

# Python\_Lab\_Assignment

October 4, 2022

## 1 PyCity Schools Analysis

- As a whole, schools with higher budgets, did not yield better test results. By contrast, schools with higher spending per student actually (\\$645 - 675) underperformed compared to schools with smaller budgets (\\$585 per student).
- As a whole, smaller and medium sized schools dramatically out-performed large sized schools on passing math performances (89-91% passing vs 67%).
- 

**1.1 As a whole, charter schools out-performed the public district schools across all metrics. However, more analysis will be required to glean if the effect is due to school practices or the fact that charter schools tend to serve smaller student populations per school.**

**Note:** Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

```
[1]: # Dependencies and Setup
import pandas as pd

# File to Load (Remember to Change These)
school_data_to_load = "data/schools_complete.csv"
student_data_to_load = "data/students_complete.csv"

# Read School and Student Data File and store into Pandas Data Frames
school_data = pd.read_csv(school_data_to_load)
student_data = pd.read_csv(student_data_to_load)

# Combine the data into a single dataset
school_data_complete = pd.merge(student_data, school_data, how="left",
                                  on=["school_name", "school_name"])
school_data_complete.head()

# school_data_complete.count()
```

```
[1]:   Student ID      student_name gender grade  school_name \
0           0      Paul Bradley      M   9th  Huang High School
```

1	1	Victor Smith	M	12th	Huang High School
2	2	Kevin Rodriguez	M	12th	Huang High School
3	3	Dr. Richard Scott	M	12th	Huang High School
4	4	Bonnie Ray	F	9th	Huang High School

	reading_score	math_score	School ID	type	size	budget
0	66	79	0	District	2917	1910635
1	94	61	0	District	2917	1910635
2	90	60	0	District	2917	1910635
3	67	58	0	District	2917	1910635
4	97	84	0	District	2917	1910635

## 1.2 District Summary

- Calculate the total number of schools
- Calculate the total number of students
- Calculate the total budget
- Calculate the average math score
- Calculate the average reading score
- Calculate the overall passing rate (overall average score), i.e. (avg. math score + avg. reading score)/2
- Calculate the percentage of students with a passing math score (70 or greater)
- Calculate the percentage of students with a passing reading score (70 or greater)
- Create a dataframe to hold the above results
- Optional: give the displayed data cleaner formatting

```
[2]: # Create a District Summary
# Total number of schools
schools = school_data_complete['school_name'].nunique()
# Total number of students
students = school_data_complete['student_name'].count()
# Total budget
budget = sum(school_data_complete['budget'].unique())
# Average math score
avg_math = school_data_complete['math_score'].mean()
# Average reading score
avg_reading = school_data_complete['reading_score'].mean()
# Overall average score
avg_all = (avg_math + avg_reading)/2
# Percentage of passing math (70 or greater)
pass_math = ((school_data_complete["math_score"]>69).sum() /
↳ school_data_complete["math_score"].count())* 100
# Percentage of passing reading (70 or greater)
```

```

pass_reading = ((school_data_complete["reading_score"]>69).sum() /
↳school_data_complete["reading_score"].count())* 100
district = {
    'Total Schools':schools,
    'Total Student':'{:,}'.format(students),
    'Total Budget':'${:,.2f}'.format(budget),
    'Average Math Score':avg_math,
    'Average Reading Score':avg_reading,
    '% Passing Math':pass_math,
    '% Passing Reading':pass_reading,
    '% Overall Passing Score':[avg_all],
}

district_summery = pd.DataFrame(district)
district_summery

```

```

[2]:      Total Schools Total Student    Total Budget  Average Math Score \
0                15         39,170  $24,649,428.00          78.985371

      Average Reading Score  % Passing Math  % Passing Reading \
0                81.87784         74.980853          85.805463

      % Overall Passing Score
0                80.431606

```

```

[3]: # Total number of schools
schools

```

```

[3]: 15

```

```

[4]: # Total number of students
students

```

```

[4]: 39170

```

```

[5]: # Total budget
budget

```

```

[5]: 24649428

```

```

[6]: # Average math score
avg_math

```

```

[6]: 78.98537145774827

```

```

[7]: # Average reading score
avg_reading

```

[7]: 81.87784018381414

```
[8]: # Overall average score
avg_all
```

[8]: 80.43160582078121

```
[9]: # Percentage of passing math (70 or greater)
pass_math
```

[9]: 74.9808526933878

```
[10]: # Percentage of passing reading (70 or greater)
pass_reading
```

[10]: 85.80546336482001

```
[11]: # Percentage of overall passing
pass_overall = (pass_math + pass_reading)/2
pass_overall
```

[11]: 80.39315802910392

### 1.3 School Summary

- Create an overview table that summarizes key metrics about each school, including:
  - School Name
  - School Type
  - Total Students
  - Total School Budget
  - Per Student Budget
  - Average Math Score
  - Average Reading Score
  - % Passing Math
  - % Passing Reading
  - Overall Passing Rate (Average of the above two)
- Create a dataframe to hold the above results

```
[12]: # Group the dataset after the schools
group_schools = school_data_complete.groupby(['school_name'])
# school type for each school
type_schools = group_schools['type'].first()
# students for each school
students_schools = group_schools.size()
# total budget for each school
budget_schools = group_schools['budget'].first()
# budget per student for each school
budget_students = budget_schools/students_schools
```

```

# average math score for each school
avg_math_schools = group_schools['math_score'].mean()
# average reading score for each school
avg_reading_schools = group_schools['reading_score'].mean()
# percentange of passing math score for each school
pass_math_schools = □
    ↳(school_data_complete[school_data_complete['math_score']>69].
    ↳groupby(['school_name']).size()/students_schools)*100
# percentange of passing math score for each school
pass_reading_schools = □
    ↳(school_data_complete[school_data_complete['reading_score']>69].
    ↳groupby(['school_name']).size()/students_schools)*100
# Calculate the overall passing score for each school
pass_overall_schools = (pass_math_schools + pass_reading_schools)/2

# Create a dataframe to hold the above results
school={
    'School Type': type_schools,
    'Total Students':students_schools,
    'Total School Budget': budget_schools,
    'Per Student Budget': budget_students,
    'Average Math Score': avg_math_schools,
    'Average Reading Score': avg_reading_schools,
    '% Passing Math': pass_math_schools,
    '% Passing Reading': pass_reading_schools,
    '% Overall Passing Rate': pass_overall_schools,
}
school_summary = pd.DataFrame(school)
school_summary.index.name = None

```

```
[13]: school_summary
```

```

[13]:
      School Type  Total Students  Total School Budget  \
Bailey High School      District          4976          3124928
Cabrera High School      Charter           1858          1081356
Figueroa High School      District          2949          1884411
Ford High School          District          2739          1763916
Griffin High School       Charter           1468           917500
Hernandez High School     District          4635          3022020
Holden High School        Charter            427           248087
Huang High School         District          2917          1910635
Johnson High School       District          4761          3094650
Pena High School          Charter            962           585858
Rodriguez High School     District          3999          2547363
Shelton High School       Charter           1761          1056600
Thomas High School        Charter          1635          1043130
Wilson High School        Charter           2283          1319574

```

Wright High School	Charter	1800	1049400
--------------------	---------	------	---------

	Per Student Budget	Average Math Score \
Bailey High School	628.0	77.048432
Cabrera High School	582.0	83.061895
Figueroa High School	639.0	76.711767
Ford High School	644.0	77.102592
Griffin High School	625.0	83.351499
Hernandez High School	652.0	77.289752
Holden High School	581.0	83.803279
Huang High School	655.0	76.629414
Johnson High School	650.0	77.072464
Pena High School	609.0	83.839917
Rodriguez High School	637.0	76.842711
Shelton High School	600.0	83.359455
Thomas High School	638.0	83.418349
Wilson High School	578.0	83.274201
Wright High School	583.0	83.682222

	Average Reading Score	% Passing Math \
Bailey High School	81.033963	66.680064
Cabrera High School	83.975780	94.133477
Figueroa High School	81.158020	65.988471
Ford High School	80.746258	68.309602
Griffin High School	83.816757	93.392371
Hernandez High School	80.934412	66.752967
Holden High School	83.814988	92.505855
Huang High School	81.182722	65.683922
Johnson High School	80.966394	66.057551
Pena High School	84.044699	94.594595
Rodriguez High School	80.744686	66.366592
Shelton High School	83.725724	93.867121
Thomas High School	83.848930	93.272171
Wilson High School	83.989488	93.867718
Wright High School	83.955000	93.333333

	% Passing Reading	% Overall Passing Rate
Bailey High School	81.933280	74.306672
Cabrera High School	97.039828	95.586652
Figueroa High School	80.739234	73.363852
Ford High School	79.299014	73.804308
Griffin High School	97.138965	95.265668
Hernandez High School	80.862999	73.807983
Holden High School	96.252927	94.379391
Huang High School	81.316421	73.500171
Johnson High School	81.222432	73.639992
Pena High School	95.945946	95.270270

Rodriguez High School	80.220055	73.293323
Shelton High School	95.854628	94.860875
Thomas High School	97.308869	95.290520
Wilson High School	96.539641	95.203679
Wright High School	96.611111	94.972222

### 1.3.1 Top Performing Schools (By Passing Rate)

- Sort and display the top five schools in overall passing rate

```
[14]: # Sort and display the top five schools in overall passing rate
top_performing_schools = school_summary.sort_values(by='% Overall Passing_
↳Rate',ascending=False)
top_performing_schools.head()
```

```
[14]:
```

	School Type	Total Students	Total School Budget \
Cabrera High School	Charter	1858	1081356
Thomas High School	Charter	1635	1043130
Pena High School	Charter	962	585858
Griffin High School	Charter	1468	917500
Wilson High School	Charter	2283	1319574

  

	Per Student Budget	Average Math Score \
Cabrera High School	582.0	83.061895
Thomas High School	638.0	83.418349
Pena High School	609.0	83.839917
Griffin High School	625.0	83.351499
Wilson High School	578.0	83.274201

  

	Average Reading Score	% Passing Math	% Passing Reading \
Cabrera High School	83.975780	94.133477	97.039828
Thomas High School	83.848930	93.272171	97.308869
Pena High School	84.044699	94.594595	95.945946
Griffin High School	83.816757	93.392371	97.138965
Wilson High School	83.989488	93.867718	96.539641

  

	% Overall Passing Rate
Cabrera High School	95.586652
Thomas High School	95.290520
Pena High School	95.270270
Griffin High School	95.265668
Wilson High School	95.203679

```
[15]: # Calculate the total school budget
# for each school
budget_schools
```

```
[15]: Bailey High School      3124928
      Cabrera High School    1081356
      Figueroa High School   1884411
      Ford High School       1763916
      Griffin High School    917500
      Hernandez High School  3022020
      Holden High School     248087
      Huang High School      1910635
      Johnson High School    3094650
      Pena High School       585858
      Rodriguez High School  2547363
      Shelton High School    1056600
      Thomas High School     1043130
      Wilson High School     1319574
      Wright High School     1049400
      Name: budget, dtype: int64
```

```
[16]: # sum of all schools
      budget_schools.sum()
```

```
[16]: 24649428
```

```
[17]: # Calculate per student budget
      budget_students
```

```
[17]: Bailey High School      628.0
      Cabrera High School    582.0
      Figueroa High School   639.0
      Ford High School       644.0
      Griffin High School    625.0
      Hernandez High School  652.0
      Holden High School     581.0
      Huang High School      655.0
      Johnson High School    650.0
      Pena High School       609.0
      Rodriguez High School  637.0
      Shelton High School    600.0
      Thomas High School     638.0
      Wilson High School     578.0
      Wright High School     583.0
      dtype: float64
```

```
[18]: # Caculate the avg math and reading score
      avg_math_schools
```

```
[18]: Bailey High School      77.048432
      Cabrera High School    83.061895
      Figueroa High School   76.711767
```



Ford High School	77.102592
Griffin High School	83.351499
Hernandez High School	77.289752
Holden High School	83.803279
Huang High School	76.629414
Johnson High School	77.072464
Pena High School	83.839917
Rodriguez High School	76.842711
Shelton High School	83.359455
Thomas High School	83.418349
Wilson High School	83.274201
Wright High School	83.682222

Name: math\_score, dtype: float64

```
[19]: # Calculate avg reading score
avg_reading_schools
```

```
[19]: Bailey High School      81.033963
Cabrera High School      83.975780
Figueroa High School     81.158020
Ford High School         80.746258
Griffin High School      83.816757
Hernandez High School    80.934412
Holden High School       83.814988
Huang High School        81.182722
Johnson High School      80.966394
Pena High School         84.044699
Rodriguez High School    80.744686
Shelton High School      83.725724
Thomas High School       83.848930
Wilson High School       83.989488
Wright High School       83.955000
Name: reading_score, dtype: float64
```

**Find the passing rate for math and reading (above 70 points)**

```
[20]: #math passing rate
pass_math_schools
```

```
[20]: school_name
Bailey High School      66.680064
Cabrera High School     94.133477
Figueroa High School    65.988471
Ford High School        68.309602
Griffin High School     93.392371
Hernandez High School   66.752967
Holden High School      92.505855
Huang High School       65.683922
```

Johnson High School	66.057551
Pena High School	94.594595
Rodriguez High School	66.366592
Shelton High School	93.867121
Thomas High School	93.272171
Wilson High School	93.867718
Wright High School	93.333333

dtype: float64

```
[21]: #read passing rate
pass_reading_schools
```

```
[21]: school_name
Bailey High School      81.933280
Cabrera High School     97.039828
Figueroa High School    80.739234
Ford High School        79.299014
Griffin High School     97.138965
Hernandez High School   80.862999
Holden High School      96.252927
Huang High School       81.316421
Johnson High School     81.222432
Pena High School        95.945946
Rodriguez High School    80.220055
Shelton High School     95.854628
Thomas High School      97.308869
Wilson High School      96.539641
Wright High School      96.611111
dtype: float64
```

```
[22]: # Calculate the overall passing rate (average of the math and reading passing
      ↪rate)
pass_overall_schools
```

```
[22]: school_name
Bailey High School      74.306672
Cabrera High School     95.586652
Figueroa High School    73.363852
Ford High School        73.804308
Griffin High School     95.265668
Hernandez High School   73.807983
Holden High School      94.379391
Huang High School       73.500171
Johnson High School     73.639992
Pena High School        95.270270
Rodriguez High School    73.293323
Shelton High School     94.860875
```

Thomas High School	95.290520
Wilson High School	95.203679
Wright High School	94.972222

dtype: float64

### 1.3.2 Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

```
[23]: # Sort and display the worst five schools in overall passing rate
worst_performing_schools = school_summary.sort_values(by='% Overall Passing_
↳Rate')
worst_performing_schools.head()
```

```
[23]:
```

	School Type	Total Students	Total School Budget \
Rodriguez High School	District	3999	2547363
Figueroa High School	District	2949	1884411
Huang High School	District	2917	1910635
Johnson High School	District	4761	3094650
Ford High School	District	2739	1763916

	Per Student Budget	Average Math Score \
Rodriguez High School	637.0	76.842711
Figueroa High School	639.0	76.711767
Huang High School	655.0	76.629414
Johnson High School	650.0	77.072464
Ford High School	644.0	77.102592

	Average Reading Score	% Passing Math \
Rodriguez High School	80.744686	66.366592
Figueroa High School	81.158020	65.988471
Huang High School	81.182722	65.683922
Johnson High School	80.966394	66.057551
Ford High School	80.746258	68.309602

	% Passing Reading	% Overall Passing Rate
Rodriguez High School	80.220055	73.293323
Figueroa High School	80.739234	73.363852
Huang High School	81.316421	73.500171
Johnson High School	81.222432	73.639992
Ford High School	79.299014	73.804308

## 1.4 Math Scores by Grade

- Create a table that lists the average Reading Score for students of each grade level (9th, 10th, 11th, 12th) at each school.
  - Create a pandas series for each grade. Hint: use a conditional statement.

- Group each series by school
- Combine the series into a dataframe
- Optional: give the displayed data cleaner formatting

```
[24]: # Create table that lists the average math score for each school of each grade_
      ↪level.
      # Calculate the average math score for students of 9th grade at each school
      math_grade_9th = school_data_complete[school_data_complete['grade']=='9th'].
      ↪groupby('school_name')['math_score'].mean()
      # Calculate the average math score for students of 10th grade at each school
      math_grade_10th = school_data_complete[school_data_complete['grade']=='10th'].
      ↪groupby('school_name')['math_score'].mean()
      # Calculate the average math score for students of 11th grade at each school
      math_grade_11th = school_data_complete[school_data_complete['grade']=='11th'].
      ↪groupby('school_name')['math_score'].mean()
      # Calculate the average math score for students of 12th grade at each school
      math_grade_12th = school_data_complete[school_data_complete['grade']=='12th'].
      ↪groupby('school_name')['math_score'].mean()

      # Create a dataframe to hold the above results
      grade_math={
          '9th':math_grade_9th,
          '10th':math_grade_10th,
          '11th':math_grade_11th,
          '12th':math_grade_12th,
      }

      math_by_grade = pd.DataFrame(grade_math)
      math_by_grade.index.name = None
      math_by_grade.head(15)
```

```
[24]:
```

	9th	10th	11th	12th
Bailey High School	77.083676	76.996772	77.515588	76.492218
Cabrera High School	83.094697	83.154506	82.765560	83.277487
Figueroa High School	76.403037	76.539974	76.884344	77.151369
Ford High School	77.361345	77.672316	76.918058	76.179963
Griffin High School	82.044010	84.229064	83.842105	83.356164
Hernandez High School	77.438495	77.337408	77.136029	77.186567
Holden High School	83.787402	83.429825	85.000000	82.855422
Huang High School	77.027251	75.908735	76.446602	77.225641
Johnson High School	77.187857	76.691117	77.491653	76.863248
Pena High School	83.625455	83.372000	84.328125	84.121547
Rodriguez High School	76.859966	76.612500	76.395626	77.690748
Shelton High School	83.420755	82.917411	83.383495	83.778976
Thomas High School	83.590022	83.087886	83.498795	83.497041
Wilson High School	83.085578	83.724422	83.195326	83.035794

Wright High School      83.264706   84.010288   83.836782   83.644986

```
[25]: # Calculate the average math score for 9th grade in each school
      math_grade_9th
```

```
[25]: Bailey High School      77.083676
      Cabrera High School     83.094697
      Figueroa High School    76.403037
      Ford High School        77.361345
      Griffin High School     82.044010
      Hernandez High School   77.438495
      Holden High School      83.787402
      Huang High School       77.027251
      Johnson High School     77.187857
      Pena High School        83.625455
      Rodriguez High School   76.859966
      Shelton High School     83.420755
      Thomas High School      83.590022
      Wilson High School      83.085578
      Wright High School      83.264706
      Name: math_score, dtype: float64
```

```
[26]: # Calculate the average math score for 10th grade in each school
      math_grade_10th
```

```
[26]: school_name
      Bailey High School      76.996772
      Cabrera High School     83.154506
      Figueroa High School    76.539974
      Ford High School        77.672316
      Griffin High School     84.229064
      Hernandez High School   77.337408
      Holden High School      83.429825
      Huang High School       75.908735
      Johnson High School     76.691117
      Pena High School        83.372000
      Rodriguez High School   76.612500
      Shelton High School     82.917411
      Thomas High School      83.087886
      Wilson High School      83.724422
      Wright High School      84.010288
      Name: math_score, dtype: float64
```

```
[27]: # Calculate the average math score for 11th grade in each school
      math_grade_11th
```

```
[27]: school_name
      Bailey High School      77.515588
      Cabrera High School    82.765560
      Figueroa High School   76.884344
      Ford High School       76.918058
      Griffin High School    83.842105
      Hernandez High School   77.136029
      Holden High School     85.000000
      Huang High School      76.446602
      Johnson High School    77.491653
      Pena High School       84.328125
      Rodriguez High School   76.395626
      Shelton High School    83.383495
      Thomas High School     83.498795
      Wilson High School     83.195326
      Wright High School     83.836782
      Name: math_score, dtype: float64
```

```
[28]: # Calculate the average math score for 12th grade in each school
      math_grade_12th
```

```
[28]: school_name
      Bailey High School      76.492218
      Cabrera High School    83.277487
      Figueroa High School   77.151369
      Ford High School       76.179963
      Griffin High School    83.356164
      Hernandez High School   77.186567
      Holden High School     82.855422
      Huang High School      77.225641
      Johnson High School    76.863248
      Pena High School       84.121547
      Rodriguez High School   77.690748
      Shelton High School    83.778976
      Thomas High School     83.497041
      Wilson High School     83.035794
      Wright High School     83.644986
      Name: math_score, dtype: float64
```

### 1.4.1 Reading Score by Grade

- Perform the same operations as above for reading scores

```
[29]: # Create table that lists the average reading score for each school of each
      ↪ grade level.
      # Calculate the average reading score for students of 9th grade at each school
      read_grade_9th = school_data_complete[school_data_complete['grade']=='9th'].
      ↪groupby('school_name')['reading_score'].mean()
```

```

# Calculate the average reading score for students of 10th grade at each school
read_grade_10th = school_data_complete[school_data_complete['grade']=='10th'].
↳groupby('school_name')['reading_score'].mean()
# Calculate the average reading score for students of 11th grade at each school
read_grade_11th = school_data_complete[school_data_complete['grade']=='11th'].
↳groupby('school_name')['reading_score'].mean()
# Calculate the average reading score for students of 12th grade at each school
read_grade_12th = school_data_complete[school_data_complete['grade']=='12th'].
↳groupby('school_name')['reading_score'].mean()

# Create a dataframe to hold the above results
reading_grade={
    '9th':read_grade_9th,
    '10th':read_grade_10th,
    '11th':read_grade_11th,
    '12th':read_grade_12th,
}

reading_score_grade = pd.DataFrame(reading_grade)
reading_score_grade.index.name = None
reading_score_grade

```

```

[29]:

```

	9th	10th	11th	12th
Bailey High School	81.303155	80.907183	80.945643	80.912451
Cabrera High School	83.676136	84.253219	83.788382	84.287958
Figueroa High School	81.198598	81.408912	80.640339	81.384863
Ford High School	80.632653	81.262712	80.403642	80.662338
Griffin High School	83.369193	83.706897	84.288089	84.013699
Hernandez High School	80.866860	80.660147	81.396140	80.857143
Holden High School	83.677165	83.324561	83.815534	84.698795
Huang High School	81.290284	81.512386	81.417476	80.305983
Johnson High School	81.260714	80.773431	80.616027	81.227564
Pena High School	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	80.993127	80.629808	80.864811	80.376426
Shelton High School	84.122642	83.441964	84.373786	82.781671
Thomas High School	83.728850	84.254157	83.585542	83.831361
Wilson High School	83.939778	84.021452	83.764608	84.317673
Wright High School	83.833333	83.812757	84.156322	84.073171

```

[30]: # Calculate the average reading score for 9th grade in each school
read_grade_9th

```

```

[30]:

```

Bailey High School	81.303155
Cabrera High School	83.676136
Figueroa High School	81.198598
Ford High School	80.632653
Griffin High School	83.369193

Hernandez High School	80.866860
Holden High School	83.677165
Huang High School	81.290284
Johnson High School	81.260714
Pena High School	83.807273
Rodriguez High School	80.993127
Shelton High School	84.122642
Thomas High School	83.728850
Wilson High School	83.939778
Wright High School	83.833333

Name: reading\_score, dtype: float64

```
[31]: # Calculate the average reading score for 10th grade in each school
      read_grade_10th
```

```
[31]: school_name
      Bailey High School      80.907183
      Cabrera High School    84.253219
      Figueroa High School   81.408912
      Ford High School       81.262712
      Griffin High School    83.706897
      Hernandez High School   80.660147
      Holden High School     83.324561
      Huang High School      81.512386
      Johnson High School    80.773431
      Pena High School       83.612000
      Rodriguez High School  80.629808
      Shelton High School    83.441964
      Thomas High School     84.254157
      Wilson High School     84.021452
      Wright High School     83.812757
      Name: reading_score, dtype: float64
```

```
[32]: # Calculate the average reading score for 11th grade in each school
      read_grade_11th
```

```
[32]: school_name
      Bailey High School      80.945643
      Cabrera High School    83.788382
      Figueroa High School   80.640339
      Ford High School       80.403642
      Griffin High School    84.288089
      Hernandez High School   81.396140
      Holden High School     83.815534
      Huang High School      81.417476
      Johnson High School    80.616027
      Pena High School       84.335938
```



```
Rodriguez High School    80.864811
Shelton High School      84.373786
Thomas High School       83.585542
Wilson High School       83.764608
Wright High School       84.156322
Name: reading_score, dtype: float64
```

```
[33]: # Calculate the average reading score for 12th grade in each school
read_grade_12th
```

```
[33]: school_name
Bailey High School      80.912451
Cabrera High School     84.287958
Figueroa High School    81.384863
Ford High School        80.662338
Griffin High School     84.013699
Hernandez High School   80.857143
Holden High School      84.698795
Huang High School       80.305983
Johnson High School     81.227564
Pena High School        84.591160
Rodriguez High School   80.376426
Shelton High School     82.781671
Thomas High School      83.831361
Wilson High School      84.317673
Wright High School      84.073171
Name: reading_score, dtype: float64
```

## 1.5 Scores by School Spending

- Create a table that breaks down school performances based on average Spending Ranges (Per Student). Use 4 reasonable bins to group school spending. Include in the table each of the following:
  - Average Math Score
  - Average Reading Score
  - % Passing Math
  - % Passing Reading
  - Overall Passing Rate (Average of the above two)

```
[34]: # Sample bins. Feel free to create your own bins.
spending_bins = [0, 585, 615, 645, 675]
group_names = ["<$585", "$585-615", "$615-645", "$645-675"]

# Create a new data frame by locating the desired columns
scores_school_spending = school_summary.loc[:, ['Average Math Score',
                                                'Average Reading Score', '% Passing Math',
                                                '% Passing Reading', '% Overall Passing
↳Rate',]]
```

```
# Create two new columns to show budget per student and to define the spending
    ↳ ranges per student
scores_school_spending['Students Spending Ranges'] = pd.cut(school_summary['Per_
    ↳ Student Budget'], spending_bins, labels=group_names)
# Create a group based off of the bins
scores_school_spending = scores_school_spending.groupby('Students Spending_
    ↳ Ranges').mean()
scores_school_spending.head()
```

```
[34]:
```

Students Spending Ranges	Average Math Score	Average Reading Score \
<\$585	83.455399	83.933814
\$585-615	83.599686	83.885211
\$615-645	79.079225	81.891436
\$645-675	76.997210	81.027843

  

Students Spending Ranges	% Passing Math	% Passing Reading \
<\$585	93.460096	96.610877
\$585-615	94.230858	95.900287
\$615-645	75.668212	86.106569
\$645-675	66.164813	81.133951

  

Students Spending Ranges	% Overall Passing Rate
<\$585	95.035486
\$585-615	95.065572
\$615-645	80.887391
\$645-675	73.649382

```
[35]: # Calculate the average math score within each spending range
math_score_spending = scores_school_spending[['Average Math Score',]]
math_score_spending
```

```
[35]:
```

Students Spending Ranges	Average Math Score
<\$585	83.455399
\$585-615	83.599686
\$615-645	79.079225
\$645-675	76.997210

```
[36]: # Calculate the percentage passing rate for math in each spending range
math_percentage_spending = scores_school_spending[['% Passing Math',]]
math_percentage_spending
```

```
[36]:
```

Students Spending Ranges	% Passing Math
--------------------------	----------------

<\$585	93.460096
\$585-615	94.230858
\$615-645	75.668212
\$645-675	66.164813

```
[37]: # Calculate the percentage passing rate for reading in each spending range
reading_score_spending = scores_school_spending[['Average Reading Score',]]
reading_score_spending
```

```
[37]:
```

Students Spending Ranges	Average Reading Score
<\$585	83.933814
\$585-615	83.885211
\$615-645	81.891436
\$645-675	81.027843

```
[38]: # Calculate the percentage overall passing rate in each spending range
reading_percentage_spending = scores_school_spending[['% Passing Reading',]]
reading_percentage_spending
```

```
[38]:
```

Students Spending Ranges	% Passing Reading
<\$585	96.610877
\$585-615	95.900287
\$615-645	86.106569
\$645-675	81.133951

### 1.5.1 Scores by School Size

- Perform the same operations as above, based on school size.

```
[39]: # Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

```
[40]: # Create a new column for the bin groups
# Create a new data frame by locating the desired columns
scores_school_size = school_summary.loc[:,['Average Math Score',
                                           'Average Reading Score','% Passing Math',
                                           '% Passing Reading','% Overall Passing_
↪Rate',]]
# Add a new columns named School Size and binning based off total students
scores_school_size['School Size']= pd.cut(school_summary['Total_
↪Students'],size_bins,labels=group_names)
# Create a group based off of the bins
scores_school_size = scores_school_size.groupby('School Size').mean()
scores_school_size.head()
```

```
[40]:
```

	Average Math Score	Average Reading Score	% Passing Math \
School Size			
Small (<1000)	83.821598	83.929843	93.550225
Medium (1000-2000)	83.374684	83.864438	93.599695
Large (2000-5000)	77.746417	81.344493	69.963361

	% Passing Reading	% Overall Passing Rate
School Size		
Small (<1000)	96.099437	94.824831
Medium (1000-2000)	96.790680	95.195187
Large (2000-5000)	82.766634	76.364998

Look for the total count of test scores that pass 70% or higher

```
[41]: # math_pass_size
math_pass_size = scores_school_size[['% Passing Math',]]
math_pass_size
```

```
[41]:
```

	% Passing Math
School Size	
Small (<1000)	93.550225
Medium (1000-2000)	93.599695
Large (2000-5000)	69.963361

```
[42]: # read_pass_size
read_pass_size = scores_school_size[['% Passing Reading',]]
read_pass_size
```

```
[42]:
```

	% Passing Reading
School Size	
Small (<1000)	96.099437
Medium (1000-2000)	96.790680
Large (2000-5000)	82.766634

```
[43]: # Calculate the overall passing rate for different school size
overall_pass_size = scores_school_size[['% Overall Passing Rate']]
overall_pass_size
```

```
[43]:
```

	% Overall Passing Rate
School Size	
Small (<1000)	94.824831
Medium (1000-2000)	95.195187
Large (2000-5000)	76.364998

### 1.5.2 Scores by School Type

- Perform the same operations as above, based on school type.

```
[44]: # Create bins and groups, school type {'Charter', 'District'}
# Create a new data frame with our desired columns
scores_schools_type = school_summary[['School Type', 'Average Math Score',
                                     'Average Reading Score', '% Passing Math',
                                     '% Passing Reading', '% Overall Passing_
                                     ↪Rate'],]
# Create a group based off of the school type
scores_schools_type = scores_schools_type.groupby('School Type').mean()
scores_schools_type.head()
```

```
[44]:
```

	Average Math Score	Average Reading Score	% Passing Math	\
School Type				
Charter	83.473852	83.896421	93.620830	
District	76.956733	80.966636	66.548453	

  

	% Passing Reading	% Overall Passing Rate
School Type		
Charter	96.586489	95.103660
District	80.799062	73.673757

Find counts of the passing 70 or higher score for the both test

```
[45]: # math pass size
math_pass_size_type = school_summary[school_summary['% Passing Math']>=70].
    ↪groupby('School Type')['% Passing Math'].count()
math_pass_size_type
```

```
[45]: School Type
Charter      8
Name: % Passing Math, dtype: int64
```

```
[46]: # reading pass size
reading_pass_size_type = school_summary[school_summary['% Passing_
    ↪Reading']>=70].groupby('School Type')['% Passing Reading'].count()
reading_pass_size_type
```

```
[46]: School Type
Charter      8
District     7
Name: % Passing Reading, dtype: int64
```

```
[47]: # Calculate the overall passing rate
overall_pass_type = scores_schools_type[['% Overall Passing Rate']]
overall_pass_type
```

```
[47]:
```

	% Overall Passing Rate
School Type	
Charter	95.103660

District

73.673757