

PyCity School Analysis

Data Analyst:

Vidal Garcia Martinez

Summary:

PyCity has 15 high schools with 39,170 students and a budget of \$24,649,428. At least 70% of students passed either their standardized math or reading exams, however, only 65% passed both exams. The schools with highest budgets have the lowest passing scores overall and, on the other hand, the schools with lowest budgets have the highest scores overall.

The top-5 performing institutions are charter schools whereas public district schools see themselves in the bottom 5. Not only that, charter schools out-performed district schools by far, with a 90% overall passing against just 53%, respectively.

Students from all schools performed better in reading than math tests. Despite that, schools with higher budgets per student had inversely proportionate passing scores overall.

In terms of student population, the school size didn't make much difference between small and medium schools but large schools had the poorest scores overall.

Conclusions:

- Big budgets per school and per capita do not guarantee better student performance
 - Charter schools have a clear advantage over district schools, so it would be interesting to know the contributing factors to their success (teaching practices, population size, etc). Remember we are talking about standardized tests.
-

In [249...]

```
# Dependencies and Setup
import pandas as pd

# Files to Load
school_data_to_load = "Resources/schools_complete.csv"
student_data_to_load = "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_name"])
school_data_complete.head()
```

Out[249]:

	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 [250...]

```
# Total number of unique schools
school_count = school_data.school_name.count()
school_count
```

Out[250]:

```
# Total students
student_count = student_data.student_name.count()
"{:,}").format(student_count)
```

Out[251]:

```
# Total budget
total_budget = school_data.budget.sum()
"${:,}").format(total_budget)
```

Out[252]:

```
# Average math score
average_math_score = student_data.math_score.mean()
average_math_score
```

Out[253]:

```
# Average reading score
average_reading_score = student_data.reading_score.mean()
average_reading_score
```

Out[254]:

```
# % passing math (the percentage of students who passed math with 70+)
passing_math_count = school_data_complete[(school_data_complete["math_score"] >= 70)].
```

```

passing_math_percentage = passing_math_count / float(student_count) * 100
str(round(passing_math_percentage)) + '%'

Out[255]: '75 %'

In [256... # % passing reading (the percentage of students who passed reading with 70+)
passing_reading_count = school_data_complete[(school_data_complete["reading_score"] >= 70).count()]
passing_reading_percentage = passing_reading_count / float(student_count) * 100
str(round(passing_reading_percentage)) + '%'

Out[256]: '86 %'

In [259... # % overall passing (the percentage of students who passed math AND reading)
passing_math_reading_count = school_data_complete[
    (school_data_complete["math_score"] >= 70) & (school_data_complete["reading_score"]
].count()["student_name"]
overall_passing_rate = passing_math_reading_count / float(student_count) * 100
str(round(overall_passing_rate)) + '%'

Out[259]: '65 %'

In [260... # 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,
                                  "Average Math Score": round(average_math_score),
                                  "Average Reading Score": round(average_reading_score),
                                  "% Passing Math": passing_math_percentage,
                                  "% Passing Reading": passing_reading_percentage,
                                  "% Overall Passing": overall_passing_rate}])

# Formatting
district_summary["Total Schools"] = district_summary["Total Schools"].map("{:,}").format()
district_summary["Total Students"] = district_summary["Total Students"].map("{:,}").format()
district_summary["Total Budget"] = district_summary["Total Budget"].map("${:,}").format()
district_summary["Average Math Score"] = district_summary["Average Math Score"]
district_summary["Average Reading Score"] = district_summary["Average Reading Score"]
district_summary["% Passing Math"] = district_summary["% Passing Math"].map("{:.0f}").format()
district_summary["% Passing Reading"] = district_summary["% Passing Reading"].map("{:.0f}").format()
district_summary["% Overall Passing"] = district_summary["% Overall Passing"].map("{:.0f}").format()

# Display the DataFrame
district_summary

```

	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	79	82	75	86	65

School Summary

```

In [261... # School Name & Type
school_types = school_data.set_index(["school_name"])["type"]
school_types

```

```
Out[261]: 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 [262...]: # Students per school
per_school_counts = school_data_complete['school_name'].value_counts()
per_school_counts
```

```
Out[262]: 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: school_name, dtype: int64
```

```
In [263...]: # Total School Budget
per_school_budget = school_data_complete.groupby(["school_name"]).mean()["budget"]
per_school_budget.map('${:,.0f}'.format)
```

```
Out[263]:
```

school_name	
Bailey High School	\$3,124,928
Cabrera High School	\$1,081,356
Figueroa High School	\$1,884,411
Ford High School	\$1,763,916
Griffin High School	\$917,500
Hernandez High School	\$3,022,020
Holden High School	\$248,087
Huang High School	\$1,910,635
Johnson High School	\$3,094,650
Pena High School	\$585,858
Rodriguez High School	\$2,547,363
Shelton High School	\$1,056,600
Thomas High School	\$1,043,130
Wilson High School	\$1,319,574
Wright High School	\$1,049,400
Name: budget, dtype: object	

```
In [264...]
```

```
# Budget per student
per_school_capita = per_school_budget / per_school_counts
per_school_capita.map('${:,0f}'.format)
```

```
Out[264]:
```

Bailey High School	\$628
Cabrera High School	\$582
Figueroa High School	\$639
Ford High School	\$644
Griffin High School	\$625
Hernandez High School	\$652
Holden High School	\$581
Huang High School	\$655
Johnson High School	\$650
Pena High School	\$609
Rodriguez High School	\$637
Shelton High School	\$600
Thomas High School	\$638
Wilson High School	\$578
Wright High School	\$583
dtype: object	

```
In [265...]
```

```
# Average Math Scores
per_school_math = school_data_complete.groupby(['school_name']).math_score.mean()
per_school_math
```

```
Out[265]:
```

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 [266...]: # Average Reading Scores  
per_school_reading = school_data_complete.groupby(['school_name']).reading_score.mean()  
per_school_reading
```

```
Out[266]: 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 [267...]: # Schools with math scores of 70 or higher  
school_passing_math = school_data_complete[school_data_complete.math_score >= 70].groupby('school_name').size()  
school_passing_math
```

```
Out[267]: school_name  
Bailey High School      3318  
Cabrera High School    1749  
Figueroa High School   1946  
Ford High School       1871  
Griffin High School    1371  
Hernandez High School  3094  
Holden High School     395  
Huang High School      1916  
Johnson High School    3145  
Pena High School       910  
Rodriguez High School  2654  
Shelton High School    1653  
Thomas High School     1525  
Wilson High School     2143  
Wright High School     1680  
Name: student_name, dtype: int64
```

```
In [268...]: # Schools with reading scores of 70 or higher  
school_passing_reading = school_data_complete[school_data_complete.reading_score >= 70].groupby('school_name').size()  
school_passing_reading
```

```
Out[268]: school_name
Bailey High School      4077
Cabrera High School    1803
Figueroa High School   2381
Ford High School       2172
Griffin High School    1426
Hernandez High School  3748
Holden High School     411
Huang High School      2372
Johnson High School    3867
Pena High School       923
Rodriguez High School  3208
Shelton High School    1688
Thomas High School     1591
Wilson High School     2204
Wright High School     1739
Name: student_name, dtype: int64
```

```
In [269...]: # Schools that passed both math and reading with scores of 70 or higher
passing_math_and_reading = school_data_complete[
    (school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"]
].groupby(['school_name']).count()['student_name']
passing_math_and_reading
```

```
Out[269]: 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
Name: student_name, dtype: int64
```

```
In [270...]: # Use the provided code to calculate the passing rates
per_school_passing_math = school_passing_math / per_school_counts * 100
per_school_passing_math
```

```
Out[270]:
```

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	

```
In [271]: per_school_passing_reading = school_passing_reading / per_school_counts * 100  
per_school_passing_reading
```

```
Out[271]:
```

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
```

```
In [272]: overall_passing_rate = passing_math_and_reading / per_school_counts * 100  
overall_passing_rate
```

```
Out[272]:
```

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
```

```
'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})

#Formatting
per_school_summary["School Type"] = per_school_summary["School Type"]
per_school_summary["Total Students"] = per_school_summary["Total Students"]
per_school_summary["Total School Budget"] = per_school_summary["Total School Budget"].map(
    lambda x: str('${:,}').format(x))
per_school_summary["Average Math Score"] = per_school_summary["Average Math Score"].map(
    lambda x: str('%.1f') % x)
per_school_summary["Average Reading Score"] = per_school_summary["Average Reading Score"].map(
    lambda x: str('%.1f') % x)
per_school_summary["% Passing Math"] = per_school_summary["% Passing Math"].map(
    lambda x: str('{:,}') % x)
per_school_summary["% Passing Reading"] = per_school_summary["% Passing Reading"].map(
    lambda x: str('{:,}') % x)
per_school_summary["% Overall Passing"] = per_school_summary["% Overall Passing"].map(
    lambda x: str('{:,}') % x)

#Display DF
per_school_summary
```

Out[273]:

School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Ove Pass
Bailey High School	District	4976	\$3,124,928	\$628	77.048432	81.033963	66.680064	81.933280
Cabrera High School	Charter	1858	\$1,081,356	\$582	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	\$1,884,411	\$639	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	\$1,763,916	\$644	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	\$917,500	\$625	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	\$3,022,020	\$652	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	\$248,087	\$581	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	\$1,910,635	\$655	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	\$3,094,650	\$650	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	\$585,858	\$609	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	\$2,547,363	\$637	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	\$1,056,600	\$600	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	\$1,043,130	\$638	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	\$1,319,574	\$578	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	\$1,049,400	\$583	83.682222	83.955000	93.333333	96.611111

Highest-Performing Schools (by % Overall Passing)

In [274...]

```
# Top 5 - Overall Passing
top_schools = per_school_summary.sort_values(['% Overall Passing'], ascending = False)
top_schools.head()
```

Out[274]:

School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Cabrera High School	Charter 1858	\$1,081,356	\$582	83.061895	83.975780	94.133477	97.039828	91.334769
Thomas High School	Charter 1635	\$1,043,130	\$638	83.418349	83.848930	93.272171	97.308869	90.948012
Griffin High School	Charter 1468	\$917,500	\$625	83.351499	83.816757	93.392371	97.138965	90.599455
Wilson High School	Charter 2283	\$1,319,574	\$578	83.274201	83.989488	93.867718	96.539641	90.582567
Pena High School	Charter 962	\$585,858	\$609	83.839917	84.044699	94.594595	95.945946	90.540541



Lowest-Performing Schools (by % Overall Passing)

In [275...]

```
# Bottom 5 - Overall Passing
top_schools = per_school_summary.sort_values(['% Overall Passing'])
top_schools.head()
```

Out[275]:

School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Ove Pass
Rodriguez High School	District 3999	\$2,547,363	\$637	76.842711	80.744686	66.366592	80.220055	52.9881
Figueroa High School	District 2949	\$1,884,411	\$639	76.711767	81.158020	65.988471	80.739234	53.2044
Huang High School	District 2917	\$1,910,635	\$655	76.629414	81.182722	65.683922	81.316421	53.5138
Hernandez High School	District 4635	\$3,022,020	\$652	77.289752	80.934412	66.752967	80.862999	53.5271
Johnson High School	District 4761	\$3,094,650	\$650	77.072464	80.966394	66.057551	81.222432	53.539

In [276...]

```
# Separate data by grades
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 get the average of each school
ninth_graders_scores = ninth_graders.groupby(['school_name']).mean()['math_score']
tenth_graders_scores = tenth_graders.groupby(['school_name']).mean()['math_score']
eleventh_graders_scores = eleventh_graders.groupby(['school_name']).mean()['math_score']
twelfth_graders_scores = twelfth_graders.groupby(['school_name']).mean()['math_score']

# Select only the "math_score"
ninth_grade_math_scores = round(ninth_graders_scores)
tenth_grader_math_scores = round(tenth_graders_scores)
eleventh_grader_math_scores = round(eleventh_graders_scores)
twelfth_grader_math_scores = round(twelfth_graders_scores)

# Consolidate grades into single DataFrame called "math_scores_by_grade"
math_scores_by_grade = pd.DataFrame({'9th':ninth_grade_math_scores.map('{:.0f}'.format),
                                      '10th':tenth_grader_math_scores.map('{:.0f}'.format),
                                      '11th':eleventh_grader_math_scores.map('{:.0f}'.format),
                                      '12th':twelfth_grader_math_scores.map('{:.0f}'.format)})

# Minor data wrangling
math_scores_by_grade.index.name = None

# Display the DataFrame
math_scores_by_grade
```

Out[276]:

	9th	10th	11th	12th
Bailey High School	77	77	78	76
Cabrera High School	83	83	83	83
Figueroa High School	76	77	77	77
Ford High School	77	78	77	76
Griffin High School	82	84	84	83
Hernandez High School	77	77	77	77
Holden High School	84	83	85	83
Huang High School	77	76	76	77
Johnson High School	77	77	77	77
Peña High School	84	83	84	84
Rodriguez High School	77	77	76	78
Shelton High School	83	83	83	84
Thomas High School	84	83	83	83
Wilson High School	83	84	83	83
Wright High School	83	84	84	84

Reading Scores by Grade

In [277...]

```
# Separate data by grades
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 get the average of each school
ninth_graders_scores = ninth_graders.groupby(['school_name']).mean()['reading_score']
tenth_graders_scores = tenth_graders.groupby(['school_name']).mean()['reading_score']
eleventh_graders_scores = eleventh_graders.groupby(['school_name']).mean()['reading_score']
twelfth_graders_scores = twelfth_graders.groupby(['school_name']).mean()['reading_score']

# Select only the "reading_score"
ninth_grade_reading_scores = round(ninth_graders_scores)
tenth_grader_reading_scores = round(tenth_graders_scores)
eleventh_grader_reading_scores = round(eleventh_graders_scores)
twelfth_grader_reading_scores = round(twelfth_graders_scores)

# Consolidate grades into single DataFrame called "reading_scores_by_grade"
reading_scores_by_grade = pd.DataFrame({'9th':ninth_grade_reading_scores.map('{:.0f}' ),
                                         '10th':tenth_grader_reading_scores.map('{:.0f}' ),
                                         '11th':eleventh_grader_reading_scores.map('{:.0f}' ),
                                         '12th':twelfth_grader_reading_scores.map('{:.0f}' )})

# 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[277]:

	9th	10th	11th	12th
Bailey High School	81	81	81	81
Cabrera High School	84	84	84	84
Figueroa High School	81	81	81	81
Ford High School	81	81	80	81
Griffin High School	83	84	84	84
Hernandez High School	81	81	81	81
Holden High School	84	83	84	85
Huang High School	81	82	81	80
Johnson High School	81	81	81	81
Pena High School	84	84	84	85
Rodriguez High School	81	81	81	80
Shelton High School	84	83	84	83
Thomas High School	84	84	84	84
Wilson High School	84	84	84	84
Wright High School	84	84	84	84

Scores by School Spending

In [278...]

```

# Establish the bins
spending_bins = [0, 585, 630, 645, 680]

# Categorize spending based on bins
labels = ["<$585", "$585-630", "$630-645", "$645-680"]

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

```

Out[278]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Over Pass
Bailey High School	District	4976	\$3,124,928	\$628	77.048432	81.033963	66.680064	81.933280	54.642
Cabrera High School	Charter	1858	\$1,081,356	\$582	83.061895	83.975780	94.133477	97.039828	91.334
Figueroa High School	District	2949	\$1,884,411	\$639	76.711767	81.158020	65.988471	80.739234	53.204
Ford High School	District	2739	\$1,763,916	\$644	77.102592	80.746258	68.309602	79.299014	54.289
Griffin High School	Charter	1468	\$917,500	\$625	83.351499	83.816757	93.392371	97.138965	90.599
Hernandez High School	District	4635	\$3,022,020	\$652	77.289752	80.934412	66.752967	80.862999	53.527
Holden High School	Charter	427	\$248,087	\$581	83.803279	83.814988	92.505855	96.252927	89.227
Huang High School	District	2917	\$1,910,635	\$655	76.629414	81.182722	65.683922	81.316421	53.513
Johnson High School	District	4761	\$3,094,650	\$650	77.072464	80.966394	66.057551	81.222432	53.539
Pena High School	Charter	962	\$585,858	\$609	83.839917	84.044699	94.594595	95.945946	90.540
Rodriguez High School	District	3999	\$2,547,363	\$637	76.842711	80.744686	66.366592	80.220055	52.988
Shelton High School	Charter	1761	\$1,056,600	\$600	83.359455	83.725724	93.867121	95.854628	89.892
Thomas High School	Charter	1635	\$1,043,130	\$638	83.418349	83.848930	93.272171	97.308869	90.948
Wilson High School	Charter	2283	\$1,319,574	\$578	83.274201	83.989488	93.867718	96.539641	90.582
Wright High School	Charter	1800	\$1,049,400	\$583	83.682222	83.955000	93.333333	96.611111	90.333

In [279]:

```
# Group the per_school_summary DataFrame by "Spending" and average the results.  
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)'])  
spending_overall_passing = school_spending_df.groupby(['Spending Ranges (Per Student)'])  
  
# Assemble the new data by spending into a DataFrame called "spending_summary"  
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": spending_overall_passing})  
  
# Display results  
spending_summary
```

Out[279]:

Spending Ranges (Per Student)	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
<\$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 [280]:

```
# Establish the bins  
size_bins = [0, 1000, 2000, 5000]  
  
# Categorize size based on bins  
labels = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]  
  
# Use `pd.cut` to categorize spending based on the bins.  
school_size_df = per_school_summary.copy()  
school_size_df["School Size"] = pd.Series(pd.cut(per_school_counts, size_bins, labels))  
school_size_df
```

Out[280]:

School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Ove Pass
Bailey High School	District	4976	\$3,124,928	\$628	77.048432	81.033963	66.680064	81.933280
Cabrera High School	Charter	1858	\$1,081,356	\$582	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	\$1,884,411	\$639	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	\$1,763,916	\$644	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	\$917,500	\$625	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	\$3,022,020	\$652	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	\$248,087	\$581	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	\$1,910,635	\$655	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	\$3,094,650	\$650	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	\$585,858	\$609	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	\$2,547,363	\$637	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	\$1,056,600	\$600	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	\$1,043,130	\$638	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	\$1,319,574	\$578	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	\$1,049,400	\$583	83.682222	83.955000	93.333333	96.611111

In [281]:

```
# Group the per_school_summary DataFrame by "Size" and average the results.
size_math_scores = school_size_df.groupby(['School Size'])["Average Math Score"].mean()
size_reading_scores = school_size_df.groupby(['School Size'])["Average Reading Score"].mean()
size_passing_math = school_size_df.groupby(['School Size'])[("% Passing Math")].mean()
size_passing_reading = school_size_df.groupby(['School Size'])[("% Passing Reading")].mean()
size_overall_passing = school_size_df.groupby(['School Size'])[("% Overall Passing")].mean()

# Assemble the new data by spending into a 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[281]:

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

Scores by School Type

In [282]:

```
# 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"].mean()
type_reading_scores = per_school_summary.groupby(['School Type'])["Average Reading Score"].mean()
type_passing_math = per_school_summary.groupby(['School Type'])[("% Passing Math")].mean()
type_passing_reading = per_school_summary.groupby(['School Type'])[("% Passing Reading")].mean()
type_overall_passing = per_school_summary.groupby(['School Type'])[("% Overall Passing")].mean()

# Use the code provided to select new column data
average_math_score_by_type = type_math_scores
average_reading_score_by_type = type_reading_scores
average_percent_passing_math_by_type = type_passing_math
average_percent_passing_reading_by_type = type_passing_reading
average_percent_overall_passing_by_type = type_overall_passing

# 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_type,
    "% Overall Passing": average_percent_overall_passing_by_type
})

# Display results
type_summary
```

Out[282]:

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