In [1]:

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

# File to Load (Remember to change the path if needed.)
school_data_to_load = "Resources/schools_complete.csv"
student_data_to_load = "Resources/students_complete.csv"

# Read the School Data and Student Data and store into a Pandas DataFrame
school_data_df = pd.read_csv(school_data_to_load)
student_data_df = pd.read_csv(student_data_to_load)

# Cleaning Student Names and Replacing Substrings in a Python String
# Add each prefix and suffix to remove to a list.
prefixes_suffixes = ["Dr. ", "Mr. ","Ms. ", "Mrs. ", "Miss ", " MD", " DDS

# Iterate through the words in the "prefixes_suffixes" list and replace th
for word in prefixes_suffixes:
    student_data_df["student_name"] = student_data_df["student_name"].str.

# Check names.
student_data_df.head(10)
```

```
C:\Users\mobi\anaconda3\envs\PythonData\lib\site-packages\ipykernel_launcher
.py:20: FutureWarning: The default value of regex will change from True to F
alse in a future version.
```

Out[1]:

	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	Richard Scott	M	12th	Huang High School	67	58
4	4	Bonnie Ray	F	9th	Huang High School	97	84
5	5	Bryan Miranda	M	9th	Huang High School	94	94
6	6	Sheena Carter	F	11th	Huang High School	82	80
7	7	Nicole Baker	F	12th	Huang High School	96	69
8	8	Michael Roth	M	10th	Huang High School	95	87
9	9	Matthew Greene	M	10th	Huang High School	96	84

Deliverable 1: Replace the reading and math scores.

Replace the 9th grade reading and math scores at Thomas High School with NaN.

In [2]:

```
# Install numpy using conda install numpy or pip install numpy.
# Step 1. Import numpy as np.
import numpy as np
```

In [3]:

```
# Step 2. Use the loc method on the student_data_df to select all the read
student_data_df.loc[(student_data_df["grade"] == "9th") & (student_data_df
student_data_df
```

Out[3]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_
0	0	Paul Bradley	M	9th	Huang High School	66.0	
1	1	Victor Smith	M	12th	Huang High School	94.0	
2	2	Kevin Rodriguez	M	12th	Huang High School	90.0	
3	3	Richard Scott	M	12th	Huang High School	67.0	
4	4	Bonnie Ray	F	9th	Huang High School	97.0	
...	...	...	...	...	...	...	
39165	39165	Donna Howard	F	12th	Thomas High School	99.0	
39166	39166	Dawn Bell	F	10th	Thomas High School	95.0	
39167	39167	Rebecca Tanner	F	9th	Thomas High School	NaN	
39168	39168	Desiree Kidd	F	10th	Thomas High School	99.0	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95.0	

39170 rows × 7 columns

In [4]:

```
#  Step 3. Refactor the code in Step 2 to replace the math scores with NaN
student_data_df.loc[(student_data_df["grade"] == "9th") & (student_data_df
student_data_df
```

Out[4]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_
0	0	Paul Bradley	M	9th	Huang High School	66.0	
1	1	Victor Smith	M	12th	Huang High School	94.0	
2	2	Kevin Rodriguez	M	12th	Huang High School	90.0	
3	3	Richard Scott	M	12th	Huang High School	67.0	
4	4	Bonnie Ray	F	9th	Huang High School	97.0	
...	...	...	...	...	...	...	
39165	39165	Donna Howard	F	12th	Thomas High School	99.0	
39166	39166	Dawn Bell	F	10th	Thomas High School	95.0	
39167	39167	Rebecca Tanner	F	9th	Thomas High School	NaN	
39168	39168	Desiree Kidd	F	10th	Thomas High School	99.0	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95.0	

39170 rows × 7 columns

In [5]:

```
#  Step 4. Check the student data for NaN's.

# check for missing data

print(f'{student_data_df.count()}')
# reading and math score missing values compared to other columns.

# Count students in 9th grade and Thomas High school.
school_data_complete_df = pd.merge(student_data_df, school_data_df, how="l

target_group_count = school_data_complete_df["student_name"].loc[(school_d
print(f'Number of 9th Graders at Thomas High School {target_group_count}')

# Count above should match both counts of all students with null reading a

print(f'Number of reading grades null {student_data_df["reading_score"].is
print(f'Number of math grades null {student_data_df["math_score"].isnull()

# ALL MATCH DATA OK
```

```
Student ID       39170
student_name     39170
gender           39170
grade            39170
school_name      39170
reading_score    38709
math_score       38709
dtype: int64
Number of 9th Graders at Thomas High School 461
Number of reading grades null 461
Number of math grades null 461
```

Deliverable 2 : Repeat the school district analysis

District Summary

In [6]:

```
# Combine the data into a single dataset
school_data_complete_df = pd.merge(student_data_df, school_data_df, how="l
school_data_complete_df.head()
```

Out[6]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score
0	0	Paul Bradley	M	9th	Huang High School	66.0	79.0
1	1	Victor Smith	M	12th	Huang High School	94.0	61.0
2	2	Kevin Rodriguez	M	12th	Huang High School	90.0	60.0
3	3	Richard Scott	M	12th	Huang High School	67.0	58.0
4	4	Bonnie Ray	F	9th	Huang High School	97.0	84.0

In [7]:
```
# Calculate the Totals (Schools and Students)
school_count = len(school_data_complete_df["school_name"].unique())
student_count = school_data_complete_df["Student ID"].count()

# Calculate the Total Budget
total_budget = school_data_df["budget"].sum()
```

In [8]:
```
# Calculate the Average Scores using the "clean_student_data".
average_reading_score = school_data_complete_df["reading_score"].mean()
average_math_score = school_data_complete_df["math_score"].mean()
```

In [9]:
```
# Step 1. Get the number of students that are in ninth grade at Thomas Hig
# These students have no grades.

nine_thomas_count = school_data_complete_df["student_name"].loc[(school_da
print(nine_thomas_count)


# Get the total student count
student_count = school_data_complete_df["Student ID"].count()


# Step 2. Subtract the number of students that are in ninth grade at
# Thomas High School from the total student count to get the new total stu

total_minus_thomas_nine = (student_count - nine_thomas_count)
print(f'The total number of students not including 9th graders from Thomas
```

```
461
The total number of students not including 9th graders from Thomas High Scho
ol is 38709
```

In [10]:

```
# Calculate the passing rates using the "clean_student_data".
passing_math_count = school_data_complete_df[(school_data_complete_df["mat
passing_reading_count = school_data_complete_df[(school_data_complete_df["
```

In [11]:

```
# Step 3. Calculate the passing percentages with the new total student cou
passing_math_percentage = (passing_math_count/total_minus_thomas_nine) * 1
passing_reading_percentage = (passing_reading_count/total_minus_thomas_nin
```

In [12]:

```
# Calculate the students who passed both reading and math.
passing_math_reading = school_data_complete_df[(school_data_complete_df["m
                                               & (school_data_complete_df[

# Calculate the number of students that passed both reading and math.
overall_passing_math_reading_count = passing_math_reading["student_name"].


# Step 4.Calculate the overall passing percentage with new total student c
overall_passing_percentage = (overall_passing_math_reading_count/total_min
```

In [13]:

```
# Create a DataFrame
district_summary_df = pd.DataFrame(
          [{"Total Schools": school_count,
          "Total Students": student_count,
          "Total Budget": total_budget,
          "Average Math Score": average_math_score,
          "Average Reading Score": average_reading_score,
          "% Passing Math": passing_math_percentage,
         "% Passing Reading": passing_reading_percentage,
        "% Overall Passing": overall_passing_percentage}])



# Format the "Total Students" to have the comma for a thousands separator.
district_summary_df["Total Students"] = district_summary_df["Total Student
# Format the "Total Budget" to have the comma for a thousands separator, a
district_summary_df["Total Budget"] = district_summary_df["Total Budget"].
# Format the columns.
district_summary_df["Average Math Score"] = district_summary_df["Average M
district_summary_df["Average Reading Score"] = district_summary_df["Averag
district_summary_df["% Passing Math"] = district_summary_df["% Passing Mat
district_summary_df["% Passing Reading"] = district_summary_df["% Passing 
district_summary_df["% Overall Passing"] = district_summary_df["% Overall 

# Display the data frame
district_summary_df
```

Out[13]:

	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	78.9	81.9	74.8	85.7	64.9

School Summary

In [14]:

```
# Determine the School Type
per_school_types = school_data_df.set_index(["school_name"])["type"]

# Calculate the total student count.
per_school_counts = school_data_complete_df["school_name"].value_counts()

# Calculate the total school budget and per capita spending
per_school_budget = school_data_complete_df.groupby(["school_name"]).mean(
# Calculate the per capita spending.
per_school_capita = per_school_budget / per_school_counts

# Calculate the average test scores.
per_school_math = school_data_complete_df.groupby(["school_name"]).mean()[
per_school_reading = school_data_complete_df.groupby(["school_name"]).mean

# Calculate the passing scores by creating a filtered DataFrame.
per_school_passing_math = school_data_complete_df[(school_data_complete_df
per_school_passing_reading = school_data_complete_df[(school_data_complete_

# Calculate the number of students passing math and passing reading by sch
per_school_passing_math = per_school_passing_math.groupby(["school_name"])
per_school_passing_reading = per_school_passing_reading.groupby(["school_n

# Calculate the percentage of passing math and reading scores per school.
per_school_passing_math = per_school_passing_math / per_school_counts * 10
per_school_passing_reading = per_school_passing_reading / per_school_count

# Calculate the students who passed both reading and math.
per_passing_math_reading = school_data_complete_df[(school_data_complete_d
                                                   & (school_data_complete_df[

# Calculate the number of students passing math and passing reading by sch
per_passing_math_reading = per_passing_math_reading.groupby(["school_name"

# Calculate the percentage of passing math and reading scores per school.
per_overall_passing_percentage = per_passing_math_reading / per_school_cou
```

In [15]:

```
# Create the DataFrame
per_school_summary_df = pd.DataFrame({
    "School Type": per_school_types,
    "Total Students": per_school_counts,
    "Total School Budget": per_school_budget,
    "Per Student Budget": per_school_capita,
    "Average Math Score": per_school_math,
    "Average Reading Score": per_school_reading,
    "% Passing Math": per_school_passing_math,
    "% Passing Reading": per_school_passing_reading,
    "% Overall Passing": per_overall_passing_percentage})


per_school_summary_df
```

Out[15]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passin Mat
Bailey High School	District	4976	3124928.0	628.0	77.048432	81.033963	66.68006
Cabrera High School	Charter	1858	1081356.0	582.0	83.061895	83.975780	94.13347
Figueroa High School	District	2949	1884411.0	639.0	76.711767	81.158020	65.98847
Ford High School	District	2739	1763916.0	644.0	77.102592	80.746258	68.30960
Griffin High School	Charter	1468	917500.0	625.0	83.351499	83.816757	93.39237
Hernandez High School	District	4635	3022020.0	652.0	77.289752	80.934412	66.75296
Holden High School	Charter	427	248087.0	581.0	83.803279	83.814988	92.50585
Huang High School	District	2917	1910635.0	655.0	76.629414	81.182722	65.68392
Johnson High School	District	4761	3094650.0	650.0	77.072464	80.966394	66.05755

Pena High School	Charter	962	585858.0	609.0	83.839917	84.044699	94.59459
Rodriguez High School	District	3999	2547363.0	637.0	76.842711	80.744686	66.36659
Shelton High School	Charter	1761	1056600.0	600.0	83.359455	83.725724	93.86712
Thomas High School	Charter	1635	1043130.0	638.0	83.350937	83.896082	66.91131
Wilson High School	Charter	2283	1319574.0	578.0	83.274201	83.989488	93.86771
Wright High School	Charter	1800	1049400.0	583.0	83.682222	83.955000	93.33333

In [16]:

```
# Format the Total School Budget and the Per Student Budget
per_school_summary_df["Total School Budget"] = per_school_summary_df["Tota
per_school_summary_df["Per Student Budget"] = per_school_summary_df["Per S

# Display the data frame
per_school_summary_df
```

Out[16]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Pas
Bailey High School	District	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.68
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.13
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.98
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.30
Griffin High	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.39

School							
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.75
Holden High School	Charter	427	$248,087.00	$581.00	83.803279	83.814988	92.50
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.68
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.05
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.59
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.36
Shelton High School	Charter	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.86
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.350937	83.896082	66.91
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.86
Wright High School	Charter	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.33

In [17]:
```
# Step 5.  Get the number of 10th-12th graders from Thomas High School (TH
# count number of students from THS that are NOT in 9th grade
THS_not_ninth_count = school_data_complete_df["student_name"].loc[(school_
THS_not_ninth_count
```

Out[17]:
```
1174
```

In [18]:
```
# Step 6. Get all the students passing math from THS
THS_passing_math = school_data_complete_df.loc[(school_data_complete_df["m
THS_passing_math.head()
```

Out[18]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_
37535	37535	Norma Mata	F	10th	Thomas High School	76.0	
37536	37536	Cody Miller	M	11th	Thomas High School	84.0	
37541	37541	Eric Stevens	M	10th	Thomas High School	80.0	
37542	37542	Elizabeth Bennett	F	11th	Thomas High School	91.0	
37544	37544	Jacqueline Harris	F	10th	Thomas High School	71.0	

In [19]:

```
# Step 7. Get all the students passing reading from THS

THS_passing_reading = school_data_complete_df.loc[(school_data_complete_df
THS_passing_reading.head()
```

Out[19]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_
37535	37535	Norma Mata	F	10th	Thomas High School	76.0	
37536	37536	Cody Miller	M	11th	Thomas High School	84.0	
37541	37541	Eric Stevens	M	10th	Thomas High School	80.0	
37542	37542	Elizabeth Bennett	F	11th	Thomas High School	91.0	
37544	37544	Jacqueline Harris	F	10th	Thomas High School	71.0	

In [20]:

```
# Step 8. Get all the students passing math and reading from THS

THS_overall_pass = school_data_complete_df.loc[(school_data_complete_df["r
# test = school_data_complete_df.loc[(school_data_complete_df["school_name
# test
# test indicates thats school_data_complete_df already does not include th

THS_overall_pass
```

Out[20]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_
37535	37535	Norma Mata	F	10th	Thomas High School	76.0	
37536	37536	Cody Miller	M	11th	Thomas High School	84.0	
37541	37541	Eric Stevens	M	10th	Thomas High School	80.0	
37542	37542	Elizabeth Bennett	F	11th	Thomas High School	91.0	
37544	37544	Jacqueline Harris	F	10th	Thomas High School	71.0	
...	...	...	...	...	...	...	
39163	39163	John Reese	M	11th	Thomas High School	90.0	
39165	39165	Donna Howard	F	12th	Thomas High School	99.0	
39166	39166	Dawn Bell	F	10th	Thomas High School	95.0	
39168	39168	Desiree Kidd	F	10th	Thomas High School	99.0	
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95.0	

1064 rows × 11 columns

In [21]:

```
# Step 9. Calculate the percentage of 10th-12th grade students passing mat
# since THS_overall_pass only includes 10th to 12th grade students anyways

# All passing math in THS_passing_math is already 10-12 because per_school

THS_math_percent_pass_10to12 = (THS_passing_math["Student ID"].count()/THS
THS_math_percent_pass_10to12
```

Out[21]:

```
93.18568994889267
```

In [22]:
```
# Step 10. Calculate the percentage of 10th-12th grade students passing re
# All passing math in THS_passing_reading is already 10-12 because per_sch

THS_reading_percent_pass_10to12 = (THS_passing_reading["Student ID"].count
THS_reading_percent_pass_10to12
```

Out[22]:
```
97.01873935264055
```

In [23]:
```
# Step 11. Calculate the overall passing percentage of 10th-12th grade fro

THS_overall_percent_pass_10to12 = (THS_overall_pass["Student ID"].count()/
THS_overall_percent_pass_10to12
```

Out[23]:
```
90.63032367972743
```

In [24]:
```
# Step 12. Replace the passing math percent for Thomas High School in the 

per_school_summary_df.at["Thomas High School", "% Passing Math"] = THS_mat
```

In [25]:
```
# Step 13. Replace the passing reading percentage for Thomas High School i
per_school_summary_df.at["Thomas High School", "% Passing Reading"] = THS_
```

In [26]:
```
# Step 14. Replace the overall passing percentage for Thomas High School i
per_school_summary_df.at["Thomas High School", "% Overall Passing"] = THS_
```

In [27]:
```
per_school_summary_df
```

Out[27]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Pas
Bailey High School	District	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.68
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.13
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.98
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.30

Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.39
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.75
Holden High School	Charter	427	$248,087.00	$581.00	83.803279	83.814988	92.50
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.68
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.05
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.59
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.36
Shelton High School	Charter	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.86
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.350937	83.896082	93.18
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.86
Wright High School	Charter	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.33

High and Low Performing Schools

In [28]:

```
# Sort and show top five schools.

per_school_summary_df.sort_values(["% Overall Passing"], ascending = False
```

Out[28]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passin Mat
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.13347
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.350937	83.896082	93.18569
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.39237
Wilson High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.86771
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.59459

In [29]:

```
# Sort and show bottom five schools.

per_school_summary_df.sort_values(["% Overall Passing"], ascending = True)
```

Out[29]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Pass M
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057

Math and Reading Scores by Grade

In [30]:
```
# Create a Series of scores by grade levels using conditionals.

ninth = school_data_complete_df[(school_data_complete_df)["grade"] == "9th
tenth = school_data_complete_df[(school_data_complete_df)["grade"] == "10t
eleventh = school_data_complete_df[(school_data_complete_df)["grade"] == "
twelfth = school_data_complete_df[(school_data_complete_df)["grade"] == "1

# Group each school Series by the school name for the average math score.
ninth_avg_school_math = ninth.groupby(["school_name"]).mean()["math_score"
tenth_avg_school_math = tenth.groupby(["school_name"]).mean()["math_score"
eleventh_avg_school_math = eleventh.groupby(["school_name"]).mean()["math_
twelfth_avg_school_math = twelfth.groupby(["school_name"]).mean()["math_sc

# Group each school Series by the school name for the average reading scor
ninth_avg_school_read = ninth.groupby(["school_name"]).mean()["reading_sco
tenth_avg_school_read = tenth.groupby(["school_name"]).mean()["reading_sco
eleventh_avg_school_read = eleventh.groupby(["school_name"]).mean()["readi
twelfth_avg_school_read = twelfth.groupby(["school_name"]).mean()["reading
```

In [31]:
```
# Combine each Series for average math scores by school into single data f
math_by_grade_school_index = pd.DataFrame({"9th":ninth_avg_school_math, "1
math_by_grade_school_index
```

Out[31]:

school_name	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	NaN	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 [32]:

```
# Combine each Series for average reading scores by school into single dat
reading_by_grade_school_index = pd.DataFrame({"9th": ninth_avg_school_read
reading_by_grade_school_index
```

Out[32]:

school_name	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	NaN	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 [33]:

```
# Format each grade column.

math_by_grade_school_index["9th"] = math_by_grade_school_index["9th"].map(
reading_by_grade_school_index["9th"] = reading_by_grade_school_index["9th"

math_by_grade_school_index["10th"] = math_by_grade_school_index["10th"].ma
reading_by_grade_school_index["10th"] = reading_by_grade_school_index["10t

math_by_grade_school_index["11th"] = math_by_grade_school_index["11th"].ma
reading_by_grade_school_index["11th"] = reading_by_grade_school_index["11t

math_by_grade_school_index["12th"] = math_by_grade_school_index["12th"].ma
reading_by_grade_school_index["12th"] = reading_by_grade_school_index["12t

# Correct order

math_by_grade_school_index = math_by_grade_school_index[["9th","10th","11t
reading_by_grade_school_index = reading_by_grade_school_index[["9th","10th
```

In [34]:
```
# Remove the index.
math_by_grade_school_index.index.name = None

# Display the data frame
math_by_grade_school_index.head()
```

Out[34]:

	9th	10th	11th	12th
Bailey High School	77.1	77.0	77.5	76.5
Cabrera High School	83.1	83.2	82.8	83.3
Figueroa High School	76.4	76.5	76.9	77.2
Ford High School	77.4	77.7	76.9	76.2
Griffin High School	82.0	84.2	83.8	83.4

In [35]:
```
## Remove the index.
reading_by_grade_school_index.index.name = None

# Display the data frame
reading_by_grade_school_index.head()
```

Out[35]:

	9th	10th	11th	12th
Bailey High School	81.3	80.9	80.9	80.9
Cabrera High School	83.7	84.3	83.8	84.3
Figueroa High School	81.2	81.4	80.6	81.4
Ford High School	80.6	81.3	80.4	80.7
Griffin High School	83.4	83.7	84.3	84.0

Scores by School Spending

In [36]:
```
# Establish the spending bins and group names.
spending_bins = [0, 585, 630, 645, 675]
group_names = ["<585$", "$586-630", "$631-645", "$646-675"]


# Categorize spending based on the bins.
per_school_summary_df["Spending Range (Per Student)"] = pd.cut(per_school_
per_school_summary_df
```

Out[36]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Pas M
Bailey High School	District	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.68
Cabrera High School	Charter	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.13
Figueroa High School	District	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.98
Ford High School	District	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.30
Griffin High School	Charter	1468	$917,500.00	$625.00	83.351499	83.816757	93.39
Hernandez High School	District	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.75
Holden High School	Charter	427	$248,087.00	$581.00	83.803279	83.814988	92.50
Huang High School	District	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.68
Johnson High School	District	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.05
Pena High School	Charter	962	$585,858.00	$609.00	83.839917	84.044699	94.59
Rodriguez High School	District	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.36
Shelton High School	Charter	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.86
Thomas High School	Charter	1635	$1,043,130.00	$638.00	83.350937	83.896082	93.18
Wilson							

High School	Charter	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.86
Wright High School	Charter	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.333

In [37]:
```
# Calculate averages for the desired columns.
math_bin = per_school_summary_df.groupby(["Spending Range (Per Student)"])
reading_bin = per_school_summary_df.groupby(["Spending Range (Per Student)
math_pass_bin = per_school_summary_df.groupby(["Spending Range (Per Studen
reading_pass_bin = per_school_summary_df.groupby(["Spending Range (Per Stu
overall_pass_bin = per_school_summary_df.groupby(["Spending Range (Per Stu
```

In [38]:
```
# Create the DataFrame
spending_summary = pd.DataFrame({
    "Average Math Score": math_bin,
    "Average Reading Score" : reading_bin,
    "% Passing Math": math_pass_bin,
    "% Passing Reading" : reading_pass_bin,
    "% Overall Pass" : overall_pass_bin
})
```

In [39]:
```
# Format the DataFrame
spending_summary["Average Math Score"] = spending_summary["Average Math Sc
spending_summary["Average Reading Score"] = spending_summary["Average Read
spending_summary["% Passing Math"] = spending_summary["% Passing Math"].ma
spending_summary["% Passing Reading"] = spending_summary["% Passing Readin
spending_summary["% Overall Pass"] = spending_summary["% Overall Pass"].ma

spending_summary
```

Out[39]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Pass
Spending Range (Per Student)					
<585$	83%	84%	93%	97%	90%
$586-630	82%	83%	87%	93%	81%
$631-645	79%	82%	73%	84%	63%
$646-675	77%	81%	66%	81%	54%

Scores by School Size

In [40]:
```
# Establish the bins.
size_bins = [0, 999, 1999, 5000]
group_names = ["Small (<1000)", "Medium (1000-1999)", "Large (2000-5000)"]

# Categorize spending based on the bins.
per_school_summary_df["School Size"] = pd.cut(per_school_summary_df["Total
```

In [41]:
```
# Calculate averages for the desired columns.
size_math_bin = per_school_summary_df.groupby(["School Size"]).mean()["Ave
size_reading_bin = per_school_summary_df.groupby(["School Size"]).mean()["
size_math_pass_bin = per_school_summary_df.groupby(["School Size"]).mean()
size_reading_pass_bin = per_school_summary_df.groupby(["School Size"]).mea
size_overall_pass_bin = per_school_summary_df.groupby(["School Size"]).mea
```

In [42]:
```
# Assemble into DataFrame.
size_summary = pd.DataFrame({
    "Average Math Score": size_math_bin,
    "Average Reading Score" : size_reading_bin,
    "% Passing Math": size_math_pass_bin,
    "% Passing Reading" : size_reading_pass_bin,
    "% Overall Pass" : size_overall_pass_bin
})
```

In [43]:
```
# Format the DataFrame
size_summary["Average Math Score"] = size_summary["Average Math Score"].ma
size_summary["Average Reading Score"] = size_summary["Average Reading Scor
size_summary["% Passing Math"] = size_summary["% Passing Math"].map("{:.0f
size_summary["% Passing Reading"] = size_summary["% Passing Reading"].map(
size_summary["% Overall Pass"] = size_summary["% Overall Pass"].map("{:.0f

size_summary
```

Out[43]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Pass
Small (<1000)	83.8%	83.9%	94%	96%	90%
Medium (1000-1999)	83.4%	83.9%	94%	97%	91%
Large (2000-5000)	77.7%	81.3%	70%	83%	58%

Scores by School Type

In [44]:

```
# Calculate averages for the desired columns.
type_math_group = per_school_summary_df.groupby(["School Type"]).mean()["A
type_reading_group = per_school_summary_df.groupby(["School Type"]).mean()
type_math_pass_group = per_school_summary_df.groupby(["School Type"]).mean
type_reading_pass_group = per_school_summary_df.groupby(["School Type"]).m
type_overall_pass_group = per_school_summary_df.groupby(["School Type"]).m
```

In [45]:

```
# Assemble into DataFrame.
type_summary = pd.DataFrame({
    "Average Math Score": type_math_group,
    "Average Reading Score" : type_reading_group,
    "% Passing Math": type_math_pass_group,
    "% Passing Reading" : type_reading_pass_group,
    "% Overall Pass" : type_overall_pass_group
})
```

In [46]:

```
# # Format the DataFrame
type_summary["Average Math Score"] = type_summary["Average Math Score"].ma
type_summary["Average Reading Score"] = type_summary["Average Reading Scor
type_summary["% Passing Math"] = type_summary["% Passing Math"].map("{:.0f
type_summary["% Passing Reading"] = type_summary["% Passing Reading"].map(
type_summary["% Overall Pass"] = type_summary["% Overall Pass"].map("{:.0f

type_summary
```

Out[46]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Pass
School Type					
Charter	83.5%	83.9%	94%	97%	90%
District	77.0%	81.0%	67%	81%	54%

In [ ]:

In [ ]:

In [ ]: