# StudentsPerformance

May 24, 2021

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

df = pd.read_csv("StudentsPerformance.csv")
df.head()
```

Out[6]:

| | 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 |

## 1 1. Find all the unique 'math score' values in the data.

In [19]:
```
df.head(2)
```

Out[19]:

| | 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 |

In [20]:
```
df.nunique()
```

Out[20]:
```
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 [21]: df['math score'].unique()#Answer
```

```
Out[21]: array([ 72,  69,  90,  47,  76,  71,  88,  40,  64,  38,  58,  65,  78,
                 50,  18,  46,  54,  66,  44,  74,  73,  67,  70,  62,  63,  56,
                 97,  81,  75,  57,  55,  53,  59,  82,  77,  33,  52,   0,  79,
                 39,  45,  60,  61,  41,  49,  30,  80,  42,  27,  43,  68,  85,
                 98,  87,  51,  99,  84,  91,  83,  89,  22, 100,  96,  94,  48,
                 35,  34,  86,  92,  37,  28,  24,  26,  95,  36,  29,  32,  93,
                 19,  23,   8])
```

## 2 Find the number of times a student had exactly '38' in math.

```
In [22]: df.head(2)
```

```
Out[22]:    gender race/ethnicity parental level of education     lunch  \
         0  female        group B           bachelor's degree  standard
         1  female        group C                some college  standard

           test preparation course  math score  reading score  writing score
         0                    none          72             72             74
         1               completed          69             90             88
```

```
In [27]: df[df['math score'] == 38]#answer
```

```
Out[27]:      gender race/ethnicity parental level of education         lunch  \
         9    female        group B                high school  free/reduced
         383  female        group E            some high school  free/reduced
         384  female        group A            some high school  free/reduced

             test preparation course  math score  reading score  writing score
         9                       none          38             60             50
         383                     none          38             49             45
         384                     none          38             43             43
```

```
In [28]: import seaborn as sns
```

```
In [29]: df.count()
```

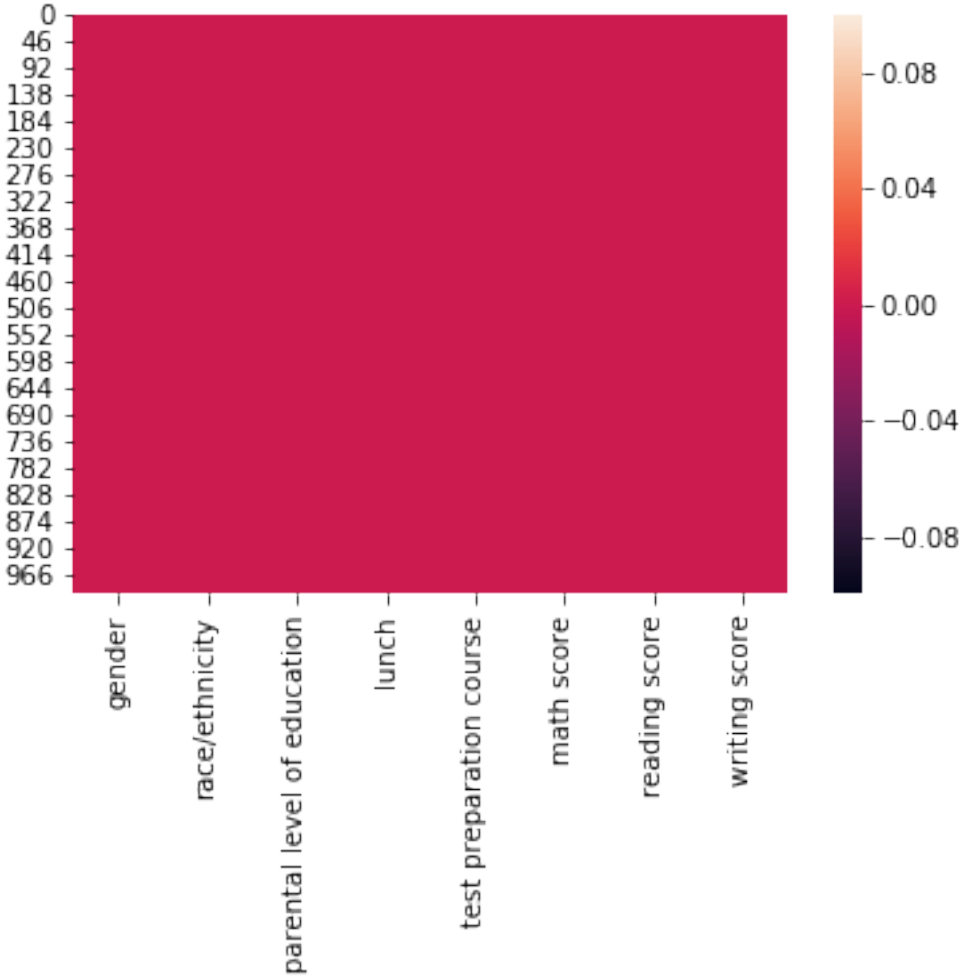
```
Out[29]: gender                       1000
         race/ethnicity               1000
         parental level of education  1000
```

```
lunch                          1000
test preparation course        1000
math score                     1000
reading score                  1000
writing score                  1000
dtype: int64
```

In [31]:
```
df.isnull().sum()
```

Out[31]:
```
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
```

In [33]:
```
sns.heatmap(df.isnull())
plt.show()
```



## 3 Q3. What does the data suggest about writing ,reading and spelling in each race.

```
In [39]: df.head(2)
```

```
Out[39]:    gender race/ethnicity parental level of education     lunch  \
         0  female        group B           bachelor's degree  standard
         1  female        group C                some college  standard

           test preparation course  math score  reading score  writing score
         0                    none          72             72             74
         1               completed          69             90             88
```

df.groupby('race/ethnicity').sum() # answer

```
In [44]: df.groupby('race/ethnicity')['math score'].sum().sort_values(ascending = False)
```

```
Out[44]: race/ethnicity
         group C    20564
         group D    17649
         group B    12056
         group E    10335
         group A     5485
         Name: math score, dtype: int64
```

```
In [45]: df.groupby('race/ethnicity')['reading score'].sum().sort_values(ascending = False)
```

```
Out[45]: race/ethnicity
         group C    22044
         group D    18348
         group B    12797
         group E    10224
         group A     5756
         Name: reading score, dtype: int64
```

```
In [46]: df.groupby('race/ethnicity')['writing score'].sum().sort_values(ascending = False)
```

```
Out[46]: race/ethnicity
         group C    21637
         group D    18378
         group B    12464
         group E     9997
         group A     5578
         Name: writing score, dtype: int64
```

## 4 Handling missing data

In [48]:
```
df.isnull().sum() # There are no null values ,which means that there are no missing dat
because it shows the number of values in each of the columns.
```

Out[48]:
```
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
```

In [ ]: