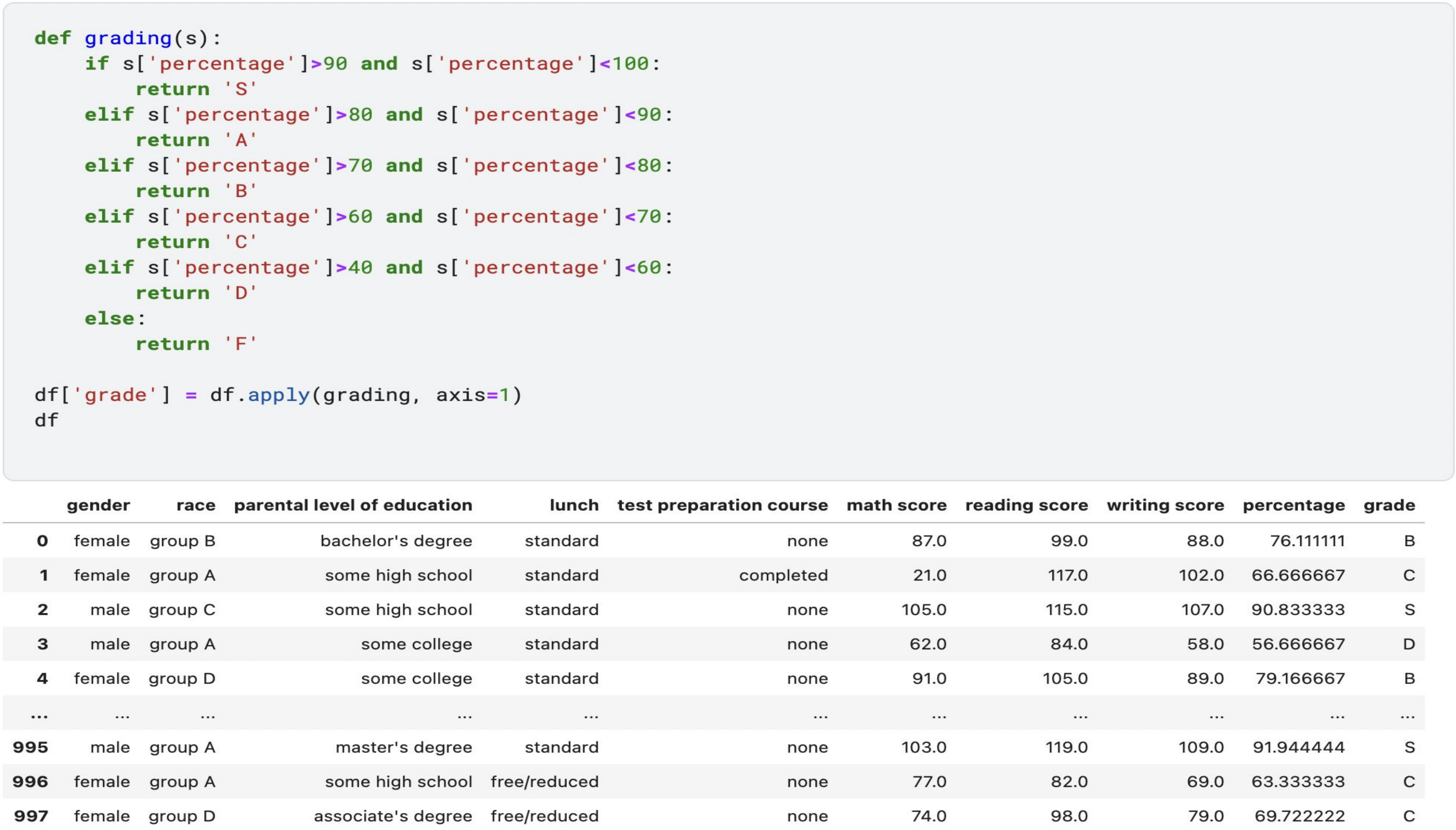


```
df['percentage'] = (df['math score']/120*100 + df['reading score']/120*100 + df['writing score']/120*100)/3
df
```

	gender	race	parental level of education	lunch	test preparation course	math score	reading score	writing score	percentage
0	female	group B	bachelor's degree	standard	none	87.0	99.0	88.0	76.111111
1	female	group A	some high school	standard	completed	21.0	117.0	102.0	66.666667
2	male	group C	some high school	standard	none	105.0	115.0	107.0	90.833333
3	male	group A	some college	standard	none	62.0	84.0	58.0	56.666667
4	female	group D	some college	standard	none	91.0	105.0	89.0	79.166667
...	...	...	...	...	...	...	...	...	...
995	male	group A	master's degree	standard	none	103.0	119.0	109.0	91.944444
996	female	group A	some high school	free/reduced	none	77.0	82.0	69.0	63.333333
997	female	group D	associate's degree	free/reduced	none	74.0	98.0	79.0	69.722222
998	male	group D	master's degree	standard	none	83.0	105.0	91.0	77.500000
999	female	group D	some high school	free/reduced	none	92.0	113.0	100.0	84.722222

996 rows × 9 columns

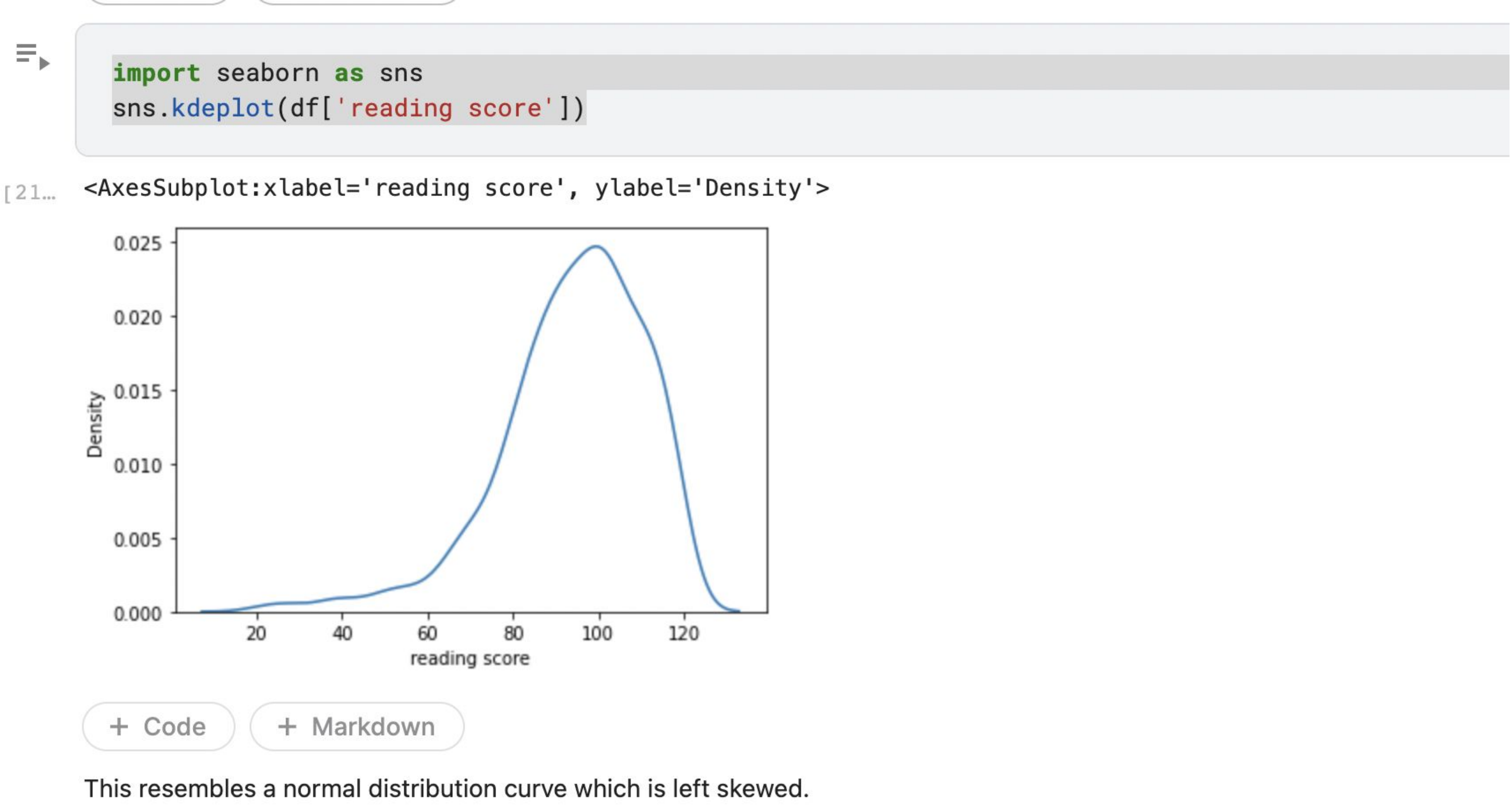
The percentage is calculated and the percentage column is added to the dataset



```python
def grading(s):
    if s['percentage']>90 and s['percentage']<100:
        return 'S'
    elif s['percentage']>80 and s['percentage']<90:
        return 'A'
    elif s['percentage']>70 and s['percentage']<80:
        return 'B'
    elif s['percentage']>60 and s['percentage']<70:
        return 'C'
    elif s['percentage']>40 and s['percentage']<60:
        return 'D'
    else:
        return 'F'

df['grade'] = df.apply(grading, axis=1)
df
```

	gender	race	parental level of education	lunch	test preparation course	math score	reading score	writing score	percentage	grade
0	female	group B	bachelor's degree	standard	none	87.0	99.0	88.0	76.111111	B
1	female	group A	some high school	standard	completed	21.0	117.0	102.0	66.666667	C
2	male	group C	some high school	standard	none	105.0	115.0	107.0	90.833333	S
3	male	group A	some college	standard	none	62.0	84.0	58.0	56.666667	D
4	female	group D	some college	standard	none	91.0	105.0	89.0	79.166667	B
...	...	...	...	...	...	...	...	...	...	...
995	male	group A	master's degree	standard	none	103.0	119.0	109.0	91.944444	S
996	female	group A	some high school	free/reduced	none	77.0	82.0	69.0	63.333333	C
997	female	group D	associate's degree	free/reduced	none	74.0	98.0	79.0	69.722222	C



```python
import seaborn as sns
sns.kdeplot(df['reading score'])
```

```
[21…  <AxesSubplot:xlabel='reading score', ylabel='Density'>
```

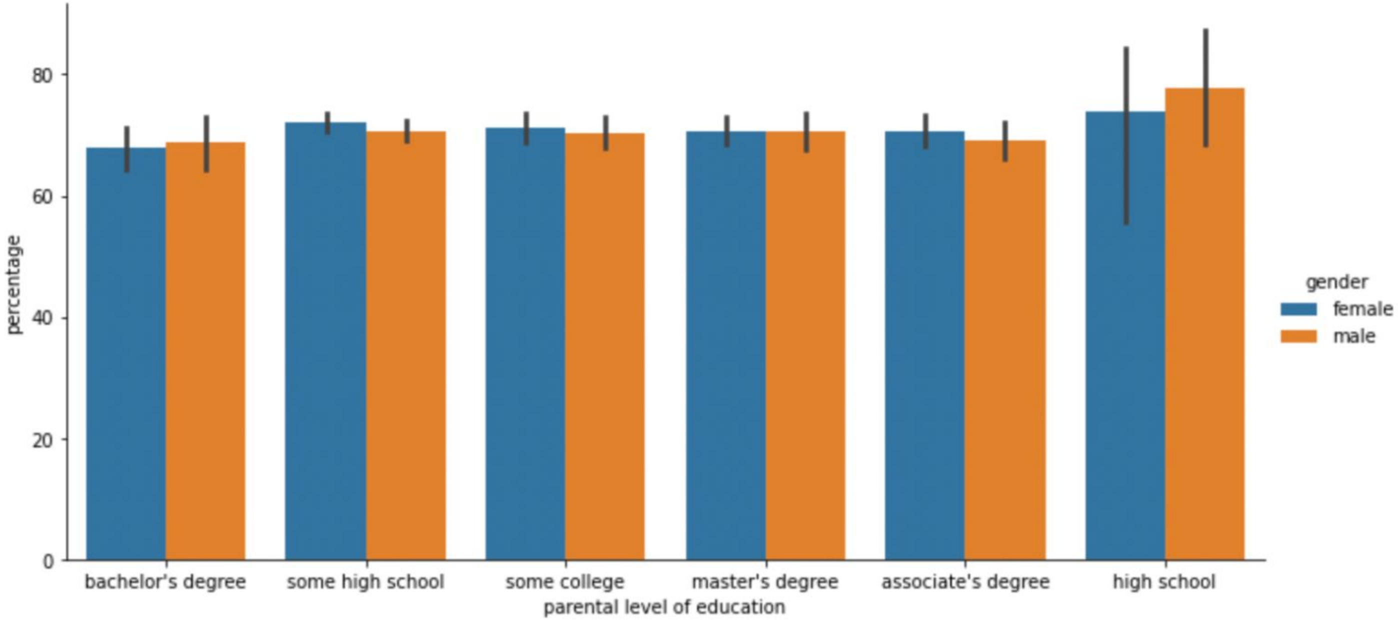
This resembles a normal distribution curve which is left skewed.

[30]:

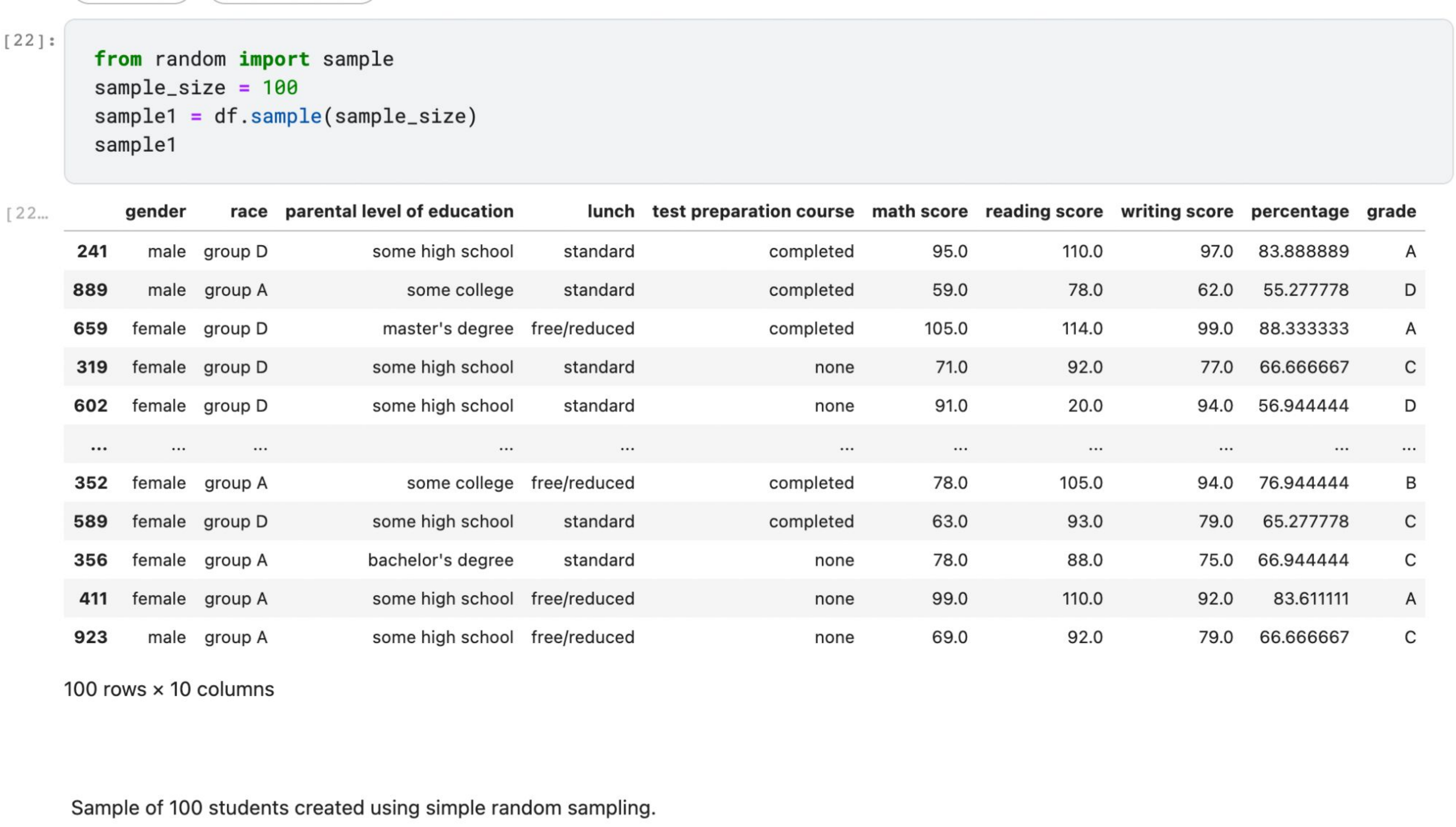
```python
import matplotlib.pyplot as plt
sns.catplot(x='parental level of education', y= 'percentage', hue = 'gender', data = df, kind = 'bar', aspect = 2
```

[30…

```
<seaborn.axisgrid.FacetGrid at 0x7f1879786710>
```



Distribution of percentage across parental level of education, for each gender.



[22]:

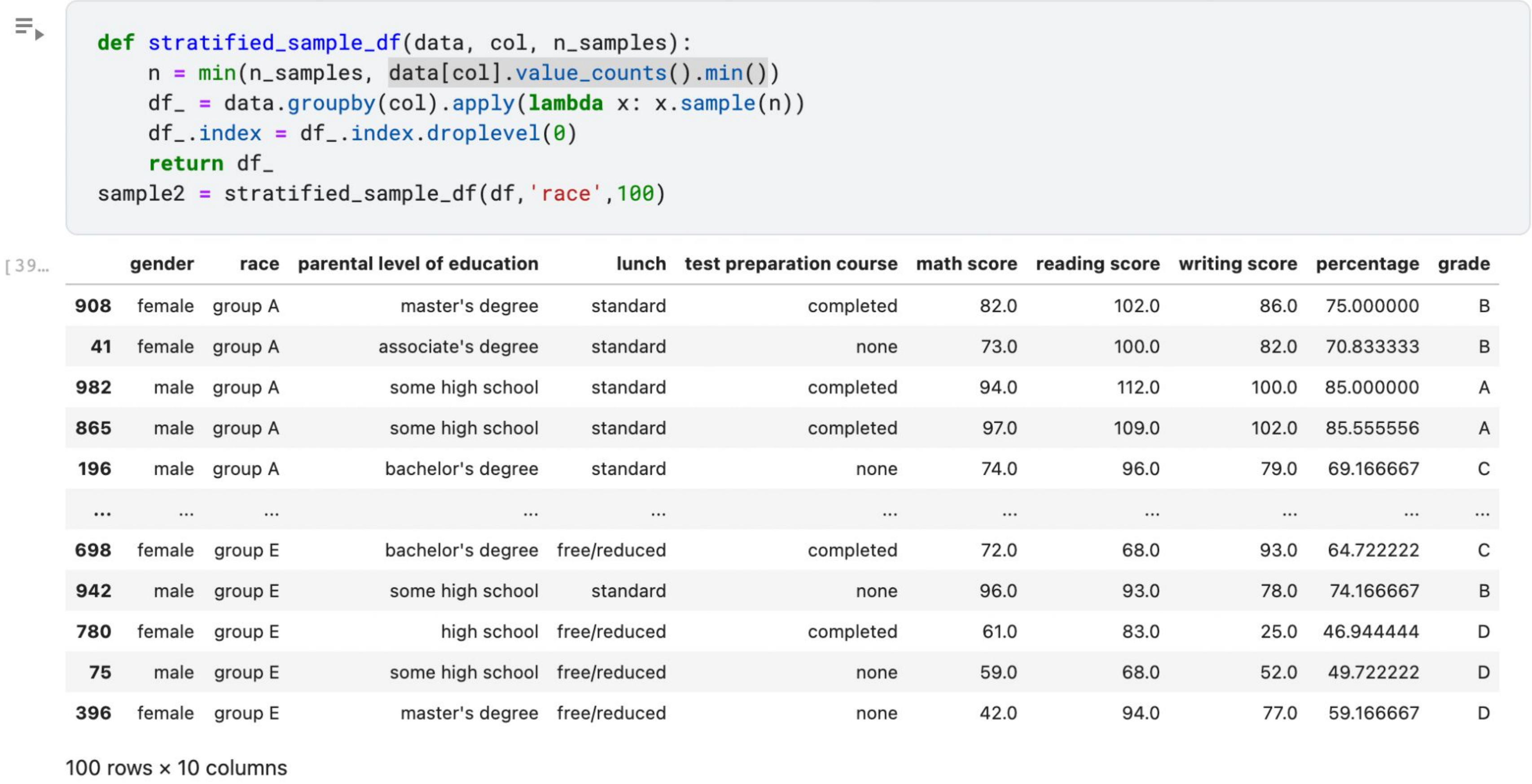
```
from random import sample
sample_size = 100
sample1 = df.sample(sample_size)
sample1
```

[22…

	gender	race	parental level of education	lunch	test preparation course	math score	reading score	writing score	percentage	grade
241	male	group D	some high school	standard	completed	95.0	110.0	97.0	83.888889	A
889	male	group A	some college	standard	completed	59.0	78.0	62.0	55.277778	D
659	female	group D	master's degree	free/reduced	completed	105.0	114.0	99.0	88.333333	A
319	female	group D	some high school	standard	none	71.0	92.0	77.0	66.666667	C
602	female	group D	some high school	standard	none	91.0	20.0	94.0	56.944444	D
...	...	...	...	...	...	...	...	...	...	...
352	female	group A	some college	free/reduced	completed	78.0	105.0	94.0	76.944444	B
589	female	group D	some high school	standard	completed	63.0	93.0	79.0	65.277778	C
356	female	group A	bachelor's degree	standard	none	78.0	88.0	75.0	66.944444	C
411	female	group A	some high school	free/reduced	none	99.0	110.0	92.0	83.611111	A
923	male	group A	some high school	free/reduced	none	69.0	92.0	79.0	66.666667	C

100 rows × 10 columns

Sample of 100 students created using simple random sampling.



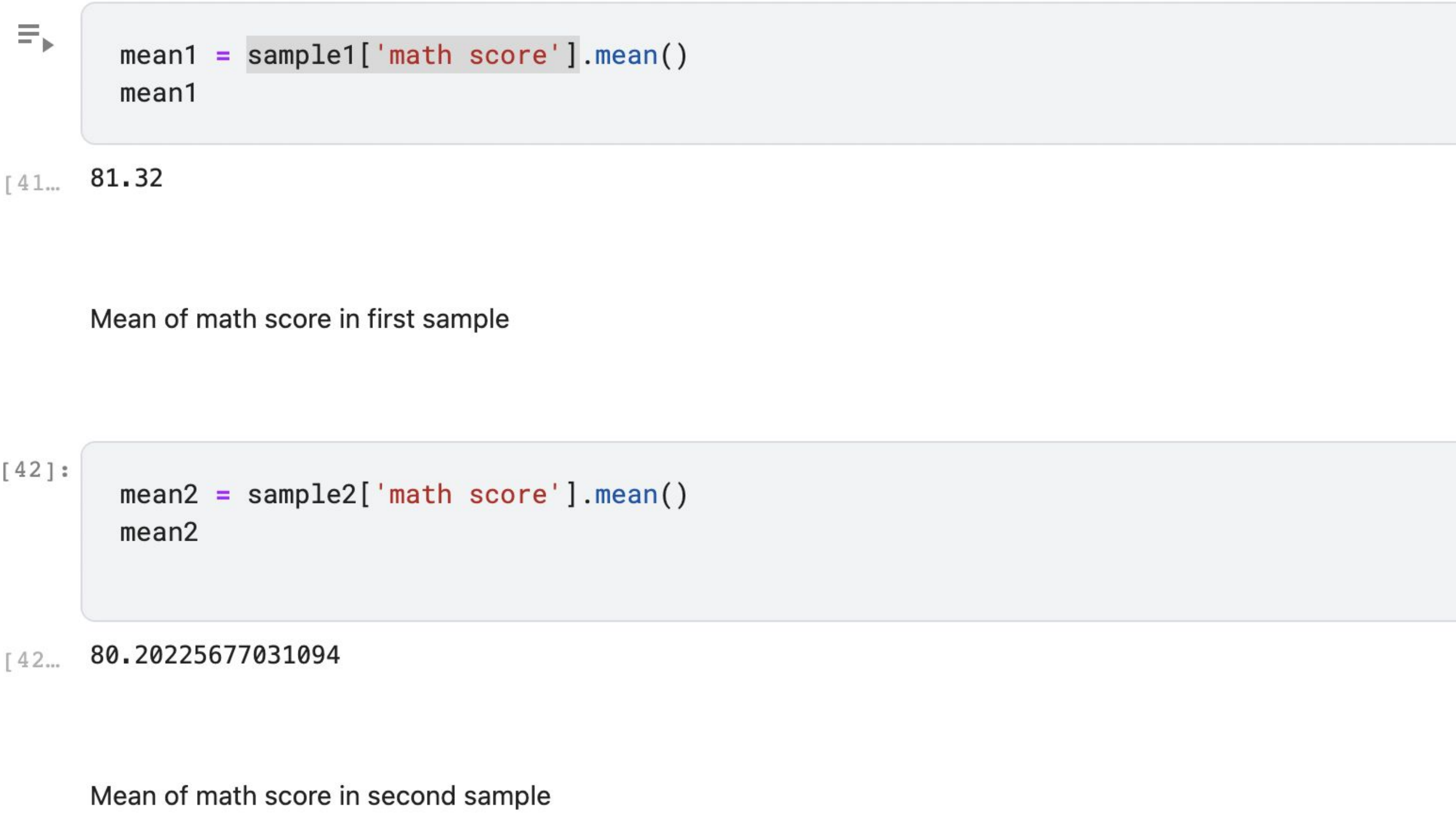
```python
def stratified_sample_df(data, col, n_samples):
    n = min(n_samples, data[col].value_counts().min())
    df_ = data.groupby(col).apply(lambda x: x.sample(n))
    df_.index = df_.index.droplevel(0)
    return df_
sample2 = stratified_sample_df(df,'race',100)
```

[39...

	gender	race	parental level of education	lunch	test preparation course	math score	reading score	writing score	percentage	grade
908	female	group A	master's degree	standard	completed	82.0	102.0	86.0	75.000000	B
41	female	group A	associate's degree	standard	none	73.0	100.0	82.0	70.833333	B
982	male	group A	some high school	standard	completed	94.0	112.0	100.0	85.000000	A
865	male	group A	some high school	standard	completed	97.0	109.0	102.0	85.555556	A
196	male	group A	bachelor's degree	standard	none	74.0	96.0	79.0	69.166667	C
...	...	...	...	...	...	...	...	...	...	...
698	female	group E	bachelor's degree	free/reduced	completed	72.0	68.0	93.0	64.722222	C
942	male	group E	some high school	standard	none	96.0	93.0	78.0	74.166667	B
780	female	group E	high school	free/reduced	completed	61.0	83.0	25.0	46.944444	D
75	male	group E	some high school	free/reduced	none	59.0	68.0	52.0	49.722222	D
396	female	group E	master's degree	free/reduced	none	42.0	94.0	77.0	59.166667	D

100 rows × 10 columns

Sample of 100 students created using Stratified random sampling, using race as strata



```
mean1 = sample1['math score'].mean()
mean1
```

[41…

```
81.32
```

Mean of math score in first sample

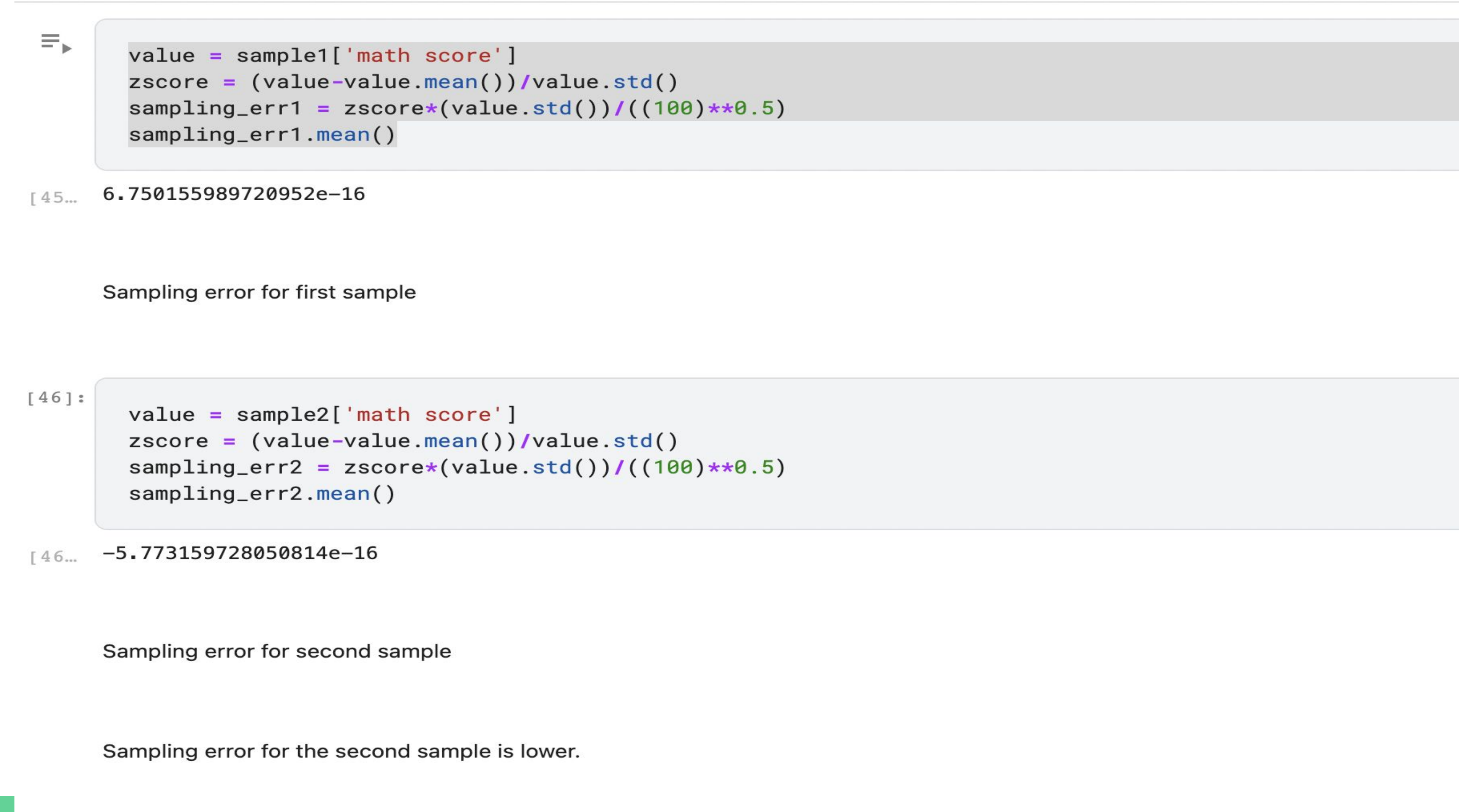
[42]:

```
mean2 = sample2['math score'].mean()
mean2
```

[42…

```
80.20225677031094
```

Mean of math score in second sample



```python
value = sample1['math score']
zscore = (value-value.mean())/value.std()
sampling_err1 = zscore*(value.std())/((100)**0.5)
sampling_err1.mean()
```

```
[45… 6.750155989720952e-16
```

Sampling error for first sample

[46]:

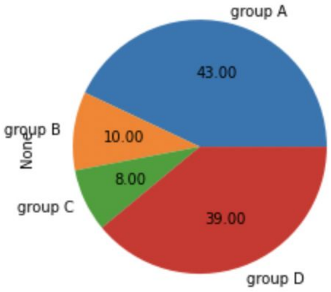
```python
value = sample2['math score']
zscore = (value-value.mean())/value.std()
sampling_err2 = zscore*(value.std())/((100)**0.5)
sampling_err2.mean()
```

```
[46… -5.773159728050814e-16
```

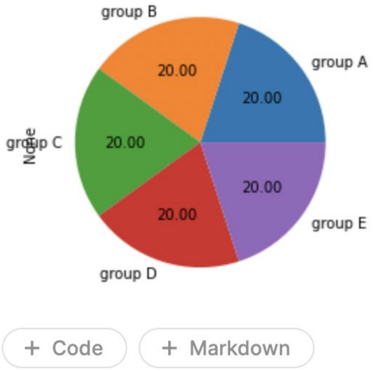
Sampling error for second sample

Sampling error for the second sample is lower.

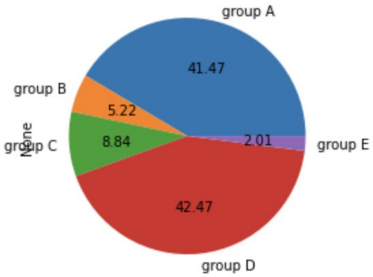
Distribution of race compared between the two samples and the population



Distribution of race in sample 1. Here the majority race is group

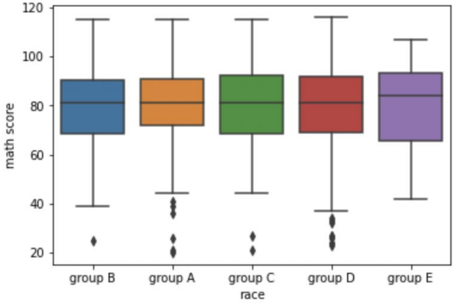


Distribution of race in sample 2.



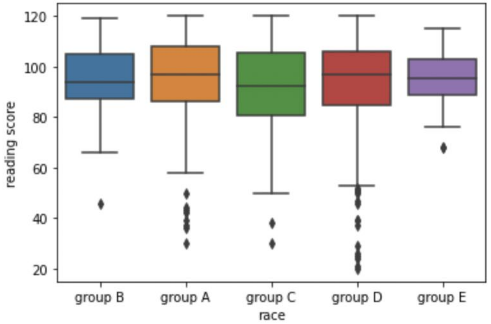
Distribution of race in population. Here the majority race is group D.

Boxplot of Race against subject scores:

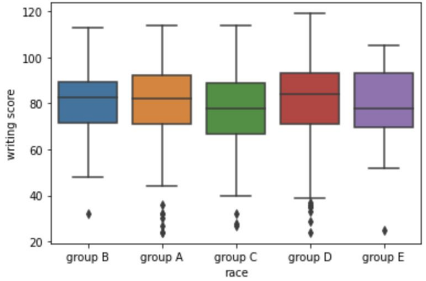


+ Code + Markdown

In race against math score, group A has the greatest number of outliers.



In race against reading score, group D has the greatest number of outliers.



+ Code + Markdown

In race against math score, group A has the greatest number of outliers.

Conclusion

The conclusions drawn from this dataset are:

- students with parents with high school level of education have relatively higher percentage.
- Group A and Group D comprise of the majority of the population of students.
- Students have scored relatively higher in reading, as compared to math and writing.