

# PyCity Schools Analysis

In the below analysis, data from 15 schools were collected and reviewed. This data took into account the performance of 39,170 students. The data also included the budgets and per student budgets for all 15 schools. With this data I was able to isolate the math, reading, and overall test scores across the school's students and determine if there were any conclusions to be drawn.

After review I identified two clear conclusions. First, the data shows Charter schools scored significantly better in math and reading. Second, I was unable to identify any correlation between school budget and per student budget and higher test scores.

In [40]:

```
import os
os.getcwd()
```

Out[40]:

```
'C:\\\\Users\\\\conne\\\\OneDrive\\\\Desktop\\\\NU-VIRT-DATA-PT-10-2023-U-LOLC\\\\02-Homework\\\\04-Data-Analysis-Pandas'
```

In [188...]

```
# Dependencies and Setup
import pandas as pd
from pathlib import Path

# File to Load (Remember to Change These)
school_data_to_load = Path('C:\\\\Users\\\\conne\\\\OneDrive\\\\Desktop\\\\NU-VIRT-DATA-PT-10-2023-U-LOLC\\\\02-Homework\\\\04-Data-Analysis-Pandas\\\\Schools.csv')
student_data_to_load = Path('C:\\\\Users\\\\conne\\\\OneDrive\\\\Desktop\\\\NU-VIRT-DATA-PT-10-2023-U-LOLC\\\\02-Homework\\\\04-Data-Analysis-Pandas\\\\Students.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_name"])
school_data_complete.head()
```

Out[188]:

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

# District Summary

In [189]:

```
# Calculate the total number of unique schools
school_count = school_data_complete["school_name"].nunique()
school_count
```

Out[189]:

15

In [190]:

```
# Calculate the total number of students
student_count = school_data_complete["Student ID"].nunique()
student_count
```

Out[190]:

39170

In [191]:

```
# Calculate the total budget
total_budget = set(school_data_complete["budget"])

total_budget_2 = pd.Series(list(total_budget))

total_budget_3 = total_budget_2.sum()

print(total_budget_3)
```

24649428

In [192]:

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

Out[192]:

78.98537145774827

In [193]:

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

Out[193]:

81.87784018381414

In [194]:

```
# Use the following to calculate the percentage of students who passed math (math score)
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[194]:

74.9808526933878

In [195]:

```
# Calculate the percentage of students who passed reading (hint: Look at how the math
passing_reading_count = school_data_complete[(school_data_complete["reading_score"] >= 70)]
passing_reading_percentage = passing_reading_count / float(student_count) * 100
passing_reading_percentage
```

Out[195]:

85.80546336482001

In [196]:

```
# Use the following to calculate the percentage of students that passed math and reading
passing_math_reading_count = school_data_complete[
    (school_data_complete["math_score"] >= 70) & (school_data_complete["reading_score"] >= 70)]
```

```
[].count()["student_name"]
overall_passing_rate = passing_math_reading_count / float(student_count) * 100
overall_passing_rate
```

Out[196]: 65.17232575950983

In [197...]

```
# Create a high-level snapshot of the district's key metrics in a DataFrame
district_summary = pd.DataFrame(
    {"Total Schools": [school_count],
     "Total Students": [student_count],
     "Total Budget": [total_budget_3],
     "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]})

# Formatting
district_summary["Total Students"] = district_summary["Total Students"].map("{:,}").format()
district_summary["Total Budget"] = district_summary["Total Budget"].map("${:, .2f}").format()

# Display the DataFrame
district_summary
```

Out[197]:

	Total 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	81.87784	74.980853	85.805463	65.172326

## School Summary

In [198...]

```
# Use the code provided to select all of the school types
school_types = school_data.set_index(["school_name"])["type"]
```

In [376...]

```
# Calculate the total student count per school
per_school_counts = school_data_complete["school_name"].value_counts()
```

In [377...]

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

In [378...]

```
# Calculate the average test scores per school
per_school_math = school_data_complete.groupby(["school_name"])["math_score"].mean()
per_school_math

per_school_reading = school_data_complete.groupby(["school_name"])["math_score"].mean()
per_school_reading
```

```
Out[378]: 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 [379...]

```
# Calculate the number of students per school with math scores of 70 or higher
students_passing_math = school_data_complete.loc[(school_data_complete["math_score"] > 70)]
students_passing_math

school_students_passing_math = students_passing_math["school_name"].value_counts()
school_students_passing_math
```

Out[379]:

```
school_name
Bailey High School    3318
Johnson High School   3145
Hernandez High School 3094
Rodriguez High School 2654
Wilson High School    2143
Figueroa High School  1946
Huang High School    1916
Ford High School     1871
Cabrera High School  1749
Wright High School   1680
Shelton High School  1653
Thomas High School   1525
Griffin High School  1371
Pena High School    910
Holden High School   395
Name: count, dtype: int64
```

In [380...]

```
# Calculate the number of students per school with math scores of 70 or higher
students_passing_reading = school_data_complete.loc[(school_data_complete["reading_score"] > 70)]
students_passing_reading

school_students_passing_reading = students_passing_reading["school_name"].value_counts()
school_students_passing_reading
```

```
Out[380]: school_name
          Bailey High School      4077
          Johnson High School     3867
          Hernandez High School   3748
          Rodriguez High School   3208
          Figueroa High School    2381
          Huang High School       2372
          Wilson High School      2204
          Ford High School        2172
          Cabrera High School     1803
          Wright High School      1739
          Shelton High School     1688
          Thomas High School      1591
          Griffin High School     1426
          Pena High School         923
          Holden High School       411
          Name: count, dtype: int64
```

In [381...]

```
# Use the provided code to calculate the number of students per school that passed both math and reading
students_passing_math_and_reading = school_data_complete[
    (school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"] >= 70)
]
school_students_passing_math_and_reading = students_passing_math_and_reading.groupby([
    "school_name"
]).size().reset_index()
```

Out[381]:

```
school_name
          Bailey High School      2719
          Cabrera High School     1697
          Figueroa High School     1569
          Ford High School        1487
          Griffin High School     1330
          Hernandez High School   2481
          Holden High School      381
          Huang High School       1561
          Johnson High School     2549
          Pena High School         871
          Rodriguez High School   2119
          Shelton High School     1583
          Thomas High School      1487
          Wilson High School      2068
          Wright High School      1626
          dtype: int64
```

In [382...]

```
# Use the provided code to calculate the passing rates
per_school_passing_math = school_students_passing_math / per_school_counts * 100
per_school_passing_reading = school_students_passing_reading / per_school_counts * 100
overall_passing_rate = school_students_passing_math_and_reading / per_school_counts * 100
```

```
Out[382]:
```

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 [383...]

```
# Create a DataFrame called `per_school_summary` with columns for the calculations above
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})

per_school_summary["Total School Budget"] = per_school_summary["Total School Budget"].map("${:,.2f}".format)
per_school_summary["Per Student Budget"] = per_school_summary["Per Student Budget"].map("${:,.2f}".format)

# Display the DataFrame
per_school_summary.head(5)
```

Out[383]:

<b>school_name</b>	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
<b>Bailey High School</b>	District	4976	\$3,124,928.00	\$628.00	77.048432	77.048432	66.680064	81.933280	52.7%
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.061895	94.133477	97.039828	91.1%
<b>Figueroa High School</b>	District	2949	\$1,884,411.00	\$639.00	76.711767	76.711767	65.988471	80.739234	53.3%
<b>Ford High School</b>	District	2739	\$1,763,916.00	\$644.00	77.102592	77.102592	68.309602	79.299014	52.7%
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.351499	93.392371	97.138965	90.0%

# Highest-Performing Schools (by % Overall Passing)

In [384...]

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

Out[384]:

<b>school_name</b>	<b>School Type</b>	<b>Total Students</b>	<b>Total School Budget</b>	<b>Per Student Budget</b>	<b>Average Math Score</b>	<b>Average Reading Score</b>	<b>% Passing Math</b>	<b>% Passing Reading</b>	<b>%</b>
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.061895	94.133477	97.039828	91
<b>Thomas High School</b>	Charter	1635	\$1,043,130.00	\$638.00	83.418349	83.418349	93.272171	97.308869	90
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.351499	93.392371	97.138965	90
<b>Wilson High School</b>	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.274201	93.867718	96.539641	90
<b>Pena High School</b>	Charter	962	\$585,858.00	\$609.00	83.839917	83.839917	94.594595	95.945946	90



# Bottom Performing Schools (By % Overall Passing)

In [385...]

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

Out[385]:

School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Math	% Passing Reading
<b>school_name</b>								
Rodriguez High School	District	3999	\$2,547,363.00	\$637.00	76.842711	76.842711	66.366592	80.220055
Figueroa High School	District	2949	\$1,884,411.00	\$639.00	76.711767	76.711767	65.988471	80.739234
Huang High School	District	2917	\$1,910,635.00	\$655.00	76.629414	76.629414	65.683922	81.316421
Hernandez High School	District	4635	\$3,022,020.00	\$652.00	77.289752	77.289752	66.752967	80.862999
Johnson High School	District	4761	\$3,094,650.00	\$650.00	77.072464	77.072464	66.057551	81.222432

◀ ▶

## Math Scores By Grade

In [386...]

```
# 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 the `math_score` column for each.
ninth_grade_math_scores = ninth_graders.groupby(["school_name"])["math_score"].mean()
tenth_grader_math_scores = tenth_graders.groupby(["school_name"])["math_score"].mean()
eleventh_grader_math_scores = eleventh_graders.groupby(["school_name"])["math_score"].mean()
twelfth_grader_math_scores = twelfth_graders.groupby(["school_name"])["math_score"].mean()

# Combine each of the scores above into single DataFrame called `math_scores_by_grade`
math_scores_by_grade = pd.DataFrame({"9th": ninth_grade_math_scores,
                                      "10th": tenth_grader_math_scores,
                                      "11th": eleventh_grader_math_scores,
                                      "12th": twelfth_grader_math_scores})

# Minor data wrangling
math_scores_by_grade.index.name = None

# Display the DataFrame
math_scores_by_grade
```

Out[386]:

	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>Peña 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 [387...]

```
# 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 the `reading_score` column for each.
ninth_grade_reading_scores = ninth_graders.groupby(["school_name"])["reading_score"].mean()
tenth_grader_reading_scores = tenth_graders.groupby(["school_name"])["reading_score"].mean()
eleventh_grader_reading_scores = eleventh_graders.groupby(["school_name"])["reading_score"].mean()
twelfth_grader_reading_scores = twelfth_graders.groupby(["school_name"])["reading_score"].mean()

# Combine each of the scores above into single DataFrame called `reading_scores_by_grade`
reading_scores_by_grade = pd.DataFrame({
    "9th": ninth_grade_reading_scores,
    "10th": tenth_grader_reading_scores,
    "11th": eleventh_grader_reading_scores,
    "12th": twelfth_grader_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[387]:

	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>Peña 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 [388...]

```
# Establish the bins
spending_bins = [0, 585, 630, 645, 680]
labels = ["<$585", "$585-630", "$630-645", "$645-680"]
```

In [389...]

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

In [390...]

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

Out[390]:

<b>school_name</b>	<b>School Type</b>	<b>Total Students</b>	<b>Total School Budget</b>	<b>Per Student Budget</b>	<b>Average Math Score</b>	<b>Average Reading Score</b>	<b>% Passing Math</b>	<b>% Passing Reading</b>	<b>%</b>
<b>Bailey High School</b>	District	4976	\$3,124,928.00	\$628.00	77.048432	77.048432	66.680064	81.933280	54
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.061895	94.133477	97.039828	91
<b>Figueroa High School</b>	District	2949	\$1,884,411.00	\$639.00	76.711767	76.711767	65.988471	80.739234	53
<b>Ford High School</b>	District	2739	\$1,763,916.00	\$644.00	77.102592	77.102592	68.309602	79.299014	54
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.351499	93.392371	97.138965	90
<b>Hernandez High School</b>	District	4635	\$3,022,020.00	\$652.00	77.289752	77.289752	66.752967	80.862999	53
<b>Holden High School</b>	Charter	427	\$248,087.00	\$581.00	83.803279	83.803279	92.505855	96.252927	89
<b>Huang High School</b>	District	2917	\$1,910,635.00	\$655.00	76.629414	76.629414	65.683922	81.316421	53
<b>Johnson High School</b>	District	4761	\$3,094,650.00	\$650.00	77.072464	77.072464	66.057551	81.222432	53
<b>Pena High School</b>	Charter	962	\$585,858.00	\$609.00	83.839917	83.839917	94.594595	95.945946	90
<b>Rodriguez High School</b>	District	3999	\$2,547,363.00	\$637.00	76.842711	76.842711	66.366592	80.220055	52
<b>Shelton High School</b>	Charter	1761	\$1,056,600.00	\$600.00	83.359455	83.359455	93.867121	95.854628	89
<b>Thomas High School</b>	Charter	1635	\$1,043,130.00	\$638.00	83.418349	83.418349	93.272171	97.308869	90
<b>Wilson High School</b>	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.274201	93.867718	96.539641	90
<b>Wright High School</b>	Charter	1800	\$1,049,400.00	\$583.00	83.682222	83.682222	93.333333	96.611111	90

In [ ]:

```
# Calculate averages for the desired columns.
spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])[
    "Spending Math Score"]
spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"])[
    "Spending Reading Score"]
spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per Student)"])[
    "Passing Math (%)"]
spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per Student)"])[
    "Passing Reading (%)"]
overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per Student)"])[
    "Overall Passing (%)"]
```

In [392...]

```
# Assemble into DataFrame
spending_summary = pd.DataFrame({"Average Math Score": spending_math_scores,
                                  "Average Reading Score": spending_reading_scores,
                                  "% Passing Math": spending_passing_math,
                                  "% Passing Reading": spending_passing_reading,
                                  "% Overall Passing": overall_passing_spending})

# Display results
spending_summary
```

Out[392]:

Spending Ranges (Per Student)	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
(0, 585]	83.455399	83.455399	93.460096	96.610877	90.369459
(585, 630]	81.899826	81.899826	87.133538	92.718205	81.418596
(630, 645]	78.518855	78.518855	73.484209	84.391793	62.857656
(645, 680]	76.997210	76.997210	66.164813	81.133951	53.526855

## Scores by School Size

In [393...]

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

In [394...]

```
# Categorize the spending based on the bins
# Use `pd.cut` on the "Total Students" column of the `per_school_summary` DataFrame.

per_school_summary["School Size"] = pd.cut(per_school_counts, size_bins, labels)
per_school_summary
```

Out[394]:

<b>school_name</b>	<b>School Type</b>	<b>Total Students</b>	<b>Total School Budget</b>	<b>Per Student Budget</b>	<b>Average Math Score</b>	<b>Average Reading Score</b>	<b>% Passing Math</b>	<b>% Passing Reading</b>	<b>%</b>
<b>Bailey High School</b>	District	4976	\$3,124,928.00	\$628.00	77.048432	77.048432	66.680064	81.933280	52
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.061895	94.133477	97.039828	91
<b>Figueroa High School</b>	District	2949	\$1,884,411.00	\$639.00	76.711767	76.711767	65.988471	80.739234	53
<b>Ford High School</b>	District	2739	\$1,763,916.00	\$644.00	77.102592	77.102592	68.309602	79.299014	54
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.351499	93.392371	97.138965	90
<b>Hernandez High School</b>	District	4635	\$3,022,020.00	\$652.00	77.289752	77.289752	66.752967	80.862999	53
<b>Holden High School</b>	Charter	427	\$248,087.00	\$581.00	83.803279	83.803279	92.505855	96.252927	89
<b>Huang High School</b>	District	2917	\$1,910,635.00	\$655.00	76.629414	76.629414	65.683922	81.316421	53
<b>Johnson High School</b>	District	4761	\$3,094,650.00	\$650.00	77.072464	77.072464	66.057551	81.222432	53
<b>Pena High School</b>	Charter	962	\$585,858.00	\$609.00	83.839917	83.839917	94.594595	95.945946	90
<b>Rodriguez High School</b>	District	3999	\$2,547,363.00	\$637.00	76.842711	76.842711	66.366592	80.220055	52
<b>Shelton High School</b>	Charter	1761	\$1,056,600.00	\$600.00	83.359455	83.359455	93.867121	95.854628	89
<b>Thomas High School</b>	Charter	1635	\$1,043,130.00	\$638.00	83.418349	83.418349	93.272171	97.308869	90
<b>Wilson High School</b>	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.274201	93.867718	96.539641	90
<b>Wright High School</b>	Charter	1800	\$1,049,400.00	\$583.00	83.682222	83.682222	93.333333	96.611111	90

In [395...]

```
# 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()
```

In [396...]

```
# Create a DataFrame called `size_summary` that breaks down school performance based on size.
# 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})
```

```
"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[396]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
(0, 1000]	83.821598	83.821598	93.550225	96.099437	89.883853
(1000, 2000]	83.374684	83.374684	93.599695	96.790680	90.621535
(2000, 5000]	77.746417	77.746417	69.963361	82.766634	58.286003

## Scores by School Type

In [397...]

```
# Group the per_school_summary DataFrame by "School Type" and average the results.
average_math_score_by_type = per_school_summary.groupby(["School Type"])["Average Math"]
average_reading_score_by_type = per_school_summary.groupby(["School Type"])["Average R
average_percent_passing_math_by_type = per_school_summary.groupby(["School Type"])[%
average_percent_passing_reading_by_type = per_school_summary.groupby(["School Type"])[%
average_percent_overall_passing_by_type = per_school_summary.groupby(["School Type"])[
```

In [398...]

```
# Assemble the new data by type into a DataFrame called `type_summary`
type_summary = pd.DataFrame({
    "Average Math Score": average_math_score_by_type,
    "Average Reading Score": average_reading_score_by_type,
    "% Passing Math": average_percent_passing_math_by_type,
    "% Passing Reading": average_percent_passing_reading_by_ty
    "% Overall Passing": average_percent_overall_passing_by_ty

# Display results
type_summary
```

Out[398]:

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

In [ ]: