This project used information from two files:

* School Data - CSV detailing the type, size, and budget of schools
* Student Data - CSV detailing student information and scores for reading and math

After merging the CSVs into DataFrame, analysis commenced to determine the detailed info below. Main conclusions from the data are…

* There is an inverse correlation in school spending per capita and performance, and it is much stronger with math. The highest spending schools had a passing rate just barely over 50%.
* Performance is strong in small and medium schools, but poor in large schools. Math performance accentuates the problem.
* Charter schools are much stronger than district schools. Charter schools perform very strong overall with 90% passing, while district schools have an alarming 54% passing rate. Math scores weigh much harder than reading scores in the discrepancy.
* Low spending, small charter schools appear have the highest performing students
* Factors of spending, size, and school type weigh much harder on math scores than reading scores
* Large district schools are the worst performers, especially in math

**District Summary**

* Total number of unique schools – 15
  + .value\_counts function on ‘school name’ column and returned length
* Total number of students
  + .value\_counts function on ‘student ID’ column and returned length
* Total budget
  + .unique function on ‘budget’ column and returned sum
* Average (mean) math score
  + .mean function on ‘math score’ column
* Average (mean) reading score
  + .mean function on ‘reading score’ column
* Percentage of students who passed math
  + .count function for number of students with ‘math score’ column >= 70, divided by total students from above and multiplied by 100
* Percentage of students who passed reading
  + .count function for number of students with ‘reading score’ column >= 70, divided by total students from above and multiplied by 100
* Percentage of students that passed both math and reading
  + .count function for number of students with both ‘math score’ and ‘reading score’ >= 70, divided by total students from above and multiplied by 100
* New DataFrame for the above calculations called district\_summary
  + pd.DataFrame function with columns for each calculation above

**School Summary**

* Return school type for each school
  + .set\_index function to create a series with index on ‘school name’ and value on ‘type’
* Total student count for each school
  + .value\_counts function on ‘school name’ to return number of rows for each school
* Per capita spending
  + First, used .groupby function to group data by school, then pulled average of budget column (every budget value the same for each row of a school)
  + Then, divided school budget by school count
* Average test scores (math)
  + .groupby function to group by school, then pulled average of ‘math score’
* Average test scores (reading)
  + .groupby function to group by school, then pulled average of ‘reading score’
* Number of schools with math scores of 70 or higher
  + Filtered data for rows where math score was >= 70, then group by school name
* Number of schools with reading scores of 70 or higher
  + Filtered data for rows where reading score was >= 70, then group by school name
* Schools that passed both math and reading with scores of 70 or higher
  + Filtered for rows where both math and reading were >= 70, then group by school name
* Passing rates
  + Students passing math from above / total school count from above
  + Students passing reading from above / total school count from above
  + Students passing both math and reading from above / total school count from above
* Create a new DataFrame for the above calculations called per\_school\_summary
  + pd.DataFrame function with columns for each of the above

**Highest-Performing Schools by Percentage of Overall Passing**

* Sort the schools by % Overall Passing in descending order
  + .sort\_values function on ‘% overall passing’ column, ascending=False
* Save the results to a DataFrame called top\_schools
* Display the first 5 rows
  + .head function

**Lowest-Performing Schools by Percentage of Overall Passing**

* Sort the schools by % Overall Passing in ascending order
  + .sort\_values function on ‘% overall passing’
* Save the results to a DataFrame called bottom\_schools
* Display the first 5 rows
  + .head function

**Math Scores by Grade**

* Separate the data by grade
  + Filtered for 9th graders by “grade” == “9th”, then did same for 10th-11th grade
* Group by "school\_name" and take the mean of each
  + .groupby function to group by ‘school name’ then returned mean on ‘math score’
* Select only the math\_score
* Combine scores above into single DataFrame called math\_scores\_by\_grade
  + pd.DataFrame with four columns on scores from 9th-12th grade

**Reading Scores by Grade**

* Separate the data by grade
  + Filtered for 9th graders by “grade” == “9th”, then did same for 10th-11th grade
* Group by "school\_name" and take the mean of each
  + .groupby function to group by ‘school name’ then returned mean on ‘reading score’
* Select only the reading\_score
* Combine scores above into single DataFrame called reading\_scores\_by\_grade
  + pd.DataFrame with four columns on scores from 9th-12th grade

**Scores by School Spending**

* Bin the data by the spending ranges
  + Created four bins off ‘per student budget’ column, converted column to float, and then pd.cut to bin onto DataFrame
* Calculate the averages
  + .mean function to average for every bin
* spending\_summary DataFrame using the binned and averaged spending data
  + pd.DataFrame to create table for each bin

**Scores by School Size**

* Bin the data by the school sizes
  + Created three bins off ‘school size’ column, then pd.cut to bin into DataFrame
* Calculate the averages
  + .mean function to average for every bin
* size\_summary DataFrame using the binned and averaged size data
  + pd.DataFrame to create table for every bin

**Scores by School Type**

* Group the per\_school\_summary DataFrame by "School Type" and average the results
  + groupby function to group data by ‘school type’ column, then returned mean
* New DataFrame called type\_summary that uses the new column data
  + pd.DataFrame to create table for both school types

**Written Report**

* Summarizes the analysis
* Draws two correct conclusions or comparisons from the calculations