

Py City Schools Analysis:

Overall Trends Observed:

- Schools with higher per-pupil budgets had lower achievement scores: Schools with budgets under 600 dollars per student have average 95 percent overall pass rate, but this falls to 73 percent for highest-funded schools (those with over 645 dollars per student.) This could be because the weaker schools are being given more resources to help with underachieving students.
- Smaller schools have higher achievement scores: Small and medium schools, with fewer than 2,000 students, average approximately 95 percent overall pass rates, while the largest schools, with 2,000 - 5,000 students, have only an average of 76 percent overall pass rates.
- Charter school are achieving at significantly higher rates, with 95 percent average overall pass rates, compared to only 73 percent for district schools. This likely correlates with size-- which is 1,524 students on average for charters, vs. 3,853 for district schools.

```

In [18]: # Dependencies and Setup
import pandas as pd
import numpy as np

# A function to format a number as a float with commas and 2 decimal places
def make_float(value):
    return '{:,.2f}'.format(value)

# A function to format a data frame column as floats with commas and 2 decimal places
def column_float(data_frame, column):
    data_frame[column] = data_frame[column].map("{:,.2f}".format)

# File 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 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
alldata = pd.merge(student_data, school_data, how="left", on=["school_name", "school_name"])

alldata.head()

```

Out[18]:

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

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. $(\text{avg. math score} + \text{avg. reading score})/2$
- 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 [19]: # Calcs values
total_schools = school_data['school_name'].count()
total_students = alldata['student_name'].count()
total_budget = school_data['budget'].sum()
avg_math = alldata['math_score'].mean()
avg_reading = alldata['reading_score'].mean()
avg_overall = (avg_math + avg_reading)/2

# Using numpy, creates column 'pass_math' with "1" if students passes then use
# s that sum to calc % passing
alldata['pass_math'] = np.where(alldata['math_score']>=70, 1, 0)

percent_pass_math = (alldata['pass_math'].sum())/total_students

# Using numpy, creates column 'pass_reading' with "1" if students passes then
# uses that sum to calc % passing
alldata['pass_reading'] = np.where(alldata['reading_score']>=70, 1, 0)

percent_pass_reading = (alldata['pass_reading'].sum())/total_students

# Formats values
total_schools = ('{:,}'.format(total_schools))
total_students = ('{:,}'.format(total_students))
total_budget = make_float(total_budget)
avg_math = make_float(avg_math)
avg_reading = make_float(avg_reading)
avg_overall = make_float(avg_overall)
percent_pass_math = make_float((100*percent_pass_math))
percent_pass_reading = make_float((100*percent_pass_reading))

# Creates list that will be made into dataframe
district_summary = [['Total Schools', total_schools],
                    ['Total Students', total_students],
                    ['Total Budget', total_budget],
                    ['Average Math Score', avg_math],
                    ['Average Reading Score', avg_reading],
                    ['Average Overall Score', avg_overall],
                    ['Percent Passing Math', percent_pass_math],
                    ['Percent Passing Reading', percent_pass_reading]]

# Create the dataframe
district_summary_df = pd.DataFrame(district_summary, columns = ['Data Point',
'Result'])

district_summary_df
```

Out[19]:

	Data Point	Result
0	Total Schools	15
1	Total Students	39,170
2	Total Budget	24,649,428.00
3	Average Math Score	78.99
4	Average Reading Score	81.88
5	Average Overall Score	80.43
6	Percent Passing Math	74.98
7	Percent Passing Reading	85.81

School Summary

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

```
In [20]: per_school = alldata.groupby(['school_name'])

# Grabs type
school_type = per_school['type'].first()

# Counts number of students
tot_students = per_school['Student ID'].count()

# Divides budget by total students to get per student
school_budget = per_school['budget'].first()
per_student_budget = (school_budget / tot_students)

# Finds average scores
sch_avg_math = per_school['math_score'].mean()
sch_avg_reading = per_school['reading_score'].mean()

# Sums up passing students and calculates the pass rates
sch_pass_math = per_school['pass_math'].sum()
sch_pass_reading = per_school['pass_reading'].sum()

sch_pct_pass_math = ((sch_pass_math / tot_students)*100)
sch_pct_pass_reading = ((sch_pass_reading / tot_students)*100)
sch_overall_pass_rate = ((sch_pct_pass_math + sch_pct_pass_reading) / 2)

# Creates dataframe
school_df = pd.DataFrame({"School Type":school_type,
                          "Total Students": tot_students,
                          "Total Budget":school_budget,
                          "Per Student Budget":per_student_budget,
                          "Avg Math Score":sch_avg_math,
                          "Avg Reading Score":sch_avg_reading,
                          "Math Pass %":sch_pct_pass_math,
                          "Reading Pass %":sch_pct_pass_reading,
                          "Overall Avg Pass %":sch_overall_pass_rate})

# Creates a copy of school_df, so that numeric values can be preserved
school_values_df = school_df.copy()

# Formats for display using mapping and the column_float function
school_df["Total Students"] = school_df["Total Students"].map("{:,}".format)
column_float(school_df,"Total Budget")
column_float(school_df,"Per Student Budget")
column_float(school_df,"Avg Math Score")
column_float(school_df,"Avg Reading Score")
column_float(school_df,"Math Pass %")
column_float(school_df,"Reading Pass %")
column_float(school_df,"Overall Avg Pass %")

school_df
```

Out[20]:

	School Type	Total Students	Total Budget	Per Student Budget	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %
school_name									
Bailey High School	District	4,976	3,124,928.00	628.00	77.05	81.03	66.68	81.93	74.31
Cabrera High School	Charter	1,858	1,081,356.00	582.00	83.06	83.98	94.13	97.04	95.59
Figueroa High School	District	2,949	1,884,411.00	639.00	76.71	81.16	65.99	80.74	73.36
Ford High School	District	2,739	1,763,916.00	644.00	77.10	80.75	68.31	79.30	73.80
Griffin High School	Charter	1,468	917,500.00	625.00	83.35	83.82	93.39	97.14	95.27
Hernandez High School	District	4,635	3,022,020.00	652.00	77.29	80.93	66.75	80.86	73.81
Holden High School	Charter	427	248,087.00	581.00	83.80	83.81	92.51	96.25	94.38
Huang High School	District	2,917	1,910,635.00	655.00	76.63	81.18	65.68	81.32	73.50
Johnson High School	District	4,761	3,094,650.00	650.00	77.07	80.97	66.06	81.22	73.64
Pena High School	Charter	962	585,858.00	609.00	83.84	84.04	94.59	95.95	95.27
Rodriguez High School	District	3,999	2,547,363.00	637.00	76.84	80.74	66.37	80.22	73.29
Shelton High School	Charter	1,761	1,056,600.00	600.00	83.36	83.73	93.87	95.85	94.86
Thomas High School	Charter	1,635	1,043,130.00	638.00	83.42	83.85	93.27	97.31	95.29
Wilson High School	Charter	2,283	1,319,574.00	578.00	83.27	83.99	93.87	96.54	95.20
Wright High School	Charter	1,800	1,049,400.00	583.00	83.68	83.95	93.33	96.61	94.97

Top Performing Schools (By Passing Rate)

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

```
In [21]: # Converts pass rate back to float, so we can sort
school_df["Overall Avg Pass %"] = school_df["Overall Avg Pass %"].astype(float)

# Sorts descending
school_df = school_df.sort_values("Overall Avg Pass %", ascending = False)

school_df.head()
```

Out[21]:

	School Type	Total Students	Total Budget	Per Student Budget	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %
school_name									
Cabrera High School	Charter	1,858	1,081,356.00	582.00	83.06	83.98	94.13	97.04	95.59
Thomas High School	Charter	1,635	1,043,130.00	638.00	83.42	83.85	93.27	97.31	95.29
Griffin High School	Charter	1,468	917,500.00	625.00	83.35	83.82	93.39	97.14	95.27
Pena High School	Charter	962	585,858.00	609.00	83.84	84.04	94.59	95.95	95.27
Wilson High School	Charter	2,283	1,319,574.00	578.00	83.27	83.99	93.87	96.54	95.20

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools


```
In [22]: # Sorts ascending
school_df = school_df.sort_values("Overall Avg Pass %", ascending = True)

school_df.head()
```

Out[22]:

	School Type	Total Students	Total Budget	Per Student Budget	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %
school_name									
Rodriguez High School	District	3,999	2,547,363.00	637.00	76.84	80.74	66.37	80.22	73.29
Figueroa High School	District	2,949	1,884,411.00	639.00	76.71	81.16	65.99	80.74	73.36
Huang High School	District	2,917	1,910,635.00	655.00	76.63	81.18	65.68	81.32	73.50
Johnson High School	District	4,761	3,094,650.00	650.00	77.07	80.97	66.06	81.22	73.64
Ford High School	District	2,739	1,763,916.00	644.00	77.10	80.75	68.31	79.30	73.80

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 [23]: # Creates series for each grade
ninth_grade = alldata[alldata["grade"] == "9th"]
tenth_grade = alldata[alldata["grade"] == "10th"]
eleventh_grade = alldata[alldata["grade"] == "11th"]
twelfth_grade = alldata[alldata["grade"] == "12th"]

# Groups by school and averages math scores
ninth_math_grouped = ninth_grade.groupby(['school_name'])['math_score'].mean()
tenth_math_grouped = tenth_grade.groupby(['school_name'])['math_score'].mean()
eleventh_math_grouped = eleventh_grade.groupby(['school_name'])['math_score'].mean()
twelfth_math_grouped = twelfth_grade.groupby(['school_name'])['math_score'].mean()

# Creates dataframe
math_by_grade_df = pd.DataFrame({"9th":ninth_math_grouped,
                                "10th":tenth_math_grouped,
                                "11th":eleventh_math_grouped,
                                "12th":twelfth_math_grouped})

# Formats the values
column_float(math_by_grade_df, "9th")
column_float(math_by_grade_df, "10th")
column_float(math_by_grade_df, "11th")
column_float(math_by_grade_df, "12th")

math_by_grade_df

```

Out[23]:

	9th	10th	11th	12th
school_name				
Bailey High School	77.08	77.00	77.52	76.49
Cabrera High School	83.09	83.15	82.77	83.28
Figueroa High School	76.40	76.54	76.88	77.15
Ford High School	77.36	77.67	76.92	76.18
Griffin High School	82.04	84.23	83.84	83.36
Hernandez High School	77.44	77.34	77.14	77.19
Holden High School	83.79	83.43	85.00	82.86
Huang High School	77.03	75.91	76.45	77.23
Johnson High School	77.19	76.69	77.49	76.86
Pena High School	83.63	83.37	84.33	84.12
Rodriguez High School	76.86	76.61	76.40	77.69
Shelton High School	83.42	82.92	83.38	83.78
Thomas High School	83.59	83.09	83.50	83.50
Wilson High School	83.09	83.72	83.20	83.04
Wright High School	83.26	84.01	83.84	83.64

Reading Score by Grade

- Perform the same operations as above for reading scores

```
In [24]: # Groups by school and averages reading scores
ninth_reading_grouped = ninth_grade.groupby(['school_name'])['reading_score'].
mean()
tenth_reading_grouped = tenth_grade.groupby(['school_name'])['reading_score'].
mean()
eleventh_reading_grouped = eleventh_grade.groupby(['school_name'])['reading_score'].mean()
twelfth_reading_grouped = twelfth_grade.groupby(['school_name'])['reading_score'].mean()

# Creates dataframe
reading_by_grade_df = pd.DataFrame({"9th":ninth_reading_grouped,
                                     "10th":tenth_reading_grouped,
                                     "11th":eleventh_reading_grouped,
                                     "12th":twelfth_reading_grouped})

# Formats values
column_float(reading_by_grade_df,"9th")
column_float(reading_by_grade_df,"10th")
column_float(reading_by_grade_df,"11th")
column_float(reading_by_grade_df,"12th")

reading_by_grade_df
```

Out[24]:

	9th	10th	11th	12th
school_name				
Bailey High School	81.30	80.91	80.95	80.91
Cabrera High School	83.68	84.25	83.79	84.29
Figueroa High School	81.20	81.41	80.64	81.38
Ford High School	80.63	81.26	80.40	80.66
Griffin High School	83.37	83.71	84.29	84.01
Hernandez High School	80.87	80.66	81.40	80.86
Holden High School	83.68	83.32	83.82	84.70
Huang High School	81.29	81.51	81.42	80.31
Johnson High School	81.26	80.77	80.62	81.23
Pena High School	83.81	83.61	84.34	84.59
Rodriguez High School	80.99	80.63	80.86	80.38
Shelton High School	84.12	83.44	84.37	82.78
Thomas High School	83.73	84.25	83.59	83.83
Wilson High School	83.94	84.02	83.76	84.32
Wright High School	83.83	83.81	84.16	84.07

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 [25]: # Copies school values dataframe into school_df, to restore numeric values
school_df = school_values_df.copy()

# Creates bins based on per student spending
bins = [0, 600, 630, 645, 661]
group_names = ["<$600", "$600-629", "$630-644", "$645-660"]

school_df["Per Student Budget Range"] = pd.cut(school_df["Per Student Budget"],
], bins, labels=group_names)

budget_ranges = school_df.groupby("Per Student Budget Range")

# Gets average scores
avg_math_by_budget = budget_ranges["Avg Math Score"].mean()
avg_reading_by_budget = budget_ranges["Avg Reading Score"].mean()
avg_math_pass = budget_ranges["Math Pass %"].mean()
avg_reading_pass = budget_ranges["Reading Pass %"].mean()
avg_overall_pass = budget_ranges["Overall Avg Pass %"].mean()

# Creates dataframe
budget_summary_df = pd.DataFrame({"Avg Math Score":avg_math_by_budget,
                                "Avg Reading Score":avg_reading_by_budget,
                                "Math Pass %":avg_math_pass,
                                "Reading Pass %":avg_reading_pass,
                                "Overall Avg Pass %":avg_overall_pass})

# Formats values
column_float(budget_summary_df,"Avg Math Score")
column_float(budget_summary_df,"Avg Reading Score")
column_float(budget_summary_df,"Math Pass %")
column_float(budget_summary_df,"Reading Pass %")
column_float(budget_summary_df,"Overall Avg Pass %")

budget_summary_df

```

Out[25]:

	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %
Per Student Budget Range					
<\$600	83.44	83.89	93.54	96.46	95.00
\$600-629	81.41	82.97	84.89	91.67	88.28
\$630-644	78.52	81.62	73.48	84.39	78.94
\$645-660	77.00	81.03	66.16	81.13	73.65

Scores by School Size

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

```

In [26]: # Copies the school values dataframe, to restore numeric values
school_df = school_values_df.copy()

# Creates bins for number of total students
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

school_df["Size Range"] = pd.cut(school_df["Total Students"], size_bins, labels=group_names)

size_ranges = school_df.groupby("Size Range")

# Calcs average scores
avg_math_by_size = size_ranges["Avg Math Score"].mean()
avg_reading_by_size = size_ranges["Avg Reading Score"].mean()
avg_math_pass_by_size = size_ranges["Math Pass %"].mean()
avg_reading_pass_by_size = size_ranges["Reading Pass %"].mean()
avg_overall_pass_by_size = size_ranges["Overall Avg Pass %"].mean()

# Creates dataframe
size_summary_df = pd.DataFrame({"Avg Math Score":avg_math_by_size,
                                "Avg Reading Score":avg_reading_by_size,
                                "Math Pass %":avg_math_pass_by_size,
                                "Reading Pass %":avg_reading_pass_by_size,
                                "Overall Avg Pass %":avg_overall_pass_by_size})

# Formats values
column_float(size_summary_df, "Avg Math Score")
column_float(size_summary_df, "Avg Reading Score")
column_float(size_summary_df, "Math Pass %")
column_float(size_summary_df, "Reading Pass %")
column_float(size_summary_df, "Overall Avg Pass %")

size_summary_df

```

Out[26]:

	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %
Size Range					
Small (<1000)	83.82	83.93	93.55	96.10	94.82
Medium (1000-2000)	83.37	83.86	93.60	96.79	95.20
Large (2000-5000)	77.75	81.34	69.96	82.77	76.36

Scores by School Type

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

```

In [27]: # Copies the school values dataframe, to restore numeric values
school_df = school_values_df.copy()

# Groups by school type
type_groups = school_df.groupby("School Type")

# Calcs average scores
avg_math_by_type = type_groups["Avg Math Score"].mean()
avg_reading_by_type = type_groups["Avg Reading Score"].mean()
avg_math_pass_by_type = type_groups["Math Pass %"].mean()
avg_reading_pass_by_type = type_groups["Reading Pass %"].mean()
avg_overall_pass_by_type = type_groups["Overall Avg Pass %"].mean()
avg_size_by_type = type_groups["Total Students"].mean()
avg_budget_by_type = type_groups["Per Student Budget"].mean()

# Creates dataframe
type_summary_df = pd.DataFrame({"Avg Math Score":avg_math_by_type,
                                "Avg Reading Score":avg_reading_by_type,
                                "Math Pass %":avg_math_pass_by_type,
                                "Reading Pass %":avg_reading_pass_by_type,
                                "Overall Avg Pass %":avg_overall_pass_by_type,
                                "Average # of Students":avg_size_by_type,
                                "Avg Per Student Budget":avg_budget_by_type,
                                })

# Formats values
column_float(type_summary_df,"Avg Math Score")
column_float(type_summary_df,"Avg Reading Score")
column_float(type_summary_df,"Math Pass %")
column_float(type_summary_df,"Reading Pass %")
column_float(type_summary_df,"Overall Avg Pass %")
column_float(type_summary_df,"Average # of Students")
column_float(type_summary_df,"Avg Per Student Budget")

type_summary_df

```

Out[27]:

	Avg Math Score	Avg Reading Score	Math Pass %	Reading Pass %	Overall Avg Pass %	Average # of Students	Avg Per Student Budget
School Type							
Charter	83.47	83.90	93.62	96.59	95.10	1,524.25	599.50
District	76.96	80.97	66.55	80.80	73.67	3,853.71	643.57

In []: