PyCity Schools Challenege:

A comprehensive analysis of Py City Schools

Author: Bailey Thompson

1) Charter schools have less students and produce higher overall passing rates, this suggests, but does not validate, correlation between number of students and overall passing rate, with number of students having a positive effect on overall passing rates. However, this finding will be challeneged by a finding I observe in #5.

2) The top five performing schools are all charter schools with less than 2,500 students, the bottom five performing schools are all district schools with more than 2,500 students.

3) The data suggests that budget per student does NOT have an effect on passing rates. On average district schools have higher levels of budget per student than charter schools but lower overall passing rates.

4) Both math and reading grades amoung classes at both charter and district schools are consistent, with math having lower scores than reading.

5) One district school, Bailey High School, is an outlier amoung the district school category due to the fact that it has the lowest budget per student, yet the highest overall passing rate. However, the deviation is less than one percentage point above the average district passing rate*.

5-1) This again suggests that budget per student does not have a correlation on overall passing rate. It also raises another suggestion; Bailey High School has the highest number of students AND the highest overall passing rate amoung district schools. This finding challenges our observation in #1, that schools with the lower number of students perform at higher rates. This observation is also demostrated with charter schools, the two charter schools with the highest number of students are present in the top five performing schools.

Conclusion: Schools with less than 2,500 students perform at higher rates than schools with more than 2,500 students. All the charter schools in this data set contain less than 2,500 students and perform at higher rates, while all district schools have more than 2,500 students and perform at lower rates. However, the positive correlation that is number of students on overall passing rate seems to have a decreasing effect per additional student because of the outlier in district schools, Bailey High School, which has the most students in this data set and the highest overall passing rates out of all the district schools.

Further analysis: Further reseach on this school district should request number of teachers at each school, average travel time to school per student, number of disciplinary actions per school, and average time students spend on both math and reading homework per student to identify any other correlations. Further research might also leverge a linear regression on the effect number of students has on overall passing rates.

Below is my code. I am using snake case for data frames and camel case for variables to help me distinguish between the two.

In [17]:
```
# Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load
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 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"])
```

In [18]:
```
# view the data sets to get a feel for the information in each one (continued in #4)
student_data.head()
school_data.head()
```

Out[18]:

	School ID	school_name	type	size	budget
0	0	Huang High School	District	2917	1910635
1	1	Figueroa High School	District	2949	1884411
2	2	Shelton High School	Charter	1761	1056600
3	3	Hernandez High School	District	4635	3022020
4	4	Griffin High School	Charter	1468	917500

In [19]:
```
school_data_complete.head()
```

Out[19]:

	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

In [20]:
```
# grouped by school df for future use
grouped_by_school = school_data_complete.set_index('school_name').groupby(['school_name'])
# counting number of schools in original school_data df
countSchools = len(school_data)
print(countSchools)
# counting total number of students in orginal student_data df
totalStudents = len(student_data)
print(totalStudents)
# sum of each school's budget using orginal school_data df
totalBudget = sum(school_data["budget"])
print(totalBudget)
# average math score using orginal student_data df
avgMath = sum(student_data["math_score"]) / totalStudents
print(avgMath)
# average reading score using orginal student_data df
avgReading = sum(student_data["reading_score"]) / totalStudents
print(avgReading)
# percent passing math using orginal student_data df
percPassMath = student_data[student_data["math_score"] > 69].count()["student_name"] / totalStudents
print(percPassMath)
#numStuPassMath = count(student_data['math_score'] > 59)
percPassRead = student_data[student_data["reading_score"] > 69].count()["student_name"] / totalStudents
print(percPassRead)
# overall passing rate
overallPassPerc = (avgMath + avgReading) / 2
print(overallPassPerc)
# printing to ensure validity and identify any errors
```

```
15
39170
24649428
78.98537145774827
81.87784018381414
0.749808526933878
0.8580546336482001
80.43160582078121
```

In [21]:
```
# giving all values a cleaner format
# total number of students
totalStudents = "{:,}".format(totalStudents)
# total budget
totalBudget = "${:,.2f}".format(totalBudget)
#print(totalBudget)
# average math score
avgMath = round(avgMath, 2)
#print(avgMath)
# average reading score
avgReading = round(avgReading, 2)
#print(avgReading)
# percentage of students passing math
percPassMath = "{:.2%}".format(percPassMath)
#print(percPassMath)
# percentage of students passing reading
percPassRead = "{:.2%}".format(percPassRead)
#print(percPassRead)
# overall percentage of students passing
overallPassPerc = round(overallPassPerc,2)
#print(overallPassPerc)
```

In [22]:
```
# creating the district summary df

district_summary = pd.DataFrame({"Total Schools": [countSchools],
                                 "Total Students": [totalStudents],
                                 "Total Budget": str(totalBudget),
                                 "Average Math Score": [avgMath],
                                 "Average Reading Score": [avgReading],
                                 "% Passing Math": [percPassMath],
                                  "% Passing Reading": [percPassRead],
                                  "% Overall Passing Rate": [overallPassPerc]})



district_summary
```

Out[22]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
0	15	39,170	$24,649,428.00	78.99	81.88	74.98%	85.81%	80.43

In [24]:
```
# School summary table:
# students at each school
studentsAtSchools = grouped_by_school['Student ID'].count()
# school type
schoolTypes = school_data.set_index('school_name')['type']
#print(schoolTypes)
# budget for each school
schoolsBudget = school_data.set_index('school_name')['budget']
#print(schoolsBudget)
# per student budget
perStudentBudget = schoolsBudget/studentsAtSchools
#print(perStudentBudget)
# average math score per school
schoolsAvgMath = round(grouped_by_school['math_score'].mean(), 2)
#print(schoolsAvgMath)
# average reading score per school
schoolsAvgReading = round(grouped_by_school['reading_score'].mean(), 2)
#print(schoolsAvgReading)
# percent passing math
percPassMathSchools = round(school_data_complete[school_data_complete["math_score"] > 69].groupby('school_n
ame')['student_name'].count() / studentsAtSchools * 100, 2)
# percent passing reading
percPassReadingSchools = round(school_data_complete[school_data_complete["reading_score"] > 69].groupby('sc
hool_name')['student_name'].count() / studentsAtSchools * 100, 2)
# percent overall passing
overallPassingSchools = round((percPassMathSchools + percPassReadingSchools) / 2, 2)
# create df
schools_summary = pd.DataFrame({"School Type": schoolTypes,
                                "Total Students": studentsAtSchools,
                                "Total School Budget": schoolsBudget,
                                "Per Student Budget": perStudentBudget,
                                "Average Math Score": schoolsAvgMath,
                                "Average Reading Score": schoolsAvgReading,
                                "% Passing Math": percPassMathSchools,
                                "% Passing Reading": percPassReadingSchools,
                                 "% Overall Passing Rate": overallPassingSchools})

# formating df

schools_summary.style.format({'Total Students': '{:,}',
                              "Total School Budget": "${:,}",
                              "Per Student Budget": "${:.0f}",
                              'Average Math Score': "{:.2f}",
                              'Average Reading Score': "{:.2f}"})
```

Out[24]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Bailey High School	District	4,976	$3,124,928	$628	77.05	81.03	66.68	81.93	74.31
Cabrera High School	Charter	1,858	$1,081,356	$582	83.06	83.98	94.13	97.04	95.58
Figueroa High School	District	2,949	$1,884,411	$639	76.71	81.16	65.99	80.74	73.36
Ford High School	District	2,739	$1,763,916	$644	77.10	80.75	68.31	79.3	73.81
Griffin High School	Charter	1,468	$917,500	$625	83.35	83.82	93.39	97.14	95.26
Hernandez High School	District	4,635	$3,022,020	$652	77.29	80.93	66.75	80.86	73.81
Holden High School	Charter	427	$248,087	$581	83.80	83.81	92.51	96.25	94.38
Huang High School	District	2,917	$1,910,635	$655	76.63	81.18	65.68	81.32	73.5
Johnson High School	District	4,761	$3,094,650	$650	77.07	80.97	66.06	81.22	73.64
Pena High School	Charter	962	$585,858	$609	83.84	84.04	94.59	95.95	95.27
Rodriguez High School	District	3,999	$2,547,363	$637	76.84	80.74	66.37	80.22	73.3
Shelton High School	Charter	1,761	$1,056,600	$600	83.36	83.73	93.87	95.85	94.86
Thomas High School	Charter	1,635	$1,043,130	$638	83.42	83.85	93.27	97.31	95.29
Wilson High School	Charter	2,283	$1,319,574	$578	83.27	83.99	93.87	96.54	95.21
Wright High School	Charter	1,800	$1,049,400	$583	83.68	83.96	93.33	96.61	94.97

In [25]:
```
# Top five performing schools
# sort by highest overall passing rate
top_5_schools = schools_summary.sort_values("% Overall Passing Rate", ascending = False)
# format
top_5_schools.head().style.format({'Total Students': '{:,}',
                           "Total School Budget": "${:,}",
                           "Per Student Budget": "${:.0f}",
                           'Average Math Score': "{:.2f}",
                           'Average Reading Score': "{:.2f}"})
```

Out[25]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Cabrera High School	Charter	1,858	$1,081,356	$582	83.06	83.98	94.13	97.04	95.58
Thomas High School	Charter	1,635	$1,043,130	$638	83.42	83.85	93.27	97.31	95.29
Pena High School	Charter	962	$585,858	$609	83.84	84.04	94.59	95.95	95.27
Griffin High School	Charter	1,468	$917,500	$625	83.35	83.82	93.39	97.14	95.26
Wilson High School	Charter	2,283	$1,319,574	$578	83.27	83.99	93.87	96.54	95.21

In [26]:
```
# sort by loweststst overall passing rate
bottom_5_schools = schools_summary.sort_values("% Overall Passing Rate")
# format
bottom_5_schools.head().style.format({'Total Students': '{:,}',
                           "Total School Budget": "${:,}",
                           "Per Student Budget": "${:.0f}",
                           'Average Math Score': "{:.2f}",
                           'Average Reading Score': "{:.2f}"})
```

Out[26]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Rodriguez High School	District	3,999	$2,547,363	$637	76.84	80.74	66.37	80.22	73.3
Figueroa High School	District	2,949	$1,884,411	$639	76.71	81.16	65.99	80.74	73.36
Huang High School	District	2,917	$1,910,635	$655	76.63	81.18	65.68	81.32	73.5
Johnson High School	District	4,761	$3,094,650	$650	77.07	80.97	66.06	81.22	73.64
Ford High School	District	2,739	$1,763,916	$644	77.10	80.75	68.31	79.3	73.81

In [27]:
```
# math and reading averages
nineMath = round(school_data_complete[school_data_complete["grade"] == '9th'].groupby('school_name')['math_
score'].mean(), 2)
nineReading = round(school_data_complete[school_data_complete["grade"] == '9th'].groupby('school_name')['re
ading_score'].mean(), 2)
tenMath = round(school_data_complete[school_data_complete["grade"] == '10th'].groupby('school_name')['math_
score'].mean(), 2)
tenReading = round(school_data_complete[school_data_complete["grade"] == '10th'].groupby('school_name')['re
ading_score'].mean(), 2)
elevenMath = round(school_data_complete[school_data_complete["grade"] == '11th'].groupby('school_name')['ma
th_score'].mean(), 2)
elevenReading = round(school_data_complete[school_data_complete["grade"] == '11th'].groupby('school_name')[
'reading_score'].mean(), 2)
twelveMath = round(school_data_complete[school_data_complete["grade"] == '12th'].groupby('school_name')['ma
th_score'].mean(), 2)
twelveReading = round(school_data_complete[school_data_complete["grade"] == '12th'].groupby('school_name')[
'reading_score'].mean(), 2)
# math df
math_by_grade = pd.DataFrame({"9th": nineMath,
                              "10th": tenMath,
                              "11th": elevenMath,
                              "12th": twelveMath})
# index name
math_by_grade.index.name = "School"

# reading df
reading_by_grade = pd.DataFrame({"9th": nineReading,
                              "10th": tenReading,
                              "11th": elevenReading,
                              "12th": twelveReading})
# index name
reading_by_grade.index.name = "School"



math_by_grade
```

Out[27]:

	9th	10th	11th	12th
School				
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

In [28]:
```
reading_by_grade
```

Out[28]:

	9th	10th	11th	12th
School				
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

In [29]:
```
# scores by school spending, using sample bins and group names
spending_bins = [0, 585, 615, 645, 675]
group_names = ["<$585", "$585-615", "$615-645", "$645-675"]
school_data_complete['spending bins'] = pd.cut(school_data_complete['budget']/school_data_complete['size'],
spending_bins, labels = group_names)

# group by spending
grouped_by_spend = school_data_complete.groupby('spending bins')

# students per spending bins
stuBySpend = grouped_by_spend['Student ID'].count()
# avg math by spending
mathAvgSpend = round(grouped_by_spend["math_score"].mean(), 2)
# avg reading by spending
readingAvgSpend = round(grouped_by_spend["reading_score"].mean(), 2)
# passing math by spending
percPassMathSpend = round(school_data_complete[school_data_complete['math_score'] > 69].groupby('spending b
ins')['Student ID'].count()/ stuBySpend *100, 2)
# passing reading by spending
percPassReadSpend = round(school_data_complete[school_data_complete['reading_score'] > 69].groupby('spendin
g bins')['Student ID'].count()/ stuBySpend *100, 2)
# overall passing rate by spending
overallPassSpend = round((percPassMathSpend + percPassReadSpend) / 2, 2)

# scores by spending df
scores_spend = pd.DataFrame({"Average Math Score": mathAvgSpend,
                             "Average Reading Score": readingAvgSpend,
                             "% Passing Math": percPassMathSpend,
                             "% Passing Reading": percPassReadSpend,
                             "% Overall Passing Rate": overallPassSpend})

# index name and desginating index
scores_spend.index.name = "Budget Per Student"
scores_spend = scores_spend.reindex(group_names)



scores_spend
```

Out[29]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Budget Per Student					
<$585	83.36	83.96	93.70	96.69	95.20
$585-615	83.53	83.84	94.12	95.89	95.00
$615-645	78.06	81.43	71.40	83.61	77.50
$645-675	77.05	81.01	66.23	81.11	73.67

In [30]:
```
# scores by school size
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
school_data_complete['size bins'] = pd.cut(school_data_complete['size'], size_bins, labels = group_names)

# group by school size
grouped_by_size = school_data_complete.groupby('size bins')
grouped_by_size.head()

# number of students per size
stuSize = grouped_by_size['Student ID'].count()
# avg math score by size
mathAvgSize = round(grouped_by_size['math_score'].mean(), 2)
# avg reading score by size
readAvgSize = round(grouped_by_size['reading_score'].mean(), 2)
# percent passing math by size
percPassMathSize = round(school_data_complete[school_data_complete['math_score'] > 69].groupby('size bins')
['Student ID'].count()/ stuSize *100, 2)
# percent passing reading by size
percPassReadSize = round(school_data_complete[school_data_complete['reading_score'] > 69].groupby('size bin
s')['Student ID'].count()/ stuSize *100, 2)
# overall passing rate by size
overallPassSize = round((percPassMathSize + percPassReadSize) / 2, 2)

# scores by size df
scores_size = pd.DataFrame({"Average Math Score": mathAvgSize,
                            "Average Reading Score": readAvgSize,
                            "% Passing Math": percPassMathSize,
                            "% Passing Reading": percPassReadSize,
                            "% Overall Passing Rate": overallPassSize})

# index name and desginating index
scores_size.index.name = "Number of Students"
scores_size = scores_size.reindex(group_names)



scores_size
```

Out[30]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Number of Students					
Small (<1000)	83.83	83.97	93.95	96.04	95.00
Medium (1000-2000)	83.37	83.87	93.62	96.77	95.20
Large (2000-5000)	77.48	81.20	68.65	82.13	75.39

In [31]:
```
grouped_by_type = school_data_complete.groupby('type')

# number of students per size
stuType = grouped_by_type['Student ID'].count()
# avg math score by size
mathAvgType = round(grouped_by_type['math_score'].mean(), 2)
# avg reading score by size
readAvgType = round(grouped_by_type['reading_score'].mean(), 2)
# percent passing math by size
percPassMathType = round(school_data_complete[school_data_complete['math_score'] > 69].groupby('type')['Stu
dent ID'].count()/ stuType *100, 2)
# percent passing reading by size
percPassReadType = round(school_data_complete[school_data_complete['reading_score'] > 69].groupby('type')['
Student ID'].count()/ stuType *100, 2)
# overall passing rate by size
overallPassType = round((percPassMathType + percPassReadType) / 2, 2)

# scores by size df
scores_type = pd.DataFrame({"Total Students": stuType,
                            "Average Math Score": mathAvgType,
                             "Average Reading Score": readAvgType,
                             "% Passing Math": percPassMathType,
                             "% Passing Reading": percPassReadType,
                             "% Overall Passing Rate": overallPassType})

# index name and desginating index
scores_type.index.name = "School Type"


scores_type
```

Out[31]:

School Type	Total Students	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Charter	12194	83.41	83.90	93.70	96.65	95.18
District	26976	76.99	80.96	66.52	80.91	73.72

End of code