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 = "data/schools_complete.csv"
student_data_to_load = "data/students_complete.csv"

# Read School and Student Data File and store into Pandas Data Frames
school_data = pd.read_csv('./data/schools_complete.csv')
student_data = pd.read_csv('./data/students_complete.csv')

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

# school_data_complete.count()
```

Out[1]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score	School ID	typ
0	0	Paul Bradley	M	9th	Huang High School	66	79	0	Distri
1	1	Victor Smith	M	12th	Huang High School	94	61	0	Distri
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0	Distri
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	0	Distri
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0	Distri

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 [ ]:
```
```

In [2]:
```
totalSchools = school_data_complete["school_name"].nunique()
```

In [3]:
```
totalStudents = school_data_complete["Student ID"].nunique()
```

In [4]:
```
totalBudget = school_data["budget"].sum()
```

In [5]:
```
averageMathScore = school_data_complete["math_score"].mean()
```

In [6]:
```
averageReadingScore = school_data_complete["reading_score"].mean()
```

In [7]:
```
percentageOverallPassingRate = (averageMathScore + averageReadingScore) / 2
```

In [8]:
```
passingMath = len(school_data_complete[school_data_complete["math_score"] >= 70])
percentagePassingMath = passingMath / totalStudents * 100
```

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 [9]:
```
grouped_school_df = school_data_complete.groupby(["school_name"])
schoolType = grouped_school_df["type"].first()
totalStudents = grouped_school_df["Student ID"].count()
```

In [10]:
```
totalSchoolBudget = grouped_school_df["budget"].first()
```

In [11]:
```
perStudentBudget = totalSchoolBudget / totalStudents
```

In [12]:
```
averageMathScore = grouped_school_df["math_score"].mean()
averageReadingScore = grouped_school_df["reading_score"].mean()
```

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

In [13]:
```
totalMathScore = grouped_school_df["math_score"].count()

passingMath = school_data_complete[school_data_complete["math_score"] >= 70].grou

percentagePassingMath = passingMath / totalStudents * 100
```

In [14]:
```
totalReadingScore = grouped_school_df["reading_score"].count()

passingRead = school_data_complete[school_data_complete["reading_score"] >= 70].g

percentagePassingReading = passingRead / totalStudents * 100
```

In [15]:
```
percentageOverallPassingRate = (percentagePassingMath + percentagePassingReading)
```

In [16]:
```
school_summary_df = pd.DataFrame({"School Type": schoolType,
      "Total Students": totalStudents,
      "Total School Budget": totalSchoolBudget,
      "Per Student Budget": perStudentBudget,
      "Average Math Score": averageMathScore,
      "Average Reading Score": averageReadingScore,
      "% Passing Math": percentagePassingMath,
      "% Passing Reading": percentagePassingReading,
      "% Overall Passing Rate": percentageOverallPassingRate})
```

In [17]:
```
school_summary_df = school_summary_df.sort_values(["% Overall Passing Rate"], asc
```

In [18]:
```
school_summary_df[["School Type",
      "Total Students",
      "Total School Budget",
      "Per Student Budget",
      "Average Math Score",
      "Average Reading Score",
      "% Passing Math",
      "% Passing Reading",
      "% Overall Passing Rate"]].head()
```

Out[18]:

school_name	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% P
Cabrera High School	Charter	1858	1081356	582.0	83.061895	83.975780	94.133477	97.039828	95.
Thomas High School	Charter	1635	1043130	638.0	83.418349	83.848930	93.272171	97.308869	95.
Pena High School	Charter	962	585858	609.0	83.839917	84.044699	94.594595	95.945946	95.
Griffin High School	Charter	1468	917500	625.0	83.351499	83.816757	93.392371	97.138965	95.
Wilson High School	Charter	2283	1319574	578.0	83.274201	83.989488	93.867718	96.539641	95.

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [19]:
```
school_summary_df = school_summary_df.sort_values(["% Overall Passing Rate"], asc

school_summary_df[["School Type",
      "Total Students",
      "Total School Budget",
      "Per Student Budget",
      "Average Math Score",
      "Average Reading Score",
      "% Passing Math",
      "% Passing Reading",
      "% Overall Passing Rate"]].head()
```

Out[19]:

school_name	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	%
Rodriguez High School	District	3999	2547363	637.0	76.842711	80.744686	66.366592	80.220055	73.
Figueroa High School	District	2949	1884411	639.0	76.711767	81.158020	65.988471	80.739234	73.
Huang High School	District	2917	1910635	655.0	76.629414	81.182722	65.683922	81.316421	73.
Johnson High School	District	4761	3094650	650.0	77.072464	80.966394	66.057551	81.222432	73.
Ford High School	District	2739	1763916	644.0	77.102592	80.746258	68.309602	79.299014	73.

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 [25]:
```
grade_summary_df = pd.DataFrame({"9th": grade9th_ds,
      "10th": grade10th_ds,
      "11th": grade11th_ds,
      "12th": grade12th_ds})

grade_summary_df[["9th", "10th", "11th", "12th"]]
```

Out[25]:

school_name	9th	10th	11th	12th
Bailey High School	77.083676	76.996772	77.515588	76.492218
Cabrera High School	83.094697	83.154506	82.765560	83.277487
Figueroa High School	76.403037	76.539974	76.884344	77.151369
Ford High School	77.361345	77.672316	76.918058	76.179963
Griffin High School	82.044010	84.229064	83.842105	83.356164
Hernandez High School	77.438495	77.337408	77.136029	77.186567
Holden High School	83.787402	83.429825	85.000000	82.855422
Huang High School	77.027251	75.908735	76.446602	77.225641
Johnson High School	77.187857	76.691117	77.491653	76.863248
Pena High School	83.625455	83.372000	84.328125	84.121547
Rodriguez High School	76.859966	76.612500	76.395626	77.690748
Shelton High School	83.420755	82.917411	83.383495	83.778976
Thomas High School	83.590022	83.087886	83.498795	83.497041
Wilson High School	83.085578	83.724422	83.195326	83.035794
Wright High School	83.264706	84.010288	83.836782	83.644986

In [21]:
```
grade9th_ds = school_data_complete.loc[school_data_complete["grade"] == "9th"].gr
```

In [22]:
```
grade10th_ds = school_data_complete.loc[school_data_complete["grade"] == "10th"].
```

In [23]:
```
grade11th_ds = school_data_complete.loc[school_data_complete["grade"] == "11th"].
```

In [24]:
```
grade12th_ds = school_data_complete.loc[school_data_complete["grade"] == "12th"].
```

Reading Score by Grade

- Perform the same operations as above for reading scores

In [30]:
```
grade_summary_df2 = pd.DataFrame({"9th": grade9th_ds2,
      "10th": grade10th_ds2,
      "11th": grade11th_ds2,
      "12th": grade12th_ds2})

grade_summary_df2[["9th", "10th", "11th", "12th"]]
```

Out[30]:

school_name	9th	10th	11th	12th
Bailey High School	81.303155	80.907183	80.945643	80.912451
Cabrera High School	83.676136	84.253219	83.788382	84.287958
Figueroa High School	81.198598	81.408912	80.640339	81.384863
Ford High School	80.632653	81.262712	80.403642	80.662338
Griffin High School	83.369193	83.706897	84.288089	84.013699
Hernandez High School	80.866860	80.660147	81.396140	80.857143
Holden High School	83.677165	83.324561	83.815534	84.698795
Huang High School	81.290284	81.512386	81.417476	80.305983
Johnson High School	81.260714	80.773431	80.616027	81.227564
Pena High School	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	80.993127	80.629808	80.864811	80.376426
Shelton High School	84.122642	83.441964	84.373786	82.781671
Thomas High School	83.728850	84.254157	83.585542	83.831361
Wilson High School	83.939778	84.021452	83.764608	84.317673
Wright High School	83.833333	83.812757	84.156322	84.073171

In [26]:
```
grade9th_ds2 = school_data_complete.loc[school_data_complete["grade"] == "9th"].g
```

In [27]:
```
ade10th_ds2 = school_data_complete.loc[school_data_complete["grade"] == "10th"].g
```

In [28]:
```
ade11th_ds2 = school_data_complete.loc[school_data_complete["grade"] == "11th"].g
```

In [29]:
```
ade12th_ds2 = school_data_complete.loc[school_data_complete["grade"] == "12th"].g
```

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 [31]:
```
# 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 [34]:
```
school_data_complete['spending_bins'] = pd.cut(school_data_complete['budget']/sch
```

In [35]:
```
by_spending = school_data_complete.groupby('spending_bins')
```

In [36]:
```
avg_math = by_spending['math_score'].mean()
```

In [37]:
```
omplete['math_score'] >= 70].groupby('spending_bins')['Student ID'].count()/by_sp
```

In [38]:
```
ete['reading_score'] >= 70].groupby('spending_bins')['Student ID'].count()/by_spe
```

In [39]:
```
ete['reading_score'] >= 70) & (school_data_complete['math_score'] >= 70)].groupby
```

Scores by School Size

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

In [40]:
```
# 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 [44]:
```
school_data_complete['size_bins'] = pd.cut(school_data_complete['size'], size_bi
by_size = school_data_complete.groupby('size_bins')
```

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

In [45]:
```
te[school_data_complete['math_score'] >= 70].groupby('size_bins')['Student ID'].c
```

In [46]:
```
hool_data_complete['reading_score'] >= 70].groupby('size_bins')['Student ID'].cou
```

In [49]:
```
_score'] >= 70) & (school_data_complete['math_score'] >= 70)].groupby('size_bins'
```

Scores by School Type

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

In [50]:
```
school_type = school_data_complete.groupby("type")
```

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

In [52]:
```
pass_math = school_data_complete[school_data_complete['math_score'] >= 70].groupb
```

In [53]:
```
pass_read = school_data_complete[school_data_complete['reading_score'] >= 70].gro
```

In [55]:
```
'reading_score'] >= 70) & (school_data_complete['math_score'] >= 70)].groupby('ty
```

In [56]:
```
scores_school_type = pd.DataFrame({
    'Total Passing Math': pass_math,
    'Total Passing Reading': pass_read,
    "% Overall Passing": overall})
```

In [57]:
```
scores_school_type.index.name = "Type of School"
```

In [58]:
```
scores_school_type
```

Out[58]:

Type of School	Total Passing Math	Total Passing Reading	% Overall Passing
Charter	0.937018	0.966459	1104300
District	0.665184	0.809052	1448500

In [ ]:
```
```