

# PyCity Schools Analysis

- As a whole, schools with higher budgets, did not yield better test results. By contrast, schools with higher spending 645-675 per student actually 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%).
- 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.

```
In [1]: # Dependencies and Setup
import pandas as pd
from pathlib import Path

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

# Read School and Student Data File and store into Pandas DataFrames
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_
school_data_complete.head()
```

Out[1]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score	School ID
0	0	Paul Bradley	M	9th	Huang High School	66	79	(
1	1	Victor Smith	M	12th	Huang High School	94	61	(
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	(
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	(
4	4	Bonnie Ray	F	9th	Huang High School	97	84	(

## District Summary

```
In [2]: # Calculate the total number of unique schools
school_count = len(school_data_complete["school_name"].unique())
school_count
```

Out[2]: 15

```
In [3]: # Calculate the total number of students
student_count = len(school_data_complete["Student ID"])
student_count
```

Out[3]: 39170

```
In [4]: # Calculate the total budget
total_budget = (school_data["budget"]).sum()
total_budget
```

Out[4]: 24649428

```
In [5]: # Calculate the average (mean) math score
average_math_score = (school_data_complete["math_score"]).mean()
average_math_score
```

Out[5]: 78.98537145774827

```
In [6]: # Calculate the average (mean) reading score
average_reading_score = (school_data_complete["reading_score"]).mean()
average_reading_score
```

Out[6]: 81.87784018381414

```
In [7]: # Use the following to calculate the percentage of students who passed math (math s
passing_math_count = school_data_complete[(school_data_complete["math_score"] >= 70)
passing_math_percentage = passing_math_count / float(student_count) * 100
passing_math_percentage
```

Out[7]: 74.9808526933878

```
In [8]: # Calculate the percentage of students who passed reading (hint: Look at how the m
passing_reading_count = school_data_complete[(school_data_complete["reading_score"]
passing_reading_percentage = passing_reading_count / float(student_count) * 100
passing_reading_percentage
```

Out[8]: 85.80546336482001

```
In [9]: # Use the following to calculate the percentage of students that passed math and re
passing_math_reading_count = school_data_complete[
    (school_data_complete["math_score"] >= 70) & (school_data_complete["reading_sco
].count()["student_name"]
```

```
overall_passing_rate = passing_math_reading_count / float(student_count) * 100
overall_passing_rate
```

Out[9]: 65.17232575950983

```
In [10]: # Create a high-level snapshot of the district's key metrics in a DataFrame
district_summary = pd.DataFrame ({ "Total number of unique schools" : [school_count],
                                   "Total students" : [student_count],
                                   "Total budget" : [total_budget],
                                   "Average math score" : [average_math_score],
                                   "Average reading score" : [average_reading_score],
                                   "% passing math" : [passing_math_percentage],
                                   "% passing reading" : [passing_reading_percentage],
                                   "% overall passing" : [overall_passing_rate]})

# BackUp & Copy
district_summary_copy = district_summary.copy()

# Formatting
district_summary_copy["Total students"] = district_summary_copy["Total students"].m
district_summary_copy["Total budget"] = district_summary_copy["Total budget"].map("
district_summary_copy["Average math score"] = district_summary_copy["Average math s
district_summary_copy["Average reading score"] = district_summary_copy["Average rea
district_summary_copy["% passing math"] = district_summary_copy["% passing math"].m
district_summary_copy["% passing reading"] = district_summary_copy["% passing readi
district_summary_copy["% overall passing"] = district_summary_copy["% overall passi

# Display the DataFrame
district_summary_copy
```

Out[10]:

	Total number of unique schools	Total students	Total budget	Average math score	Average reading score	% passing math	% passing reading	% overall passing
0	15	39,170	\$24,649,428.00	78.985371	81.877840	74.981%	85.805%	65.172%

## School Summary

```
In [11]: # Use the code provided to select the school type
school_types = school_data.set_index(["school_name"])["type"]
school_types
```

```
Out[11]: school_name
Huang High School      District
Figueroa High School   District
Shelton High School    Charter
Hernandez High School  District
Griffin High School    Charter
Wilson High School     Charter
Cabrera High School    Charter
Bailey High School     District
Holden High School     Charter
Pena High School       Charter
Wright High School     Charter
Rodriguez High School  District
Johnson High School   District
Ford High School       District
Thomas High School     Charter
Name: type, dtype: object
```

```
In [12]: # Calculate the total student count
per_school_counts = school_data_complete["school_name"].value_counts()
per_school_counts
```

```
Out[12]: school_name
Bailey High School      4976
Johnson High School    4761
Hernandez High School   4635
Rodriguez High School   3999
Figueroa High School    2949
Huang High School       2917
Ford High School        2739
Wilson High School      2283
Cabrera High School     1858
Wright High School      1800
Shelton High School     1761
Thomas High School      1635
Griffin High School     1468
Pena High School        962
Holden High School      427
Name: count, dtype: int64
```

```
In [13]: # Calculate the total school budget and per capita spending
per_school_budget = school_data_complete.groupby(["school_name"])["budget"].mean()
per_school_capita = per_school_budget / per_school_counts
per_school_capita
```

```
Out[13]: school_name
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
```

```
In [14]: # Calculate the average test scores
per_school_math = school_data_complete.groupby(["school_name"])["math_score"].mean()
per_school_reading = school_data_complete.groupby(["school_name"])["reading_score"]
```

```
In [15]: #Print to Ensure Accurate Reading (Math)
print(per_school_math)
```

```
school_name
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
```

```
In [16]: #Print to Ensure Accurate Reading (Reading)
print(per_school_reading)
```

```

school_name
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

```

```

In [17]: # Calculate the number of schools with math scores of 70 or higher
school_passing_math = school_data_complete[school_data_complete["math_score"]>=70]

```

```

In [18]: #Print to Ensure Accurate Reading (Math)
print(school_passing_math)

```

	Student ID	student_name	gender	grade	school_name \
0	0	Paul Bradley	M	9th	Huang High School
4	4	Bonnie Ray	F	9th	Huang High School
5	5	Bryan Miranda	M	9th	Huang High School
6	6	Sheena Carter	F	11th	Huang High School
8	8	Michael Roth	M	10th	Huang High School
...	...	...	...	...	...
39165	39165	Donna Howard	F	12th	Thomas High School
39166	39166	Dawn Bell	F	10th	Thomas High School
39167	39167	Rebecca Tanner	F	9th	Thomas High School
39168	39168	Desiree Kidd	F	10th	Thomas High School
39169	39169	Carolyn Jackson	F	11th	Thomas High School

  

	reading_score	math_score	School ID	type	size	budget
0	66	79	0	District	2917	1910635
4	97	84	0	District	2917	1910635
5	94	94	0	District	2917	1910635
6	82	80	0	District	2917	1910635
8	95	87	0	District	2917	1910635
...	...	...	...	...	...	...
39165	99	90	14	Charter	1635	1043130
39166	95	70	14	Charter	1635	1043130
39167	73	84	14	Charter	1635	1043130
39168	99	90	14	Charter	1635	1043130
39169	95	75	14	Charter	1635	1043130

[29370 rows x 11 columns]

```

In [19]: # Calculate the number of schools with reading scores of 70 or higher
school_passing_reading = school_data_complete[school_data_complete['reading_score']

```

```
In [20]: #Print to Ensure Accurate Reading (Reading)
print(school_passing_reading)
```

	Student ID	student_name	gender	grade	school_name	\
1	1	Victor Smith	M	12th	Huang High School	
2	2	Kevin Rodriguez	M	12th	Huang High School	
4	4	Bonnie Ray	F	9th	Huang High School	
5	5	Bryan Miranda	M	9th	Huang High School	
6	6	Sheena Carter	F	11th	Huang High School	
...	...	...	...	...	...	
39165	39165	Donna Howard	F	12th	Thomas High School	
39166	39166	Dawn Bell	F	10th	Thomas High School	
39167	39167	Rebecca Tanner	F	9th	Thomas High School	
39168	39168	Desiree Kidd	F	10th	Thomas High School	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	

  

	reading_score	math_score	School ID	type	size	budget
1	94	61	0	District	2917	1910635
2	90	60	0	District	2917	1910635
4	97	84	0	District	2917	1910635
5	94	94	0	District	2917	1910635
6	82	80	0	District	2917	1910635
...	...	...	...	...	...	...
39165	99	90	14	Charter	1635	1043130
39166	95	70	14	Charter	1635	1043130
39167	73	84	14	Charter	1635	1043130
39168	99	90	14	Charter	1635	1043130
39169	95	75	14	Charter	1635	1043130

[33610 rows x 11 columns]

```
In [21]: # Use the provided code to calculate the schools that passed both math and reading
passing_math_and_reading = school_data_complete[
    (school_data_complete["reading_score"] >= 70) & (school_data_complete["math_sco
    ]
```

```
In [22]: #Print to Ensure Accurate Data
print(passing_math_and_reading)
```

	Student ID	student_name	gender	grade	school_name	\
4	4	Bonnie Ray	F	9th	Huang High School	
5	5	Bryan Miranda	M	9th	Huang High School	
6	6	Sheena Carter	F	11th	Huang High School	
8	8	Michael Roth	M	10th	Huang High School	
9	9	Matthew Greene	M	10th	Huang High School	
...	...	...	...	...	...	
39165	39165	Donna Howard	F	12th	Thomas High School	
39166	39166	Dawn Bell	F	10th	Thomas High School	
39167	39167	Rebecca Tanner	F	9th	Thomas High School	
39168	39168	Desiree Kidd	F	10th	Thomas High School	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	

	reading_score	math_score	School ID	type	size	budget
4	97	84	0	District	2917	1910635
5	94	94	0	District	2917	1910635
6	82	80	0	District	2917	1910635
8	95	87	0	District	2917	1910635
9	96	84	0	District	2917	1910635
...	...	...	...	...	...	...
39165	99	90	14	Charter	1635	1043130
39166	95	70	14	Charter	1635	1043130
39167	73	84	14	Charter	1635	1043130
39168	99	90	14	Charter	1635	1043130
39169	95	75	14	Charter	1635	1043130

[25528 rows x 11 columns]

```
In [23]: # Use the provided code to calculate the passing rates
per_school_passing_math = school_passing_math.groupby(["school_name"]).count()["stu
per_school_passing_reading = school_passing_reading.groupby(["school_name"]).count(
overall_passing_rate = passing_math_and_reading.groupby(["school_name"]).count()["s
```

```
In [24]: #Print to Ensure Accurate Data
print(per_school_passing_math)
print(per_school_passing_reading)
print(overall_passing_rate)
```



```

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

```

```

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

```

```

school_name
Bailey High School      54.642283
Cabrera High School     91.334769
Figueroa High School    53.204476
Ford High School        54.289887
Griffin High School     90.599455
Hernandez High School   53.527508
Holden High School      89.227166
Huang High School       53.513884
Johnson High School     53.539172
Pena High School        90.540541
Rodriguez High School   52.988247
Shelton High School     89.892107
Thomas High School      90.948012
Wilson High School      90.582567
Wright High School      90.333333
dtype: float64

```

```

In [25]: # Create a DataFrame called `per_school_summary` with columns for the calculations
per_school_summary = pd.DataFrame ({"School Type" : school_types,
                                     "Total Students" : per_school_counts,
                                     "Total School Budget" : per_school_budget,

```

```
        "Per Student Budget" : per_school_capita,  
        "Average Math Score" : per_school_math,  
        "Average Reading Score" : per_school_reading,  
        "% Passing Math" : per_school_passing_math,  
        "% Passing Reading" : per_school_passing_reading,  
        "% Overall Passing" : overall_passing_rate})  
  
# Backup & Copy  
per_school_summary_copy = per_school_summary.copy()  
  
# Formatting  
per_school_summary_copy["Total School Budget"] = per_school_summary_copy["Total Sch  
per_school_summary_copy["Per Student Budget"] = per_school_summary_copy["Per Studen  
per_school_summary_copy["Average Math Score"] = per_school_summary_copy["Average Ma  
per_school_summary_copy["Average Reading Score"] = per_school_summary_copy["Average  
per_school_summary_copy["% Passing Math"] = per_school_summary_copy["% Passing Math  
per_school_summary_copy["% Passing Reading"] = per_school_summary_copy["% Passing R  
per_school_summary_copy["% Overall Passing"] = per_school_summary_copy["% Overall P  
  
# Display the DataFrame  
per_school_summary_copy
```

Out[25]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
school_name								
Bailey High School	District	4976	\$3,124,928.00	\$628.00	77.048432	81.033963	66.680%	81.033963
Cabrera High School	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.975780	94.133%	97.000000
Figueroa High School	District	2949	\$1,884,411.00	\$639.00	76.711767	81.158020	65.988%	80.000000
Ford High School	District	2739	\$1,763,916.00	\$644.00	77.102592	80.746258	68.310%	79.000000
Griffin High School	Charter	1468	\$917,500.00	\$625.00	83.351499	83.816757	93.392%	97.000000
Hernandez High School	District	4635	\$3,022,020.00	\$652.00	77.289752	80.934412	66.753%	80.000000
Holden High School	Charter	427	\$248,087.00	\$581.00	83.803279	83.814988	92.506%	96.000000
Huang High School	District	2917	\$1,910,635.00	\$655.00	76.629414	81.182722	65.684%	81.000000
Johnson High School	District	4761	\$3,094,650.00	\$650.00	77.072464	80.966394	66.058%	81.000000
Pena High School	Charter	962	\$585,858.00	\$609.00	83.839917	84.044699	94.595%	95.000000
Rodriguez High School	District	3999	\$2,547,363.00	\$637.00	76.842711	80.744686	66.367%	80.000000
Shelton High School	Charter	1761	\$1,056,600.00	\$600.00	83.359455	83.725724	93.867%	95.000000
Thomas High School	Charter	1635	\$1,043,130.00	\$638.00	83.418349	83.848930	93.272%	97.000000
Wilson High School	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.989488	93.868%	96.000000
Wright High School	Charter	1800	\$1,049,400.00	\$583.00	83.682222	83.955000	93.333%	96.000000

## Highest-Performing Schools (by % Overall Passing)

```
In [26]: # Sort the schools by `% Overall Passing` in descending order and display the top 5
top_schools = per_school_summary.sort_values("% Overall Passing", ascending = False)
```

```
top_schools.head(5)
```

Out[26]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Passing Rate
school_name								
Cabrera High School	Charter	1858	1081356.0	582.0	83.061895	83.975780	94.133477	97.03
Thomas High School	Charter	1635	1043130.0	638.0	83.418349	83.848930	93.272171	97.30
Griffin High School	Charter	1468	917500.0	625.0	83.351499	83.816757	93.392371	97.13
Wilson High School	Charter	2283	1319574.0	578.0	83.274201	83.989488	93.867718	96.53
Pena High School	Charter	962	585858.0	609.0	83.839917	84.044699	94.594595	95.94

## Bottom Performing Schools (By % Overall Passing)

In [27]:

```
# Sort the schools by `% Overall Passing` in ascending order and display the top 5
bottom_schools = per_school_summary.sort_values("% Overall Passing", ascending = True)
bottom_schools.head(5)
```

Out[27]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Passing Rate
school_name								
Rodriguez High School	District	3999	2547363.0	637.0	76.842711	80.744686	66.366592	80.22
Figueroa High School	District	2949	1884411.0	639.0	76.711767	81.158020	65.988471	80.73
Huang High School	District	2917	1910635.0	655.0	76.629414	81.182722	65.683922	81.31
Hernandez High School	District	4635	3022020.0	652.0	77.289752	80.934412	66.752967	80.86
Johnson High School	District	4761	3094650.0	650.0	77.072464	80.966394	66.057551	81.22

## Math Scores by Grade

```
In [28]: # Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] == "11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] == "12th")]
# Group by "school_name" and take the mean of each & Add only Math Score
ninth_graders_math_scores = ninth_graders.groupby(["school_name"])["math_score"].me
tenth_graders_math_scores = tenth_graders.groupby(["school_name"])["math_score"].me
eleventh_graders_math_scores = eleventh_graders.groupby(["school_name"])["math_scor
twelfth_graders_math_scores = twelfth_graders.groupby(["school_name"])["math_score"]

# Combine each of the scores above into single DataFrame called `math_scores_by_grade
math_scores_by_grade = pd.DataFrame ({"9th" : ninth_graders_math_scores,
                                     "10th" : tenth_graders_math_scores,
                                     "11th" : eleventh_graders_math_scores,
                                     "12th" : twelfth_graders_math_scores})

# Minor data wrangling
math_scores_by_grade.index.name = None

# Display the DataFrame
math_scores_by_grade
```

```
Out[28]:
```

	9th	10th	11th	12th
<b>Bailey High School</b>	77.083676	76.996772	77.515588	76.492218
<b>Cabrera High School</b>	83.094697	83.154506	82.765560	83.277487
<b>Figueroa High School</b>	76.403037	76.539974	76.884344	77.151369
<b>Ford High School</b>	77.361345	77.672316	76.918058	76.179963
<b>Griffin High School</b>	82.044010	84.229064	83.842105	83.356164
<b>Hernandez High School</b>	77.438495	77.337408	77.136029	77.186567
<b>Holden High School</b>	83.787402	83.429825	85.000000	82.855422
<b>Huang High School</b>	77.027251	75.908735	76.446602	77.225641
<b>Johnson High School</b>	77.187857	76.691117	77.491653	76.863248
<b>Pena High School</b>	83.625455	83.372000	84.328125	84.121547
<b>Rodriguez High School</b>	76.859966	76.612500	76.395626	77.690748
<b>Shelton High School</b>	83.420755	82.917411	83.383495	83.778976
<b>Thomas High School</b>	83.590022	83.087886	83.498795	83.497041
<b>Wilson High School</b>	83.085578	83.724422	83.195326	83.035794
<b>Wright High School</b>	83.264706	84.010288	83.836782	83.644986

## Reading Score by Grade

```
In [29]: # Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] == "11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] == "12th")]
# Group by "school_name" and take the mean of each & Add only Math Score
ninth_graders_reading_scores = ninth_graders.groupby(["school_name"])["reading_score"].mean()
tenth_graders_reading_scores = tenth_graders.groupby(["school_name"])["reading_score"].mean()
eleventh_graders_reading_scores = eleventh_graders.groupby(["school_name"])["reading_score"].mean()
twelfth_graders_reading_scores = twelfth_graders.groupby(["school_name"])["reading_score"].mean()

# Combine each of the scores above into single DataFrame called `math_scores_by_grade`
reading_scores_by_grade = pd.DataFrame({"9th" : ninth_graders_reading_scores,
                                         "10th" : tenth_graders_reading_scores,
                                         "11th" : eleventh_graders_reading_scores,
                                         "12th" : twelfth_graders_reading_scores})

# Minor data wrangling
reading_scores_by_grade = reading_scores_by_grade[["9th", "10th", "11th", "12th"]]
reading_scores_by_grade.index.name = None

# Display the DataFrame
reading_scores_by_grade
```

```
Out[29]:
```

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

## Scores by School Spending

```
In [30]: # Establish the bins
        spending_bins = [0, 585, 630, 645, 680]
        spending_labels = ["<$585", "$585-630", "$630-645", "$645-680"]
        spending_bins
```

```
Out[30]: [0, 585, 630, 645, 680]
```

```
In [31]: # Create a copy of the school summary since it has the "Per Student Budget"
        school_spending_df = per_school_summary.copy()
        school_spending_df
```

Out[31]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Pa: Rea
school_name								
Bailey High School	District	4976	3124928.0	628.0	77.048432	81.033963	66.680064	81.93
Cabrera High School	Charter	1858	1081356.0	582.0	83.061895	83.975780	94.133477	97.03
Figueroa High School	District	2949	1884411.0	639.0	76.711767	81.158020	65.988471	80.73
Ford High School	District	2739	1763916.0	644.0	77.102592	80.746258	68.309602	79.29
Griffin High School	Charter	1468	917500.0	625.0	83.351499	83.816757	93.392371	97.13
Hernandez High School	District	4635	3022020.0	652.0	77.289752	80.934412	66.752967	80.86
Holden High School	Charter	427	248087.0	581.0	83.803279	83.814988	92.505855	96.25
Huang High School	District	2917	1910635.0	655.0	76.629414	81.182722	65.683922	81.31
Johnson High School	District	4761	3094650.0	650.0	77.072464	80.966394	66.057551	81.22
Pena High School	Charter	962	585858.0	609.0	83.839917	84.044699	94.594595	95.94
Rodriguez High School	District	3999	2547363.0	637.0	76.842711	80.744686	66.366592	80.22
Shelton High School	Charter	1761	1056600.0	600.0	83.359455	83.725724	93.867121	95.85
Thomas High School	Charter	1635	1043130.0	638.0	83.418349	83.848930	93.272171	97.30
Wilson High School	Charter	2283	1319574.0	578.0	83.274201	83.989488	93.867718	96.53
Wright High School	Charter	1800	1049400.0	583.0	83.682222	83.955000	93.333333	96.61

```
In [32]: # Use `pd.cut` to categorize spending based on the bins.
school_spending_df["Spending Ranges (Per Student)"] = pd.cut(school_spending_df["Per
                                                                spending_bins, labels=
school_spending_df
```



Out[32]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Passing Math
school_name								
Bailey High School	District	4976	3124928.0	628.0	77.048432	81.033963	66.680064	81.93
Cabrera High School	Charter	1858	1081356.0	582.0	83.061895	83.975780	94.133477	97.03
Figueroa High School	District	2949	1884411.0	639.0	76.711767	81.158020	65.988471	80.73
Ford High School	District	2739	1763916.0	644.0	77.102592	80.746258	68.309602	79.29
Griffin High School	Charter	1468	917500.0	625.0	83.351499	83.816757	93.392371	97.13
Hernandez High School	District	4635	3022020.0	652.0	77.289752	80.934412	66.752967	80.86
Holden High School	Charter	427	248087.0	581.0	83.803279	83.814988	92.505855	96.25
Huang High School	District	2917	1910635.0	655.0	76.629414	81.182722	65.683922	81.31
Johnson High School	District	4761	3094650.0	650.0	77.072464	80.966394	66.057551	81.22
Pena High School	Charter	962	585858.0	609.0	83.839917	84.044699	94.594595	95.94
Rodriguez High School	District	3999	2547363.0	637.0	76.842711	80.744686	66.366592	80.22
Shelton High School	Charter	1761	1056600.0	600.0	83.359455	83.725724	93.867121	95.85
Thomas High School	Charter	1635	1043130.0	638.0	83.418349	83.848930	93.272171	97.30
Wilson High School	Charter	2283	1319574.0	578.0	83.274201	83.989488	93.867718	96.53
Wright High School	Charter	1800	1049400.0	583.0	83.682222	83.955000	93.333333	96.61

```
In [33]: # Calculate averages for the desired columns.
spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])
spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])
spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per Student)"])
```

```

spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per Student)"])["Average Reading Score"].mean()
overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per Student)"])["% Overall Passing"].mean()

```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\4024963217.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])["Average Math Score"].mean()

```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\4024963217.py:3: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])["Average Reading Score"].mean()

```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\4024963217.py:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per Student)"])["% Passing Math"].mean()

```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\4024963217.py:5: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per Student)"])["% Passing Reading"].mean()

```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\4024963217.py:6: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per Student)"])["% Overall Passing"].mean()

```

In [34]: `# Display Data`  
`spending_math_scores`

Out[34]: Spending Ranges (Per Student)

<\$585	83.455399
\$585-630	81.899826
\$630-645	78.518855
\$645-680	76.997210

Name: Average Math Score, dtype: float64

In [35]: `# Display Data`  
`spending_reading_scores`

Out[35]: Spending Ranges (Per Student)

<\$585	83.933814
\$585-630	83.155286
\$630-645	81.624473
\$645-680	81.027843

Name: Average Reading Score, dtype: float64

In [36]: `# Display Data`  
`spending_passing_math`

```
Out[36]: Spending Ranges (Per Student)
<$585      93.460096
$585-630   87.133538
$630-645   73.484209
$645-680   66.164813
Name: % Passing Math, dtype: float64
```

```
In [37]: # Display Data
spending_passing_reading
```

```
Out[37]: Spending Ranges (Per Student)
<$585      96.610877
$585-630   92.718205
$630-645   84.391793
$645-680   81.133951
Name: % Passing Reading, dtype: float64
```

```
In [38]: # Display Data
overall_passing_spending
```

```
Out[38]: Spending Ranges (Per Student)
<$585      90.369459
$585-630   81.418596
$630-645   62.857656
$645-680   53.526855
Name: % Overall Passing, dtype: float64
```

```
In [39]: # Assemble into DataFrame
spending_summary = reading_scores_by_grade = pd.DataFrame ({
    "Average Math Score" :
    "Average Reading Score"
    "% Passing Math" : spen
    "% Passing Reading" : s
    "% Overall Passing" : o

# Display results
spending_summary
```

Out[39]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Spending Ranges (Per Student)					
<\$585	83.455399	83.933814	93.460096	96.610877	90.369459
\$585-630	81.899826	83.155286	87.133538	92.718205	81.418596
\$630-645	78.518855	81.624473	73.484209	84.391793	62.857656
\$645-680	76.997210	81.027843	66.164813	81.133951	53.526855

# Scores by School Size

```
In [40]: # Establish the bins.
size_bins = [0, 1000, 2000, 5000]
size_labels = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

In [41]: # Categorize the spending based on the bins
# Use `pd.cut` on the "Total Students" column of the `per_school_summary` DataFrame
school_size_df = per_school_summary.copy()
per_school_summary["School Size"] = pd.cut(school_size_df["Total Students"], size_b
per_school_summary
```

Out[41]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Passing Rate
school_name								
<b>Bailey High School</b>	District	4976	3124928.0	628.0	77.048432	81.033963	66.680064	81.93
<b>Cabrera High School</b>	Charter	1858	1081356.0	582.0	83.061895	83.975780	94.133477	97.03
<b>Figueroa High School</b>	District	2949	1884411.0	639.0	76.711767	81.158020	65.988471	80.73
<b>Ford High School</b>	District	2739	1763916.0	644.0	77.102592	80.746258	68.309602	79.29
<b>Griffin High School</b>	Charter	1468	917500.0	625.0	83.351499	83.816757	93.392371	97.13
<b>Hernandez High School</b>	District	4635	3022020.0	652.0	77.289752	80.934412	66.752967	80.86
<b>Holden High School</b>	Charter	427	248087.0	581.0	83.803279	83.814988	92.505855	96.25
<b>Huang High School</b>	District	2917	1910635.0	655.0	76.629414	81.182722	65.683922	81.31
<b>Johnson High School</b>	District	4761	3094650.0	650.0	77.072464	80.966394	66.057551	81.22
<b>Pena High School</b>	Charter	962	585858.0	609.0	83.839917	84.044699	94.594595	95.94
<b>Rodriguez High School</b>	District	3999	2547363.0	637.0	76.842711	80.744686	66.366592	80.22
<b>Shelton High School</b>	Charter	1761	1056600.0	600.0	83.359455	83.725724	93.867121	95.85
<b>Thomas High School</b>	Charter	1635	1043130.0	638.0	83.418349	83.848930	93.272171	97.30
<b>Wilson High School</b>	Charter	2283	1319574.0	578.0	83.274201	83.989488	93.867718	96.53

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	Passing Reading
<b>school_name</b>								
<b>Wright High School</b>	Charter	1800	1049400.0	583.0	83.682222	83.955000	93.333333	96.61

```
In [42]: # Calculate averages for the desired columns.
size_math_scores = per_school_summary.groupby(["School Size"])["Average Math Score"].mean()
size_reading_scores = per_school_summary.groupby(["School Size"])["Average Reading Score"].mean()
size_passing_math = per_school_summary.groupby(["School Size"])["% Passing Math"].mean()
size_passing_reading = per_school_summary.groupby(["School Size"])["% Passing Reading"].mean()
size_overall_passing = per_school_summary.groupby(["School Size"])["% Overall Passing"].mean()
```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\2584247256.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_math_scores = per_school_summary.groupby(["School Size"])["Average Math Score"].mean()
```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\2584247256.py:3: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_reading_scores = per_school_summary.groupby(["School Size"])["Average Reading Score"].mean()
```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\2584247256.py:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_passing_math = per_school_summary.groupby(["School Size"])["% Passing Math"].mean()
```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\2584247256.py:5: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_passing_reading = per_school_summary.groupby(["School Size"])["% Passing Reading"].mean()
```

C:\Users\wes05\AppData\Local\Temp\ipykernel\_11772\2584247256.py:6: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_overall_passing = per_school_summary.groupby(["School Size"])["% Overall Passing"].mean()
```

```
In [43]: # Display Data
size_math_scores
```

```
Out[43]: School Size
Small (<1000)      83.821598
Medium (1000-2000) 83.374684
Large (2000-5000)  77.746417
Name: Average Math Score, dtype: float64
```

```
In [44]: # Display Data
size_reading_scores
```

```
Out[44]: School Size
Small (<1000)      83.929843
Medium (1000-2000) 83.864438
Large (2000-5000)  81.344493
Name: Average Reading Score, dtype: float64
```

```
In [45]: # Display Data
size_passing_math
```

```
Out[45]: School Size
Small (<1000)      93.550225
Medium (1000-2000) 93.599695
Large (2000-5000)  69.963361
Name: % Passing Math, dtype: float64
```

```
In [46]: # Display Data
size_passing_reading
```

```
Out[46]: School Size
Small (<1000)      96.099437
Medium (1000-2000) 96.790680
Large (2000-5000)  82.766634
Name: % Passing Reading, dtype: float64
```

```
In [47]: # Display Data
size_overall_passing
```

```
Out[47]: School Size
Small (<1000)      89.883853
Medium (1000-2000) 90.621535
Large (2000-5000)  58.286003
Name: % Overall Passing, dtype: float64
```

```
In [48]: # Create a DataFrame called `size_summary` that breaks down school performance base
# Use the scores above to create a new DataFrame called `size_summary`
size_summary = pd.DataFrame ({ "Average Math Score" : size_math_scores,
                                "Average Reading Score" : size_reading_scores,
                                "% Passing Math" : size_passing_math,
                                "% Passing Reading" : size_passing_reading,
                                "% Overall Passing" : size_overall_passing})

# Display results
size_summary
```

Out[48]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
<b>School Size</b>					
<b>Small (&lt;1000)</b>	83.821598	83.929843	93.550225	96.099437	89.883853
<b>Medium (1000-2000)</b>	83.374684	83.864438	93.599695	96.790680	90.621535
<b>Large (2000-5000)</b>	77.746417	81.344493	69.963361	82.766634	58.286003

## Scores by School Type

```
In [49]: # Group the per_school_summary DataFrame by "School Type" and average the results.
type_math_scores = per_school_summary.groupby(["School Type"])["Average Math Score"]
type_reading_scores = per_school_summary.groupby(["School Type"])["Average Reading Score"]
type_passing_math = per_school_summary.groupby(["School Type"])["% Passing Math"]
type_passing_reading = per_school_summary.groupby(["School Type"])["% Passing Reading"]
type_overall_passing = per_school_summary.groupby(["School Type"])["% Overall Passing"]
```

```
In [50]: # Display Data
type_math_scores
```

```
Out[50]: School Type
Charter      83.473852
District     76.956733
Name: Average Math Score, dtype: float64
```

```
In [51]: # Display Data
type_reading_scores
```

```
Out[51]: School Type
Charter      83.896421
District     80.966636
Name: Average Reading Score, dtype: float64
```

```
In [52]: # Display Data
type_passing_math
```

```
Out[52]: School Type
Charter      93.620830
District     66.548453
Name: % Passing Math, dtype: float64
```

```
In [53]: # Display Data
type_passing_reading
```

```
Out[53]: School Type
Charter      96.586489
District     80.799062
Name: % Passing Reading, dtype: float64
```



```
In [54]: # Display Data
type_overall_passing
```

```
Out[54]: School Type
Charter    90.432244
District   53.672208
Name: % Overall Passing, dtype: float64
```

```
In [55]: # Assemble the new data by type into a DataFrame called `type_summary`
type_summary = pd.DataFrame({"Average Math Score" : type_math_scores,
                             "Average Reading Score" : type_reading_scores,
                             "% Passing Math" : type_passing_math,
                             "% Passing Reading" : type_passing_reading,
                             "% Overall Passing" : type_overall_passing})

# Display results
type_summary
```

```
Out[55]:
```

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
School Type					
Charter	83.473852	83.896421	93.620830	96.586489	90.432244
District	76.956733	80.966636	66.548453	80.799062	53.672208

```
In [ ]:
```