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 [35]:

```
# Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load (Remember to Change These)
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 Data Frames
school_data = pd.read_csv(school_data_to_load)
student_data = pd.read_csv(student_data_to_load)
school_df = pd.DataFrame(school_data)
student_df = pd.DataFrame(student_data)
# Combine the data into a single dataset
school_data_complete = pd.merge(student_data, school_data, how="left", on=["school_name", "school_name"])

school_data.head()

student_df.head()
school_df.head()
```

Out[35]:

	School ID	school_name	type	size	budget
0	0	Huang High School	District	2917	1910635
1	1	Figueroa High School	District	2949	1884411
2	2	Shelton High School	Charter	1761	1056600
3	3	Hernandez High School	District	4635	3022020
4	4	Griffin High School	Charter	1468	917500

In [ ]:

```
```

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 [30]:

```
total_school1 =school_df['school_name'].nunique()
total_school1
total_students1 = student_df['Student ID'].count()
total_students1
total_budget1 =school_df['budget'].sum()
total_budget1
avg_math1 = student_df['math_score'].mean()
avg_read1 = student_df['reading_score'].mean()
avg_read1
overall_pass_rate1 = (avg_math1+avg_read1)/2
math_pass1 = student_df.loc[student_df['math_score']>=70]['math_score'].count
()
read_pass1 = student_df.loc[student_df['reading_score']>=70]['reading_score'].
count()

District_Summary = ({'Total Schools': [total_school1],
                     'Total Students':[total_students1],
                     'Total_Budegt': [total_budget1],
                     'Average Math Score': [avg_math1],
                     'Average Raeding Score': [avg_read1],
                     '% Passing Math': [math_pass1],
                     '% Passing Read': [read_pass1],
                     'Overall Passing rate': [overall_pass_rate1]
                    })
District_Summary_df =pd.DataFrame(District_Summary)
District_Summary_df.round(2)
```

Out[30]:

	Total Schools	Total Students	Total_Budegt	Average Math Score	Average Raeding Score	% Passing Math	% Passing Read	Overall Passing rate
0	15	39170	24649428	78.99	81.88	29370	33610	80.43

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

Top Performing Schools (By Passing Rate)

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

In [40]:

```
# School Summary

school_name = school_df["school_name"]
school_types = school_df["type"]
total_school_budget = school_df["budget"]
total_students = student_df.groupby(["school_name"])["student_name"].count().rename("Student Counts")


total_student_df = pd.DataFrame(total_students)


pd_new = pd.DataFrame({"school_name": school_name,
                      "school type": school_types,
                       "Total School Budget": total_school_budget,

                      })

total_students
#new2=pd_new.join(total_student_df('school_name'), on ='school_name')
new3_df = pd.merge(pd_new, total_student_df, on= 'school_name')

new3_df['Per Student Budget'] = new3_df["Total School Budget"]/ new3_df["Student Counts"]

math = student_df.groupby("school_name")["math_score"].sum().rename("Math Score")
avg_math = math/total_students

read=student_df.groupby("school_name")["reading_score"].sum().rename("Reading Score")
avg_read = read/total_students

avg_math_df = pd.DataFrame(avg_math)
avg_read_df =pd.DataFrame(avg_read)

new4_df = pd.merge(new3_df, avg_math_df, on= 'school_name')
new4_df.rename(columns ={0:"Average Math Score"}, inplace = True)
new5_df =pd.merge(new4_df, avg_read_df, on= 'school_name')
new5_df.rename(columns ={0:"Average Reading Score"}, inplace = True)
pass_math = student_df.loc[student_df['math_score'] >= 70]['math_score']

pass_math = student_df[student_df['math_score'] >= 70].groupby('school_name')['Student ID'].count()/total_students
pass_read = student_df[student_df['reading_score'] >= 70].groupby('school_name')['Student ID'].count()/total_students
pass_perc_math = pass_math*100
pass_perc_read = pass_read*100
pass_perc_math_df =pd.DataFrame(pass_perc_math)
new6_df = pd.merge(new5_df, pass_perc_math_df, on= 'school_name')

pass_perc_read_df =pd.DataFrame(pass_perc_read)
new7_df = pd.merge(new6_df, pass_perc_read_df, on= 'school_name')


new7_df.rename(columns={"0_x": "% Passing Math",
                                    "0_y": "% Passing Read"}, inplace= True)


new7_df["Overall Passing rate"] =  (new7_df["% Passing Math"]+new7_df["% Passi
ng Read"])/2
#total_student_df
#new3_df
#math
#new5_df
list(new6_df.columns.values)
pass_math

pass_perc_math

new7_df
list(new7_df.columns.values)
new7_df.set_index("school_name")

# Sorting for top five passing rate

new8_df= new7_df.sort_values("Overall Passing rate",ascending = False)
# Top five schools in overall pass rate
new8_df.head(5).round(2)
```

Out[40]:

	school_name	school type	Total School Budget	Student Counts	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Read	Ov Pas
6	Cabrera High School	Charter	1081356	1858	582.0	83.06	83.98	94.13	97.04	9
14	Thomas High School	Charter	1043130	1635	638.0	83.42	83.85	93.27	97.31	9
9	Pena High School	Charter	585858	962	609.0	83.84	84.04	94.59	95.95	9
4	Griffin High School	Charter	917500	1468	625.0	83.35	83.82	93.39	97.14	9
5	Wilson High School	Charter	1319574	2283	578.0	83.27	83.99	93.87	96.54	9

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [41]:

```
# Bottom Performing School

new8_df.tail(5).round(2)
```

Out[41]:

	school_name	school type	Total School Budget	Student Counts	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Read	Ov Pas
13	Ford High School	District	1763916	2739	644.0	77.10	80.75	68.31	79.30	7
12	Johnson High School	District	3094650	4761	650.0	77.07	80.97	66.06	81.22	7
0	Huang High School	District	1910635	2917	655.0	76.63	81.18	65.68	81.32	7
1	Figueroa High School	District	1884411	2949	639.0	76.71	81.16	65.99	80.74	7
11	Rodriguez High School	District	2547363	3999	637.0	76.84	80.74	66.37	80.22	7

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 [42]:

```
ninth = student_df.loc[student_df['grade'] == "9th"]
ninth_math=ninth.groupby('school_name')["math_score"].mean()
tenth = student_df.loc[student_df['grade'] == "10th"]
tenth_math=tenth.groupby('school_name')["math_score"].mean()
eleven =  student_df.loc[student_df['grade'] == "11th"]
eleven_math=eleven.groupby('school_name')["math_score"].mean()

twelve = student_df.loc[student_df['grade'] == "12th"]
twelve_math=twelve.groupby('school_name')["math_score"].mean()

df3 = pd.DataFrame({"9th":ninth_math,
                   "10th": tenth_math,
                   "11th": eleven_math,
                   "12th": twelve_math})

df3.round(2)
```

Out[42]:

	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 [43]:

```
ninth = student_df.loc[student_df['grade'] == "9th"]
ninth_read=ninth.groupby('school_name')["reading_score"].mean()
tenth = student_df.loc[student_df['grade'] == "10th"]
tenth_read=tenth.groupby('school_name')["reading_score"].mean()
eleven =  student_df.loc[student_df['grade'] == "11th"]
eleven_read=eleven.groupby('school_name')["reading_score"].mean()

twelve = student_df.loc[student_df['grade'] == "12th"]
twelve_read=twelve.groupby('school_name')["reading_score"].mean()

df4 = pd.DataFrame({"9th":ninth_read,
                   "10th": tenth_read,
                   "11th": eleven_read,
                   "12th": twelve_read})

df4.round(2)
```

Out[43]:

	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 [44]:

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

In [46]:

```
# assign new name to the dataframe
new10_df = new9_df.copy()
new10_df['student_bins'] = pd.cut(new10_df['Per Student Budget'], spending_bin
s, labels= group_names)

new11_df =new10_df[["student_bins", "Average Math Score", "Average Reading Sco
re","% Passing Math", "% Passing Read", "Overall Passing rate" ]]
#new11_df = pd.DataFrame(zip(new10_df.student_bins, new10_df.Average Math Scor
e, new10_df.Average Reading Score, new10_df.%Passing reading, new10_df.%Passin
g reading, new10_df.% Overall Passing rate))
new10_df.head()
list(new10_df.columns.values)

school_spending = new10_df.groupby(['student_bins'])["student_bins", "Average
 Math Score", "Average Reading Score","% Passing Math", "% Passing Read", "Ove
rall Passing rate" ].mean()
school_spending.head()
list(new9_df.columns.values)
school_spending.head().round(2)
```

Out[46]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Read	Overall Passing rate
student_bins					
<$585	83.46	83.93	93.46	96.61	95.04
$585-615	83.60	83.89	94.23	95.90	95.07
$615-645	79.08	81.89	75.67	86.11	80.89
$645-675	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 [50]:

```
# Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names1 = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

In [51]:

```
school_size_col = school_df['size']

new9_df['School Size']= school_size_col


new9_df['Size Bins'] = pd.cut(new9_df['School Size'], size_bins, labels= group
_names1)

school_size_df =new9_df[["Size Bins", "Average Math Score", "Average Reading S
core","% Passing Math", "% Passing Read", "Overall Passing rate" ]]

school_size_score = school_size_df.groupby(['Size Bins'])[ "Average Math Scor
e", "Average Reading Score","% Passing Math", "% Passing Read", "Overall Passi
ng rate" ].mean()
school_size_score.head().round(2)


#school_size_df.head()
```

Out[51]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Read	Overall Passing rate
Size Bins					
Small (<1000)	83.821598	83.929843	93.550225	96.099437	94.824831
Medium (1000-2000)	83.374684	83.864438	93.599695	96.790680	95.195187
Large (2000-5000)	77.746417	81.344493	69.963361	82.766634	76.364998

Scores by School Type

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

In [52]:

```
school_type_df = new9_df[["school type", "Average Math Score", "Average Readin
g Score","% Passing Math", "% Passing Read", "Overall Passing rate" ]]

school_type_df_group = school_type_df.groupby(['school type'])["Average Math S
core", "Average Reading Score","% Passing Math", "% Passing Read", "Overall Pa
ssing rate" ].mean()
```

In [53]:
```
school_type_df_group
```

Out[53]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Read	Overall Passing rate
school type					
Charter	83.473852	83.896421	93.620830	96.586489	95.103660
District	76.956733	80.966636	66.548453	80.799062	73.673757

In [ ]:
```
```