In [1]:

```python
# Dependencies and Setup
import pandas as pd

# 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 DataFrames
school_data = pd.read_csv(school_data_to_load)
student_data = pd.read_csv(student_data_to_load)

# Combine the data into a single dataset.
school_data_complete = pd.merge(student_data, school_data, how="left", on=["school_name", "school_name"])
```

In [2]:

```python
# Create dataframe to store district summary
district_summary_df = pd.DataFrame([{
    # Get total number of unique schools
    "Total Schools" : school_data_complete["School ID"].nunique(),
    # Get total number of students
    "Total Students" : school_data_complete["Student ID"].count(),
    # Get total budget of each school
    "Total Budget" : school_data_complete.drop_duplicates("School ID")["budget"].sum(),
    # Get average math score
    "Average Math Score" : school_data_complete["math_score"].mean(),
    # Get average reading score
    "Average Reading Score" : school_data_complete["reading_score"].mean(),
    # Get passing math percentage
    "% Passing Math" : school_data_complete["Student ID"].loc[school_data_complete["math_score"] >= 70].count() / school_data_complete["Stu
    # Get passing reading percentage
    "% Passing Reading" : school_data_complete["Student ID"].loc[school_data_complete["reading_score"] >= 70].count() / school_data_complet
    # Get overall passing percentage
    "% Overall Passing" : school_data_complete["Student ID"].loc[(school_data_complete["reading_score"] >= 70) & (school_data_complete["mat
}])

# Convert format
district_summary_df["Total Students"] = district_summary_df["Total Students"].map("{:,}".format)
district_summary_df["Total Budget"] = district_summary_df["Total Budget"].astype(float).map("${:,.2f}".format)

district_summary_df
```

Out[2]:

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

In [3]:

```
# Group by school
group_school = school_data_complete.groupby("school_name")
# Group students passed math by school
group_pass_math = school_data_complete.loc[school_data_complete["math_score"] >= 70].groupby("school_name")
# Group students passed reading by school
group_pass_reading = school_data_complete.loc[school_data_complete["reading_score"] >= 70].groupby("school_name")
# Group students passed both by school
group_pass_overall = school_data_complete.loc[(school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"] >= 70)].g

# Create dataframe to hold result
school_summary_df = pd.DataFrame({
    # Get school types
    "School Type" : group_school.first()["type"],
    # Get total number of students
    "Total Students" : group_school["Student ID"].count(),
    # Get each school budget
    "Total School Budget" : group_school["budget"].mean(),
    # Get budgets per student
    "Per Student Budget" : group_school["budget"].mean() / group_school["Student ID"].count(),
    # Get avg math scores by school
    "Average Math Score" : group_school["math_score"].mean(),
    # Get avg reading scores by school
    "Average Reading Score" : group_school["reading_score"].mean(),
    # Get math passing rates by school
    "% Passing Math" : group_pass_math["Student ID"].count() / group_school["Student ID"].count() * 100,
    # Get reading passing rates by school
    "% Passing Reading" : group_pass_reading["Student ID"].count() / group_school["Student ID"].count() * 100,
    # Get overall passing rates by school
    "% Overall Passing" : group_pass_overall["Student ID"].count() / group_school["Student ID"].count() * 100
})

# Set school_name colume to index to get "Per Student Budget" into school_data_complete dataframe
school_data_complete.set_index("school_name", inplace = True)
school_data_complete["Per Student Budget"] = school_summary_df["Per Student Budget"]
school_data_complete.reset_index(inplace = True)

# Convert formats
school_summary_df["Total School Budget"] = school_summary_df["Total School Budget"].map("${:,.2f}".format)
school_summary_df["Per Student Budget"] = school_summary_df["Per Student Budget"].map("${:,.2f}".format)

school_summary_df
```

Out [3]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
school_name									
Bailey High School	District	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.680064	81.933280	54.642283
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	91.334769
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	53.204476
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602	79.299014	54.289887
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	90.599455
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752967	80.862999	53.527508
Holden High School	Charter	427	$248,087.00	$581.00	83.803279	83.814988	92.505855	96.252927	89.227166
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	53.513884
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	53.539172
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	90.540541
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	52.988247
Shelton High School	Charter	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.867121	95.854628	89.892107
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	90.948012
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	90.582567
Wright High School	Charter	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.333333	96.611111	90.333333

In [4]:

```python
# Descending sort
school_summary_df = school_summary_df.sort_values(by = ["% Overall Passing"], ascending = False)
school_summary_df.head()
```

Out[4]:

school_name	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	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	91.334769
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	90.948012
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	90.599455
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	90.582567
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	90.540541

In [5]:

```python
# Ascending sort
school_summary_df = school_summary_df.sort_values(by = ["% Overall Passing"])
school_summary_df.head()
```

Out[5]:

school_name	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	52.988247
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	53.204476
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	53.513884
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752967	80.862999	53.527508
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	53.539172

In [6]:

```python
# Group by "school_name", "grade" to get avg math, reading scores by grade, school
scores_by_grade_df = school_data_complete[["math_score", "reading_score", "school_name", "grade"]].groupby(["school_name", "grade"]).mean()
# "grade" index to column
scores_by_grade_df.reset_index(level = "grade", inplace = True)

# Make dataframe to hold the math scores by grade result
math_by_grade_df = pd.DataFrame(index = sorted(school_data_complete["school_name"].unique()))

math_by_grade_df["9th"] = scores_by_grade_df["math_score"].loc[scores_by_grade_df["grade"] == "9th"]
math_by_grade_df["10th"] = scores_by_grade_df["math_score"].loc[scores_by_grade_df["grade"] == "10th"]
math_by_grade_df["11th"] = scores_by_grade_df["math_score"].loc[scores_by_grade_df["grade"] == "11th"]
math_by_grade_df["12th"] = scores_by_grade_df["math_score"].loc[scores_by_grade_df["grade"] == "12th"]

math_by_grade_df
```

Out[6]:

	9th	10th	11th	12th
Bailey High School	77.083676	76.996772	77.515588	76.492218
Cabrera High School	83.094697	83.154506	82.765560	83.277487
Figueroa High School	76.403037	76.539974	76.884344	77.151369
Ford High School	77.361345	77.672316	76.918058	76.179963
Griffin High School	82.044010	84.229064	83.842105	83.356164
Hernandez High School	77.438495	77.337408	77.136029	77.186567
Holden High School	83.787402	83.429825	85.000000	82.855422
Huang High School	77.027251	75.908735	76.446602	77.225641
Johnson High School	77.187857	76.691117	77.491653	76.863248
Pena High School	83.625455	83.372000	84.328125	84.121547
Rodriguez High School	76.859966	76.612500	76.395626	77.690748
Shelton High School	83.420755	82.917411	83.383495	83.778976
Thomas High School	83.590022	83.087886	83.498795	83.497041
Wilson High School	83.085578	83.724422	83.195326	83.035794
Wright High School	83.264706	84.010288	83.836782	83.644986

In [7]:

```python
# Make dataframe to hold the reading scores by grade result
reading_by_grade_df = pd.DataFrame(index = sorted(school_data_complete["school_name"].unique()))

reading_by_grade_df["9th"] = scores_by_grade_df["reading_score"].loc[scores_by_grade_df["grade"] == "9th"]
reading_by_grade_df["10th"] = scores_by_grade_df["reading_score"].loc[scores_by_grade_df["grade"] == "10th"]
reading_by_grade_df["11th"] = scores_by_grade_df["reading_score"].loc[scores_by_grade_df["grade"] == "11th"]
reading_by_grade_df["12th"] = scores_by_grade_df["reading_score"].loc[scores_by_grade_df["grade"] == "12th"]

reading_by_grade_df
```

Out[7]:

	9th	10th	11th	12th
Bailey High School	81.303155	80.907183	80.945643	80.912451
Cabrera High School	83.676136	84.253219	83.788382	84.287958
Figueroa High School	81.198598	81.408912	80.640339	81.384863
Ford High School	80.632653	81.262712	80.403642	80.662338
Griffin High School	83.369193	83.706897	84.288089	84.013699
Hernandez High School	80.866860	80.660147	81.396140	80.857143
Holden High School	83.677165	83.324561	83.815534	84.698795
Huang High School	81.290284	81.512386	81.417476	80.305983
Johnson High School	81.260714	80.773431	80.616027	81.227564
Pena High School	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	80.993127	80.629808	80.864811	80.376426
Shelton High School	84.122642	83.441964	84.373786	82.781671
Thomas High School	83.728850	84.254157	83.585542	83.831361
Wilson High School	83.939778	84.021452	83.764608	84.317673
Wright High School	83.833333	83.812757	84.156322	84.073171

In [8]:

```python
# Set bins and labels
bins = [0, 585, 629, 644, 675]
labels = ["<$585", "$585-629", "$630-644", "$645-675"]

# Binning data
school_data_complete["Spending Ranges (Per Student)"] = pd.cut(school_data_complete["Per Student Budget"], bins, labels = labels)

# Group data by Spending Ranges (Per Student) to get total number of students by Spending Ranges
group_budget = school_data_complete.groupby("Spending Ranges (Per Student)")
# Group data passed math to get total number of passed students
group_budget_math = school_data_complete.loc[school_data_complete["math_score"] >= 70].groupby("Spending Ranges (Per Student)")
# Group data passed reading to get total number of passed students
group_budget_reading = school_data_complete.loc[school_data_complete["reading_score"] >= 70].groupby("Spending Ranges (Per Student)")
# Group data overall passed to get total number of passed students
group_budget_overall = school_data_complete.loc[(school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"] >= 70)]

# Create dataframe to store result
scores_by_spending_df = pd.DataFrame({
    # Avg math scores by spending ranges
    "Average Math Score" : group_budget["math_score"].mean(),
    # Avg reading scores by spending ranges
    "Average Reading Score" : group_budget["reading_score"].mean(),
    # Get passing math rates
    "% Passing Math" : group_budget_math["Student ID"].count() / group_budget["Student ID"].count() * 100,
    # Get passing reading rates
    "% Passing Reading" : group_budget_reading["Student ID"].count() / group_budget["Student ID"].count() * 100,
    # Get overall passing rates
    "% Overall Passing" : group_budget_overall["Student ID"].count() / group_budget["Student ID"].count() * 100
})

scores_by_spending_df
```

Out[8]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Spending Ranges (Per Student)					
<$585	83.363065	83.964039	93.702889	96.686558	90.640704
$585-629	79.982873	82.312643	79.109851	88.513145	70.939239
$630-644	77.821056	81.301007	70.623565	82.600247	58.841194
$645-675	77.049297	81.005604	66.230813	81.109397	53.528791

In [9]:

```
# Set bins and labels
bins_size = [0, 1000, 2000, 5000]
labels_size = ["Small(<1000)", "Medium(1000-1999)", "Large(2000-5000)"]

# Binning data
school_data_complete["School Size"] = pd.cut(school_data_complete["size"], bins_size, labels = labels_size)

# Group data by School Size to get total number of students by School Size
group_size = school_data_complete.groupby("School Size")
# Group data passed math to get total number of passed students
group_size_math = school_data_complete.loc[school_data_complete["math_score"] >= 70].groupby("School Size")
# Group data passed reading to get total number of passed students
group_size_reading = school_data_complete.loc[school_data_complete["reading_score"] >= 70].groupby("School Size")
# Group data overall passed to get total number of passed students
group_size_overall = school_data_complete.loc[(school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"] >= 70)].g

# Create dataframe to store result
scores_by_size_df = pd.DataFrame({
    # Avg math scores by school size
    "Average Math Score" : group_size["math_score"].mean(),
    # Avg reading scores by school size
    "Average Reading Score" : group_size["reading_score"].mean(),
    # Get passing math rates
    "% Passing Math" : group_size_math["Student ID"].count() / group_size["Student ID"].count() * 100,
    # Get passing reading rates
    "% Passing Reading" : group_size_reading["Student ID"].count() / group_size["Student ID"].count() * 100,
    # Get overall passing rates
    "% Overall Passing" : group_size_overall["Student ID"].count() / group_size["Student ID"].count() * 100
})

scores_by_size_df
```

Out[9]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Small(<1000)	83.828654	83.974082	93.952484	96.040317	90.136789
Medium(1000-1999)	83.372682	83.867989	93.616522	96.773058	90.624267
Large(2000-5000)	77.477597	81.198674	68.652380	82.125158	56.574046

In [10]:

```
# Group data by School Size to get total number of students by type
group_type = school_data_complete.groupby("type")
# Group data passed math to get total number of passed students
group_type_math = school_data_complete.loc[school_data_complete["math_score"] >= 70].groupby("type")
# Group data passed reading to get total number of passed students
group_type_reading = school_data_complete.loc[school_data_complete["reading_score"] >= 70].groupby("type")
# Group data overall passed to get total number of passed students
group_type_overall = school_data_complete.loc[(school_data_complete["reading_score"] >= 70) & (school_data_complete["math_score"] >= 70)].g

# Create dataframe to store result
scores_by_type_df = pd.DataFrame({
    # Avg math scores by school size
    "Average Math Score" : group_type["math_score"].mean(),
    # Avg reading scores by school size
    "Average Reading Score" : group_type["reading_score"].mean(),
    # Get passing math rates
    "% Passing Math" : group_type_math["Student ID"].count() / group_type["Student ID"].count() * 100,
    # Get passing reading rates
    "% Passing Reading" : group_type_reading["Student ID"].count() / group_type["Student ID"].count() * 100,
    # Get overall passing rates
    "% Overall Passing" : group_type_overall["Student ID"].count() / group_type["Student ID"].count() * 100
})

scores_by_type_df
```

Out[10]:

type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Charter	83.406183	83.902821	93.701821	96.645891	90.560932
District	76.987026	80.962485	66.518387	80.905249	53.695878

Result

- Students of large schools passing rates are relatively lower than other schools
- Charter school students' academic performances are better than district type school students