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 [1]:
```
# 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)

# Combine the data into a single dataset
School_Data_Complete = pd.merge(Student_Data, School_Data, how="left", on=["school_name", "school_name"])
```

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 [2]:
```
complete_df = School_Data_Complete
School_df = School_Data
#complete_df.head()
#School_df.head()
```

In [3]:
```
Total_Schools = complete_df['school_name'].nunique()

Total_Students = complete_df['student_name'].count()

Total_Budget = School_Data["budget"].sum()

Average_Math_Score = complete_df['math_score'].sum()/Total_Students

Average_Reading_Score = complete_df['reading_score'].sum()/Total_Students

Overall_Passing_Rate = (Average_Math_Score + Average_Reading_Score)/2

Math_70 = complete_df['math_score'] >= 70
Percent_Passing_Math = (Math_70.sum()/Total_Students)*100

Read_70 = complete_df['reading_score'] >= 70
Percent_Passing_Reading = (Read_70.sum()/Total_Students)*100


District_Summary = pd.DataFrame(
    {"Total Schools": [Total_Schools],
     "Total Students": [Total_Students],
     "Total Budget": [Total_Budget],
     "Average Math Score": [Average_Math_Score],
     "Average Reading Score": [Average_Reading_Score],
     "% Passing Math": [Percent_Passing_Math],
     "% Passing Reading": [Percent_Passing_Reading],
     "% Overall Passing Rate": [Overall_Passing_Rate],
     })

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

District_Summary
```

Out[3]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
0	15	39,170	$24,649,428.00	78.985371	81.87784	74.980853	85.805463	80.431606

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 [4]:
```
#Execute the groupby on complete dataframe with respect to size and budget
#and call it grouped_data
grouped_data = complete_df.groupby(['school_name']).agg\
({'size':'max','budget':'max'})

# Define a function with which to aggregate groupby data with respect to
# function that calculates the percentage of values between 70 and 100 (pass level)
def pct_between(s,low,high):
    return s.between(low,high).mean()

#Average Math Score
A0 = complete_df.groupby('school_name')['math_score'].agg('mean')
#Average Reading Score
B0 = complete_df.groupby('school_name')['reading_score'].agg('mean')
#Percentage passing Math
C0 = complete_df.groupby('school_name')['math_score'].agg(pct_between,70,100)
#Percentage passing Reading
D0 = complete_df.groupby('school_name')['reading_score'].agg(pct_between,70,100)
#Overall Passing Percentage
E0 = (C0 + D0)/2
#Extract the 'type' column from school_data
F0 = School_Data.set_index("school_name")['type']

#Set up a Dataframe using the data collected above.
Performers = pd.DataFrame(
    {"School Type" :F0,
     "Average Math Score": A0,
     "Average Reading Score": B0,
     "% Passing Math": C0,
     "% Passing Reading": D0,
     "% Overall Passing Rate": E0
     })

#Rename as Final, the merged dataframes, Performers & grouped_data
Final = grouped_data.merge(Performers, left_index=True, right_index=True)

#Multiplying by 100 the % values
Final['% Passing Math'] = Final['% Passing Math']*100
Final['% Passing Reading'] = Final['% Passing Reading']*100
Final['% Overall Passing Rate'] = Final['% Overall Passing Rate']*100
```

```
#calculate the Per Student Budget by dividing the budget by size column
Final["Per Student Budget"]= Final['budget']/Final['size']

#Rename the column headers
Final = Final.rename(columns={"size":"Total Students", "budget":"Total School Budget",\
                              "student_name":"Total Students"})
#Format the values
Final["Total School Budget"] = Final["Total School Budget"].map("${:,.2f}".format)
Final["Per Student Budget"] = Final["Per Student Budget"].map("${:.2f}".format)

#Reorder the columns as required
Final = Final[['School Type','Total Students','Total School Budget',
               'Per Student Budget','Average Math Score','Average Reading Score',
               '% Passing Math','% Passing Reading','% Overall Passing Rate']]
#Sort the values according to the instructions - top 5
Final.sort_values('% Overall Passing Rate',ascending=False).head(5)
```

Out[4]:

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
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	95.586652
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	95.290520
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	95.270270
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	95.265668
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	95.203679

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [5]:
```
#Sort the values according to the instructions - lowest 5
Final.sort_values('% Overall Passing Rate',ascending=True).head(5)
```

Out[5]:

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
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	73.293323
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	73.363852
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	73.500171
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	73.639992
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602	79.299014	73.804308

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 [6]:
```
#Define a df based on required grade and grouped by school_name and returning a formatted mean value for math_score

Ninth_Grade_Math = Student_Data.loc[Student_Data['grade'] == '9th'].groupby('school_name')['math_score'].agg('mean').m
ap("{:,.2f}".format)
Tenth_Grade_Math = Student_Data.loc[Student_Data['grade'] == '10th'].groupby('school_name')['math_score'].agg('mean').
map("{:,.2f}".format)
Eleventh_Grade_Math = Student_Data.loc[Student_Data['grade'] == '11th'].groupby('school_name')['math_score'].agg('mea
n').map("{:,.2f}".format)
Twelvth_Grade_Math = Student_Data.loc[Student_Data['grade'] == '12th'].groupby('school_name')['math_score'].agg('mean'
).map("{:,.2f}".format)


#Define the Dataframe using the dataframes calculated above
MathByGrade = pd.DataFrame(
    {"9th": Ninth_Grade_Math,
     "10th": Tenth_Grade_Math,
     "11th": Eleventh_Grade_Math,
     "12th": Twelvth_Grade_Math
     })

MathByGrade
```

Out[6]:

school_name	9th	10th	11th	12th
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 [7]:
```
#Define a df based on required grade and grouped by school_name and returning a formatted mean value for reading_score

Ninth_Grade_Read = Student_Data.loc[Student_Data['grade'] == '9th'].groupby('school_name')['reading_score'].agg('mean'
).map("{:,.2f}".format)
Tenth_Grade_Read = Student_Data.loc[Student_Data['grade'] == '10th'].groupby('school_name')['reading_score'].agg('mea
n').map("{:,.2f}".format)
Eleventh_Grade_Read = Student_Data.loc[Student_Data['grade'] == '11th'].groupby('school_name')['reading_score'].agg('m
ean').map("{:,.2f}".format)
Twelvth_Grade_Read = Student_Data.loc[Student_Data['grade'] == '12th'].groupby('school_name')['reading_score'].agg('me
an').map("{:,.2f}".format)

#Define the Dataframe using the dataframes calculated above
ReadByGrade = pd.DataFrame(
    {"9th": Ninth_Grade_Read,
     "10th": Tenth_Grade_Read,
     "11th": Eleventh_Grade_Read,
     "12th": Twelvth_Grade_Read
     })

ReadByGrade
```

Out[7]:

school_name	9th	10th	11th	12th
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 [8]:
```
# Sample bins. Feel free to create your own bins.
spending_bins = [0, 585, 615, 645, 675]
group_names = ["$0 to 585", "$585-615", "$615-645", "$645-675"]
```

In [9]:
```
complete_df["spending_bins"] = pd.cut(complete_df['budget']/complete_df['size'],\
                                      spending_bins, labels=group_names)

# Define a function with which to aggregate groupby data with respect to
# function that calculates the percentage of values between 70 and 100 (pass level)
def pct_between(s,low,high):
    return s.between(low,high).mean()

# Average Math Score
A1 = complete_df.groupby('spending_bins')['math_score'].agg('mean')
# Average Reading Score
B1 = complete_df.groupby('spending_bins')['reading_score'].agg('mean')
# % Passing Math
C1 = complete_df.groupby('spending_bins')['math_score'].agg(pct_between,70,100)
# % Passing Reading
D1 = complete_df.groupby('spending_bins')['reading_score'].agg(pct_between,70,100)

E1 = (C1 + D1)/2

Spend_Bins = pd.DataFrame(
    {"Average Math Score": A1,
     "Average Reading Score": B1,
     "% Passing Math": C1*100,
     "% Passing Reading": D1*100,
     "% Overall Passing Rate": E1*100
     })

Spend_Bins
```

Out[9]:

spending_bins	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
$0 to 585	83.363065	83.964039	93.702889	96.686558	95.194724
$585-615	83.529196	83.838414	94.124128	95.886889	95.005509
$615-645	78.061635	81.434088	71.400428	83.614770	77.507599
$645-675	77.049297	81.005604	66.230813	81.109397	73.670105

Scores by School Size

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

In [10]:
```
# 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 [11]:
```
complete_df["size_bins"] = pd.cut(complete_df['size'],\
                                  size_bins, labels=group_names)

# Define a function with which to aggregate groupby data with respect to
# function that calculates the percentage of values between 70 and 100 (pass level)
def pct_between(s,low,high):
    return s.between(low,high).mean()

# Average Math Score
A2 = complete_df.groupby('size_bins')['math_score'].agg('mean')
# Average Reading Score
B2 = complete_df.groupby('size_bins')['reading_score'].agg('mean')
# % Passing Math
C2 = complete_df.groupby('size_bins')['math_score'].agg(pct_between,70,100)
# % Passing Reading
D2 = complete_df.groupby('size_bins')['reading_score'].agg(pct_between,70,100)

E2 = (C2 + D2)/2

Size_Bins = pd.DataFrame(
    {"Average Math Score": A2,
     "Average Reading Score": B2,
     "% Passing Math": C2*100,
     "% Passing Reading": D2*100,
     "% Overall Passing Rate": E2*100
     })

Size_Bins
```

Out[11]:

size_bins	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Small (<1000)	83.828654	83.974082	93.952484	96.040317	94.996400
Medium (1000-2000)	83.372682	83.867989	93.616522	96.773058	95.194790
Large (2000-5000)	77.477597	81.198674	68.652380	82.125158	75.388769

Scores by School Type

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

In [12]:
```
# Define a function with which to aggregate groupby data with respect to
# function that calculates the percentage of values between 70 and 100 (pass level)
def pct_between(s,low,high):
    return s.between(low,high).mean()

# Average Math Score
A3 = complete_df.groupby('type')['math_score'].agg('mean')
# Average Reading Score
B3 = complete_df.groupby('type')['reading_score'].agg('mean')
# % Passing Math
C3 = complete_df.groupby('type')['math_score'].agg(pct_between,70,100)
# % Passing Reading
D3 = complete_df.groupby('type')['reading_score'].agg(pct_between,70,100)

# Overall Pass Percentage
E3 = 100*(C3 + D3)/2

School_Type = pd.DataFrame(
    {"Average Math Score": A3,
     "Average Reading Score": B3,
     "% Passing Math": C3*100,
     "% Passing Reading": D3*100,
     "% Overall Passing Rate": E3
     })

School_Type
```

Out[12]:

type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Charter	83.406183	83.902821	93.701821	96.645891	95.173856
District	76.987026	80.962485	66.518387	80.905249	73.711818

Observable trends:

a) The Top 5 Performing schools with respect to an overall pass rate are "Charter" in contrast with the bottom 5 that are all "District". The Charter schools have an overall pass rate that is 20% greater than District Schools.

b) There is no sgnificant difference in the average math and reading scores from grade to grade for each school. Neither any overall improvement nor reduction in performance for students from grade to grade.

c) Spending per student levels were not proprational to the overal passing rate. In fact the lowest spending had a higher overall passing rate.

d) Schools with a total number of students within the 2000-5000 range have a 20% lower overall pass rate than those schools with student size less than 2000.