

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [1]:

```
# Dependencies and Setup
import pandas as pd

# File to Load (Remember to Change These)
school_data_to_load = "schools_complete.csv"
student_data_to_load = "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_dataComp = pd.merge(student_data, school_data, how="left", on=["school_name", "school_name"])
```

In [2]:

```
school_dataComp.head()
```

Out[2]:

	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

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 [3]:

```

total_schools = len(school_dataComp["school_name"].unique())
total_students = school_dataComp["student_name"].count()
total_budget = school_data["budget"].sum()
avg_math = school_dataComp["math_score"].mean()
avg_reading = school_dataComp["reading_score"].mean()

passing_math = [num for num in school_dataComp["math_score"] if num >= 70]
passing_mathPerc = (len(passing_math) / len(school_dataComp["math_score"])) * 100

passing_reading = [num for num in school_dataComp["reading_score"] if num >= 70]
passing_readingPerc = (len(passing_reading) / len(school_dataComp["reading_score"])) * 100

overallPass = student_data[(student_data["math_score"] >=70) & (student_data["reading_score"] >= 70)]
overallPassC = len(overallPass)
passingAvgPerc = overallPassC / total_students *100

district_summary = pd.DataFrame({"Total Schools": [total_schools],
                                "Total Students": [total_students],
                                "Total Budget": [total_budget],
                                "Average Math Score": [avg_math],
                                "Average Reading Score": [avg_reading],
                                "% Passing Math": [passing_mathPerc],
                                "% Passing Reading": [passing_readingPerc],
                                "Overall Pass Rate": [passingAvgPerc]})

district_summary["Total Students"] = district_summary["Total Students"].map("{:,}".format)
district_summary["Total Budget"] = district_summary["Total Budget"].map("${:,.2f}".format)
district_summary["% Passing Math"] = district_summary["% Passing Math"].map("{:.2f}%".format)
district_summary["% Passing Reading"] = district_summary["% Passing Reading"].map("{:.2f}%".format)
district_summary["Overall Pass Rate"] = district_summary["Overall Pass Rate"].map("{:.2f}%".format)

district_summary

```

Out[3]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Pass Rate
0	15	39,170	\$24,649,428.00	78.985371	81.87784	74.98%	85.81%	65.17%

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 [4]:

```
school_type = school_data.set_index(["school_name"])[ "type"]  
school_type
```

Out[4]:

school_name	
Huang High School	District
Figueroa High School	District
Shelton High School	Charter
Hernandez High School	District
Griffin High School	Charter
Wilson High School	Charter
Cabrera High School	Charter
Bailey High School	District
Holden High School	Charter
Pena High School	Charter
Wright High School	Charter
Rodriguez High School	District
Johnson High School	District
Ford High School	District
Thomas High School	Charter

Name: type, dtype: object

In [5]:

```
school_students = school_dataComp["school_name"].value_counts()  
school_students
```

Out[5]:

Bailey High School	4976
Johnson High School	4761
Hernandez High School	4635
Rodriguez High School	3999
Figueroa High School	2949
Huang High School	2917
Ford High School	2739
Wilson High School	2283
Cabrera High School	1858
Wright High School	1800
Shelton High School	1761
Thomas High School	1635
Griffin High School	1468
Pena High School	962
Holden High School	427

Name: school_name, dtype: int64

In [6]:

```
school_budget = school_data.groupby(["school_name"])[ "budget" ].sum()  
school_budget
```

Out[6]:

school_name	
Bailey High School	3124928
Cabrera High School	1081356
Figueroa High School	1884411
Ford High School	1763916
Griffin High School	917500
Hernandez High School	3022020
Holden High School	248087
Huang High School	1910635
Johnson High School	3094650
Pena High School	585858
Rodriguez High School	2547363
Shelton High School	1056600
Thomas High School	1043130
Wilson High School	1319574
Wright High School	1049400

Name: budget, dtype: int64

In [7]:

```
per_student_budget = school_budget / school_students  
per_student_budget
```

Out[7]:

Bailey High School	628.0
Cabrera High School	582.0
Figueroa High School	639.0
Ford High School	644.0
Griffin High School	625.0
Hernandez High School	652.0
Holden High School	581.0
Huang High School	655.0
Johnson High School	650.0
Pena High School	609.0
Rodriguez High School	637.0
Shelton High School	600.0
Thomas High School	638.0
Wilson High School	578.0
Wright High School	583.0

dtype: float64

In [8]:

```
school_math = school_dataComp.groupby(["school_name"])["math_score"].mean()  
school_math
```

Out[8]:

school_name	
Bailey High School	77.048432
Cabrera High School	83.061895
Figueroa High School	76.711767
Ford High School	77.102592
Griffin High School	83.351499
Hernandez High School	77.289752
Holden High School	83.803279
Huang High School	76.629414
Johnson High School	77.072464
Pena High School	83.839917
Rodriguez High School	76.842711
Shelton High School	83.359455
Thomas High School	83.418349
Wilson High School	83.274201
Wright High School	83.682222

Name: math_score, dtype: float64

In [9]:

```
school_read = school_dataComp.groupby(["school_name"])["reading_score"].mean()  
school_read
```

Out[9]:

school_name	
Bailey High School	81.033963
Cabrera High School	83.975780
Figueroa High School	81.158020
Ford High School	80.746258
Griffin High School	83.816757
Hernandez High School	80.934412
Holden High School	83.814988
Huang High School	81.182722
Johnson High School	80.966394
Pena High School	84.044699
Rodriguez High School	80.744686
Shelton High School	83.725724
Thomas High School	83.848930
Wilson High School	83.989488
Wright High School	83.955000

Name: reading_score, dtype: float64

In [10]:

```
mathpass_school = school_dataComp[(school_dataComp["math_score"] >= 70)]  
school_mathpass_pct = mathpass_school.groupby(["school_name"])["student_name"].count() / school_students *100  
school_mathpass_pct
```

Out[10]:

Bailey High School	66.680064
Cabrera High School	94.133477
Figueroa High School	65.988471
Ford High School	68.309602
Griffin High School	93.392371
Hernandez High School	66.752967
Holden High School	92.505855
Huang High School	65.683922
Johnson High School	66.057551
Pena High School	94.594595
Rodriguez High School	66.366592
Shelton High School	93.867121
Thomas High School	93.272171
Wilson High School	93.867718
Wright High School	93.333333

dtype: float64

In [11]:

```
readpass_school = school_dataComp[(school_dataComp["reading_score"] >= 70)]
school_readpass_pct = readpass_school.groupby(["school_name"])["student_name"].count() / school_students * 100
school_readpass_pct
```

Out[11]:

Bailey High School	81.933280
Cabrera High School	97.039828
Figueroa High School	80.739234
Ford High School	79.299014
Griffin High School	97.138965
Hernandez High School	80.862999
Holden High School	96.252927
Huang High School	81.316421
Johnson High School	81.222432
Pena High School	95.945946
Rodriguez High School	80.220055
Shelton High School	95.854628
Thomas High School	97.308869
Wilson High School	96.539641
Wright High School	96.611111

dtype: float64

In [12]:

```
school_overallPass = school_dataComp[(school_dataComp["math_score"] >= 70) & (school_dataComp["reading_score"] >= 70)]
school_overall_pass = school_overallPass.groupby(["school_name"])["student_name"].count() / school_students * 100
school_overall_pass
```

Out[12]:

Bailey High School	54.642283
Cabrera High School	91.334769
Figueroa High School	53.204476
Ford High School	54.289887
Griffin High School	90.599455
Hernandez High School	53.527508
Holden High School	89.227166
Huang High School	53.513884
Johnson High School	53.539172
Pena High School	90.540541
Rodriguez High School	52.988247
Shelton High School	89.892107
Thomas High School	90.948012
Wilson High School	90.582567
Wright High School	90.333333

dtype: float64

In [13]:

```
school_summary = pd.DataFrame({
    "School Type": school_type,
    "Students": school_students,
    "School Budget": school_budget,
    "Budget per Student": per_student_budget,
    "Average Math Score": school_math,
    "Average Reading Score": school_read,
    "% Passing Math": school_mathpass_pct,
    "% Passing Reading": school_readpass_pct,
    "Overall Pass Rate": school_overall_pass
})

school_summary = school_summary[["School Type", "Students", "School Budget", "Budget per Student",
    "Average Math Score", "Average Reading Score", "% Passing Math",
    "% Passing Reading", "Overall Pass Rate"]]

school_summary["School Budget"] = school_summary["School Budget"].map("${:,.2f}".format)
school_summary["Budget per Student"] = school_summary["Budget per Student"].map("${:,.2f}".format)
school_summary["Average Math Score"] = school_summary["Average Math Score"].map("{:.2f}".format)
school_summary["Average Reading Score"] = school_summary["Average Reading Score"].map("{:.2f}".format)
school_summary["% Passing Math"] = school_summary["% Passing Math"].map("{:.2f}%".format)
school_summary["% Passing Reading"] = school_summary["% Passing Reading"].map("{:.2f}%".format)
school_summary["Overall Pass Rate"] = school_summary["Overall Pass Rate"].map("{:.2f}%".format)

school_summary
```

Out[13]:

	School Type	Students	School Budget	Budget per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Pass Rate
Bailey High School	District	4976	\$3,124,928.00	\$628.00	77.05	81.03	66.68%	81.93%	54.64%
Cabrera High School	Charter	1858	\$1,081,356.00	\$582.00	83.06	83.98	94.13%	97.04%	91.33%
Figueroa High School	District	2949	\$1,884,411.00	\$639.00	76.71	81.16	65.99%	80.74%	53.20%
Ford High School	District	2739	\$1,763,916.00	\$644.00	77.10	80.75	68.31%	79.30%	54.29%
Griffin High School	Charter	1468	\$917,500.00	\$625.00	83.35	83.82	93.39%	97.14%	90.60%
Hernandez High School	District	4635	\$3,022,020.00	\$652.00	77.29	80.93	66.75%	80.86%	53.53%
Holden High School	Charter	427	\$248,087.00	\$581.00	83.80	83.81	92.51%	96.25%	89.23%
Huang High School	District	2917	\$1,910,635.00	\$655.00	76.63	81.18	65.68%	81.32%	53.51%
Johnson High School	District	4761	\$3,094,650.00	\$650.00	77.07	80.97	66.06%	81.22%	53.54%
Pena High School	Charter	962	\$585,858.00	\$609.00	83.84	84.04	94.59%	95.95%	90.54%
Rodriguez High School	District	3999	\$2,547,363.00	\$637.00	76.84	80.74	66.37%	80.22%	52.99%
Shelton High School	Charter	1761	\$1,056,600.00	\$600.00	83.36	83.73	93.87%	95.85%	89.89%
Thomas High School	Charter	1635	\$1,043,130.00	\$638.00	83.42	83.85	93.27%	97.31%	90.95%
Wilson High School	Charter	2283	\$1,319,574.00	\$578.00	83.27	83.99	93.87%	96.54%	90.58%
Wright High School	Charter	1800	\$1,049,400.00	\$583.00	83.68	83.95	93.33%	96.61%	90.33%

Top Performing Schools (By % Overall Passing)

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

In [14]:

```
top_schools = school_summary.sort_values("Overall Pass Rate", ascending=False)
top_schools.head()
```

Out[14]:

	School Type	Students	School Budget	Budget per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Pass Rate
Cabrera High School	Charter	1858	\$1,081,356.00	\$582.00	83.06	83.98	94.13%	97.04%	91.33%
Thomas High School	Charter	1635	\$1,043,130.00	\$638.00	83.42	83.85	93.27%	97.31%	90.95%
Griffin High School	Charter	1468	\$917,500.00	\$625.00	83.35	83.82	93.39%	97.14%	90.60%
Wilson High School	Charter	2283	\$1,319,574.00	\$578.00	83.27	83.99	93.87%	96.54%	90.58%
Pena High School	Charter	962	\$585,858.00	\$609.00	83.84	84.04	94.59%	95.95%	90.54%

Bottom Performing Schools (By % Overall Passing)

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

In [15]:

```
bottom_schools = school_summary.sort_values("Overall Pass Rate")
bottom_schools.head()
```

Out[15]:

	School Type	Students	School Budget	Budget per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Pass Rate
Rodriguez High School	District	3999	\$2,547,363.00	\$637.00	76.84	80.74	66.37%	80.22%	52.99%
Figueroa High School	District	2949	\$1,884,411.00	\$639.00	76.71	81.16	65.99%	80.74%	53.20%
Huang High School	District	2917	\$1,910,635.00	\$655.00	76.63	81.18	65.68%	81.32%	53.51%
Hernandez High School	District	4635	\$3,022,020.00	\$652.00	77.29	80.93	66.75%	80.86%	53.53%
Johnson High School	District	4761	\$3,094,650.00	\$650.00	77.07	80.97	66.06%	81.22%	53.54%

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 [16]:

```
math9=student_data.loc[student_data["grade"] == "9th"].groupby(["school_name"])["math_score"].mean()
math10=student_data.loc[student_data["grade"] == "10th"].groupby(["school_name"])["math_score"].mean()
math11=student_data.loc[student_data["grade"] == "11th"].groupby(["school_name"])["math_score"].mean()
math12=student_data.loc[student_data["grade"] == "12th"].groupby(["school_name"])["math_score"].mean()

mathbygrade=pd.DataFrame({"9th":math9, "10th":math10, "11th":math11, "12th":math12})

mathbygrade["9th"] = mathbygrade["9th"].map("{:.2f}".format)
mathbygrade["10th"] = mathbygrade["10th"].map("{:.2f}".format)
mathbygrade["11th"] = mathbygrade["11th"].map("{:.2f}".format)
mathbygrade["12th"] = mathbygrade["12th"].map("{:.2f}".format)

mathbygrade
```

Out[16]:

	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

Reading Score by Grade

- Perform the same operations as above for reading scores

In [17]:

```
read9=student_data.loc[student_data["grade"] == "9th"].groupby(["school_name"])["reading_score"].mean()
read10=student_data.loc[student_data["grade"] == "10th"].groupby(["school_name"])["reading_score"].mean()
read11=student_data.loc[student_data["grade"] == "11th"].groupby(["school_name"])["reading_score"].mean()
read12=student_data.loc[student_data["grade"] == "12th"].groupby(["school_name"])["reading_score"].mean()

readbygrade=pd.DataFrame({"9th":read9, "10th":read10, "11th":read11, "12th":read12})

readbygrade["9th"] = readbygrade["9th"].map("{:.2f}".format)
readbygrade["10th"] = readbygrade["10th"].map("{:.2f}".format)
readbygrade["11th"] = readbygrade["11th"].map("{:.2f}".format)
readbygrade["12th"] = readbygrade["12th"].map("{:.2f}".format)

readbygrade
```

Out[17]:

	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

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 [18]:

```
school_summary.head()
```

Out[18]:

	School Type	Students	School Budget	Budget per Student	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Pass Rate
Bailey High School	District	4976	\$3,124,928.00	\$628.00	77.05	81.03	66.68%	81.93%	54.64%
Cabrera High School	Charter	1858	\$1,081,356.00	\$582.00	83.06	83.98	94.13%	97.04%	91.33%
Figueroa High School	District	2949	\$1,884,411.00	\$639.00	76.71	81.16	65.99%	80.74%	53.20%
Ford High School	District	2739	\$1,763,916.00	\$644.00	77.10	80.75	68.31%	79.30%	54.29%
Griffin High School	Charter	1468	\$917,500.00	\$625.00	83.35	83.82	93.39%	97.14%	90.60%

In [48]:

```
bins = [0, 584.99, 629.99, 644.99, 675.99]
group_names = ["<$584", "$585-629", "$630-644", "$645-675"]

school_dataComp['spending_bins'] = pd.cut(school_dataComp['budget']/school_dataComp['size'], bins, labels = group_names)

bySpending = school_dataComp.groupby("spending_bins")

avgMath = bySpending["math_score"].mean()
avgRead = bySpending["reading_score"].mean()
passMath = school_dataComp[school_dataComp["math_score"] >= 70].groupby("spending_bins")["Student ID"].count()/bySpending["Student ID"].count()
passRead = school_dataComp[school_dataComp["reading_score"] >= 70].groupby("spending_bins")["Student ID"].count()/bySpending["Student ID"].count()
overall = school_dataComp[(school_dataComp["reading_score"] >= 70) & (school_dataComp["math_score"] >= 70)].groupby("spending_bins")["Student ID"].count()/bySpending["Student ID"].count()

scores_byspend = pd.DataFrame({
    "Average Math Score": avgMath,
    "Average Reading Score": avgRead,
    "% Passing Math": passMath,
    "% Passing Reading": passRead,
    "Overall Passing Rate": overall
})

scores_byspend.index.name = "Spending Ranges (Per Student)"
scores_byspend = scores_byspend.reindex(group_names)

scores_byspend["Average Math Score"] = scores_byspend["Average Math Score"].map("{:.1f}".format)
scores_byspend["Average Reading Score"] = scores_byspend["Average Reading Score"].map("{:.1f}".format)
scores_byspend["% Passing Math"] = scores_byspend["% Passing Math"].map("{:.1%}".format)
scores_byspend["% Passing Reading"] = scores_byspend["% Passing Reading"].map("{:.1%}".format)
scores_byspend["Overall Passing Rate"] = scores_byspend["Overall Passing Rate"].map("{:.1%}".format)

scores_byspend
```

Out[48]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Spending Ranges (Per Student)					
<\$584	83.4	84.0	93.7%	96.7%	90.6%
\$585-629	80.0	82.3	79.1%	88.5%	70.9%
\$630-644	77.8	81.3	70.6%	82.6%	58.8%
\$645-675	77.0	81.0	66.2%	81.1%	53.5%

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

In [56]:

```
bins = [0, 999, 1999, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

school_dataComp["size_bins"] = pd.cut(school_dataComp["size"], bins, labels = group_names)

bySize = school_dataComp.groupby("size_bins")

avgMath = bySize["math_score"].mean()
avgRead = bySize["reading_score"].mean()
passMath = school_dataComp[school_dataComp["math_score"] >= 70].groupby("size_bins")["Student ID"].count()/bySize["Student ID"].count()
passread = school_dataComp[school_dataComp["reading_score"] >= 70].groupby("size_bins")["Student ID"].count()/bySize["Student ID"].count()
overall = school_dataComp[(school_dataComp["reading_score"] >= 70) & (school_dataComp["math_score"] >= 70)].groupby("size_bins")["Student ID"].count()/bySize["Student ID"].count()

scores_bysize = pd.DataFrame({
    "Average Math Score": avgMath,
    "Average Reading Score": avgRead,
    "% Passing Math": passMath,
    "% Passing Reading": passread,
    "Overall Passing Rate": overall
})

scores_bysize.index.name = "Total Students"
scores_bysize = scores_bysize.reindex(group_names)

scores_bysize["Average Math Score"] = scores_bysize["Average Math Score"].map("{:.1f}".format)
scores_bysize["Average Reading Score"] = scores_bysize["Average Reading Score"].map("{:.1f}".format)
scores_bysize["% Passing Math"] = scores_bysize["% Passing Math"].map("{:.1%}".format)
scores_bysize["% Passing Reading"] = scores_bysize["% Passing Reading"].map("{:.1%}".format)
scores_bysize["Overall Passing Rate"] = scores_bysize["Overall Passing Rate"].map("{:.1%}".format)

scores_bysize
```

Out[56]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	Overall Passing Rate
Total Students					
Small (<1000)	83.8	84.0	94.0%	96.0%	90.1%
Medium (1000-2000)	83.4	83.9	93.6%	96.8%	90.6%
Large (2000-5000)	77.5	81.2	68.7%	82.1%	56.6%

Scores by School Type

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

In [78]:

```

byType = school_dataComp.groupby("type")

avgMath = byType["math_score"].mean()
avgRead = byType["reading_score"].mean()
passMath = school_dataComp[school_dataComp["math_score"] >= 70].groupby("type")["Student ID"].count()/byType["Student ID"].count()
passRead = school_dataComp[school_dataComp["reading_score"] >= 70].groupby("type")["Student ID"].count()/byType["Student ID"].count()
overall = school_dataComp[(school_dataComp["reading_score"] >= 70) & (school_dataComp["math_score"] >= 70)].groupby("type")["Student ID"].count()/byType["Student ID"].count()

scores_bytype = pd.DataFrame({
    "Average Math Score": avgMath,
    "Average Reading Score": avgRead,
    "% Passing Math": passMath,
    "% Passing Reading": passRead,
    "Overall Passing Rate": overall
})

scores_bytype.index.name = "Type of School"

scores_bytype["Average Math Score"] = scores_bytype["Average Math Score"].map("{:.1f}".format)
scores_bytype["Average Reading Score"] = scores_bytype["Average Reading Score"].map("{:.1f}".format)
scores_bytype["% Passing Math"] = scores_bytype["% Passing Math"].map("{:.1%}".format)
scores_bytype["% Passing Reading"] = scores_bytype["% Passing Reading"].map("{:.1%}".format)
scores_bytype["Overall Passing Rate"] = scores_bytype["Overall Passing Rate"].map("{:.1%}".format)

scores_bytype

```

Out[78]:

	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%

In []: