

# PyCity Schools Analysis

- As a whole, schools with higher budgets, did not yield better test results. By contrast, schools with higher spending per student actually (\$645 - 675) underperformed compared to schools with smaller budgets (\$585 per student).
  - As a whole, smaller and medium sized schools dramatically out-performed large sized schools on passing math performances (89-91% passing vs 67%).
  - As a whole, charter schools out-performed the public district schools across all metrics. However, more analysis will be required to glean if the effect is due to school practices or the fact that charter schools tend to serve smaller student populations per school.
- 

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

```

In [44]: # Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load (Remember to Change These)
school_data_to_load = "schools_complete-Copy1.csv"
student_data_to_load = "students_complete-Copy1.csv"

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

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

# school_data_complete.count()

```

Out [44]:

	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	0
1	1	Victor Smith	M	12th	Huang High School	94	61	0
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	0
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0

## District Summary

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

In [48]: *# Create a District Summary*

```
District_Summary = pd.DataFrame({"Total Number School": [total_schools],
                                "Total Students" : [student_total],
                                "Total Budget" : [total_budget],
                                "Average Math Score": [avg_math_score],
                                "Average Reading Score": [avg_read_score],
                                "Overall Passing Rate": [Overall_Passing_Rate],
                                "% Passing Math": [perc_math_pass],
                                "% Passing Reading" : [perc_read_pass]})

District_Summary
```

Out[48]:

	Total Number School	Total Students	Total Budget	Average Math Score	Average Reading Score	Overall Passing Rate	% Passing Math	% Passing Reading
0	15	39170	24649428	78.985371	81.87784	80.431606	72.392137	82.971662

In [39]: *# Total number of schools*

```
total_schools = school_merge.school_name.nunique()
total_schools
```

Out[39]: 15

```
In [10]: # Total number of students
student_total = school_merge['Student ID'].count()
student_total
```

Out[10]: 39170

```
In [47]: # Total budget
total_budget = school_data["budget"].sum()
total_budget
```

Out[47]: 24649428

```
In [14]: # Average math score

avg_math_score = school_merge['math_score'].mean()
avg_math_score
```

Out[14]: 78.98537145774827

```
In [15]: # Average reading score
avg_read_score = school_merge['reading_score'].mean()
avg_read_score
```

Out[15]: 81.87784018381414

```
In [25]: # Overall average score
overall_score = avg_read_score + avg_math_score
overall_score

Overall_Passing_Rate = overall_score/2
print(Overall_Passing_Rate)
```

80.43160582078121

```
In [49]: # Percentage of passing math (70 or greater)
perc_math_pass=(school_merge["math_score"]>70).sum() / school_merge["m
print(perc_math_pass)

perc_read_pass = (school_merge["reading_score"]>70).sum()/school_merge
print(perc_read_pass)

District_Summary = pd.DataFrame({"Total Number School":[total_schools]
                                "Total Students" : [student_total],
                                "Total Budget" : [total_budget],
                                "Average Math Score":[avg_math_score]
                                "Average Reading Score":[avg_read_sco
                                "Overall Passing Rate":[Overall_Passi
                                "% Passing Math": [perc_math_pass],
                                "% Passing Reading" : [perc_read_pass

District_Summary
```

```
72.39213683941792
82.97166198621395
```

Out[49]:

	Total Number School	Total Students	Total Budget	Average Math Score	Average Reading Score	Overall Passing Rate	% Passing Math	% Passing Reading
0	15	39170	24649428	78.985371	81.87784	80.431606	72.392137	82.971662

## School Summary

```
In [50]: # Create an overview table that summarizes key metrics about each school
* School Name
* School Type
* Total Students
* Total School Budget
* Per Student Budget
* Average Math Score
* Average Reading Score
* % Passing Math
* % Passing Reading
* Overall Passing Rate (Average of the above two)

* Create a dataframe to hold the above results
school_types = school_data.set_index(['school_name'])['type']
```

Input In [50]

\* Create an overview table that summarizes key metrics about each school, including:

^  
SyntaxError: invalid syntax

```

In [236]: school_types = school_data.set_index(['school_name'])['type']
students_per_sch = school_merge['school_name'].value_counts()
budget_per_sch = school_merge.groupby(['school_name'])['budget'].mean()
per_stu_budg = budget_per_sch / students_per_sch
avg_math_stu = school_merge.groupby(['school_name'])['math_score'].mean()
avg_read_stu = school_merge.groupby(['school_name'])['reading_score'].mean()

pas_math_score = school_merge.loc[school_merge['math_score'] >= 70]
group_math_score = pas_math_score['school_name'].value_counts()
math_percent = group_math_score / students_per_sch * 100

pas_read_score = school_merge.loc[school_merge['reading_score'] >= 70]
group_read_score = pas_read_score['school_name'].value_counts()
read_percent = group_read_score / students_per_sch * 100

overall_pas_percent = math_percent + read_percent / students_per_sch

school_merge.head()

```

Out[236]:

	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	0
1	1	Victor Smith	M	12th	Huang High School	94	61	0
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	0
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0

In [85]:

```
Key_Metric_Per_School = pd.DataFrame({'School Type': school_types,
                                     'Total Students': students_per_sch,
                                     'Budget for School': budget_per_sch,
                                     'Budget Per Student': per_stu_budg,
                                     'Average Math Score': avg_math_stu,
                                     'Average Reading Score': avg_read,
                                     '% Passing Math': math_percent,
                                     '% Passing Reading': read_percent,
                                     'Overall Passing Rate': overall_pa

Key_Metric_Per_School
```

Out[85]:

	School Type	Total Students	Budget for School	Budget Per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
<b>Bailey High School</b>	District	4976	3124928.0	628.0	77.048432	81.87784	66.680064	81.933280
<b>Cabrera High School</b>	Charter	1858	1081356.0	582.0	83.061895	81.87784	94.133477	97.039828
<b>Figueroa High School</b>	District	2949	1884411.0	639.0	76.711767	81.87784	65.988471	80.739234
<b>Ford High School</b>	District	2739	1763916.0	644.0	77.102592	81.87784	68.309602	79.299014
<b>Griffin High School</b>	Charter	1468	917500.0	625.0	83.351499	81.87784	93.392371	97.138965
<b>Hernandez High School</b>	District	4635	3022020.0	652.0	77.289752	81.87784	66.752967	80.862999
<b>Holden High School</b>	Charter	427	248087.0	581.0	83.803279	81.87784	92.505855	96.252927
<b>Huang High School</b>	District	2917	1910635.0	655.0	76.629414	81.87784	65.683922	81.316421
<b>Johnson High School</b>	District	4761	3094650.0	650.0	77.072464	81.87784	66.057551	81.222432
<b>Pena High School</b>	Charter	962	585858.0	609.0	83.839917	81.87784	94.594595	95.945946
<b>Rodriguez High School</b>	District	3999	2547363.0	637.0	76.842711	81.87784	66.366592	80.220055



<b>Shelton High School</b>	Charter	1761	1056600.0	600.0	83.359455	81.87784	93.867121	95.854628
<b>Thomas High School</b>	Charter	1635	1043130.0	638.0	83.418349	81.87784	93.272171	97.308869
<b>Wilson High School</b>	Charter	2283	1319574.0	578.0	83.274201	81.87784	93.867718	96.539641
<b>Wright High School</b>	Charter	1800	1049400.0	583.0	83.682222	81.87784	93.333333	96.611111

## Top Performing Schools (By Passing Rate)

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

```
In [92]: # Sort and display the top five schools in overall passing rat
Top_Five_Passing = Key_Metric_Per_School.sort_values(['% Passing Math'])
Top_Five_Passing.head(5)
```

Out[92]:

	School Type	Total Students	Budget for School	Budget Per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	F
<b>Pena High School</b>	Charter	962	585858.0	609.0	83.839917	81.87784	94.594595	95.945946	94.
<b>Cabrera High School</b>	Charter	1858	1081356.0	582.0	83.061895	81.87784	94.133477	97.039828	94.
<b>Wilson High School</b>	Charter	2283	1319574.0	578.0	83.274201	81.87784	93.867718	96.539641	93.
<b>Shelton High School</b>	Charter	1761	1056600.0	600.0	83.359455	81.87784	93.867121	95.854628	93.
<b>Griffin High School</b>	Charter	1468	917500.0	625.0	83.351499	81.87784	93.392371	97.138965	93.

In [96]: *# Calculate total school budget*

```
budget_per_sch = school_merge.groupby(['school_name'])['budget'].mean()  
  
budget_per_sch
```

Out[96]:

school_name	
Bailey High School	3124928.0
Cabrera High School	1081356.0
Figueroa High School	1884411.0
Ford High School	1763916.0
Griffin High School	917500.0
Hernandez High School	3022020.0
Holden High School	248087.0
Huang High School	1910635.0
Johnson High School	3094650.0
Pena High School	585858.0
Rodriguez High School	2547363.0
Shelton High School	1056600.0
Thomas High School	1043130.0
Wilson High School	1319574.0
Wright High School	1049400.0

Name: budget, dtype: float64

In [97]: *# Calculate per student budget*

```
per_stu_budg = budget_per_sch / students_per_sch  
  
per_stu_budg
```

Out[97]:

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 [98]: *# Caculate the avg math and reading score*

```
pas_math_score = school_merge.loc[school_merge['math_score']>=70]
group_math_score = pas_math_score['school_name'].value_counts()
math_percent = group_math_score/ students_per_sch*100

pas_read_score = school_merge.loc[school_merge['reading_score']>=70]
group_read_score = pas_read_score['school_name'].value_counts()
read_percent = group_read_score/ students_per_sch*100

print(read_percent)
print(math_percent)
```

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
Name: school_name, dtype: float64	
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
Name: school_name, dtype: float64	

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

```
In [100]: # Find the total counts of math result

group_math_score = pas_math_score['school_name'].value_counts()

# Find the counts for math result in each school that pass 70 or higher
pas_math_score = school_merge.loc[school_merge['math_score']>=70]

# Calculate the math passing rate

math_percent = group_math_score / students_per_sch*100
```

```
In [101]: # Find the total counts of read result

group_read_score = pas_read_score['school_name'].value_counts()

# Find the counts for read result in each school that pass 70 or higher
pas_read_score = school_merge.loc[school_merge['reading_score']>=70]

# Calculate the read passing rate

read_percent = group_read_score / students_per_sch*100
```

```
In [102]: # Calculate the overall passing rate (average of the math and reading

overall_pas_percent = math_percent + read_percent / students_per_sch
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

## Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

```
In [111]: # Sort and display the worst five schools in overall passing rate

Worst_5_Passing_Rate = Key_Metric_Per_School.sort_values(['% Passing M

Worst_5_Passing_Rate.head(5
```

Out[111]:

	School Type	Total Students	Budget for School	Budget Per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	
<b>Huang High School</b>	District	2917	1910635.0	655.0	76.629414	81.87784	65.683922	81.316421	(
<b>Figueroa High School</b>	District	2949	1884411.0	639.0	76.711767	81.87784	65.988471	80.739234	(
<b>Johnson High School</b>	District	4761	3094650.0	650.0	77.072464	81.87784	66.057551	81.222432	(
<b>Rodriguez High School</b>	District	3999	2547363.0	637.0	76.842711	81.87784	66.366592	80.220055	(
<b>Bailey High School</b>	District	4976	3124928.0	628.0	77.048432	81.87784	66.680064	81.933280	(

## Math Scores by Grade

- Create a table that lists the average Reading Score for students of each grade level (9th, 10th, 11th, 12th) at each school.
  - Create a pandas series for each grade. Hint: use a conditional statement.
  - Group each series by school
  - Combine the series into a dataframe
  - Optional: give the displayed data cleaner formatting

```
In [144]: # Create table that lists the average math score for each school of ea
math_by_grade = pd.DataFrame({'9th': math_9,
                              '10th': math_10,
                              '11th': math_11,
                              '12th': math_12})
math_by_grade
```

Out[144]:

	9th	10th	11th	12th
school_name				
Bailey High School	77.083676	76.996772	76.996772	76.492218
Cabrera High School	83.094697	83.154506	83.154506	83.277487
Figueroa High School	76.403037	76.539974	76.539974	77.151369
Ford High School	77.361345	77.672316	77.672316	76.179963
Griffin High School	82.044010	84.229064	84.229064	83.356164
Hernandez High School	77.438495	77.337408	77.337408	77.186567
Holden High School	83.787402	83.429825	83.429825	82.855422
Huang High School	77.027251	75.908735	75.908735	77.225641
Johnson High School	77.187857	76.691117	76.691117	76.863248
Pena High School	83.625455	83.372000	83.372000	84.121547
Rodriguez High School	76.859966	76.612500	76.612500	77.690748
Shelton High School	83.420755	82.917411	82.917411	83.778976
Thomas High School	83.590022	83.087886	83.087886	83.497041
Wilson High School	83.085578	83.724422	83.724422	83.035794
Wright High School	83.264706	84.010288	84.010288	83.644986

```
In [138]: # Calculate the average math score for 9th grade in each school
math_9 = student_data.loc[student_data['grade'] == '9th'].groupby(['s
```

```
In [139]: # Calculate the average math score for 10th grade in each school
math_10 = student_data.loc[student_data['grade'] == '10th'].groupby(['s
```

```
In [140]: # Calculate the average math score for 11th grade in each school
math_11 = student_data.loc[student_data['grade'] == '10th'].groupby(['s
```

```
In [143]: # Calculate the average math score for 12th grade in each school  
math_12 = student_data.loc[student_data['grade']=='12th'].groupby(['s
```

## Reading Score by Grade

- Perform the same operations as above for reading scores

```
In [154]: # Create table that lists the average reading score for each school of
read_score_by_grade = pd.DataFrame({'9th': read_9,
                                     '10th': read_10,
                                     '11th': read_11,
                                     '12th': read_12})
read_score_by_grade
```

Out[154]:

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

```
In [148]: # Calculate the average reading score for 9th grade in each school
read_9 = student_data.loc[student_data['grade'] == '9th'].groupby(['sc
```



```
In [149]: # Calculate the average reading score for 10th grade in each school

read_10 = student_data.loc[student_data['grade'] == '10th'].groupby(['s
```

```
In [150]: # Calculate the average reading score for 11th grade in each school
read_11 = student_data.loc[student_data['grade'] == '9th'].groupby(['s
```

```
In [151]: # Calculate the average reading score for 12th grade in each school
read_12 = student_data.loc[student_data['grade'] == '12th'].groupby(['s
```

## Scores by School Spending

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

```
In [169]: # Sample bins. Feel free to create your own bins.
bins = [0, 585, 615, 645, 675]
group_values = ["<$585", "$585-615", "$615-645", "$645-675"]
```

```
In [178]: # Create a new column to show budget per student in each row

Key_Metric_Per_School["Spending Ranges Per Student"] = pd.cut(Key_Metric_Per_School['Spending Per Student'], bins, labels=group_values)
scoresvia_sch = Key_Metric_Per_School.drop(columns = ['School Type', 'Spending Ranges Per Student'])
```

```
In [182]: # Calculate the average math score within each spending range

scores_spent = scoresvia_sch.groupby(['Spending Ranges Per Student'])

scores_spent.mean()
```

Out[182]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Spending Ranges Per Student					
<\$585	83.455399	81.87784	93.460096	96.610877	93.553497
\$585-615	83.599686	81.87784	94.230858	95.900287	94.307942
\$615-645	79.079225	81.87784	75.668212	86.106569	75.704636
\$645-675	76.997210	81.87784	66.164813	81.133951	66.185607

```
In [183]: # Calculate the percentage passing rate for math in each spending range
```

```
In [184]: # Calculate the percentage passing rate for reading in each spending range
```

```
In [185]: # Calculate the percentage overall passing rate in each spending range
```

## Scores by School Size

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

```
In [208]: # Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

```
In [209]: # Create a new column for the bin groups
Key_Metric_Per_School['School Size']=pd.cut(Key_Metric_Per_School['Tot
avgby = Key_Metric_Per_School.drop(columns=['School Type', 'Total Stud
avgby.head()
avg_bysize = avgby.groupby(['School Size'])
```

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

```
In [229]: # math_pass_size
school_merge['size_bins']=pd.cut(school_merge['size'], size_bins, label
by_spent = school_merge.groupby('size_bins')
pass_math = school_merge[school_merge['math_score'] >= 70].groupby('si
pass_math.head()
```

```
Out[229]: size_bins
Small (<1000)      1305
Medium (1000-2000)  7978
Large (2000-5000)  20087
Name: Student ID, dtype: int64
```

```
In [232]: # read_pass_size

pass_reading = school_merge[school_merge['reading_score'] >= 70].group
pass_reading
```

```
Out[232]: size_bins
Small (<1000)      1334
Medium (1000-2000)  8247
Large (2000-5000)  24029
Name: Student ID, dtype: int64
```

```
In [226]: # Calculate the overall passing rate for different school size
avg_bysize.mean()
```

Out[226]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
School Size					
Small (<1000)	83.821598	81.87784	93.550225	96.099437	93.712801
Medium (1000-2000)	83.374684	81.87784	93.599695	96.790680	93.656899
Large (2000-5000)	77.746417	81.87784	69.963361	82.766634	69.988051

## Scores by School Type

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

```
In [235]: # Create bins and groups, school type {'Charter', 'District'}  
pass_reading = school_merge[school_merge['reading_score'] >= 70].group
```

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

```
In [240]: # math pass size  
  
pass_math = school_merge[school_merge['math_score'] >= 70].groupby('ty  
pass_math
```

```
Out[240]: type  
Charter      11426  
District     17944  
Name: Student ID, dtype: int64
```

```
In [238]: # reading pass size  
  
pass_reading = school_merge[school_merge['reading_score'] >= 70].group  
pass_reading
```

```
Out[238]: type  
Charter      11785  
District     21825  
Name: Student ID, dtype: int64
```

In [200]: *# Calculate the overall passing rate*

```
type_score = Key_Metric_Per_School.drop(columns=['Total Students', 'Bu
type_score.head()
```

Out[200]:

	School Type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate	Spending Ranges Per Student)	Spending Ranges Per Student
<b>Bailey High School</b>	District	77.048432	81.87784	66.680064	81.933280	66.696530	\$615-645	\$615-645
<b>Cabrera High School</b>	Charter	83.061895	81.87784	94.133477	97.039828	94.185705	<\$585	<\$585
<b>Figueroa High School</b>	District	76.711767	81.87784	65.988471	80.739234	66.015849	\$615-645	\$615-645
<b>Ford High School</b>	District	77.102592	81.87784	68.309602	79.299014	68.338554	\$615-645	\$615-645
<b>Griffin High School</b>	Charter	83.351499	81.87784	93.392371	97.138965	93.458542	\$615-645	\$615-645

In [202]: `school_typescore = type_score.groupby(['School Type'])`  
`school_typescore.mean()`

Out[202]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
<b>School Type</b>					
<b>Charter</b>	83.473852	81.87784	93.620830	96.586489	93.702512
<b>District</b>	76.956733	81.87784	66.548453	80.799062	66.570630

In [ ]:

