

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

- Above is final analysis, below is stepwise Jupyter Notebook interwoven with the assignment instructions.

In []:

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

# 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 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_
sDcPristine=school_data_complete.copy()
# school_data_complete.head()
```

In []:

```
student_data.head()
```

Out[]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score
0	0	Paul Bradley	M	9th	Huang High School	66	79
1	1	Victor Smith	M	12th	Huang High School	94	61
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58
4	4	Bonnie Ray	F	9th	Huang High School	97	84

In []:

```
school_data.head()
```

Out[]:

	School ID	school_name	type	size	budget
0	0	Huang High School	District	2917	1910635

	School ID	school_name	type	size	budget
1	1	Figueroa High School	District	2949	1884411
2	2	Shelton High School	Charter	1761	1056600
3	3	Hernandez High School	District	4635	3022020

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

```
In [ ]: school_data_complete.describe()
```

```
Out[ ]:
```

	Student ID	reading_score	math_score	School ID	size	budget
count	39170.000000	39170.000000	39170.000000	39170.000000	39170.000000	3.917000e+04
mean	19584.500000	81.87784	78.985371	6.978172	3332.957110	2.117241e+06
std	11307.549359	10.23958	12.309968	4.444329	1323.914069	8.749987e+05
min	0.000000	63.00000	55.000000	0.000000	427.000000	2.480870e+05
25%	9792.250000	73.00000	69.000000	3.000000	1858.000000	1.081356e+06
50%	19584.500000	82.00000	79.000000	7.000000	2949.000000	1.910635e+06
75%	29376.750000	91.00000	89.000000	11.000000	4635.000000	3.022020e+06
max	39169.000000	99.00000	99.000000	14.000000	4976.000000	3.124928e+06

```
In [ ]: school_data_complete.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 39170 entries, 0 to 39169
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Student ID      39170 non-null  int64
1   student_name    39170 non-null  object
2   gender          39170 non-null  object
3   grade           39170 non-null  object
```

```
4  school_name      39170 non-null object
5  reading_score    39170 non-null int64
6  math_score       39170 non-null int64
7  School ID       39170 non-null int64
8  type             39170 non-null object
9  size             39170 non-null int64
10 budget           39170 non-null int64
dtypes: int64(6), object(5)
```

```
In [ ]: # * Calculate the total number of schools
numSch=school_data['school_name'].count()
numSch
```

Out[]: 15

```
In [ ]: # * Calculate the total number of students
numStu=school_data['size'].sum()
numStu
```

Out[]: 39170

```
In [ ]: # * Calculate the total budget
df = pd.DataFrame(columns=['X', 'Y'])
df=school_data_complete.sort_values(by="School ID").groupby("school_name", as_index=False).first()
ttlBud=df['budget'].sum()
ttlBud
```

Out[]: 24649428

```
In [ ]: TTLstudents=school_data_complete['size'].sum()
TTLstudents
```

Out[]: 130551930

```
In [ ]: # * Calculate the average math score
MEANmath=school_data_complete['math_score'].mean()
MEANmath
```

Out[]: 78.98537145774827

```
In [ ]: # * Calculate the average reading score
MEANreading=school_data_complete['reading_score'].mean()
MEANreading
```

Out[]: 81.87784018381414

```
In [ ]: # * Calculate the overall passing rate (overall average score), i.e. (avg. math score + avg. readi
# # Create bins and bin labels for the Math Scores
math_bins = [0, 69, 100]
math_labels = [0, 1]

# # Bin the math score column
# # cut() returns a Pandas Series containing each of the binned column's values translated into th

mathCt=pd.cut(school_data_complete["math_score"], math_bins, labels=math_labels)
mathdf=pd.DataFrame(mathCt)
# mathdf.describe()
vc=pd.DataFrame(mathdf["math_score"].value_counts())
vc
tlStu=vc.sum()
numPassing=vc.iloc[0,:]
passRateM=(numPassing/tlStu)
passRateM
# PASS RATE MATH=
```

```
Out[ ]: math_score    0.749809
dtype: float64
```

```
In [ ]: # Create bins and bin labels for the Reading Scores
read_bins = [0, 69, 100]
read_labels = [0, 1]
# # Bin the math score column
# # cut() returns a Pandas Series containing each of the binned column's values translated into th
readCt=pd.cut(school_data_complete["reading_score"], read_bins, labels=read_labels)
readdf=pd.DataFrame(readCt)
readdf.describe()
vcr=pd.DataFrame(readdf["reading_score"].value_counts())
# vcr
tlStuR=vcr.sum()
numPassing=vcr.iloc[0,:]
passRateR=(numPassing/tlStuR)
passRateR
# READING PASS RATE=
```

```
Out[ ]: reading_score    0.858055
dtype: float64
```

```
In [ ]: # * Overall passing rate
# avg_math_score = school_data_complete["math_score"].mean()
# avg_reading_score = school_data_complete["reading_score"].mean()
overall_pass = (MEANmath + MEANreading) / 2
overall_pass #Overall Pass Rate
```

```
Out[ ]: 80.43160582078121
```

```
In [ ]: # * Create a dataframe to hold the above results
listData=[numSch,numStu,ttlBud,MEANmath,MEANreading,passRateM,passRateR,overall_pass]
labels=['Total Schools','Total Students','Total Budget','Average Math Score','Average Reading Score']
df1=pd.DataFrame(listData,labels).transpose()
df1
```

```
Out[ ]:
```

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
0	15	39170	24649428	78.9854	81.8778	math_score	reading_score	80.4316

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

```
In [ ]: # sDcPristine.columns
```

```
In [ ]: # Create an overview table that summarizes key metrics about each school, including:
# School Name
# School Type
# Total Students
# Total School Budget
cypherListSum=sDcPristine[['school_name','School ID', 'type', 'size', 'budget']]
schoolSum1= cypherListSum.sort_values(by="School ID").groupby("school_name", as_index=False).first
# schoolSum1
```

```
In [ ]: # sDcPristine.columns
```

```
In [ ]: cypherListSum2 = sDcPristine[['school_name','reading_score', 'math_score','size']]
schoolSum2 = cypherListSum2.groupby(['school_name'])
meanSch=schoolSum2.mean()
meanSch
meanSch.columns=['Avrg_Reading_Score', 'Avrg_Math_Score', '#_Students']
meanSch.reset_index()
```

```
Out[ ]:
```

	school_name	Avrg_Reading_Score	Avrg_Math_Score	#_Students
0	Bailey High School	81.033963	77.048432	4976.0
1	Cabrera High School	83.975780	83.061895	1858.0
2	Figueroa High School	81.158020	76.711767	2949.0
3	Ford High School	80.746258	77.102592	2739.0
4	Griffin High School	83.816757	83.351499	1468.0
5	Hernandez High School	80.934412	77.289752	4635.0
6	Holden High School	83.814988	83.803279	427.0

	school_name	Avrg_Reading_Score	Avrg_Math_Score	#_Students
7	Huang High School	81.182722	76.629414	2917.0
8	Johnson High School	80.966394	77.072464	4761.0
9	Pena High School	84.044699	83.839917	962.0
10	Rodriguez High School	80.744686	76.842711	3999.0
11	Shelton High School	83.725724	83.359455	1761.0
12	Thomas High School	83.848930	83.418349	1635.0
13	Wilson High School	83.989488	83.274201	2283.0

```
In [ ]: cypherListSum3 = sDcPristine[['school_name', 'student_name', 'grade', 'reading_score', 'math_score']]
schoolSum3=cypherListSum3[cypherListSum3["reading_score"]>=70]
r = schoolSum3.groupby(['school_name'])
rNumPassing=pd.DataFrame(r["reading_score"].count())
# rNumPassing.index
rNumPassing.columns=['Read_Pass']
rNumPassing
rNumPassing.reset_index()
```

```
Out [ ]:
```

	school_name	Read_Pass
0	Bailey High School	4077
1	Cabrera High School	1803
2	Figueroa High School	2381
3	Ford High School	2172
4	Griffin High School	1426
5	Hernandez High School	3748
6	Holden High School	411
7	Huang High School	2372
8	Johnson High School	3867
9	Pena High School	923
10	Rodriguez High School	3208
11	Shelton High School	1688
12	Thomas High School	1591
13	Wilson High School	2204
14	Wright High School	1739

```
In [ ]: cypherListSum4 = sDcPristine[['school_name', 'student_name', 'grade', 'reading_score', 'math_score']]
schoolSum4=cypherListSum4[cypherListSum3["math_score"]>=70]
m = schoolSum4.groupby(['school_name'])
mNumPassing=pd.DataFrame(m["math_score"].count())
mNumPassing.columns=['Math_Pass']
mNumPassing# mNumPassing.index
mNumPassing.reset_index()
```

```
Out [ ]:
```

	school_name	Math_Pass
--	-------------	-----------

	school_name	Math_Pass
0	Bailey High School	3318
1	Cabrera High School	1749
2	Figueroa High School	1946
3	Ford High School	1871
4	Griffin High School	1371
5	Hernandez High School	3094
6	Holden High School	395
7	Huang High School	1916
8	Johnson High School	3145
9	Pena High School	910
10	Rodriguez High School	2654
11	Shelton High School	1653
12	Thomas High School	1525
13	Wilson High School	2143

```
In [ ]: merSum=pd.merge(schoolSum1,meanSch,how="inner",on=['school_name'])
merSum1=pd.merge(merSum, rNumPassing, how="inner", on=["school_name"])
school_SummaryTtl=pd.merge(merSum1, mNumPassing, how="inner",on=["school_name"])
school_SummaryTtl['Pass_Rate_Reading']=school_SummaryTtl['Read_Pass'] / school_SummaryTtl['#_Students']
school_SummaryTtl['Pass_Rate_Math']=school_SummaryTtl['Math_Pass'] / school_SummaryTtl['#_Students']
school_SummaryTtl['Overall_Pass_Rate']=(school_SummaryTtl['Avrg_Reading_Score'] +school_SummaryTtl['Avrg_Math_Score'])/2
school_SummaryTtl['Budget_per_Student']=school_SummaryTtl['budget'] / school_SummaryTtl['#_Students']
school_SummaryTtl
```

	school_name	School ID	type	size	budget	Avrg_Reading_Score	Avrg_Math_Score	#_Students	Read_Pass	Mat
0	Bailey High School	7	District	4976	3124928	81.033963	77.048432	4976.0	4077	
1	Cabrera High School	6	Charter	1858	1081356	83.975780	83.061895	1858.0	1803	
2	Figueroa High School	1	District	2949	1884411	81.158020	76.711767	2949.0	2381	
3	Ford High School	13	District	2739	1763916	80.746258	77.102592	2739.0	2172	
4	Griffin High School	4	Charter	1468	917500	83.816757	83.351499	1468.0	1426	
5	Hernandez High School	3	District	4635	3022020	80.934412	77.289752	4635.0	3748	
6	Holden High School	8	Charter	427	248087	83.814988	83.803279	427.0	411	
7	Huang High School	0	District	2917	1910635	81.182722	76.629414	2917.0	2372	
8	Johnson High School	12	District	4761	3094650	80.966394	77.072464	4761.0	3867	

	school_name	School ID	type	size	budget	Avrg_Reading_Score	Avrg_Math_Score	#_Students	Read_Pass	Mat
9	Pena High School	9	Charter	962	585858	84.044699	83.839917	962.0	923	
10	Rodriguez High School	11	District	3999	2547363	80.744686	76.842711	3999.0	3208	
11	Shelton High School	2	Charter	1761	1056600	83.725724	83.359455	1761.0	1688	
12	Thomas High School	14	Charter	1635	1043130	83.848930	83.418349	1635.0	1591	
13	Wilson High School	5	Charter	2283	1319574	83.989488	83.274201	2283.0	2204	

Top Performing Schools (By Passing Rate)

- Sort and display the top five schools in overall passing rate

```
In [ ]: school_SummaryTtl.drop(columns=['#_Students', 'Read_Pass', 'Math_Pass'], axis=1, inplace=True)
bestWorst=school_SummaryTtl.sort_values(by=['Overall_Pass_Rate'], ascending=False)
```

```
In [ ]: bestWorst.head()
```

```
Out[ ]:
```

	school_name	School ID	type	size	budget	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rat
9	Pena High School	9	Charter	962	585858	84.044699	83.839917	0.959459	C
14	Wright High School	10	Charter	1800	1049400	83.955000	83.682222	0.966111	C
6	Holden High School	8	Charter	427	248087	83.814988	83.803279	0.962529	C
12	Thomas High School	14	Charter	1635	1043130	83.848930	83.418349	0.973089	C
13	Wilson High School	5	Charter	2283	1319574	83.989488	83.274201	0.965396	C

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

```
In [ ]: bestWorst.tail()
```

```
Out[ ]:
```

	school_name	School ID	type	size	budget	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rat
8	Johnson High School	12	District	4761	3094650	80.966394	77.072464	0.812224	0
2	Figueroa High School	1	District	2949	1884411	81.158020	76.711767	0.807392	0

	school_name	School ID	type	size	budget	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rat
3	Ford High School	13	District	2739	1763916	80.746258	77.102592	0.792990	0
7	Huang High School	0	District	2917	1910635	81.182722	76.629414	0.813164	0

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 [ ]: freshman=sDcPristine[sDcPristine['grade']=='9th']
sophomore=sDcPristine[sDcPristine['grade']=='10th']
junior=sDcPristine[sDcPristine['grade']=='11th']
senior=sDcPristine[sDcPristine['grade']=='12th']
```

```
In [ ]: fmM = pd.DataFrame(freshman.groupby('school_name')['math_score'].mean())
fmM.columns=["9th_Average_Math_Score"]
fmM.reset_index()
```

```
Out[ ]:
```

	school_name	9th_Average_Math_Score
0	Bailey High School	77.083676
1	Cabrera High School	83.094697
2	Figueroa High School	76.403037
3	Ford High School	77.361345
4	Griffin High School	82.044010
5	Hernandez High School	77.438495
6	Holden High School	83.787402
7	Huang High School	77.027251
8	Johnson High School	77.187857
9	Pena High School	83.625455
10	Rodriguez High School	76.859966
11	Shelton High School	83.420755
12	Thomas High School	83.590022
13	Wilson High School	83.085578
14	Wright High School	83.264706

```
In [ ]: soM = pd.DataFrame(sophomore.groupby('school_name')['math_score'].mean())
soM.columns=["10th_Average_Math_Score"]
soM.reset_index()
# byGrade=pd.concat([fmM, soM], verify_integrity=True, sort=False, axis=1, join='outer')
merSum2=pd.merge(fmM, soM, how="inner", on=["school_name"])
merSum2
```

Out []:

	9th_Average_Math_Score	10th_Average_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
Hernandez High School	77.438495	77.337408
Holden High School	83.787402	83.429825
Huang High School	77.027251	75.908735
Johnson High School	77.187857	76.691117
Pena High School	83.625455	83.372000
Rodriguez High School	76.859966	76.612500
Shelton High School	83.420755	82.917411
Thomas High School	83.590022	83.087886
Wilson High School	83.085578	83.724422
Wright High School	83.264706	84.010288

```
In [ ]: jrM = pd.DataFrame(junior.groupby('school_name')['math_score'].mean())
jrM.columns=["11th_Average_Math_Score"]
jrM.reset_index()
merSum3=pd.merge(merSum2, jrM, how="inner", on=["school_name"])
merSum3
```

Out []:

	9th_Average_Math_Score	10th_Average_Math_Score	11th_Average_Math_Score
school_name			
Bailey High School	77.083676	76.996772	77.515588
Cabrera High School	83.094697	83.154506	82.765560
Figueroa High School	76.403037	76.539974	76.884344
Ford High School	77.361345	77.672316	76.918058
Griffin High School	82.044010	84.229064	83.842105
Hernandez High School	77.438495	77.337408	77.136029
Holden High School	83.787402	83.429825	85.000000
Huang High School	77.027251	75.908735	76.446602
Johnson High School	77.187857	76.691117	77.491653

	9th_Average_Math_Score	10th_Average_Math_Score	11th_Average_Math_Score
school_name			
Pena High School	83.625455	83.372000	84.328125
Rodriguez High School	76.859966	76.612500	76.395626
Shelton High School	83.420755	82.917411	83.383495
Thomas High School	83.590022	83.087886	83.498795
Wilson High School	83.085578	83.724422	83.195326

In []:

```
srM = pd.DataFrame(senior.groupby('school_name')['math_score'].mean())
srM.columns=["12th_Average_Math_Score"]
srM.reset_index()
merSum4=pd.merge(merSum3, srM, how="inner", on=["school_name"])
merSum4
```

Out[]:

	9th_Average_Math_Score	10th_Average_Math_Score	11th_Average_Math_Score	12th_Average_Math_Score
school_name				
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

In []:

Reading Score by Grade

- Perform the same operations as above for reading scores

In []:

```
fm = pd.DataFrame(freshman.groupby('school_name')['reading_score'].mean())
fm.columns=["9th_Average_Reading_Score"]
fm.reset_index()
```

Out[]:

	school_name	9th_Average_Reading_Score
0	Bailey High School	81.303155
1	Cabrera High School	83.676136
2	Figueroa High School	81.198598
3	Ford High School	80.632653
4	Griffin High School	83.369193
5	Hernandez High School	80.866860
6	Holden High School	83.677165
7	Huang High School	81.290284
8	Johnson High School	81.260714
9	Pena High School	83.807273
10	Rodriguez High School	80.993127
11	Shelton High School	84.122642
12	Thomas High School	83.728850
13	Wilson High School	83.939778
14	Wright High School	83.833333

In []:

```
sm = pd.DataFrame(sophomore.groupby('school_name')['reading_score'].mean())
sm.columns=["10th_Average_Reading_Score"]
sm.reset_index()
merSumA=pd.merge(fm, sm, how="inner", on=["school_name"])
merSumA.reset_index()
```

Out[]:

	school_name	9th_Average_Reading_Score	10th_Average_Reading_Score
0	Bailey High School	81.303155	80.907183
1	Cabrera High School	83.676136	84.253219
2	Figueroa High School	81.198598	81.408912
3	Ford High School	80.632653	81.262712
4	Griffin High School	83.369193	83.706897
5	Hernandez High School	80.866860	80.660147
6	Holden High School	83.677165	83.324561

	school_name	9th_Average_Reading_Score	10th_Average_Reading_Score
7	Huang High School	81.290284	81.512386
8	Johnson High School	81.260714	80.773431
9	Pena High School	83.807273	83.612000
10	Rodriguez High School	80.993127	80.629808
11	Shelton High School	84.122642	83.441964
12	Thomas High School	83.728850	84.254157
13	Wilson High School	83.939778	84.021452

```
In [ ]: jr = pd.DataFrame(junior.groupby('school_name')['reading_score'].mean())
jr.columns=["11th_Average_Reading_Score"]
jr.reset_index()
merSumB=pd.merge(merSumA, jr, how="inner", on=["school_name"])
merSumB.reset_index()
```

	school_name	9th_Average_Reading_Score	10th_Average_Reading_Score	11th_Average_Reading_Score
0	Bailey High School	81.303155	80.907183	80.945643
1	Cabrera High School	83.676136	84.253219	83.788382
2	Figueroa High School	81.198598	81.408912	80.640339
3	Ford High School	80.632653	81.262712	80.403642
4	Griffin High School	83.369193	83.706897	84.288089
5	Hernandez High School	80.866860	80.660147	81.396140
6	Holden High School	83.677165	83.324561	83.815534
7	Huang High School	81.290284	81.512386	81.417476
8	Johnson High School	81.260714	80.773431	80.616027
9	Pena High School	83.807273	83.612000	84.335938
10	Rodriguez High School	80.993127	80.629808	80.864811
11	Shelton High School	84.122642	83.441964	84.373786
12	Thomas High School	83.728850	84.254157	83.585542
13	Wilson High School	83.939778	84.021452	83.764608
14	Wright High School	83.833333	83.812757	84.156322

```
In [ ]: sr= pd.DataFrame(senior.groupby('school_name')['reading_score'].mean())
sr.columns=["12th_Average_Reading_Score"]
sr.reset_index()
merSumC=pd.merge(merSumB, sr, how="inner", on=["school_name"])
merSumC.reset_index()
```

	school_name	9th_Average_Reading_Score	10th_Average_Reading_Score	11th_Average_Reading_Score	12th_Average_Reading_Score
0	Bailey High School	81.303155	80.907183	80.945643	
1	Cabrera High School	83.676136	84.253219	83.788382	

	school_name	9th_Average_Reading_Score	10th_Average_Reading_Score	11th_Average_Reading_Score	12th_Average_Reading_Score
2	Figueroa High School	81.198598	81.408912	80.640339	
3	Ford High School	80.632653	81.262712	80.403642	
4	Griffin High School	83.369193	83.706897	84.288089	
5	Hernandez High School	80.866860	80.660147	81.396140	
6	Holden High School	83.677165	83.324561	83.815534	
7	Huang High School	81.290284	81.512386	81.417476	
8	Johnson High School	81.260714	80.773431	80.616027	
9	Pena High School	83.807273	83.612000	84.335938	
10	Rodriguez High School	80.993127	80.629808	80.864811	
11	Shelton High School	84.122642	83.441964	84.373786	
12	Thomas High School	83.728850	84.254157	83.585542	
13	Wilson High School	83.939778	84.021452	83.764608	
	Wright High School	81.198598	81.408912	80.640339	

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 [ ]: # Sample bins. Feel free to create your own bins.
spending_bins = [0, 585, 615, 645, 675]
group_names = ["Low", "Mid-Low", "Medium-High", "High"]

bPSS=pd.DataFrame()
bPSS["Budget_per_Student Summary"] = pd.cut(school_SummaryTtl["Budget_per_Student"], bins=spending
bPSS.index=['Bailey High School', 'Cabrera High School', 'Figueroa High School',
'Ford High School', 'Griffin High School', 'Hernandez High School',
'Holden High School', 'Huang High School', 'Johnson High School',
'Pena High School', 'Rodriguez High School', 'Shelton High School',
'Thomas High School', 'Wilson High School', 'Wright High School']
bPSS['school_name']=['Bailey High School', 'Cabrera High School', 'Figueroa High School',
'Ford High School', 'Griffin High School', 'Hernandez High School',
'Holden High School', 'Huang High School', 'Johnson High School',
'Pena High School', 'Rodriguez High School', 'Shelton High School',
'Thomas High School', 'Wilson High School', 'Wright High School']
merSum5=pd.merge(bPSS, bestWorst, how="inner")
merSum5.columns=['Scores_by_School_Spending', 'School_Name', 'School ID', 'Type', 'Size',
'Budget', 'Avrg_Reading_Score', 'Avrg_Math_Score', 'Pass_Rate_Reading',
'Pass_Rate_Math', 'Overall_Pass_Rate', 'Budget_per_Student']
merSum5.drop(['School ID',"Type",'Size','Budget'], inplace=True, axis=1)
merSum5.sort_values(by=['Overall_Pass_Rate'],ascending=False, inplace=True)
merSum5.reset_index(drop=True)

```

Out []:	Scores_by_School_Spending	School_Name	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rate_M
0	Mid-Low	Pena High School	84.044699	83.839917	0.959459	0.945
1	Low	Wright High School	83.955000	83.682222	0.966111	0.933
2	Low	Holden High School	83.814988	83.803279	0.962529	0.925
3	Medium-High	Thomas High School	83.848930	83.418349	0.973089	0.932
4	Low	Wilson High School	83.989488	83.274201	0.965396	0.938
5	Medium-High	Griffin High School	83.816757	83.351499	0.971390	0.933
6	Mid-Low	Shelton High School	83.725724	83.359455	0.958546	0.938
7	Low	Cabrera High School	83.975780	83.061895	0.970398	0.941
8	High	Hernandez High School	80.934412	77.289752	0.808630	0.667
9	Medium-High	Bailey High School	81.033963	77.048432	0.819333	0.666
10	High	Johnson High School	80.966394	77.072464	0.812224	0.660
11	Medium-High	Figueroa High School	81.158020	76.711767	0.807392	0.659
12	Medium-High	Ford High School	80.746258	77.102592	0.792990	0.683
13	High	Huang High School	81.182722	76.629414	0.813164	0.656

14	Medium-High	Rodriguez High School	80.744686	76.842711	0.802201	0.663
----	-------------	-----------------------	-----------	-----------	----------	-------

Scores by School Size

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

In []:

```
# 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)"]

sBSS=pd.DataFrame()
sBSS["Scores_by_School_Size"] = pd.cut(school_SummaryTtl["size"], bins=size_bins, labels=group_names)
sBSS.index=['Bailey High School', 'Cabrera High School', 'Figueroa High School',
            'Ford High School', 'Griffin High School', 'Hernandez High School',
            'Holden High School', 'Huang High School', 'Johnson High School',
            'Pena High School', 'Rodriguez High School', 'Shelton High School',
            'Thomas High School', 'Wilson High School', 'Wright High School']
sBSS['school_name']=['Bailey High School', 'Cabrera High School', 'Figueroa High School',
                    'Ford High School', 'Griffin High School', 'Hernandez High School',
                    'Holden High School', 'Huang High School', 'Johnson High School',
                    'Pena High School', 'Rodriguez High School', 'Shelton High School',
                    'Thomas High School', 'Wilson High School', 'Wright High School']

sBSS
merSum6=pd.merge(sBSS, bestWorst, how="inner")
merSum6.columns=['Scores_by_School_Size', 'School_Name', 'School ID', 'Type', 'Size',
                 'Budget', 'Avrg_Reading_Score', 'Avrg_Math_Score', 'Pass_Rate_Reading',
                 'Pass_Rate_Math', 'Overall_Pass_Rate', 'Budget_per_Student']
merSum6.style.hide_columns(['Avrg_Reading_Score', 'Avrg_Math_Score', 'Pass_Rate_Reading', 'Pass_Rate_Math'])
merSum6.reset_index(drop=True)
merSum6.drop(['School ID', 'Size', 'Budget', 'Budget_per_Student'], inplace=True, axis=1)
merSum6.sort_values(by=['Overall_Pass_Rate'], ascending=False, inplace=True)
merSum6.reset_index(drop=True)
# merSum6
```

Out []:

	Scores_by_School_Size	School_Name	Type	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rate_Math
0	Small (<1000)	Pena High School	Charter	84.044699	83.839917	0.959459	0.663
1	Medium (1000-2000)	Wright High School	Charter	83.955000	83.682222	0.966111	0.663
2	Small (<1000)	Holden High School	Charter	83.814988	83.803279	0.962529	0.663
3	Medium (1000-2000)	Thomas High School	Charter	83.848930	83.418349	0.973089	0.663
4	Large (2000-5000)	Wilson High School	Charter	83.989488	83.274201	0.965396	0.663
5	Medium (1000-2000)	Griffin High School	Charter	83.816757	83.351499	0.971390	0.663
6	Medium (1000-2000)	Shelton High School	Charter	83.725724	83.359455	0.958546	0.663
7	Medium (1000-2000)	Cabrera High School	Charter	83.975780	83.061895	0.970398	0.663
8	Large (2000-5000)	Hernandez High School	District	80.934412	77.289752	0.808630	0.663

	Scores_by_School_Size	School_Name	Type	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rate_Math
9	Large (2000-5000)	Bailey High School	District	81.033963	77.048432	0.819333	0.819333
10	Large (2000-5000)	Johnson High School	District	80.966394	77.072464	0.812224	0.812224
11	Large (2000-5000)	Figueroa High School	District	81.158020	76.711767	0.807392	0.807392
12	Large (2000-5000)	Ford High School	District	80.746258	77.102592	0.792990	0.792990
13	Large (2000-5000)	Huang High School	District	81.182722	76.629414	0.813164	0.813164
14	Large (2000-5000)	Rodriguez High School	District	80.744686	76.842711	0.802201	0.802201

In []:

Scores by School Type

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

In []:

```
merSum6.columns=['Scores_by_School_Type', 'School_Name', 'Type', 'Avrg_Reading_Score', 'Avrg_Math_Score', 'Pass_Rate_Reading', 'Pass_Rate_Math', 'Overall_Pass_Rate']
# merSum6.style.hide_columns(['Avrg_Reading_Score', 'Avrg_Math_Score', 'Pass_Rate_Reading', 'Pass_Rate_Math'])
merSum6.sort_values(by=['Overall_Pass_Rate'], ascending=False, inplace=True)
merSum6.reset_index(drop=True)
# merSum6
```

Out[]:

	Scores_by_School_Type	School_Name	Type	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rate_Math
0	Small (<1000)	Pena High School	Charter	84.044699	83.839917	0.959459	0.959459
1	Medium (1000-2000)	Wright High School	Charter	83.955000	83.682222	0.966111	0.966111
2	Small (<1000)	Holden High School	Charter	83.814988	83.803279	0.962529	0.962529
3	Medium (1000-2000)	Thomas High School	Charter	83.848930	83.418349	0.973089	0.973089
4	Large (2000-5000)	Wilson High School	Charter	83.989488	83.274201	0.965396	0.965396
5	Medium (1000-2000)	Griffin High School	Charter	83.816757	83.351499	0.971390	0.971390
6	Medium (1000-2000)	Shelton High School	Charter	83.725724	83.359455	0.958546	0.958546
7	Medium (1000-2000)	Cabrera High School	Charter	83.975780	83.061895	0.970398	0.970398
8	Large (2000-5000)	Hernandez High School	District	80.934412	77.289752	0.808630	0.808630
9	Large (2000-5000)	Bailey High School	District	81.033963	77.048432	0.819333	0.819333

	Scores_by_School_Type	School_Name	Type	Avrg_Reading_Score	Avrg_Math_Score	Pass_Rate_Reading	Pass_Rate_Math
10	Large (2000-5000)	Johnson High School	District	80.966394	77.072464	0.812224	0.792990
11	Large (2000-5000)	Figueroa High School	District	81.158020	76.711767	0.807392	0.792990
12	Large (2000-5000)	Ford High School	District	80.746258	77.102592	0.792990	0.792990
13	Large (2000-5000)	Huang High School	District	81.182722	76.629414	0.813164	0.792990

```
In [ ]: # Conclusions
## 1) BUDGET--Overall passing rate seemed inversely proportional to amount of money spent per Student
## 2) SIZE-- With one exception,large Schools had the worst Overall Passing Rates
## 3) TYPE-- Charter schools consistently had the best Overall Pass Rates
```