

# 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.

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

# Load in the files
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=["school_name", "school_name"])

# Display the count of each series in the DataFrame
print(school_data_complete.count())

# Display the count of columns and rows
school_data_complete.shape

Student ID      39170
student_name    39170
gender          39170
```

```
grade          39170
school_name    39170
reading_score  39170
math_score     39170
School ID     39170
type           39170
size           39170
budget         39170
dtype: int64

(39170, 11)
```

## 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

```
# Create a District Summary

# Total number of schools
number_of_schools = school_data_complete['School ID'].nunique()
print(number_of_schools)

15

# Total number of students
number_of_students = school_data_complete['Student ID'].nunique()
print(number_of_students)

39170
```

```

# Total budget
total_budget = school_data_complete['budget'].sum()
print(total_budget)

82932329558

# Average math score
avg_math_score = school_data_complete['math_score'].mean()
print(avg_math_score)

78.98537145774827

# Average reading score
avg_reading_score = school_data_complete['reading_score'].mean()
print(avg_reading_score)

81.87784018381414

# Overall passing rate
overall_passing_rate = (avg_math_score + avg_reading_score) / 2
print(overall_passing_rate)

80.43160582078121

# Percentage of students passing math (70 or greater)

# Find count of students with math score >= 70
students_passing_math = (school_data_complete['math_score'] >=
70).sum()
print(students_passing_math)

# Find percentage of students passing math
percent_passing_math = (students_passing_math / number_of_students) *
100
print(percent_passing_math)

29370
74.9808526933878

# Percentage of students passing reading (70 or greater)

# Find count of students with reading score >= 70
students_passing_reading = (school_data_complete['reading_score'] >=
70).sum()
print(students_passing_reading)

# Find percentage of students passing reading
percent_passing_reading = (students_passing_reading /
number_of_students) * 100
print(percent_passing_reading)

```

```

33610
85.80546336482001

# Create a DataFrame holding all of the newly calculated values

# Create district summary dictionary
district_summary = {
    'number_of_schools': number_of_schools,
    'number_of_students': number_of_students,
    'total_budget': total_budget,
    'avg_math_score': avg_math_score,
    'avg_reading_score': avg_reading_score,
    'overall_passing_rate': overall_passing_rate,
    'percent_passing_math': percent_passing_math,
    'percent_passing_reading': percent_passing_reading
}

# Convert dictionary to DataFrame
df_district_summary = pd.DataFrame([district_summary])
display(df_district_summary)

```

	number_of_schools	number_of_students	total_budget	avg_math_score
0	15	39170	82932329558	78.985371

  

	avg_reading_score	overall_passing_rate	percent_passing_math	\
0	81.87784	80.431606	74.980853	

  

	percent_passing_reading
0	85.805463

## 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

```
# Sort and display the top five schools in overall passing rate
```

```
# Calculate total school budget
```

```
total_school_budget = school_data_complete.groupby(  
    'school_name').first()['budget']  
print(total_school_budget)
```

```
school_name  
Bailey High School      3124928  
Cabrera High School     1081356  
Figueroa High School    1884411  
Ford High School        1763916  
Griffin High School      917500  
Hernandez High School   3022020  
Holden High School       248087  
Huang High School        1910635  
Johnson High School     3094650  
Pena High School         585858  
Rodriguez High School    2547363  
Shelton High School      1056600  
Thomas High School       1043130  
Wilson High School       1319574  
Wright High School       1049400  
Name: budget, dtype: int64
```

```
# Calculate per student budget
```

```
# Calculate count of students per school
```

```
student_counts = school_data_complete['school_name'].value_counts()
```

```
# Calculate budget per student
```

```
budget_per_student = total_school_budget / student_counts  
print(budget_per_student)
```

```
Bailey High School      628.0  
Cabrera High School     582.0  
Figueroa High School    639.0  
Ford High School        644.0  
Griffin High School      625.0  
Hernandez High School   652.0  
Holden High School       581.0  
Huang High School        655.0  
Johnson High School     650.0  
Pena High School         609.0  
Rodriguez High School    637.0  
Shelton High School      600.0  
Thomas High School       638.0  
Wilson High School       578.0
```

```
Wright High School      583.0  
dtype: float64
```

```
# Caculate the avg math and reading score
```

```
# Calculate avg math score per school
```

```
avg_math_score_per_school = school_data_complete.groupby(  
    'school_name')['math_score'].mean()  
print(avg_math_score_per_school)
```

```
# Calculate avg reading score per school
```

```
avg_reading_score_per_school = school_data_complete.groupby(  
    'school_name')['reading_score'].mean()  
print(avg_reading_score)
```

```
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  
81.87784018381414
```

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

```
# Find the total count of math scores by school
```

```
math_score_count_by_school = school_data_complete.groupby(  
    'school_name')['math_score'].count()  
print(math_score_count_by_school)
```

```
# Find the count for math result in each school that pass with 70 or higher
```

```
passing_math = school_data_complete['math_score'] >= 70  
passing_math_scores = school_data_complete[passing_math]  
passing_math_score_count_per_school = passing_math_scores.groupby(  
    'school_name')['math_score'].count()  
print(passing_math_score_count_per_school)
```

```
# Calculate the math passing rate
```

```

passing_math_rate_per_school = (passing_math_score_count_per_school /
                                math_score_count_by_school) * 100
print(passing_math_rate_per_school)

```

```

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: math_score, dtype: int64

```

```

school_name
Bailey High School      3318
Cabrera High School     1749
Figueroa High School    1946
Ford High School        1871
Griffin High School     1371
Hernandez High School   3094
Holden High School      395
Huang High School       1916
Johnson High School     3145
Pena High School        910
Rodriguez High School   2654
Shelton High School     1653
Thomas High School      1525
Wilson High School      2143
Wright High School      1680
Name: math_score, dtype: int64

```

```

school_name
Bailey High School      66.680064
Cabrera High School     94.133477
Figueroa High School    65.988471
Ford High School        68.309602
Griffin High School     93.392371
Hernandez High School   66.752967
Holden High School      92.505855
Huang High School       65.683922
Johnson High School     66.057551
Pena High School        94.594595
Rodriguez High School   66.366592

```

Shelton High School	93.867121
Thomas High School	93.272171
Wilson High School	93.867718
Wright High School	93.333333

Name: math\_score, dtype: float64

*# Find the total count of read result*

```
reading_score_count_by_school = school_data_complete.groupby(
    'school_name')['reading_score'].count()
print(reading_score_count_by_school)
```

*# Find the count for read result in each school that pass 70 or higher*

```
passing_reading = school_data_complete['reading_score'] >= 70
passing_reading_scores = school_data_complete[passing_reading]
passing_reading_score_count_per_school =
passing_reading_scores.groupby(
    'school_name')['reading_score'].count()
print(passing_reading_score_count_per_school)
```

*# Calculate the read passing rate*

```
passing_reading_rate_per_school =
    (passing_reading_score_count_per_school /
     reading_score_count_by_school) *
100
print(passing_reading_rate_per_school)
```

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: reading\_score, dtype: int64

school_name	
Bailey High School	4077
Cabrera High School	1803
Figueroa High School	2381
Ford High School	2172
Griffin High School	1426
Hernandez High School	3748



Holden High School	411
Huang High School	2372
Johnson High School	3867
Pena High School	923
Rodriguez High School	3208
Shelton High School	1688
Thomas High School	1591
Wilson High School	2204
Wright High School	1739

Name: reading\_score, dtype: int64  
school\_name

Bailey High School	81.933280
Cabrera High School	97.039828
Figueroa High School	80.739234
Ford High School	79.299014
Griffin High School	97.138965
Hernandez High School	80.862999
Holden High School	96.252927
Huang High School	81.316421
Johnson High School	81.222432
Pena High School	95.945946
Rodriguez High School	80.220055
Shelton High School	95.854628
Thomas High School	97.308869
Wilson High School	96.539641
Wright High School	96.611111

Name: reading\_score, dtype: float64

*# Calculate the overall passing rate (average of the math and reading passing rate)*

overall\_passing\_rate\_per\_school = (passing\_math\_rate\_per\_school +  
passing\_reading\_rate\_per\_school) /

2

print(overall\_passing\_rate\_per\_school)

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
dtype: float64
```

```
# Sort and display the top five schools in overall passing rate
top_schools =
overall_passing_rate_per_school.sort_values(ascending=False)
top_five_schools = top_schools.head()
print(top_five_schools)
```

```
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
dtype: float64
```

## Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

```
# Sort and display the worst five schools in overall passing rate
bottom_schools =
overall_passing_rate_per_school.sort_values(ascending=True)
bottom_five_schools = bottom_schools.head()
print(bottom_five_schools)
```

```
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
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

```
# Create table that lists the average math score for each school of each grade level.
```

```
# Calculate the average math score for 9th grade in each school

# Filter the data for 9th graders
ninth_graders = school_data_complete[school_data_complete['grade'] ==
'9th']

# Calculate the average math score for 9th graders by school
avg_9th_grade_math_score = ninth_graders.groupby(
    'school_name')['math_score'].mean()
print(avg_9th_grade_math_score)
```

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

Name: math\_score, dtype: float64

```
# Calculate the average math score for 10th grade in each school

# Filter the data for 10th graders
tenth_graders = school_data_complete[school_data_complete['grade'] ==
'10th']

# Calculate the average math score for 10th graders by school
avg_10th_grade_math_score = tenth_graders.groupby(
    'school_name')['math_score'].mean()
display(avg_10th_grade_math_score)
```

school_name	
Bailey High School	76.996772
Cabrera High School	83.154506
Figueroa High School	76.539974
Ford High School	77.672316
Griffin High School	84.229064
Hernandez High School	77.337408
Holden High School	83.429825
Huang High School	75.908735
Johnson High School	76.691117

```
Pena High School      83.372000
Rodriguez High School 76.612500
Shelton High School   82.917411
Thomas High School    83.087886
Wilson High School    83.724422
Wright High School    84.010288
Name: math_score, dtype: float64
```

```
# Calculate the average math score for 11th grade in each school
```

```
# Filter the data for 11th graders
```

```
eleventh_graders = school_data_complete[school_data_complete['grade']
==
                                     '11th']
```

```
# Calculate the average math score for 11th graders by school
```

```
avg_11th_grade_math_score = eleventh_graders.groupby(
    'school_name')['math_score'].mean()
display(avg_11th_grade_math_score)
```

```
school_name
Bailey High School      77.515588
Cabrera High School     82.765560
Figueroa High School    76.884344
Ford High School        76.918058
Griffin High School     83.842105
Hernandez High School   77.136029
Holden High School      85.000000
Huang High School       76.446602
Johnson High School     77.491653
Pena High School        84.328125
Rodriguez High School   76.395626
Shelton High School     83.383495
Thomas High School      83.498795
Wilson High School      83.195326
Wright High School      83.836782
Name: math_score, dtype: float64
```

```
# Calculate the average math score for 12th grade in each school
```

```
# Filter the data for 12th graders
```

```
twelfth_graders = school_data_complete[school_data_complete['grade']
== '12th']
```

```
# Calculate the average math score for 12th graders by school
```

```
avg_12th_grade_math_score = twelfth_graders.groupby(
    'school_name')['math_score'].mean()
display(avg_12th_grade_math_score)
```

```
school_name
Bailey High School      76.492218
```

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

Name: math\_score, dtype: float64

*# Create a DataFrame using the newly calculated series*

*# Create a dict with the newly calculated series*

```
school_avg_math_scores_by_grade = {
    'avg_9th_grade_math_score': avg_9th_grade_math_score,
    'avg_10th_grade_math_score': avg_10th_grade_math_score,
    'avg_11th_grade_math_score': avg_11th_grade_math_score,
    'avg_12th_grade_math_score': avg_12th_grade_math_score
}
```

*# Convert dict into DataFrame*

```
df_school_avg_math_scores_by_grade = pd.DataFrame(
    school_avg_math_scores_by_grade)
display(df_school_avg_math_scores_by_grade.head())
```

	avg_9th_grade_math_score
avg_10th_grade_math_score \	
school_name	

Bailey High School	77.083676
76.996772	
Cabrera High School	83.094697
83.154506	
Figueroa High School	76.403037
76.539974	
Ford High School	77.361345
77.672316	
Griffin High School	82.044010
84.229064	

	avg_11th_grade_math_score
avg_12th_grade_math_score	
school_name	

Bailey High School	77.515588
76.492218	
Cabrera High School	82.765560
83.277487	
Figueroa High School	76.884344
77.151369	
Ford High School	76.918058
76.179963	
Griffin High School	83.842105
83.356164	

## Reading Score by Grade

- Perform the same operations as above for reading scores

*# Create table that lists the average reading score for each school of each grade level.*

*# Calculate the average reading score for 9th grade in each school*

```
avg_9th_grade_reading_score = ninth_graders.groupby(
    'school_name')['reading_score'].mean()
display(avg_9th_grade_reading_score)
```

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

Name: reading\_score, dtype: float64

*# Calculate the average reading score for 10th grade in each school*

```
avg_10th_grade_reading_score = tenth_graders.groupby(
    'school_name')['reading_score'].mean()
display(avg_10th_grade_reading_score)
```

school_name	
Bailey High School	80.907183
Cabrera High School	84.253219
Figueroa High School	81.408912
Ford High School	81.262712

Griffin High School	83.706897
Hernandez High School	80.660147
Holden High School	83.324561
Huang High School	81.512386
Johnson High School	80.773431
Pena High School	83.612000
Rodriguez High School	80.629808
Shelton High School	83.441964
Thomas High School	84.254157
Wilson High School	84.021452
Wright High School	83.812757

Name: reading\_score, dtype: float64

```
# Calculate the average reading score for 11th grade in each school
avg_11th_grade_reading_score = eleventh_graders.groupby(
    'school_name')['reading_score'].mean()
display(avg_11th_grade_reading_score)
```

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

Name: reading\_score, dtype: float64

```
# Calculate the average reading score for 12th grade in each school
avg_12th_grade_reading_score = twelfth_graders.groupby(
    'school_name')['reading_score'].mean()
display(avg_12th_grade_reading_score)
```

school_name	
Bailey High School	80.912451
Cabrera High School	84.287958
Figueroa High School	81.384863
Ford High School	80.662338
Griffin High School	84.013699
Hernandez High School	80.857143
Holden High School	84.698795
Huang High School	80.305983

```

Johnson High School      81.227564
Pena High School         84.591160
Rodriguez High School    80.376426
Shelton High School      82.781671
Thomas High School       83.831361
Wilson High School       84.317673
Wright High School       84.073171
Name: reading_score, dtype: float64

```

```
# Create a DataFrame that holds all the newly calculated values
```

```
# Create a dict with the newly calculated series
```

```

school_avg_reading_scores_by_grade = {
    'avg_9th_grade_reading_score': avg_9th_grade_reading_score,
    'avg_10th_grade_reading_score': avg_10th_grade_reading_score,
    'avg_11th_grade_reading_score': avg_11th_grade_reading_score,
    'avg_12th_grade_reading_score': avg_12th_grade_reading_score
}

```

```
# Convert dict into DataFrame
```

```

df_school_avg_reading_scores_by_grade = pd.DataFrame(
    school_avg_reading_scores_by_grade)
display(df_school_avg_reading_scores_by_grade.head())

```

```

                                avg_9th_grade_reading_score \
school_name
Bailey High School                        81.303155
Cabrera High School                      83.676136
Figueroa High School                     81.198598
Ford High School                         80.632653
Griffin High School                      83.369193

```

```

                                avg_10th_grade_reading_score \
school_name
Bailey High School                        80.907183
Cabrera High School                      84.253219
Figueroa High School                     81.408912
Ford High School                         81.262712
Griffin High School                      83.706897

```

```

                                avg_11th_grade_reading_score \
school_name
Bailey High School                        80.945643
Cabrera High School                      83.788382
Figueroa High School                     80.640339
Ford High School                         80.403642
Griffin High School                      84.288089

```

```

                                avg_12th_grade_reading_score
school_name

```



Bailey High School	80.912451
Cabrera High School	84.287958
Figueroa High School	81.384863
Ford High School	80.662338
Griffin High School	84.013699

## 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)

```
# 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"]

# Create a new column to show budget per student in each row

# Calculate the spending per student
school_data_complete['spending_per_student'] = school_data_complete[
    'budget'] / school_data_complete['size']

# Categorize spending based on the bins
school_data_complete['Spending Ranges (Per Student)'] = pd.cut(
    school_data_complete['spending_per_student'],
    spending_bins,
    labels=group_names,
    include_lowest=True)

# Create a new column to define the spending ranges per student
grouped_spending = school_data_complete.groupby(
    'Spending Ranges (Per Student)')

# Calculate the average math score within each spending range
average_math_score = grouped_spending['math_score'].mean()

# Calculate the average reading score within each spending range
average_reading_score = grouped_spending['reading_score'].mean()

# Calculate the percentage passing rate for math in each spending
range
percentage_passing_math = school_data_complete[
    school_data_complete['math_score'] >= 70].groupby(
    'Spending Ranges (Per Student)').count()['student_name']
```

```

# Calculate the percentage passing rate for reading in each spending
range
percentage_passing_reading = school_data_complete[
    school_data_complete['reading_score'] >= 70].groupby(
    'Spending Ranges (Per Student)').count()['student_name']

# Calculate the percentage overall passing rate in each spending range
percentage_overall_passing_rate = (percentage_passing_reading +
    percentage_passing_math) / 2

# Assemble into DataFrame
scores_by_schools_spending = pd.DataFrame({
    'Spending Ranges (Per Student)':
    average_math_score.index,
    'Average Math Score':
    average_math_score,
    'Average Reading Score':
    average_reading_score,
    '% Passing Math':
    percentage_passing_math,
    '% Passing Reading':
    percentage_passing_reading,
    '% Overall Passing':
    percentage_overall_passing_rate
})

display(scores_by_schools_spending)

```

Spending Ranges (Per Student) \	
Spending Ranges (Per Student)	
<\$585	<\$585
\$585-615	\$585-615
\$615-645	\$615-645
\$645-675	\$645-675

  

Average Math Score    Average Reading	
Score \	Spending Ranges (Per Student)
<\$585	83.363065
83.964039	
\$585-615	83.529196
83.838414	
\$615-645	78.061635
81.434088	
\$645-675	77.049297
81.005604	

  

% Passing Math    % Passing Reading \	
Spending Ranges (Per Student)	

<\$585	5967	6157
\$585-615	2563	2611
\$615-645	12685	14855
\$645-675	8155	9987

% Overall Passing	
Spending Ranges (Per Student)	
<\$585	6062.0
\$585-615	2587.0
\$615-645	13770.0
\$645-675	9071.0

## Scores by School Size

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

```
# 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)"]

# Create a new column for the bin groups

school_data_complete['School Size'] = pd.cut(
    school_data_complete['size'], size_bins, labels=group_names,
    include_lowest=True)

# Group by school size
grouped_size = school_data_complete.groupby('School Size')

# Calculate the average scores and passing rates for each size category
average_math_score_size = grouped_size['math_score'].mean()
average_reading_score_size = grouped_size['reading_score'].mean()
```

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

```
# math_pass_size
# Calculate the total count of students passing math and reading separately
math_pass_size =
school_data_complete[school_data_complete['math_score'] >=
70].groupby(
    'School Size').count()['student_name']

# read_pass_size
read_pass_size =
school_data_complete[school_data_complete['reading_score'] >=
70].groupby(
    'School Size').count()['student_name']
```

```

# Calculate the percentage of students passing math and reading
percentage_passing_math_size = (math_pass_size / grouped_size['Student ID'].count()) * 100
percentage_passing_reading_size = (read_pass_size / grouped_size['Student ID'].count()) * 100

# Calculate the overall passing rate for different school size
overall_pass_rate_size = (
    percentage_passing_math_size + percentage_passing_reading_size) / 2

# Assemble into DataFrame
size_summary = pd.DataFrame({
    "Average Math Score": average_math_score_size,
    "Average Reading Score": average_reading_score_size,
    "% Passing Math": percentage_passing_math_size,
    "% Passing Reading": percentage_passing_reading_size,
    "% Overall Passing": overall_pass_rate_size
})

display(size_summary)

```

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

## Scores by School Type

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

```

# Group by school type
grouped_type = school_data_complete.groupby('type')

# Calculate the average scores for each school type
average_math_score_type = grouped_type['math_score'].mean()
average_reading_score_type = grouped_type['reading_score'].mean()

```

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

```

# math pass size
math_pass_size =
school_data_complete[school_data_complete['math_score'] >=
70].groupby(
    'type').count()['student_name']

# reading pass size
read_pass_size =
school_data_complete[school_data_complete['reading_score'] >=
70].groupby(
    'type').count()['student_name']

# Calculate the percentage of students passing math and reading
percentage_passing_math_type = (
    math_pass_size / grouped_type['Student ID'].count()) * 100
percentage_passing_reading_type = (
    read_pass_size / grouped_type['Student ID'].count()) * 100

# Calculate the overall passing rate
# Calculate the overall passing rate for different school types
overall_pass_rate_type = (
    percentage_passing_math_type + percentage_passing_reading_type) /
2

# Assemble into DataFrame
type_summary = pd.DataFrame({
    "Average Math Score": average_math_score_type,
    "Average Reading Score": average_reading_score_type,
    "% Passing Math": percentage_passing_math_type,
    "% Passing Reading": percentage_passing_reading_type,
    "% Overall Passing": overall_pass_rate_type
})

display(type_summary)

```

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