

Pandas-Challenge, PyCitySchools

Well done! Having spent years analyzing financial records for big banks, you've finally scratched your idealistic itch and joined the education sector. In your latest role, you've become the Chief Data Scientist for your city's school district. In this capacity, you'll be helping the school board and mayor make strategic decisions regarding future school budgets and priorities.

```

In [58]: # Dependencies and Setup
import pandas as pd
import numpy as np

#Examine CSV files to review data

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

# Read School and Student Data File and store into two separate 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 using merge function
school_data_complete_df = pd.merge(student_data, school_data, how="left", on=["school_name", "school_name"])
school_data_complete_df

```

Out[58]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score	School ID	type	size	budget
0	0	Paul Bradley	M	9th	Huang High School	66	79	0	District	2917	1910635
1	1	Victor Smith	M	12th	Huang High School	94	61	0	District	2917	1910635
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0	District	2917	1910635
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	0	District	2917	1910635
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0	District	2917	1910635
...
39165	39165	Donna Howard	F	12th	Thomas High School	99	90	14	Charter	1635	1043130
39166	39166	Dawn Bell	F	10th	Thomas High School	95	70	14	Charter	1635	1043130
39167	39167	Rebecca Tanner	F	9th	Thomas High School	73	84	14	Charter	1635	1043130
39168	39168	Desiree Kidd	F	10th	Thomas High School	99	90	14	Charter	1635	1043130
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95	75	14	Charter	1635	1043130

39170 rows × 11 columns

1. 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 percentage of students with a passing math score (70 or greater)
- Calculate the percentage of students with a passing reading score (70 or greater)
- Calculate the percentage of students who passed math **and** reading (% Overall Passing)
- Create a dataframe to hold the above results
- Optional: give the displayed data cleaner formatting

```
In [59]: #total schools
total_schools = len(school_data["school_name"])

#total Students
total_students = school_data_complete_df["Student ID"].count()

#total_budget
total_budget = school_data["budget"].sum()

#test score averages
avg_math_score = school_data_complete_df["math_score"].mean()
avg_reading_score = school_data_complete_df["reading_score"].mean()
#Percent Passed

percent_passed_math = ((school_data_complete_df.loc[school_data_complete_df["math_score"] >= 70, "math_score"]
]).count()/total_students)
percent_passed_reading = ((school_data_complete_df.loc[school_data_complete_df["reading_score"] >= 70, "reading_score"]
]).count()/total_students)
#must pass both, so out of those who passed math, how many passed reading?
first_passed_math = school_data_complete_df.loc[school_data_complete_df["math_score"] >= 70, ["reading_score"]
]
then_passed_reading = (first_passed_math.loc[first_passed_math["reading_score"] >= 70, "reading_score"]).count()
percent_overall_passed = (then_passed_reading/total_students)

# Create district summary dataframe
district_summary = pd.DataFrame({
    "Total Schools": [total_schools],
    "Total Students": [total_students],
    "Total Budget": [total_budget],
    "Average Math Score": [avg_math_score],
    "Average Reading Score": [avg_reading_score],
    "% Passing Math": [percent_passed_math],
    "% Passing Reading": [percent_passed_reading],
    "% Overall Passing": [percent_overall_passed]})

#Format, style, display
district_summary.style.format({
    "Total Budget": "${:,.2f}",
    "Total Students": "{:,}",
    "Average Reading Score": "{:.1f}",
    "Average Math Score": "{:.1f}",
```

```

"% Passing Math": "{:.1%}",
"% Passing Reading": "{:.1%}",
"% Overall Passing": "{:.1%}"})

```

Out[59]:

	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	79.0	81.9	75.0%	85.8%	65.2%

2. 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 (The percentage of students that passed math **and** reading.)
- Create a dataframe to hold the above results

```
In [60]: #School name set to index
school_groups = school_data_complete_df.set_index('school_name').groupby(['school_name'])

#school type
school_type = school_data.set_index('school_name')['type']

# total students by school
students_by_school = school_groups['Student ID'].count()

# school budget
budget = school_data.set_index('school_name')['budget']

#per student budget
budget_per_student = school_data.set_index('school_name')['budget']/school_data.set_index('school_name')['size']

#avg scores by school
average_math_scores = school_groups['math_score'].mean()
average_reading_scores = school_groups['reading_score'].mean()

# % Passing Scores by School
percent_passing_math_by_school = (school_data_complete_df[school_data_complete_df["math_score"] >= 70].groupby("school_name")["student_name"].count())/students_by_school
percent_passing_reading_by_school = (school_data_complete_df[school_data_complete_df["reading_score"] >= 70].groupby("school_name")["student_name"].count())/students_by_school
percent_overall_passing_by_school = (school_data_complete_df[(school_data_complete_df["reading_score"] >= 70) & (school_data_complete_df["math_score"] >= 70)]).groupby("school_name")["student_name"].count()/students_by_school

#Create summary school Dataframe
school_summary = pd.DataFrame({
    "School Type": school_type,
    "Total Students": students_by_school,
    "Total School Budget": budget,
    "Per Student Budget": budget_per_student,
    "Average Math Score": average_math_scores,
    "Average Reading Score": average_reading_scores,
    "% Passing Math": percent_passing_math_by_school*100,
    "% Passing Reading": percent_passing_reading_by_school*100,
    "% Overall Passing": percent_overall_passing_by_school*100})

#format, format, display
```

```
school_summary.style.format({  
  "Total Students": "{:,}",  
  "Total School Budget": "${:,.2f}",  
  "Per Student Budget": "${:,.2f}",  
  "Average Math Score": "{:.1f}",  
  "Average Reading Score": "{:.1f}",  
  "% Passing Math": "{:.2f}%",  
  "% Passing Reading": "{:.2f}%",  
  "% Overall Passing": "{:.2f}%"})
```

Out[60]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Bailey High School	District	4,976	\$3,124,928.00	\$628.00	77.0	81.0	66.68%	81.93%	54.64%
Cabrera High School	Charter	1,858	\$1,081,356.00	\$582.00	83.1	84.0	94.13%	97.04%	91.33%
Figueroa High School	District	2,949	\$1,884,411.00	\$639.00	76.7	81.2	65.99%	80.74%	53.20%
Ford High School	District	2,739	\$1,763,916.00	\$644.00	77.1	80.7	68.31%	79.30%	54.29%
Griffin High School	Charter	1,468	\$917,500.00	\$625.00	83.4	83.8	93.39%	97.14%	90.60%
Hernandez High School	District	4,635	\$3,022,020.00	\$652.00	77.3	80.9	66.75%	80.86%	53.53%
Holden High School	Charter	427	\$248,087.00	\$581.00	83.8	83.8	92.51%	96.25%	89.23%
Huang High School	District	2,917	\$1,910,635.00	\$655.00	76.6	81.2	65.68%	81.32%	53.51%
Johnson High School	District	4,761	\$3,094,650.00	\$650.00	77.1	81.0	66.06%	81.22%	53.54%
Pena High School	Charter	962	\$585,858.00	\$609.00	83.8	84.0	94.59%	95.95%	90.54%
Rodriguez High School	District	3,999	\$2,547,363.00	\$637.00	76.8	80.7	66.37%	80.22%	52.99%
Shelton High School	Charter	1,761	\$1,056,600.00	\$600.00	83.4	83.7	93.87%	95.85%	89.89%
Thomas High School	Charter	1,635	\$1,043,130.00	\$638.00	83.4	83.8	93.27%	97.31%	90.95%
Wilson High School	Charter	2,283	\$1,319,574.00	\$578.00	83.3	84.0	93.87%	96.54%	90.58%
Wright High School	Charter	1,800	\$1,049,400.00	\$583.00	83.7	84.0	93.33%	96.61%	90.33%

3. Top Performing Schools (By % Overall Passing)

- Sort and display the top five performing schools by % overall passing.

```
In [61]: school_summary.sort_values("% Overall Passing", ascending=False).head().style.format({
    "Total Students": "{:,}",
    "Total School Budget": "${:,.2f}",
    "Per Student Budget": "${:,.2f}",
    "Average Math Score": "{:.1f}",
    "Average Reading Score": "{:.1f}",
    "% Passing Math": "{:.2f}%",
    "% Passing Reading": "{:.2f}%",
    "% Overall Passing": "{:.2f}%"})
```

Out[61]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Cabrera High School	Charter	1,858	\$1,081,356.00	\$582.00	83.1	84.0	94.13%	97.04%	91.33%
Thomas High School	Charter	1,635	\$1,043,130.00	\$638.00	83.4	83.8	93.27%	97.31%	90.95%
Griffin High School	Charter	1,468	\$917,500.00	\$625.00	83.4	83.8	93.39%	97.14%	90.60%
Wilson High School	Charter	2,283	\$1,319,574.00	\$578.00	83.3	84.0	93.87%	96.54%	90.58%
Pena High School	Charter	962	\$585,858.00	\$609.00	83.8	84.0	94.59%	95.95%	90.54%

4. Bottom Performing Schools (By % Overall Passing)

- Sort and display the five worst-performing schools by % overall passing.

```
In [53]: school_summary.sort_values("% Overall Passing", ascending=False).tail().style.format({
    "Total Students": "{:,}",
    "Total School Budget": "${:,.2f}",
    "Per Student Budget": "${:,.2f}",
    "Average Math Score": "{:.1f}",
    "Average Reading Score": "{:.1f}",
    "% Passing Math": "{:.2f}%",
    "% Passing Reading": "{:.2f}%",
    "% Overall Passing": "{:.2f}%"})
```

Out[53]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Johnson High School	District	4,761	\$3,094,650.00	\$650.00	77.1	81.0	66.06%	81.22%	53.54%
Hernandez High School	District	4,635	\$3,022,020.00	\$652.00	77.3	80.9	66.75%	80.86%	53.53%
Huang High School	District	2,917	\$1,910,635.00	\$655.00	76.6	81.2	65.68%	81.32%	53.51%
Figueroa High School	District	2,949	\$1,884,411.00	\$639.00	76.7	81.2	65.99%	80.74%	53.20%
Rodriguez High School	District	3,999	\$2,547,363.00	\$637.00	76.8	80.7	66.37%	80.22%	52.99%

5. 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 [56]: #Create series for each grade and group each series by school
ninth_grade_math = student_data.loc[student_data["grade"] == "9th"].groupby("school_name")["math_score"].mean()
tenth_grade_math = student_data.loc[student_data["grade"] == "10th"].groupby("school_name")["math_score"].mean()
eleventh_grade_math = student_data.loc[student_data["grade"] == "11th"].groupby("school_name")["math_score"].mean()
twelfth_grade_math = student_data.loc[student_data["grade"] == "12th"].groupby("school_name")["math_score"].mean()

#Combine series into a dataframe
math_by_grade = pd.DataFrame({"9th": ninth_grade_math, "10th": tenth_grade_math, "11th": eleventh_grade_math, "12th": twelfth_grade_math})
math_by_grade.index.name = "School Name"

math_by_grade.style.format({"9th": "{:.2f}", "10th": "{:.2f}", "11th": "{:.2f}", "12th": "{:.2f}"})
```

Out[56]:

	9th	10th	11th	12th
School Name				
Bailey High School	77.08	77.00	77.52	76.49
Cabrera High School	83.09	83.15	82.77	83.28
Figueroa High School	76.40	76.54	76.88	77.15
Ford High School	77.36	77.67	76.92	76.18
Griffin High School	82.04	84.23	83.84	83.36
Hernandez High School	77.44	77.34	77.14	77.19
Holden High School	83.79	83.43	85.00	82.86
Huang High School	77.03	75.91	76.45	77.23
Johnson High School	77.19	76.69	77.49	76.86
Pena High School	83.63	83.37	84.33	84.12
Rodriguez High School	76.86	76.61	76.40	77.69
Shelton High School	83.42	82.92	83.38	83.78
Thomas High School	83.59	83.09	83.50	83.50
Wilson High School	83.09	83.72	83.20	83.04
Wright High School	83.26	84.01	83.84	83.64

6. Reading Score by Grade

- Perform the same operations as above for reading scores

```
In [41]: #Create pandas series for each grade and group each series by school
ninth_grade_reading = student_data.loc[student_data["grade"] == "9th"].groupby("school_name")["reading_score"].mean()
tenth_grade_reading = student_data.loc[student_data["grade"] == "10th"].groupby("school_name")["reading_score"].mean()
eleventh_grade_reading = student_data.loc[student_data["grade"] == "11th"].groupby("school_name")["reading_score"].mean()
twelfth_grade_reading = student_data.loc[student_data["grade"] == "12th"].groupby("school_name")["reading_score"].mean()

#Combine series into a dataframe and format
reading_by_grade = pd.DataFrame({"9th": ninth_grade_reading, "10th": tenth_grade_reading, "11th": eleventh_grade_reading, "12th": twelfth_grade_reading})
reading_by_grade.index.name = "School Name"
reading_by_grade.style.format({"9th": "{:.2f}", "10th": "{:.2f}", "11th": "{:.2f}", "12th": "{:.2f}"})
```

Out[41]:

	9th	10th	11th	12th
School Name				
Bailey High School	81.30	80.91	80.95	80.91
Cabrera High School	83.68	84.25	83.79	84.29
Figueroa High School	81.20	81.41	80.64	81.38
Ford High School	80.63	81.26	80.40	80.66
Griffin High School	83.37	83.71	84.29	84.01
Hernandez High School	80.87	80.66	81.40	80.86
Holden High School	83.68	83.32	83.82	84.70
Huang High School	81.29	81.51	81.42	80.31
Johnson High School	81.26	80.77	80.62	81.23
Pena High School	83.81	83.61	84.34	84.59
Rodriguez High School	80.99	80.63	80.86	80.38
Shelton High School	84.12	83.44	84.37	82.78
Thomas High School	83.73	84.25	83.59	83.83
Wilson High School	83.94	84.02	83.76	84.32
Wright High School	83.83	83.81	84.16	84.07

7. 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 [48]: #create bins
bins = [0, 584.999, 614.99, 644.99, 999.99]

#name groups
funding_groups = ["< $585", "$585 - 614", "$615 - 644", "> $644"]

#budget per student by bin
school_data_complete_df["budget_bins"] = pd.cut(school_data_complete_df["budget"]/school_data_complete_df["size"], bins, labels = funding_groups)

#group by spending
by_school_budget = school_data_complete_df.groupby("budget_bins")

#test scores grouped by budget
avg_math_score = by_school_spending["math_score"].mean()
avg_reading_score = by_school_spending["reading_score"].mean()
percent_passing_math = school_data_complete_df[school_data_complete_df["math_score"] >= 70].groupby("budget_bins")["Student ID"].count()/by_school_budget["Student ID"].count()
percent_passing_reading = school_data_complete_df[school_data_complete_df["reading_score"] >= 70].groupby("budget_bins")["Student ID"].count()/by_school_budget["Student ID"].count()
Overall_passing_rate = school_data_complete_df[(school_data_complete_df["reading_score"] >= 70) & (school_data_complete_df["math_score"] >= 70)].groupby("budget_bins")["Student ID"].count()/by_school_budget["Student ID"].count()

# create dataframe
scores_by_spending = pd.DataFrame({
    "Average Math Score": avg_math_score,
    "Average Reading Score": avg_reading_score,
    "% Passing Math": percent_passing_math*100,
    "% Passing Reading": percent_passing_reading*100,
    "Overall Passing Rate": Overall_passing_rate*100})

#indexes
scores_by_spending.index.name = "Per Student Budget"
scores_by_spending = scores_by_spending.reindex(funding_groups)

#formatting
scores_by_spending.style.format({"Average Math Score": "{:.2f}",
                                "Average Reading Score": "{:.2f}",
                                "% Passing Math": "{:.2f}%",
                                "% Passing Reading": "{:.2f}%",
                                "Overall Passing Rate": "{:.2f}%"})

```

Out[48]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Per Student Budget					
< \$585	83.36	83.96	93.70%	96.69%	90.64%
\$585 - 614	83.53	83.84	94.12%	95.89%	90.12%
\$615 - 644	78.06	81.43	71.40%	83.61%	60.29%
> \$644	77.05	81.01	66.23%	81.11%	53.53%

8. Scores by School Size

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


```

In [62]: #create bins
bins = [0, 1000, 2000, 5000]

#label bins
groups = ["Small(<1000)", "Medium (1000-2000)", "Large (>2000)"]

#establish bins by size and group data by bins
school_data_complete_df["size_bins"] = pd.cut(school_data_complete_df["size"], bins, labels = groups)

#group by size
scores_by_size = school_data_complete_df.groupby("size_bins")

#test scores grouped by size
math_averages= scores_by_size["math_score"].mean()
reading_averages = scores_by_size["math_score"].mean()
passed_math_percent = school_data_complete_df[school_data_complete_df["math_score"] >= 70].groupby("size_bin
s")["Student ID"].count()/scores_by_size["Student ID"].count()
passed_reading_percent = school_data_complete_df[school_data_complete_df["reading_score"] >= 70].groupby("siz
e_bins")["Student ID"].count()/scores_by_size["Student ID"].count()
overall_percent = school_data_complete_df[(school_data_complete_df["reading_score"] >= 70) & (school_data_com
plete_df["reading_score"] >= 70)].groupby("size_bins")["Student ID"].count()/scores_by_size["Student ID"].cou
nt()

# create dataframe
scores_by_size = pd.DataFrame({
    "Average Math Score": math_averages,
    "Average Reading Score": reading_averages,
    "% Passing Math": passed_math_percent*100,
    "% Passing Reading": passed_reading_percent*100,
    "Overall Passing Rate": overall_percent*100})

#indexes
scores_by_size.index.name = "School Size"
scores_by_size = scores_by_size.reindex(groups)

# #formatting
scores_by_size.style.format({
    "Average Math Score": "{:.2f}",
    "Average Reading Score": "{:.2f}",
    "% Passing Math": "{:.1f}%",
    "% Passing Reading": "{:.1f}%",
    "Overall Passing Rate": "{:.1f}%"})

```

Out[62]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
School Size					
Small(<1000)	83.83	83.83	94.0%	96.0%	96.0%
Medium (1000-2000)	83.37	83.37	93.6%	96.8%	96.8%
Large (>2000)	77.48	77.48	68.7%	82.1%	82.1%

9. Scores by School Type

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

```

In [64]: #Group by school type
school_by_type = school_data_complete_df.groupby("type")

#Test scores grouped by School Type
avg_math_score = school_by_type["math_score"].mean()
avg_reading_score = school_by_type["reading_score"].mean()
percent_passed_math = school_data_complete_df[school_data_complete_df["math_score"] >= 70].groupby("type")["Student ID"].count()/school_by_type["Student ID"].count()
percent_passed_read = school_data_complete_df[school_data_complete_df["reading_score"] >= 70].groupby("type")["Student ID"].count()/school_by_type["Student ID"].count()
percent_passed_overall = school_data_complete_df[(school_data_complete_df["reading_score"] >= 70) & (school_data_complete_df["math_score"] >= 70)].groupby("type")["Student ID"].count()/school_by_type["Student ID"].count()

#create DataFrame
scores_by_type = pd.DataFrame({
    "Average Math Score": avg_math_score,
    "Average Reading Score": avg_reading_score,
    "% Passing Math": percent_passed_math,
    "% Passing Reading": percent_passed_read,
    "Overall Passing Rate": percent_passed_overall})

#define index
scores_by_type.index.name = "Type of School"

#formatting
scores_by_type.style.format({
    "Average Math Score": "{:.1f}",
    "Average Reading Score": "{:.1f}",
    "% Passing Math": "{:.1%}",
    "% Passing Reading": "{:.1%}",
    "Overall Passing Rate": "{:.1%}"})

```

Out[64]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Type of School					
Charter	83.4	83.9	93.7%	96.6%	90.6%
District	77.0	81.0	66.5%	80.9%	53.7%

Sources:

<https://pandas.pydata.org/docs/> (<https://pandas.pydata.org/docs/>) <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.drop.html> (<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.drop.html>) <https://stackoverflow.com/questions/35661968/add-a-percent-sign-to-a-dataframe-column-in-python> (<https://stackoverflow.com/questions/35661968/add-a-percent-sign-to-a-dataframe-column-in-python>) <https://pandas.pydata.org/pandas-docs/stable/reference/series.html> (<https://pandas.pydata.org/pandas-docs/stable/reference/series.html>) <https://stackoverflow.com/questions/38107304/pandas-changing-a-specific-row-to-percentages/38107465#38107465> (<https://stackoverflow.com/questions/38107304/pandas-changing-a-specific-row-to-percentages/38107465#38107465>) https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html?highlight=syntax (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html?highlight=syntax) https://thispointer.com/pandas-sort-rows-or-columns-in-dataframe-based-on-values-using-dataframe-sort_values/#:~:text=Sort%20Dataframe%20rows%20based%20on%20columns%20in%20Descending%20Order,along%20with%20by%20arguments%20 (https://thispointer.com/pandas-sort-rows-or-columns-in-dataframe-based-on-values-using-dataframe-sort_values/#:~:text=Sort%20Dataframe%20rows%20based%20on%20columns%20in%20Descending%20Order,along%20with%20by%20arguments%20) https://pyformat.info/#conversion_flags (https://pyformat.info/#conversion_flags)

Observations

1. Charter schools had higher scores than district schools. This is seen in the top and bottom performing school based on percent of students that passed overall. The top five performing are charter schools and the bottom five are district schools. However, This trend is seen again in scores by school type. While there's less than a six point different in the average math schools and about three point difference in the average reading score, significant differences are seen in percent passing math and the overall passing rate.
2. A school's budget is not an accurate determiner of a quality education. The budget for the top performing school ranged from 585, 858.00to1,319,574.00 and the bottom performing schools had budgets between 1, 884, 411.00and3,094,650.00. The assumption was Johnson High School with a budget of \$3,094,650.00 was a large school, thus the budget per student would be low. However, upon examiner the budget per student calculations, Johnson High School has one of the highest budgets per student. This trend is seen again in the test scores grouped by school spending. From this calculation, budget per student and overall passing are inversely related.
3. More student pass reading than reading. Generally, reading scores are higher than math scores by looking over the scores by grade, but looking at the test scores by school type this seems to be a trend only in district schools, specifically those of medium to large size. This should be compared to state and national scores over time to detect trends.

These observations suggest charter schools are of higher education quality and operate more efficiently than district schools. Charter schools with 1000-2000 students grossly outperform the district schools of the same size with larger budgets. I suggest to the Mayor and Schoolboard to examine, adapt and implement the teaching methods used at Charter schools. Additionally, a more detailed analysis on charter and district school budgets is needed to compare how each type of school uses and manages their budget. Factors like the quality of food served in the cafetertia or the duration of the school day can impact student performace and school budgets. An analysis of instructional staff would also be helpful to look at the caliber of teaching, the educational level of teachers, the student-to-teacher ratio, teaching styles, and etc. Additionally, salaries of those employed by the school should be carefully analyzed. For instance, if teachers are paid more at charter schools, they will take more pride in their job and will be more attentive to student needs than underpaid instructors at district schools. I must request a thorough demographic analysis be performed of the student population across the district to look at household size, household income, geographic distribution, learning disability, mental health & wellness of household, education level of head of household, number of daily meals, among other variables. The quality of a student's homelife commonly factors into academic performance. These additional analyses will provide a detailed, holistic view of the schools and their students to know how to best distribute funds to better serve students.