PyCity Schools Analysis

- Your analysis here

In [1]:
```
# Dependencies and Setup
import pandas as pd
from pathlib import Path

# File to Load (Remember to Change These)
school_data_to_load = Path("Resources/schools_complete.csv")
student_data_to_load = Path("Resources/students_complete.csv")

# Read School and Student Data File and store into Pandas DataFrames
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_
school_data_complete.head()
```

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

In [2]:
```
# Calculate the total number of unique schools
school_count =  len(school_data_complete["school_name"].unique())
school_count
```

Out[2]:
```
15
```

In [3]:
```
# Calculate the total number of students
student_count =  len(school_data_complete["Student ID"].unique())
student_count
```

Out[3]:
```
39170
```

In [4]:
```
# Calculate the total budget
total_budget = school_data["budget"].sum()
```

```
total_budget
```

Out[4]:
```
24649428
```

In [5]:
```
# Calculate the average (mean) math score
average_math_score = school_data_complete["math_score"].mean()
average_math_score
```

Out[5]:
```
78.98537145774827
```

In [6]:
```
# Calculate the average (mean) reading score
average_reading_score = school_data_complete["reading_score"].mean()

average_reading_score
```

Out[6]:
```
81.87784018381414
```

In [7]:
```
# Use the following to calculate the percentage of students who passed math (math s
passing_math_count = school_data_complete[(school_data_complete["math_score"] >= 7(
passing_math_percentage = passing_math_count / float(student_count) * 100
passing_math_percentage
```

Out[7]:
```
74.9808526933878
```

In [10]:
```
# Calculate the percentage of students who passed reading (hint: look at how the ma
passing_reading_count = student_data.loc[student_data['reading_score'] >= 70]['read
passing_reading_percentage = passing_reading_count/student_count*100
passing_reading_percentage
```

Out[10]:
```
85.80546336482001
```

In [11]:
```
# Use the following to calculate the percentage of students that passed math and re
passing_math_reading_count = school_data_complete[
    (school_data_complete["math_score"] >= 70) & (school_data_complete["reading_sc
].count()["student_name"]
overall_passing_rate = passing_math_reading_count /  float(student_count) * 100
overall_passing_rate
```

Out[11]:
```
65.17232575950983
```

In [15]:
```
# Create a high-level snapshot of the district's key metrics in a DataFrame
district_summary = pd.DataFrame({"Total Schools": [school_count],
                                 "Total Students": [student_count],
                                 "Total Budget" : total_budget,
                                 "Average Math Score": average_math_score,
                                 "Average Reading Score": average_reading_score,
                                 "% Passing Math":passing_math_percentage,
                                 "% Passing Reading":passing_reading_percentage,
                                 "% Overall Passing":overall_passing_rate})

# Formatting
district_summary["Total Students"] = district_summary["Total Students"].map("{:,}"
district_summary["Total Budget"] = district_summary["Total Budget"].map("${:,.2f}"

# Display the DataFrame
district_summary
```

Out[15]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
0	15	39,170	$24,649,428.00	78.985371	81.87784	74.980853	85.805463	65.172326

School Summary

In [18]:
```
# Use the code provided to select all of the school types
school_types = school_data.set_index('school_name')['type']
```

In [22]:
```
# Calculate the total student count per school
#groups by school
by_school = school_data_complete.set_index('school_name').groupby(['school_name'])

per_school_counts = by_school['Student ID'].count()
```

In [28]:
```
# Calculate the total school budget and per capita spending per school
per_school_budget = school_data.set_index('school_name')['budget']

per_school_capita = school_data.set_index('school_name')['budget']/school_data.set_
```

In [30]:
```
# Calculate the average test scores per school
per_school_math = by_school['math_score'].mean()
per_school_reading = by_school['reading_score'].mean()
```

In [77]:
```
# Calculate the number of students per school with math scores of 70 or higher
students_passing_math = school_data_complete[school_data_complete['math_score'] >=
school_students_passing_math = students_passing_math.groupby('school_name')['Stude
```

In [78]:
```
# Calculate the number of students per school with reading scores of 70 or higher
students_passing_reading = school_data_complete[school_data_complete['reading_score
school_students_passing_reading = students_passing_reading.groupby('school_name')[
#school_data_complete[school_data_complete['reading_score'] >= 70].groupby('school_
```

In [79]:
```
# Use the provided code to calculate the number of students per school that passed
students_passing_math_and_reading = school_data_complete[
    (school_data_complete["reading_score"] >= 70) & (school_data_complete["math_sc
]
school_students_passing_math_and_reading = students_passing_math_and_reading.groupb
```

In [80]:
```
# Use the provided code to calculate the passing rates
per_school_passing_math = school_students_passing_math / per_school_counts * 100
per_school_passing_reading = school_students_passing_reading / per_school_counts *
overall_passing_rate = school_students_passing_math_and_reading / per_school_counts
```

In [81]:
```
# Create a DataFrame called `per_school_summary` with columns for the calculations
per_school_summary =  pd.DataFrame({
    "School Type": school_types,
    "Total Students": per_school_counts,
    "Per Student Budget": per_school_capita,
    "Total School Budget": per_school_budget,
    "Average Math Score": per_school_math,
    "Average Reading Score": per_school_reading,
    '% Passing Math': per_school_passing_math,
    '% Passing Reading': per_school_passing_reading,
    "Overall Passing Rate": overall_passing_rate
})
```

```
# Formatting
per_school_summary["Total School Budget"] = per_school_summary["Total School Budge
per_school_summary["Per Student Budget"] = per_school_summary["Per Student Budget"

# Display the DataFrame
per_school_summary
```

Out[81]:

school_name	School Type	Total Students	Per Student Budget	Total School Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
Bailey High School	District	4976	$628.00	$3,124,928.00	77.048432	81.033963	66.680064	81.93328(
Cabrera High School	Charter	1858	$582.00	$1,081,356.00	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	$639.00	$1,884,411.00	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	$644.00	$1,763,916.00	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	$625.00	$917,500.00	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	$652.00	$3,022,020.00	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	$581.00	$248,087.00	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	$655.00	$1,910,635.00	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	$650.00	$3,094,650.00	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	$609.00	$585,858.00	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	$637.00	$2,547,363.00	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	$600.00	$1,056,600.00	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	$638.00	$1,043,130.00	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	$578.00	$1,319,574.00	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	$583.00	$1,049,400.00	83.682222	83.955000	93.333333	96.611111

Highest-Performing Schools (by % Overall Passing)

In [83]:
```
#data munging
sch_summary = per_school_summary[['School Type',
                                  'Total Students',
```

```
                            'Total School Budget',
                            'Per Student Budget',
                            'Average Math Score',
                            'Average Reading Score',
                            '% Passing Math',
                            '% Passing Reading',
                            'Overall Passing Rate']]

#data formatting
sch_summary.style.format({'Total Students': '{:,}',
                            "Total School Budget": "${:,}",
                            "Per Student Budget": "${:.0f}",
                            'Average Math Score': "{:.1f}",
                            'Average Reading Score': "{:.1f}",
                            "% Passing Math": "{:.1%}",
                            "% Passing Reading": "{:.1%}",
                            "Overall Passing Rate": "{:.1%}"})

# Sort the schools by `% Overall Passing` in descending order and display the top 5

top_schools = sch_summary.sort_values("Overall Passing Rate", ascending = False)
top_schools.head(5)
```

Out[83]:

school_name	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946

Bottom Performing Schools (By % Overall Passing)

In [85]:

```
# Sort the schools by `% Overall Passing` in ascending order and display the top 5
bottom_schools = top_schools.tail()
bottom_schools = bottom_schools.sort_values('Overall Passing Rate')
bottom_schools.head(5)
```

Out[85]:

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	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752967	80.862999
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432

Math Scores by Grade

In [92]:

```
# Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] == "11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] == "12th")]

# Group by `school_name` and take the mean of the `math_score` column for each.
ninth_grade_math_scores = ninth_graders.groupby('school_name')["math_score"].mean(
tenth_grader_math_scores = tenth_graders.groupby('school_name')["math_score"].mean
eleventh_grader_math_scores = eleventh_graders.groupby('school_name')["math_score"
twelfth_grader_math_scores = twelfth_graders.groupby('school_name')["math_score"].

# Combine each of the scores above into single DataFrame called `math_scores_by_gr
math_scores_by_grade =  pd.DataFrame({
        "9th": ninth_grade_math_scores,
        "10th": tenth_grader_math_scores,
        "11th": eleventh_grader_math_scores,
        "12th": twelfth_grader_math_scores
})



# Minor data wrangling
math_scores_by_grade.index.name = None

# Display the DataFrame
math_scores_by_grade
```

Out[92]:

	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

Reading Score by Grade

In [91]:

```
# Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] == "11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] == "12th")]

# Group by `school_name` and take the mean of the the `reading_score` column for ea
ninth_grade_reading_scores = ninth_graders.groupby('school_name')["reading_score"]
tenth_grader_reading_scores = tenth_graders.groupby('school_name')["reading_score"
eleventh_grader_reading_scores = eleventh_graders.groupby('school_name')["reading_s
twelfth_grader_reading_scores = twelfth_graders.groupby('school_name')["reading_sc

# Combine each of the scores above into single DataFrame called `reading_scores_by_
reading_scores_by_grade = pd.DataFrame({
        "9th": ninth_grade_reading_scores,
        "10th": tenth_grader_reading_scores,
        "11th": eleventh_grader_reading_scores,
        "12th": twelfth_grader_reading_scores
})

# Minor data wrangling
reading_scores_by_grade = reading_scores_by_grade[["9th", "10th", "11th", "12th"]]
reading_scores_by_grade.index.name = None

# Display the DataFrame
reading_scores_by_grade
```

Out[91]:

	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

Scores by School Spending

In [155…

```
# Establish the bins
spending_bins = [0, 585, 630, 645, 680]
labels = ["<$585", "$585-630", "$630-645", "$645-680"]
```

In [168…

```
# Create a copy of the school summary since it has the "Per Student Budget"
school_spending_df = per_school_summary.copy()
```

In [169…

```
print(school_spending_df["Per Student Budget"].dtype)
school_spending_df["Per Student Budget"] = school_spending_df["Per Student Budget"
school_spending_df["Per Student Budget"] = pd.to_numeric(school_spending_df["Per S
```

```
object
```

In [172…

```
# Use `pd.cut` to categorize spending based on the bins.
per_school_summary["Spending Ranges (Per Student)"] = pd.cut(school_spending_df["P

school_spending_df
```

Out[172]:

school_name	School Type	Total Students	Per Student Budget	Total School Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
Bailey High School	District	4976	628.0	$3,124,928.00	77.048432	81.033963	66.680064	81.93328(
Cabrera High School	Charter	1858	582.0	$1,081,356.00	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	639.0	$1,884,411.00	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	644.0	$1,763,916.00	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	625.0	$917,500.00	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	652.0	$3,022,020.00	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	581.0	$248,087.00	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	655.0	$1,910,635.00	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	650.0	$3,094,650.00	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	609.0	$585,858.00	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	637.0	$2,547,363.00	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	600.0	$1,056,600.00	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	638.0	$1,043,130.00	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	578.0	$1,319,574.00	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	583.0	$1,049,400.00	83.682222	83.955000	93.333333	96.611111

In [173…

```
#  Calculate averages for the desired columns.
spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per Student)"
spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per Student
spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per Student)"
spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per Stude
overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per Stude
```

In [174…

```
# Assemble into DataFrame
spending_summary = pd.DataFrame({
    "Average Math Score": spending_math_scores,
    "Average Reading Score": spending_reading_scores,
    '% Passing Math': spending_passing_math,
    '% Passing Reading': spending_passing_reading,
```

```
    "Overall Passing Rate": overall_passing_spending

})


# Display results
spending_summary
```

Out[174]:

Spending Ranges (Per Student)	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
<$585	83.455399	83.933814	93.460096	96.610877	90.369459
$585-630	81.899826	83.155286	87.133538	92.718205	81.418596
$630-645	78.518855	81.624473	73.484209	84.391793	62.857656
$645-680	76.997210	81.027843	66.164813	81.133951	53.526855

Scores by School Size

In [123…

```
# Establish the bins.
size_bins = [0, 1000, 2000, 5000]
labels = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

In [130…

```
# Categorize the spending based on the bins
# Use `pd.cut` on the "Total Students" column of the `per_school_summary` DataFrame

per_school_summary["School Size"] = pd.cut(school_spending_df["Total Students"], s
per_school_summary
```

Out[130]:

school_name	School Type	Total Students	Per Student Budget	Total School Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
Bailey High School	District	4976	$628.00	$3,124,928.00	77.048432	81.033963	66.680064	81.93328(
Cabrera High School	Charter	1858	$582.00	$1,081,356.00	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	$639.00	$1,884,411.00	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	$644.00	$1,763,916.00	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	$625.00	$917,500.00	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	$652.00	$3,022,020.00	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	$581.00	$248,087.00	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	$655.00	$1,910,635.00	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	$650.00	$3,094,650.00	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	$609.00	$585,858.00	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	$637.00	$2,547,363.00	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	$600.00	$1,056,600.00	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	$638.00	$1,043,130.00	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	$578.00	$1,319,574.00	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	$583.00	$1,049,400.00	83.682222	83.955000	93.333333	96.611111

In [133…

```
# Calculate averages for the desired columns.
size_math_scores = per_school_summary.groupby(["School Size"])["Average Math Score
```

```
size_reading_scores = per_school_summary.groupby(["School Size"])["Average Reading
size_passing_math = per_school_summary.groupby(["School Size"])["% Passing Math"].
size_passing_reading = per_school_summary.groupby(["School Size"])["% Passing Read
size_overall_passing = per_school_summary.groupby(["School Size"])["Overall Passing
```

In [134…

```
# Create a DataFrame called `size_summary` that breaks down school performance base
# Use the scores above to create a new DataFrame called `size_summary`
size_summary = pd.DataFrame({
    "Average Math Score": size_math_scores,
    "Average Reading Score": size_reading_scores,
    '% Passing Math': size_passing_math,
    '% Passing Reading': size_passing_reading,
    "Overall Passing Rate": size_overall_passing

})

# Display results
size_summary
```

Out[134]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Small (<1000)	83.821598	83.929843	93.550225	96.099437	89.883853
Medium (1000-2000)	83.374684	83.864438	93.599695	96.790680	90.621535
Large (2000-5000)	77.746417	81.344493	69.963361	82.766634	58.286003

Scores by School Type

In [136…

```
# Group the per_school_summary DataFrame by "School Type" and average the results.
average_math_score_by_type = per_school_summary.groupby(["School Type"])["Average M
average_reading_score_by_type = per_school_summary.groupby(["School Type"])["Averag
average_percent_passing_math_by_type = per_school_summary.groupby(["School Type"])
average_percent_passing_reading_by_type = per_school_summary.groupby(["School Type
average_percent_overall_passing_by_type = per_school_summary.groupby(["School Type
```

In [137…

```
# Assemble the new data by type into a DataFrame called `type_summary`
type_summary = pd.DataFrame({
    "Average Math Score": average_math_score_by_type,
    "Average Reading Score": average_reading_score_by_type,
    '% Passing Math': average_percent_passing_math_by_type,
    '% Passing Reading': average_percent_passing_reading_by_type,
    "Overall Passing Rate": average_percent_overall_passing_by_type})


# Display results
type_summary
```

Out[137]:

School Type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Charter	83.473852	83.896421	93.620830	96.586489	90.432244
District	76.956733	80.966636	66.548453	80.799062	53.672208

In [ ]: