

```
In [126]: # Dependencies and Setup
import pandas as pd

# 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", " D", " DS", " DVM", " PhD"]

# Iterate through the words in the "prefixes_suffixes" list and replace them with an empty space, "".
for word in prefixes_suffixes:
    student_data_df["student_name"] = student_data_df["student_name"].str.replace(word, "")

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

Out[126]:

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

```
In [128]: # Step 2. Use the loc method on the student_data_df to select all the re
ading scores from the 9th grade
#at Thomas High School and replace them with NaN.
student_data_df.loc[(student_data_df["school_name"] == "Thomas High Scho
ol")
                    & (student_data_df["grade"] == "9th"),["reading_scor
e"]] = np.nan
```

```
In [129]: # Step 3. Refactor the code in Step 2 to replace the math scores with N
aN.
student_data_df.loc[(student_data_df["school_name"] == "Thomas High Scho
ol")
                    & (student_data_df["grade"] == "9th"),["math_score"
]] = np.nan
```

```
In [130]: # Step 4. Check the student data for NaN's.
student_data_df.tail(10)
```

Out[130]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score
<b>39160</b>	39160	Katie Weaver	F	11th	Thomas High School	89.0	86.0
<b>39161</b>	39161	April Reyes	F	10th	Thomas High School	70.0	84.0
<b>39162</b>	39162	Derek Weeks	M	12th	Thomas High School	94.0	77.0
<b>39163</b>	39163	John Reese	M	11th	Thomas High School	90.0	75.0
<b>39164</b>	39164	Joseph Anthony	M	9th	Thomas High School	NaN	NaN
<b>39165</b>	39165	Donna Howard	F	12th	Thomas High School	99.0	90.0
<b>39166</b>	39166	Dawn Bell	F	10th	Thomas High School	95.0	70.0
<b>39167</b>	39167	Rebecca Tanner	F	9th	Thomas High School	NaN	NaN
<b>39168</b>	39168	Desiree Kidd	F	10th	Thomas High School	99.0	90.0
<b>39169</b>	39169	Carolyn Jackson	F	11th	Thomas High School	95.0	75.0

## Deliverable 2 : Repeat the school district analysis

## District Summary

```
In [131]: # Combine the data into a single dataset
school_data_complete_df = pd.merge(student_data_df, school_data_df, how=
"left", on=["school_name", "school_name"])
school_data_complete_df.head()
```

Out[131]:

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

```
In [132]: # 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 [133]: # 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 [134]: # Step 1. Get the number of students that are in ninth grade at Thomas High School.
# These students have no grades.
ninth_graders_at_THS = school_data_complete_df.loc[(school_data_complete_df["school_name"] == "Thomas High School")
                                                    & (school_data_complete_df["grade"] == "9th")
                                                    ], "Student ID"].count()
print(ninth_graders_at_THS)

# 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 student count.
new_total_student_count = student_count - ninth_graders_at_THS
print(new_total_student_count)

461
38709
```

```
In [135]: # Calculate the passing rates using the "clean_student_data".
passing_math_count = school_data_complete_df[(school_data_complete_df["math_score"] >= 70)].count()["student_name"]
passing_reading_count = school_data_complete_df[(school_data_complete_df["reading_score"] >= 70)].count()["student_name"]
```

```
In [136]: # Step 3. Calculate the passing percentages with the new total student count.
passing_math_percentage = passing_math_count / new_total_student_count * 100
print(passing_math_percentage)

passing_reading_percentage = passing_reading_count / new_total_student_count * 100
print(passing_reading_percentage)

74.76039164018704
85.6596657108166
```

```
In [137]: # Calculate the students who passed both reading and math.
passing_math_reading = school_data_complete_df[(school_data_complete_df[
"math_score"] >= 70)
                                                & (school_data_complete_d
f["reading_score"] >= 70)]

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

# Step 4. Calculate the overall passing percentage with new total student
count.
overall_passing_percentage = overall_passing_math_reading_count / new_to
tal_student_count * 100
overall_passing_percentage
```

```
Out[137]: 64.85571830840374
```

```
In [138]: # 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 Students"].map("{:,}".format)
# Format the "Total Budget" to have the comma for a thousands separator, a decimal separator and a "$".
district_summary_df["Total Budget"] = district_summary_df["Total Budget"].map("${:,.2f}".format)
# Format the columns.
district_summary_df["Average Math Score"] = district_summary_df["Average Math Score"].map("{:.1f}".format)
district_summary_df["Average Reading Score"] = district_summary_df["Average Reading Score"].map("{:.1f}".format)
district_summary_df["% Passing Math"] = district_summary_df["% Passing Math"].map("{:.1f}".format)
district_summary_df["% Passing Reading"] = district_summary_df["% Passing Reading"].map("{:.1f}".format)
district_summary_df["% Overall Passing"] = district_summary_df["% Overall Passing"].map("{:.1f}".format)

# Display the data frame
district_summary_df
```

Out[138]:

	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 [139]: # 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
()["budget"]
# 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
()["math_score"]
per_school_reading = school_data_complete_df.groupby(["school_name"]).mean
()["reading_score"]

# Calculate the passing scores by creating a filtered DataFrame.
per_school_passing_math = school_data_complete_df[(school_data_complete_
df["math_score"] >= 70)]
per_school_passing_reading = school_data_complete_df[(school_data_comple
te_df["reading_score"] >= 70)]

# Calculate the number of students passing math and passing reading by s
chool.
per_school_passing_math = per_school_passing_math.groupby(["school_name"
]).count()["student_name"]
per_school_passing_reading = per_school_passing_reading.groupby(["school
_name"]).count()["student_name"]

# Calculate the percentage of passing math and reading scores per schoo
l.
per_school_passing_math = per_school_passing_math / per_school_counts *
100
per_school_passing_reading = per_school_passing_reading / per_school_cou
nts * 100

# Calculate the students who passed both reading and math.
per_passing_math_reading = school_data_complete_df[(school_data_complete
_df["reading_score"] >= 70)
                                                    & (school_data_complete_d
f["math_score"] >= 70)]

# Calculate the number of students passing math and passing reading by s
chool.
per_passing_math_reading = per_passing_math_reading.groupby(["school_nam
e"]).count()["student_name"]

# Calculate the percentage of passing math and reading scores per schoo
l.
per_overall_passing_percentage = per_passing_math_reading / per_school_c
ounts * 100

```

```
In [140]: # 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.head()
```



```
In [141]: # Format the Total School Budget and the Per Student Budget
per_school_summary_df["Total School Budget"] = per_school_summary_df["To
tal School Budget"].map("${:, .2f}".format)
per_school_summary_df["Per Student Budget"] = per_school_summary_df["Per
Student Budget"].map("${:, .2f}".format)

# Display the data frame
per_school_summary_df
```

Out[141]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
<b>Bailey High School</b>	District	4976	\$3,124,928.00	\$628.00	77.048432	81.033963	66.680064	81.933280
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.975780	94.133477	97.039828
<b>Figueroa High School</b>	District	2949	\$1,884,411.00	\$639.00	76.711767	81.158020	65.988471	80.739234
<b>Ford High School</b>	District	2739	\$1,763,916.00	\$644.00	77.102592	80.746258	68.309602	79.299014
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.816757	93.392371	97.138965
<b>Hernandez High School</b>	District	4635	\$3,022,020.00	\$652.00	77.289752	80.934412	66.752967	80.862999
<b>Holden High School</b>	Charter	427	\$248,087.00	\$581.00	83.803279	83.814988	92.505855	96.252927
<b>Huang High School</b>	District	2917	\$1,910,635.00	\$655.00	76.629414	81.182722	65.683922	81.316421
<b>Johnson High School</b>	District	4761	\$3,094,650.00	\$650.00	77.072464	80.966394	66.057551	81.222432
<b>Pena High School</b>	Charter	962	\$585,858.00	\$609.00	83.839917	84.044699	94.594595	95.945946
<b>Rodriguez High School</b>	District	3999	\$2,547,363.00	\$637.00	76.842711	80.744686	66.366592	80.220055
<b>Shelton High School</b>	Charter	1761	\$1,056,600.00	\$600.00	83.359455	83.725724	93.867121	95.854628
<b>Thomas High School</b>	Charter	1635	\$1,043,130.00	\$638.00	83.350937	83.896082	66.911315	69.663609
<b>Wilson High School</b>	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.989488	93.867718	96.539641
<b>Wright High School</b>	Charter	1800	\$1,049,400.00	\$583.00	83.682222	83.955000	93.333333	96.611111

```
In [142]: # Step 5. Get the number of 10th-12th graders from Thomas High School (THS).
THS_10_11_12 = school_data_complete_df.loc[(school_data_complete_df["school_name"] == "Thomas High School")
                                             & (school_data_complete_df["grade"] != "9th")
                                             ], "student_name"].count()

THS_10_11_12
```

Out[142]: 1174

```
In [143]: # Step 6. Get all the students passing math from THS
THS_passing_math = school_data_complete_df.loc[(school_data_complete_df["school_name"] == "Thomas High School")
                                                & (school_data_complete_df["math_score"] >= 70), "student_name"].count()
THS_passing_math
```

Out[143]: 1094

```
In [144]: # Step 7. Get all the students passing reading from THS
THS_passing_reading = school_data_complete_df.loc[(school_data_complete_df["school_name"] == "Thomas High School")
                                                  & (school_data_complete_df["reading_score"] >= 70), "student_name"].count()
THS_passing_reading
```

Out[144]: 1139

```
In [145]: # Step 8. Get all the students passing math and reading from THS
THS_passing_math_reading = school_data_complete_df.loc[(school_data_complete_df["school_name"] == "Thomas High School")
                                                        & (school_data_complete_df["reading_score"] >= 70)
                                                        & (school_data_complete_df["math_score"] >= 70), "student_name"].count()
THS_passing_math_reading
```

Out[145]: 1064

```
In [146]: # Step 9. Calculate the percentage of 10th-12th grade students passing math from Thomas High School.
THS_passing_math_percentage = THS_passing_math / THS_10_11_12 * 100
THS_passing_math_percentage
```

Out[146]: 93.18568994889267

```
In [147]: # Step 10. Calculate the percentage of 10th-12th grade students passing reading from Thomas High School.
THS_passing_reading_percentage = THS_passing_reading / THS_10_11_12 * 100
THS_passing_reading_percentage
```

Out[147]: 97.01873935264055

```
In [148]: # Step 11. Calculate the overall passing percentage of 10th-12th grade from Thomas High School.  
THS_passing_math_reading_percentage = THS_passing_math_reading / THS_10_11_12 * 100  
THS_passing_math_reading_percentage
```

```
Out[148]: 90.63032367972743
```

```
In [149]: # Step 12. Replace the passing math percent for Thomas High School in the per_school_summary_df.  
per_school_summary_df.loc["Thomas High School",["% Passing Math"]] = THS_passing_math_percentage
```

```
In [150]: # Step 13. Replace the passing reading percentage for Thomas High School in the per_school_summary_df.  
per_school_summary_df.loc["Thomas High School",["% Passing Reading"]] = THS_passing_reading_percentage
```

```
In [151]: # Step 14. Replace the overall passing percentage for Thomas High School in the per_school_summary_df.  
per_school_summary_df.loc["Thomas High School",["% Overall Passing"]] = THS_passing_math_reading_percentage
```

In [152]: per\_school\_summary\_df

Out[152]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading
<b>Bailey High School</b>	District	4976	\$3,124,928.00	\$628.00	77.048432	81.033963	66.680064	81.933280
<b>Cabrera High School</b>	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.975780	94.133477	97.039828
<b>Figueroa High School</b>	District	2949	\$1,884,411.00	\$639.00	76.711767	81.158020	65.988471	80.739234
<b>Ford High School</b>	District	2739	\$1,763,916.00	\$644.00	77.102592	80.746258	68.309602	79.299014
<b>Griffin High School</b>	Charter	1468	\$917,500.00	\$625.00	83.351499	83.816757	93.392371	97.138965
<b>Hernandez High School</b>	District	4635	\$3,022,020.00	\$652.00	77.289752	80.934412	66.752967	80.862999
<b>Holden High School</b>	Charter	427	\$248,087.00	\$581.00	83.803279	83.814988	92.505855	96.252927
<b>Huang High School</b>	District	2917	\$1,910,635.00	\$655.00	76.629414	81.182722	65.683922	81.316421
<b>Johnson High School</b>	District	4761	\$3,094,650.00	\$650.00	77.072464	80.966394	66.057551	81.222432
<b>Pena High School</b>	Charter	962	\$585,858.00	\$609.00	83.839917	84.044699	94.594595	95.945946
<b>Rodriguez High School</b>	District	3999	\$2,547,363.00	\$637.00	76.842711	80.744686	66.366592	80.220055
<b>Shelton High School</b>	Charter	1761	\$1,056,600.00	\$600.00	83.359455	83.725724	93.867121	95.854628
<b>Thomas High School</b>	Charter	1635	\$1,043,130.00	\$638.00	83.350937	83.896082	93.185690	97.018739
<b>Wilson High School</b>	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.989488	93.867718	96.539641
<b>Wright High School</b>	Charter	1800	\$1,049,400.00	\$583.00	83.682222	83.955000	93.333333	96.611111

# High and Low Performing Schools

```
In [153]: # Sort and show top five schools.
top_schools = per_school_summary_df.sort_values(["% Overall Passing"],as
cending = False)
top_schools.head(5)
```

Out[153]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	
Cabrera High School	Charter	1858	\$1,081,356.00	\$582.00	83.061895	83.975780	94.133477	97.039828	9
Thomas High School	Charter	1635	\$1,043,130.00	\$638.00	83.350937	83.896082	93.185690	97.018739	9
Griffin High School	Charter	1468	\$917,500.00	\$625.00	83.351499	83.816757	93.392371	97.138965	9
Wilson High School	Charter	2283	\$1,319,574.00	\$578.00	83.274201	83.989488	93.867718	96.539641	9
Pena High School	Charter	962	\$585,858.00	\$609.00	83.839917	84.044699	94.594595	95.945946	9

```
In [154]: # Sort and show bottom five schools.
bottom_schools = per_school_summary_df.sort_values(["% Overall Passing"
],ascending = True)
bottom_schools.head(5)
```

Out[154]:

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

## Math and Reading Scores by Grade

```
In [155]: # Create a Series of scores by grade levels using conditionals.
ninth_graders = school_data_complete_df[(school_data_complete_df["grade"] == "9th")]
tenth_graders = school_data_complete_df[(school_data_complete_df["grade"] == "10th")]
eleventh_graders = school_data_complete_df[(school_data_complete_df["grade"] == "11th")]
twelfth_graders = school_data_complete_df[(school_data_complete_df["grade"] == "12th")]

# Group each school Series by the school name for the average math score.
ninth_grade_math_scores = ninth_graders.groupby(["school_name"]).mean()["math_score"]
tenth_grade_math_scores = tenth_graders.groupby(["school_name"]).mean()["math_score"]
eleventh_grade_math_scores = eleventh_graders.groupby(["school_name"]).mean()["math_score"]
twelfth_grade_math_scores = twelfth_graders.groupby(["school_name"]).mean()["math_score"]

# Group each school Series by the school name for the average reading score.
ninth_grade_reading_scores = ninth_graders.groupby(["school_name"]).mean()["reading_score"]
tenth_grade_reading_scores = tenth_graders.groupby(["school_name"]).mean()["reading_score"]
eleventh_grade_reading_scores = eleventh_graders.groupby(["school_name"]).mean()["reading_score"]
twelfth_grade_reading_scores = twelfth_graders.groupby(["school_name"]).mean()["reading_score"]
```

```
In [156]: # Combine each Series for average math scores by school into single data frame.
math_scores_by_grade = pd.DataFrame({
    "9th": ninth_grade_math_scores,
    "10th": tenth_grade_math_scores,
    "11th": eleventh_grade_math_scores,
    "12th": twelfth_grade_math_scores})
```

```
In [157]: # Combine each Series for average reading scores by school into single data frame.
reading_scores_by_grade = pd.DataFrame({
    "9th": ninth_grade_reading_scores,
    "10th": tenth_grade_reading_scores,
    "11th": eleventh_grade_reading_scores,
    "12th": twelfth_grade_reading_scores})
```



```
In [158]: # Format each grade column.
math_scores_by_grade["9th"] = math_scores_by_grade["9th"].map("{:.1f}".format)
math_scores_by_grade["10th"] = math_scores_by_grade["10th"].map("{:.1f}".format)
math_scores_by_grade["11th"] = math_scores_by_grade["11th"].map("{:.1f}".format)
math_scores_by_grade["12th"] = math_scores_by_grade["12th"].map("{:.1f}".format)

reading_scores_by_grade["9th"] = reading_scores_by_grade["9th"].map("{:.1f}".format)
reading_scores_by_grade["10th"] = reading_scores_by_grade["10th"].map("{:.1f}".format)
reading_scores_by_grade["11th"] = reading_scores_by_grade["11th"].map("{:.1f}".format)
reading_scores_by_grade["12th"] = reading_scores_by_grade["12th"].map("{:.1f}".format)
```

```
In [159]: # Remove the index.
math_scores_by_grade.index.name = None

# Display the data frame
math_scores_by_grade
```

Out[159]:

	9th	10th	11th	12th
<b>Bailey High School</b>	77.1	77.0	77.5	76.5
<b>Cabrera High School</b>	83.1	83.2	82.8	83.3
<b>Figueroa High School</b>	76.4	76.5	76.9	77.2
<b>Ford High School</b>	77.4	77.7	76.9	76.2
<b>Griffin High School</b>	82.0	84.2	83.8	83.4
<b>Hernandez High School</b>	77.4	77.3	77.1	77.2
<b>Holden High School</b>	83.8	83.4	85.0	82.9
<b>Huang High School</b>	77.0	75.9	76.4	77.2
<b>Johnson High School</b>	77.2	76.7	77.5	76.9
<b>Pena High School</b>	83.6	83.4	84.3	84.1
<b>Rodriguez High School</b>	76.9	76.6	76.4	77.7
<b>Shelton High School</b>	83.4	82.9	83.4	83.8
<b>Thomas High School</b>	nan	83.1	83.5	83.5
<b>Wilson High School</b>	83.1	83.7	83.2	83.0
<b>Wright High School</b>	83.3	84.0	83.8	83.6

```
In [160]: ## Remove the index.
reading_scores_by_grade.index.name = None

# Display the data frame
reading_scores_by_grade
```

Out[160]:

	9th	10th	11th	12th
<b>Bailey High School</b>	81.3	80.9	80.9	80.9
<b>Cabrera High School</b>	83.7	84.3	83.8	84.3
<b>Figueroa High School</b>	81.2	81.4	80.6	81.4
<b>Ford High School</b>	80.6	81.3	80.4	80.7
<b>Griffin High School</b>	83.4	83.7	84.3	84.0
<b>Hernandez High School</b>	80.9	80.7	81.4	80.9
<b>Holden High School</b>	83.7	83.3	83.8	84.7
<b>Huang High School</b>	81.3	81.5	81.4	80.3
<b>Johnson High School</b>	81.3	80.8	80.6	81.2
<b>Pena High School</b>	83.8	83.6	84.3	84.6
<b>Rodriguez High School</b>	81.0	80.6	80.9	80.4
<b>Shelton High School</b>	84.1	83.4	84.4	82.8
<b>Thomas High School</b>	nan	84.3	83.6	83.8
<b>Wilson High School</b>	83.9	84.0	83.8	84.3
<b>Wright High School</b>	83.8	83.8	84.2	84.1

## Scores by School Spending

```
In [161]: per_school_capita.describe()
```

```
Out[161]: count      15.000000
mean       620.066667
std        28.544368
min        578.000000
25%       591.500000
50%       628.000000
75%       641.500000
max       655.000000
dtype: float64
```

```
In [162]: # Establish the spending bins and group names.
spending_bins = [0, 585, 630, 645, 675]
group_names = ["<$584", "$585-629", "$630-644", "$645-675"]
```

```
In [163]: # Categorize spending based on the bins.
per_school_summary_df["Spending Ranges (Per Student)"] = pd.cut(per_scho
ol_capita, spending_bins, labels = group_names)
```

```
In [164]: # Calculate averages for the desired columns.
spending_math_scores = per_school_summary_df.groupby(["Spending Ranges
(Per Student)"]).mean()["Average Math Score"]
spending_reading_scores = per_school_summary_df.groupby(["Spending Range
s (Per Student)"]).mean()["Average Reading Score"]
spending_passing_math = per_school_summary_df.groupby(["Spending Ranges
(Per Student)"]).mean()["% Passing Math"]
spending_passing_reading = per_school_summary_df.groupby(["Spending Rang
es (Per Student)"]).mean()["% Passing Reading"]
overall_passing_spending = per_school_summary_df.groupby(["Spending Rang
es (Per Student)"]).mean()["% Overall Passing"]
```

```
In [165]: # Create the DataFrame
spending_summary_df = pd.DataFrame({
    "Average Math Score" : spending_math_scores,
    "Average Reading Score": spending_reading_scores,
    "% Passing Math": spending_passing_math,
    "% Passing Reading": spending_passing_reading,
    "% Overall Passing": overall_passing_spending})
```

```
In [166]: # Format the DataFrame
spending_summary_df["Average Math Score"] = spending_summary_df["Average
Math Score"].map("{:.1f}".format)
spending_summary_df["Average Reading Score"] = spending_summary_df["Aver
age Reading Score"].map("{:.1f}".format)
spending_summary_df["% Passing Math"] = spending_summary_df["% Passing M
ath"].map("{:.0f}".format)
spending_summary_df["% Passing Reading"] = spending_summary_df["% Passin
g Reading"].map("{:.0f}".format)
spending_summary_df["% Overall Passing"] = spending_summary_df["% Overal
l Passing"].map("{:.0f}".format)

spending_summary_df
```

Out[166]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Spending Ranges (Per Student)					
<\$584	83.5	83.9	93	97	90
\$585-629	81.9	83.2	87	93	81
\$630-644	78.5	81.6	73	84	63
\$645-675	77.0	81.0	66	81	54

## Scores by School Size

```
In [167]: # Establish the bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

# Categorize spending based on the bins.
per_school_summary_df["School Size"] = pd.cut(per_school_summary_df["Total Students"], size_bins, labels = group_names)
```

```
In [168]: # Calculate averages for the desired columns.
size_math_scores = per_school_summary_df.groupby(["School Size"]).mean()["Average Math Score"]
size_reading_scores = per_school_summary_df.groupby(["School Size"]).mean()["Average Reading Score"]
size_passing_math = per_school_summary_df.groupby(["School Size"]).mean()["% Passing Math"]
size_passing_reading = per_school_summary_df.groupby(["School Size"]).mean()["% Passing Reading"]
size_overall_passing = per_school_summary_df.groupby(["School Size"]).mean()["% Overall Passing"]
```

```
In [169]: # Assemble into DataFrame.
size_summary_df = pd.DataFrame({
    "Average Math Score": size_math_scores,
    "Average Reading Score": size_reading_scores,
    "% Passing Math": size_passing_math,
    "% Passing Reading": size_passing_reading,
    "% Overall Passing": size_overall_passing})
```

```
In [170]: # Format the DataFrame
size_summary_df["Average Math Score"] = size_summary_df["Average Math Score"].map("{:.1f}".format)
size_summary_df["Average Reading Score"] = size_summary_df["Average Reading Score"].map("{:.1f}".format)
size_summary_df["% Passing Math"] = size_summary_df["% Passing Math"].map("{:.0f}".format)
size_summary_df["% Passing Reading"] = size_summary_df["% Passing Reading"].map("{:.0f}".format)
size_summary_df["% Overall Passing"] = size_summary_df["% Overall Passing"].map("{:.0f}".format)

size_summary_df
```

Out[170]:

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

## Scores by School Type

```
In [171]: # Calculate averages for the desired columns.
type_math_scores = per_school_summary_df.groupby(["School Type"]).mean()[
    "Average Math Score"]
type_reading_scores = per_school_summary_df.groupby(["School Type"]).mean(
    )["Average Reading Score"]
type_passing_math = per_school_summary_df.groupby(["School Type"]).mean(
    )["% Passing Math"]
type_passing_reading = per_school_summary_df.groupby(["School Type"]).mean(
    )["% Passing Reading"]
type_overall_passing = per_school_summary_df.groupby(["School Type"]).mean(
    )["% Overall Passing"]
```

```
In [172]: # Assemble into DataFrame.
type_summary_df = pd.DataFrame({
    "Average Math Score" : type_math_scores,
    "Average Reading Score": type_reading_scores,
    "% Passing Math": type_passing_math,
    "% Passing Reading": type_passing_reading,
    "% Overall Passing": type_overall_passing})
```

```
In [173]: # Format the DataFrame
type_summary_df["Average Math Score"] = type_summary_df["Average Math Score"].map(
    "{:.1f}".format)
type_summary_df["Average Reading Score"] = type_summary_df["Average Reading Score"].map(
    "{:.1f}".format)
type_summary_df["% Passing Math"] = type_summary_df["% Passing Math"].map(
    "{:.0f}".format)
type_summary_df["% Passing Reading"] = type_summary_df["% Passing Reading"].map(
    "{:.0f}".format)
type_summary_df["% Overall Passing"] = type_summary_df["% Overall Passing"].map(
    "{:.0f}".format)

type_summary_df
```

Out[173]:

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