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(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=["sch
school_data_complete

# school_data_complete.count()

#school_data_complete.to_csv("school_data_complete.csv")
```

Out[1]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score	Sch
0	0	Paul Bradley	M	9th	Huang High School	66	79	
1	1	Victor Smith	M	12th	Huang High School	94	61	
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	
4	4	Bonnie Ray	F	9th	Huang High School	97	84	
...	...	...	...	...	...	...	...	...
39165	39165	Donna Howard	F	12th	Thomas High School	99	90	
39166	39166	Dawn Bell	F	10th	Thomas High School	95	70	
39167	39167	Rebecca Tanner	F	9th	Thomas High School	73	84	
39168	39168	Desiree Kidd	F	10th	Thomas High School	99	90	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95	75	

39170 rows × 11 columns

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]:
```
# Create a District Summary
school_data_complete.describe()
#school_data_complete.max()
# I am not sure of what is expected in this cell but if I just need to explain
# content : I would say that it contains students informations like : Id, grade
```

Out[2]:

	Student ID	reading_score	math_score	School ID	size	budget
count	39170.000000	39170.00000	39170.000000	39170.000000	39170.000000	3.917000e+04
mean	19584.500000	81.87784	78.985371	6.978172	3332.957110	2.117241e+06
std	11307.549359	10.23958	12.309968	4.444329	1323.914069	8.749987e+05
min	0.000000	63.00000	55.000000	0.000000	427.000000	2.480870e+05
25%	9792.250000	73.00000	69.000000	3.000000	1858.000000	1.081356e+06
50%	19584.500000	82.00000	79.000000	7.000000	2949.000000	1.910635e+06
75%	29376.750000	91.00000	89.000000	11.000000	4635.000000	3.022020e+06
max	39169.000000	99.00000	99.000000	14.000000	4976.000000	3.124928e+06

In [3]:
```
# Total number of schools

len(pd.unique(school_data_complete['School ID']))
# There are 15 schools
```

Out[3]:
```
15
```

In [4]:
```
# Total number of students

school_data_complete['Student ID'].count()
# the total number of student is the total number of rows : 39170 students
```

Out[4]:
```
39170
```

In [5]:
```
# Total budget

school_data_complete['budget'].sum()

# The total budget is 82932329558
```

Out[5]:
```
82932329558
```

In [6]:
```
# Average math score

school_data_complete['math_score'].describe()

# Average math score is the mean of math_score, which is : 78.985371
```

Out[6]:
```
count    39170.000000
mean        78.985371
std         12.309968
min         55.000000
25%         69.000000
50%         79.000000
75%         89.000000
max         99.000000
Name: math_score, dtype: float64
```

In [7]:
```
# Average reading score

school_data_complete['reading_score'].describe()

# Average reading score is the mean of reading_score, which is 81.87784
```

Out[7]:
```
count    39170.00000
mean        81.87784
std         10.23958
min         63.00000
25%         73.00000
50%         82.00000
75%         91.00000
max         99.00000
Name: reading_score, dtype: float64
```

In [8]:
```
# Overall average score

school_data_complete['average_score'] = (school_data_complete['reading_score']+
school_data_complete['average_score'].describe()

# the average score is 80.431606
```

Out[8]:
```
count    39170.000000
mean        80.431606
std          8.124914
min         59.000000
25%         75.000000
50%         80.500000
75%         86.500000
max         99.000000
Name: average_score, dtype: float64
```

In [9]:
```
# Percentage of passing math (70 or greater)

math_pass = school_data_complete[school_data_complete['math_score']>= 70]
(math_pass.size/school_data_complete.size)*100
# percentage of passing math is 74.98085269338778%
```

Out[9]:
```
74.9808526933878
```

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 [10]:
```
# Assuming that for passing student must have 70 or higher at math score AND re
```

In [11]:
```
Number_Of_Student_BySchool = school_data_complete.groupby('school_name')['stude
Number_Of_Student_BySchool
```

Out[11]:
```
school_name
Bailey High School       4976
Cabrera High School      1858
Figueroa High School     2949
Ford High School         2739
Griffin High School      1468
Hernandez High School    4635
Holden High School        427
Huang High School        2917
Johnson High School      4761
Pena High School          962
Rodriguez High School    3999
Shelton High School      1761
Thomas High School       1635
Wilson High School       2283
Wright High School       1800
Name: student_name, dtype: int64
```

In [12]:
```
#  Sort and display the top five schools in overall passing rate
```

In [13]:
```
passing_math_BySchool = math_pass.groupby('school_name')['student_name'].count(
passing_reading_BySchool = school_data_complete[school_data_complete['reading_s
percentage_passing_math_BySchool = (passing_math_BySchool/Number_Of_Student_ByS
percentage_passing_reading_BySchool = (passing_reading_BySchool/Number_Of_Stude
overall_passing_rate = (percentage_passing_math_BySchool + percentage_passing_r
overall_passing_rate.sort_values(ascending  = False ).head(5)
```

Out[13]:
```
school_name
Cabrera High School    95.586652
Thomas High School     95.290520
Pena High School       95.270270
Griffin High School    95.265668
Wilson High School     95.203679
Name: student_name, dtype: float64
```

In [14]:
```
# Calculate total school budget

Total_Budget = school_data_complete.groupby('school_name')['budget'].sum()
pd.DataFrame(Total_Budget)
```

Out[14]:

school_name	budget
Bailey High School	15549641728
Cabrera High School	2009159448
Figueroa High School	5557128039
Ford High School	4831365924
Griffin High School	1346890000
Hernandez High School	14007062700
Holden High School	105933149
Huang High School	5573322295
Johnson High School	14733628650
Pena High School	563595396
Rodriguez High School	10186904637
Shelton High School	1860672600
Thomas High School	1705517550
Wilson High School	3012587442
Wright High School	1888920000

In [15]:
```
# Calculate per student budget

Total_Budget_per_student = Total_Budget/Number_Of_Student_BySchool
Total_Budget_per_student
```

Out[15]:
```
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
dtype: float64
```

In [16]:
```
# Cacluate the avg math and reading score
```

```
avg_math_score_BySchool = school_data_complete.groupby('school_name')['math_sco
avg_reading_score_BySchool = school_data_complete.groupby('school_name')['readi
avg_math_score_BySchool,avg_reading_score_BySchool
```

Out[16]:
```
(school_name
 Bailey High School       77.048432
 Cabrera High School      83.061895
 Figueroa High School     76.711767
 Ford High School         77.102592
 Griffin High School      83.351499
 Hernandez High School    77.289752
 Holden High School       83.803279
 Huang High School        76.629414
 Johnson High School      77.072464
 Pena High School         83.839917
 Rodriguez High School    76.842711
 Shelton High School      83.359455
 Thomas High School       83.418349
 Wilson High School       83.274201
 Wright High School       83.682222
 Name: math_score, dtype: float64,
 school_name
 Bailey High School       81.033963
 Cabrera High School      83.975780
 Figueroa High School     81.158020
 Ford High School         80.746258
 Griffin High School      83.816757
 Hernandez High School    80.934412
 Holden High School       83.814988
 Huang High School        81.182722
 Johnson High School      80.966394
 Pena High School         84.044699
 Rodriguez High School    80.744686
 Shelton High School      83.725724
 Thomas High School       83.848930
 Wilson High School       83.989488
 Wright High School       83.955000
 Name: reading_score, dtype: float64)
```

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

In [17]:
```
# Calculate the overall passing rate (average of the math and reading passing r

passing_math_BySchool = math_pass.groupby('school_name')['student_name'].count(
passing_reading_BySchool = school_data_complete[school_data_complete['reading_s
percentage_passing_math_BySchool = (passing_math_BySchool/Number_Of_Student_BySc
percentage_passing_reading_BySchool = (passing_reading_BySchool/Number_Of_Stude
overall_passing_rate = (percentage_passing_math_BySchool + percentage_passing_r
overall_passing_rate
#overall_passing_rate.sort_values(ascending  = False ).head(5)
```

Out[17]:
```
school_name
Bailey High School       74.306672
Cabrera High School      95.586652
Figueroa High School     73.363852
Ford High School         73.804308
Griffin High School      95.265668
Hernandez High School    73.807983
Holden High School       94.379391
Huang High School        73.500171
Johnson High School      73.639992
Pena High School         95.270270
Rodriguez High School    73.293323
Shelton High School      94.860875
Thomas High School       95.290520
Wilson High School       95.203679
Wright High School       94.972222
Name: student_name, dtype: float64
```

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [18]:
```
#  Sort and display the worst five schools in overall passing rate

overall_passing_rate.sort_values(ascending  = True).head(5)
```

Out[18]:
```
school_name
Rodriguez High School    73.293323
Figueroa High School     73.363852
Huang High School        73.500171
Johnson High School      73.639992
Ford High School         73.804308
Name: student_name, dtype: float64
```

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 [19]:
```
# Create table that lists the average math score for each school of each grade
average_math_score = school_data_complete.groupby(['school_name','grade'])['mat
df_average_math_score = average_math_score.to_frame()

df_average_math_score
```

Out[19]:

school_name	grade	math_score
Bailey High School	10th	76.996772
	11th	77.515588
	12th	76.492218
	9th	77.083676
Cabrera High School	10th	83.154506
	11th	82.765560
	12th	83.277487
	9th	83.094697
Figueroa High School	10th	76.539974
	11th	76.884344
	12th	77.151369
	9th	76.403037
Ford High School	10th	77.672316
	11th	76.918058
	12th	76.179963
	9th	77.361345
Griffin High School	10th	84.229064
	11th	83.842105
	12th	83.356164
	9th	82.044010
Hernandez High School	10th	77.337408
	11th	77.136029
	12th	77.186567
	9th	77.438495
Holden High School	10th	83.429825
	11th	85.000000
	12th	82.855422
	9th	83.787402
Huang High School	10th	75.908735
	11th	76.446602
	12th	77.225641
	9th	77.027251
Johnson High School	10th	76.691117
	11th	77.491653

		math_score
school_name	grade	
	12th	76.863248
	9th	77.187857
Pena High School	10th	83.372000
	11th	84.328125
	12th	84.121547
	9th	83.625455
Rodriguez High School	10th	76.612500
	11th	76.395626
	12th	77.690748
	9th	76.859966
Shelton High School	10th	82.917411
	11th	83.383495
	12th	83.778976
	9th	83.420755
Thomas High School	10th	83.087886
	11th	83.498795
	12th	83.497041
	9th	83.590022
Wilson High School	10th	83.724422
	11th	83.195326
	12th	83.035794
	9th	83.085578
Wright High School	10th	84.010288
	11th	83.836782
	12th	83.644986
	9th	83.264706

In [20]:
```
len(pd.unique(school_data_complete['School ID']))
```

Out[20]: 15

In [21]:
```
# Calculate the average math score for 9th grade in each school

df_average_math_score.loc(axis=0)[:,"9th", :]
```

Out[21]:

		math_score
school_name	grade	
Bailey High School	9th	77.083676
Cabrera High School	9th	83.094697
Figueroa High School	9th	76.403037
Ford High School	9th	77.361345
Griffin High School	9th	82.044010
Hernandez High School	9th	77.438495
Holden High School	9th	83.787402
Huang High School	9th	77.027251
Johnson High School	9th	77.187857
Pena High School	9th	83.625455
Rodriguez High School	9th	76.859966
Shelton High School	9th	83.420755
Thomas High School	9th	83.590022
Wilson High School	9th	83.085578
Wright High School	9th	83.264706

In [22]:
```
# Calculate the average math score for 10th grade in each school


df_average_math_score.loc(axis=0)[:,"10th", :]
```

Out[22]:

		math_score
school_name	grade	
Bailey High School	10th	76.996772
Cabrera High School	10th	83.154506
Figueroa High School	10th	76.539974
Ford High School	10th	77.672316
Griffin High School	10th	84.229064
Hernandez High School	10th	77.337408
Holden High School	10th	83.429825
Huang High School	10th	75.908735
Johnson High School	10th	76.691117
Pena High School	10th	83.372000
Rodriguez High School	10th	76.612500
Shelton High School	10th	82.917411
Thomas High School	10th	83.087886
Wilson High School	10th	83.724422
Wright High School	10th	84.010288

In [23]:
```
# Calculate the average math score for 11th grade in each school


df_average_math_score.loc(axis=0)[:,"11th", :]
```

Out[23]:

		math_score
school_name	grade	
Bailey High School	11th	77.515588
Cabrera High School	11th	82.765560
Figueroa High School	11th	76.884344
Ford High School	11th	76.918058
Griffin High School	11th	83.842105
Hernandez High School	11th	77.136029
Holden High School	11th	85.000000
Huang High School	11th	76.446602
Johnson High School	11th	77.491653
Pena High School	11th	84.328125
Rodriguez High School	11th	76.395626
Shelton High School	11th	83.383495
Thomas High School	11th	83.498795
Wilson High School	11th	83.195326
Wright High School	11th	83.836782

In [24]:
```
# Calculate the average math score for 12th grade in each school

df_average_math_score.loc(axis=0)[:,"12th", :]
```

Out[24]:

school_name	grade	math_score
Bailey High School	12th	76.492218
Cabrera High School	12th	83.277487
Figueroa High School	12th	77.151369
Ford High School	12th	76.179963
Griffin High School	12th	83.356164
Hernandez High School	12th	77.186567
Holden High School	12th	82.855422
Huang High School	12th	77.225641
Johnson High School	12th	76.863248
Pena High School	12th	84.121547
Rodriguez High School	12th	77.690748
Shelton High School	12th	83.778976
Thomas High School	12th	83.497041
Wilson High School	12th	83.035794
Wright High School	12th	83.644986

Reading Score by Grade

- Perform the same operations as above for reading scores

In [25]:
```
# Create table that lists the average reading score for each school of each gra
average_reading_score = school_data_complete.groupby(['school_name','grade'])['
df_average_reading_score = average_reading_score.to_frame()

df_average_reading_score
```

Out[25]:

		reading_score
school_name	grade	
Bailey High School	10th	80.907183
	11th	80.945643
	12th	80.912451
	9th	81.303155
Cabrera High School	10th	84.253219
	11th	83.788382
	12th	84.287958
	9th	83.676136
Figueroa High School	10th	81.408912
	11th	80.640339
	12th	81.384863
	9th	81.198598
Ford High School	10th	81.262712
	11th	80.403642
	12th	80.662338
	9th	80.632653
Griffin High School	10th	83.706897
	11th	84.288089
	12th	84.013699
	9th	83.369193
Hernandez High School	10th	80.660147
	11th	81.396140
	12th	80.857143
	9th	80.866860
Holden High School	10th	83.324561
	11th	83.815534
	12th	84.698795
	9th	83.677165
Huang High School	10th	81.512386
	11th	81.417476
	12th	80.305983
	9th	81.290284
Johnson High School	10th	80.773431
	11th	80.616027

		reading_score
school_name	grade	
	12th	81.227564
	9th	81.260714
Pena High School	10th	83.612000
	11th	84.335938
	12th	84.591160
	9th	83.807273
Rodriguez High School	10th	80.629808
	11th	80.864811
	12th	80.376426
	9th	80.993127
Shelton High School	10th	83.441964
	11th	84.373786
	12th	82.781671
	9th	84.122642
Thomas High School	10th	84.254157
	11th	83.585542
	12th	83.831361
	9th	83.728850
Wilson High School	10th	84.021452
	11th	83.764608
	12th	84.317673
	9th	83.939778
Wright High School	10th	83.812757
	11th	84.156322
	12th	84.073171
	9th	83.833333

In [26]:
```
# Calculate the average reading score for 9th grade in each school

df_average_reading_score.loc(axis=0)[:,"9th", :]
```

Out[26]:

		reading_score
school_name	grade	
Bailey High School	9th	81.303155
Cabrera High School	9th	83.676136
Figueroa High School	9th	81.198598
Ford High School	9th	80.632653
Griffin High School	9th	83.369193
Hernandez High School	9th	80.866860
Holden High School	9th	83.677165
Huang High School	9th	81.290284
Johnson High School	9th	81.260714
Pena High School	9th	83.807273
Rodriguez High School	9th	80.993127
Shelton High School	9th	84.122642
Thomas High School	9th	83.728850
Wilson High School	9th	83.939778
Wright High School	9th	83.833333

In [27]:
```
# Calculate the average reading score for 10th grade in each school


df_average_reading_score.loc(axis=0)[:,"10th", :]
```

Out[27]:

		reading_score
school_name	grade	
Bailey High School	10th	80.907183
Cabrera High School	10th	84.253219
Figueroa High School	10th	81.408912
Ford High School	10th	81.262712
Griffin High School	10th	83.706897
Hernandez High School	10th	80.660147
Holden High School	10th	83.324561
Huang High School	10th	81.512386
Johnson High School	10th	80.773431
Pena High School	10th	83.612000
Rodriguez High School	10th	80.629808
Shelton High School	10th	83.441964
Thomas High School	10th	84.254157
Wilson High School	10th	84.021452
Wright High School	10th	83.812757

In [28]:
```
# Calculate the average reading score for 11th grade in each school

df_average_reading_score.loc(axis=0)[:,"11th", :]
```

Out[28]:

		reading_score
school_name	grade	
Bailey High School	11th	80.945643
Cabrera High School	11th	83.788382
Figueroa High School	11th	80.640339
Ford High School	11th	80.403642
Griffin High School	11th	84.288089
Hernandez High School	11th	81.396140
Holden High School	11th	83.815534
Huang High School	11th	81.417476
Johnson High School	11th	80.616027
Pena High School	11th	84.335938
Rodriguez High School	11th	80.864811
Shelton High School	11th	84.373786
Thomas High School	11th	83.585542
Wilson High School	11th	83.764608
Wright High School	11th	84.156322

In [29]:
```
# Calculate the average reading score for 12th grade in each school

df_average_reading_score.loc(axis=0)[:,"12th", :]
```

Out[29]:

		reading_score
school_name	grade	
Bailey High School	12th	80.912451
Cabrera High School	12th	84.287958
Figueroa High School	12th	81.384863
Ford High School	12th	80.662338
Griffin High School	12th	84.013699
Hernandez High School	12th	80.857143
Holden High School	12th	84.698795
Huang High School	12th	80.305983
Johnson High School	12th	81.227564
Pena High School	12th	84.591160
Rodriguez High School	12th	80.376426
Shelton High School	12th	82.781671
Thomas High School	12th	83.831361
Wilson High School	12th	84.317673
Wright High School	12th	84.073171

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 [30]:
```
school_data_complete['budget'].describe()
```

Out[30]:
```
count    3.917000e+04
mean     2.117241e+06
std      8.749987e+05
min      2.480870e+05
25%      1.081356e+06
50%      1.910635e+06
75%      3.022020e+06
max      3.124928e+06
Name: budget, dtype: float64
```

In [31]:
```
# 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 [32]:
```
# Create a new column to show budget per student in each row
```

In [33]:
```
# Create a new column to show budget per student in each row
scores_by_school_spending = {}


scores_by_school_spending['student_name'] = school_data_complete['student_name'
scores_by_school_spending['budget_per_student'] = school_data_complete['budget'


df_scores_by_school_spending = pd.DataFrame.from_dict(scores_by_school_spendinc

df_scores_by_school_spending
```

Out[33]:

	student_name	budget_per_student
0	Paul Bradley	1910635
1	Victor Smith	1910635
2	Kevin Rodriguez	1910635
3	Dr. Richard Scott	1910635
4	Bonnie Ray	1910635
...	...	...
39165	Donna Howard	1043130
39166	Dawn Bell	1043130
39167	Rebecca Tanner	1043130
39168	Desiree Kidd	1043130
39169	Carolyn Jackson	1043130

39170 rows × 2 columns

In [34]:
```
df_scores_by_school_spending['budget_per_student']
```

Out[34]:
```
0         1910635
1         1910635
2         1910635
3         1910635
4         1910635
           ...
39165     1043130
39166     1043130
39167     1043130
39168     1043130
39169     1043130
Name: budget_per_student, Length: 39170, dtype: int64
```

In [35]:
```
df_scores_by_school_spending.iloc[0][1]
```

Out[35]:
```
1910635
```

In [36]:
```
len(df_scores_by_school_spending['budget_per_student'])
```

Out[36]:
```
39170
```

In [37]:
```
# Create a new column to define the spending ranges per student

group_names = []

for x in range (len(df_scores_by_school_spending['budget_per_student'])):

    if (df_scores_by_school_spending.iloc[x][1]<=1081356) :
        group_names.append('<$1081356')

    elif (1081356<df_scores_by_school_spending.iloc[x][1]<=1910635) :
        group_names.append('$1081356-1910635')

    elif (1910635<df_scores_by_school_spending.iloc[x][1]<=3022020) :
        group_names.append('$1910635-3022020')

    elif (df_scores_by_school_spending.iloc[x][1]>3022020) :
        group_names.append('>$3022020')
```

In [38]:
```
df_scores_by_school_spending.insert(2,'group_names', group_names)
```

In [39]:
```
df_scores_by_school_spending
```

Out[39]:

	student_name	budget_per_student	group_names
0	Paul Bradley	1910635	$1081356-1910635
1	Victor Smith	1910635	$1081356-1910635
2	Kevin Rodriguez	1910635	$1081356-1910635
3	Dr. Richard Scott	1910635	$1081356-1910635
4	Bonnie Ray	1910635	$1081356-1910635
...	...	...	...
39165	Donna Howard	1043130	<$1081356
39166	Dawn Bell	1043130	<$1081356
39167	Rebecca Tanner	1043130	<$1081356
39168	Desiree Kidd	1043130	<$1081356
39169	Carolyn Jackson	1043130	<$1081356

39170 rows × 3 columns

In [40]:
```
# Calculate the average math score within each spending range


df_scores_by_school_spending.insert(3,'math_score', school_data_complete['math_
```

In [41]:
```
df_scores_by_school_spending.groupby('group_names')['math_score'].mean()
```

Out[41]:
```
group_names
$1081356-1910635    78.164034
$1910635-3022020    77.082696
<$1081356           83.436586
>$3022020           77.060183
Name: math_score, dtype: float64
```

In [42]:
```
df_scores_by_school_spending
```

Out[42]:

	student_name	budget_per_student	group_names	math_score
0	Paul Bradley	1910635	$1081356-1910635	79
1	Victor Smith	1910635	$1081356-1910635	61
2	Kevin Rodriguez	1910635	$1081356-1910635	60
3	Dr. Richard Scott	1910635	$1081356-1910635	58
4	Bonnie Ray	1910635	$1081356-1910635	84
...	...	...	...	...
39165	Donna Howard	1043130	<$1081356	90
39166	Dawn Bell	1043130	<$1081356	70
39167	Rebecca Tanner	1043130	<$1081356	84
39168	Desiree Kidd	1043130	<$1081356	90
39169	Carolyn Jackson	1043130	<$1081356	75

39170 rows × 4 columns

In [43]:
```
# Calculate the percentage passing rate for math in each spending range

math_pass_ = df_scores_by_school_spending[df_scores_by_school_spending['math_sc

m = (math_pass_.groupby('group_names').count()/df_scores_by_school_spending.grc
m
```

Out[43]:

group_names	student_name	budget_per_student	math_score
$1081356-1910635	72.336517	72.336517	72.336517
$1910635-3022020	66.574010	66.574010	66.574010
<$1081356	93.663606	93.663606	93.663606
>$3022020	66.375680	66.375680	66.375680

In [44]:
```
# Calculate the percentage passing rate for reading in each spending range

df_scores_by_school_spending.insert(4,'reading_score', school_data_complete['re
```

In [45]:
```
reading_pass_ = df_scores_by_school_spending[df_scores_by_school_spending['reac
r = (reading_pass_.groupby('group_names').count()/df_scores_by_school_spending.
r
```

Out[45]:

group_names	student_name	budget_per_student	math_score	reading_score
$1081356-1910635	83.844600	83.844600	83.844600	83.844600
$1910635-3022020	80.565207	80.565207	80.565207	80.565207
<$1081356	96.670366	96.670366	96.670366	96.670366
>$3022020	81.585704	81.585704	81.585704	81.585704

In [46]:
```
# Calculate the percentage overall passing rate in each spending range

o = (m+r)/2
o
```

Out[46]:

group_names	budget_per_student	math_score	reading_score	student_name
$1081356-1910635	78.090558	78.090558	NaN	78.090558
$1910635-3022020	73.569609	73.569609	NaN	73.569609
<$1081356	95.166986	95.166986	NaN	95.166986
>$3022020	73.980692	73.980692	NaN	73.980692

Scores by School Size

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

In [47]:
```
# 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 [48]:
```
# Create a new column for the bin groups
#size_bins = [0, 1000, 2000, 5000]
#group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

scores_by_school_size = {}


scores_by_school_size['student_name'] = school_data_complete['student_name']
scores_by_school_size['size'] = school_data_complete['size']


df_scores_by_school_size = pd.DataFrame.from_dict(scores_by_school_size)

#df_scores_by_school_size


group_names_ = []

for x in range (len(df_scores_by_school_size['size'])):

    if (df_scores_by_school_size.iloc[x][1]< 1000) :
```

```
        group_names_.append('<1000')

    elif (1000<df_scores_by_school_size.iloc[x][1]<=2000) :
        group_names_.append('1000-2000')

    elif (2000<df_scores_by_school_size.iloc[x][1]<=5000) :
        group_names_.append('2000-5000')

    elif (df_scores_by_school_size.iloc[x][1]>5000) :
        group_names_.append('>5000')

df_scores_by_school_size.insert(2,'group_names', group_names_)
```

In [49]:
```
df_scores_by_school_size
```

Out[49]:

	student_name	size	group_names
0	Paul Bradley	2917	2000-5000
1	Victor Smith	2917	2000-5000
2	Kevin Rodriguez	2917	2000-5000
3	Dr. Richard Scott	2917	2000-5000
4	Bonnie Ray	2917	2000-5000
...	...	...	...
39165	Donna Howard	1635	1000-2000
39166	Dawn Bell	1635	1000-2000
39167	Rebecca Tanner	1635	1000-2000
39168	Desiree Kidd	1635	1000-2000
39169	Carolyn Jackson	1635	1000-2000

39170 rows × 3 columns

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

In [50]:
```
df_scores_by_school_size.insert(3,'math_score', school_data_complete['math_scor
#df_scores_by_school_size.groupby('group_names').mean()
```

In [51]:
```
# math_pass_size
math_pass_size_ = df_scores_by_school_size[df_scores_by_school_size['math_score

math_pass_size_
```

Out[51]:

	student_name	size	group_names	math_score
0	Paul Bradley	2917	2000-5000	79
4	Bonnie Ray	2917	2000-5000	84
5	Bryan Miranda	2917	2000-5000	94
6	Sheena Carter	2917	2000-5000	80
8	Michael Roth	2917	2000-5000	87
...	...	...	...	...
39165	Donna Howard	1635	1000-2000	90
39166	Dawn Bell	1635	1000-2000	70
39167	Rebecca Tanner	1635	1000-2000	84
39168	Desiree Kidd	1635	1000-2000	90
39169	Carolyn Jackson	1635	1000-2000	75

29370 rows × 4 columns

In [52]:
```
df_scores_by_school_size.groupby('group_names').count()
```

Out[52]:

group_names	student_name	size	math_score
1000-2000	8522	8522	8522
2000-5000	29259	29259	29259
<1000	1389	1389	1389

In [53]:
```
m_= (math_pass_size_.groupby('group_names').count()/df_scores_by_school_size.gr

m_
#math_pass_size.groupby('group_names').count()
```

Out[53]:

group_names	student_name	size	math_score
1000-2000	93.616522	93.616522	93.616522
2000-5000	68.652380	68.652380	68.652380
<1000	93.952484	93.952484	93.952484

In [54]:
```
# read_pass_size
df_scores_by_school_size.insert(4,'reading_score', school_data_complete['readir


reading__pass = df_scores_by_school_size[df_scores_by_school_size['reading_scor
r_= (reading__pass.groupby('group_names').count()/df_scores_by_school_size.grou

r_
```

Out[54]:

group_names	student_name	size	math_score	reading_score
1000-2000	96.773058	96.773058	96.773058	96.773058
2000-5000	82.125158	82.125158	82.125158	82.125158
<1000	96.040317	96.040317	96.040317	96.040317

In [55]:
```
# Calculate the overall passing rate for different school size

o_ = (m_+r_)/2
o_
```

Out[55]:

group_names	math_score	reading_score	size	student_name
1000-2000	95.194790	NaN	95.194790	95.194790
2000-5000	75.388769	NaN	75.388769	75.388769
<1000	94.996400	NaN	94.996400	94.996400

Scores by School Type

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

In [56]:
```
# Create bins and groups, school type {'Charter', 'District'}

scores_by_school_type = {}


scores_by_school_type['student_name'] = school_data_complete['student_name']
scores_by_school_type['type'] = school_data_complete['type']


df_scores_by_school_type = pd.DataFrame.from_dict(scores_by_school_type)



group__names_ = []

for x in range (len(df_scores_by_school_type['type'])):

    if (df_scores_by_school_type.iloc[x][1] == 'Charter') :
        group__names_.append('Charter')

    else :
        group__names_.append('District')

df_scores_by_school_type.insert(2,'group_names', group__names_)
```

In [57]:
```
df_scores_by_school_type
```

Out[57]:

	student_name	type	group_names
0	Paul Bradley	District	District
1	Victor Smith	District	District
2	Kevin Rodriguez	District	District
3	Dr. Richard Scott	District	District
4	Bonnie Ray	District	District
...	...	...	...
39165	Donna Howard	Charter	Charter
39166	Dawn Bell	Charter	Charter
39167	Rebecca Tanner	Charter	Charter
39168	Desiree Kidd	Charter	Charter
39169	Carolyn Jackson	Charter	Charter

39170 rows × 3 columns

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

In [58]:
```
# math pass size
df_scores_by_school_type.insert(3,'math_score', school_data_complete['math_scor
math__pass_size_ = df_scores_by_school_type[df_scores_by_school_type['math_scor


_m_= (math__pass_size_.groupby('group_names').count()/df_scores_by_school_type.

_m_
```

Out[58]:

group_names	student_name	type	math_score
Charter	93.701821	93.701821	93.701821
District	66.518387	66.518387	66.518387

In [59]:
```
# reading pass size
df_scores_by_school_type.insert(4,'reading_score', school_data_complete['readir
reading__pass_size_ = df_scores_by_school_type[df_scores_by_school_type['readir


_r_= (reading__pass_size_.groupby('group_names').count()/df_scores_by_school_ty

_r_
```

Out[59]:

group_names	student_name	type	math_score	reading_score
Charter	96.645891	96.645891	96.645891	96.645891
District	80.905249	80.905249	80.905249	80.905249

In [60]:
```
# Calculate the overall passing rate
_o_ = (_m_ + _r_)/2
_o_
```

Out[60]:

group_names	math_score	reading_score	student_name	type
Charter	95.173856	NaN	95.173856	95.173856
District	73.711818	NaN	73.711818	73.711818