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 = "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"])
school_data_complete
```

Out[1]:

	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
...	...	...	...	...	...	...	...	...	...	...	...
39165	39165	Donna Howard	F	12th	Thomas High School	99	90	14	Charter	1635	1043130
39166	39166	Dawn Bell	F	10th	Thomas High School	95	70	14	Charter	1635	1043130
39167	39167	Rebecca Tanner	F	9th	Thomas High School	73	84	14	Charter	1635	1043130
39168	39168	Desiree Kidd	F	10th	Thomas High School	99	90	14	Charter	1635	1043130
39169	39169	Carolyn Jackson	F	11th	Thomas High School	95	75	14	Charter	1635	1043130

39170 rows × 11 columns

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 [2]:

```
# Get unique school names and count of that by using len() function
total_schools = len(school_data_complete["School ID"].unique())

# Get total Number of students by applying count() function on student id
total_students = len(school_data_complete["Student ID"].unique())

# Total budget : Get the total budget by using sum() function on Budget
total_budget = school_data["budget"].sum()

# Get Avg Scores by using mean function
avg_math_score = school_data_complete["math_score"].mean()
avg_reading_score = school_data_complete["reading_score"].mean()

# Calculating percentage of students with a passing math score (70 or greater)
stu_math_passing_count = school_data_complete.loc[school_data_complete["math_score"]>=70,:]["Student ID"].count()
stu_math_pass_percent = (stu_math_passing_count / total_students) * 100

# Calculating percentage of students with a passing reading score (70 or greater)
stu_reading_passing_count = school_data_complete.loc[school_data_complete["reading_score"]>=70,:]["Student ID"].count()
stu_reading_pass_percent = (stu_reading_passing_count / total_students) * 100

# Calculating the percentage of students who passed math and reading (% Overall Passing)
overall_passing_count = school_data_complete.loc[(school_data_complete["math_score"]>=70) & (school_data_complete["reading_sco
overall_passing_percent = (overall_passing_count/total_students)*100

# Create a summary data frame with obtained values
district_summary_df = pd.DataFrame({"Total Schools" : [total_schools],
                                    "Total Students" : [total_students],
                                    "Total Budget" : [total_budget],
                                    "Average Math Score" : [avg_math_score],
                                    "Average Reading Score" : [avg_reading_score],
                                    "% Passing Math" : [stu_math_pass_percent],
                                    "% Passing Reading" : [stu_reading_pass_percent],
                                    "% Overall Passing" : [overall_passing_percent]
                                   })

# Formating of columns
district_summary_df["Total Students"] = district_summary_df["Total Students"].map("{:,}".format)
district_summary_df["Total Budget"] = district_summary_df["Total Budget"].map("${:,}".format)

# Display Data Frame
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	78.985371	81.87784	74.980853	85.805463	65.172326

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 [3]:

```
# Groupby Object by School Name
grouped_school = school_data_complete.groupby("school_name")

# Get the school type by school name from original DF
school_type = school_data.set_index("school_name")["type"]

# Get the total students on grouped by school
sch_total_student = grouped_school["Student ID"].count()

# Get the budget  by school name from original DF
school_budget = school_data.set_index("school_name")["budget"]

# Calculating per student budget
sch_per_stu_budget = school_budget / sch_total_student

# Calculating Average math & reading score
sch_avg_math_score = grouped_school["math_score"].mean()
sch_avg_read_score = grouped_school["reading_score"].mean()

# Calculating % passing math by school
math_pass_sch_count = school_data_complete.loc[school_data_complete["math_score"] >= 70].groupby("school_name")["Student ID"].
sch_math_pass_percent = math_pass_sch_count / sch_total_student * 100

# Calculating % passing reading by school
read_pass_sch_count = school_data_complete.loc[school_data_complete["reading_score"] >= 70].groupby("school_name")["Student ID
sch_read_pass_percent = read_pass_sch_count / sch_total_student * 100

# Calculating overall % passing
sch_overall_passing_count = school_data_complete.loc[(school_data_complete["math_score"]>=70) & (school_data_complete["reading_
sch_overall_pass_percent = sch_overall_passing_count / sch_total_student * 100

# Create a data frame with values obtained above
school_summary_df = pd.DataFrame({"School Type" : school_type,
                                  "Total Students" : sch_total_student,
                                  "Total School Budget" : school_budget,
                                  "Per Student Budget" : sch_per_stu_budget,
                                  "Average Math Score" : sch_avg_math_score,
                                  "Average Reading Score" : sch_avg_read_score,
                                  "% Passing Math" : sch_math_pass_percent,
                                  "% Passing Reading" : sch_read_pass_percent,
                                  "% Overall Passing" : sch_overall_pass_percent
                                 })

# Formating the columns
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)

# Display data frame
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
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

Top Performing Schools (By % Overall Passing)

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

In [4]:
```
# Sory the above summary DF by % Overall Passing in descending order
school_summary_df = school_summary_df.sort_values(by = "% Overall Passing" , ascending = False)

# Displaye top 5 records of sorted data frame
school_summary_df.head()
```

Out[4]:

	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

Bottom Performing Schools (By % Overall Passing)

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

In [5]:
```
# Sory the above summary DF by % Overall Passing in ascending order
school_summary_df = school_summary_df.sort_values(by = "% Overall Passing" , ascending = True)

# Displaye top 5 records of sorted data frame
school_summary_df.head()
```

Out[5]:

	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

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 [6]:

```
# calculate average of math score if grade matches and grouped by school name
avg_math_grade9 = school_data_complete.loc[school_data_complete["grade"] == "9th"].groupby("school_name")["math_score"].mean()
avg_math_grade10 = school_data_complete.loc[school_data_complete["grade"] == "10th"].groupby("school_name")["math_score"].mean
avg_math_grade11 = school_data_complete.loc[school_data_complete["grade"] == "11th"].groupby("school_name")["math_score"].mean
avg_math_grade12 = school_data_complete.loc[school_data_complete["grade"] == "12th"].groupby("school_name")["math_score"].mean

# Create Data Frame with values obtained from above
math_score_grade_summary = pd.DataFrame({"9th" : avg_math_grade9,
                                         "10th" : avg_math_grade10,
                                         "11th" : avg_math_grade11,
                                         "12th" : avg_math_grade12
                                         })
# Set index to None
math_score_grade_summary.index.name = None

# Display data frame
math_score_grade_summary
```

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

Reading Score by Grade

- Perform the same operations as above for reading scores

In [7]:

```
# calculate average of reading score if grade matches and grouped by school name
avg_read_grade9 = school_data_complete.loc[school_data_complete["grade"] == "9th"].groupby("school_name")["reading_score"].mean
avg_read_grade10 = school_data_complete.loc[school_data_complete["grade"] == "10th"].groupby("school_name")["reading_score"].me
avg_read_grade11 = school_data_complete.loc[school_data_complete["grade"] == "11th"].groupby("school_name")["reading_score"].me
avg_read_grade12 = school_data_complete.loc[school_data_complete["grade"] == "12th"].groupby("school_name")["reading_score"].me

# Create Data Frame with values obtained from above
reading_score_grade_summary = pd.DataFrame({"9th" : avg_read_grade9,
                                            "10th" : avg_read_grade10,
                                            "11th" : avg_read_grade11,
                                            "12th" : avg_read_grade12
                                           })
# Set index to None
reading_score_grade_summary.index.name = None

# Display data frame
reading_score_grade_summary
```

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

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 [8]:
```
#Get the Min & Max budget per student
min_budget = school_summary_df["Per Student Budget"].min()
max_budget = school_summary_df["Per Student Budget"].max()
min_budget, max_budget
```

Out[8]:
```
('$578.00', '$655.00')
```

In [9]:

```
# Convert the values to Int.
#If data type is not changed then will get an error of operation is not supported between instances of 'int' and 'str'
school_summary_df["Per Student Budget"] = school_summary_df["Per Student Budget"].str.replace("$","").astype(float)
school_summary_df["Per Student Budget"] = pd.to_numeric(school_summary_df["Per Student Budget"])
school_summary_df["Average Math Score"] = pd.to_numeric(school_summary_df["Average Math Score"])
school_summary_df["Average Reading Score"] = pd.to_numeric(school_summary_df["Average Reading Score"])

# Create bins and labels
bins = [0, 584, 629, 644, 675]
budget_labels = ["<$585", "$585-629", "$630-644", "$645-675"]

# Categorized schools as per sudent budget bns & add a column as Spending Ranges(Per Student)
school_summary_df["Spending Ranges(Per Student)"] = pd.cut(school_summary_df["Per Student Budget"], bins, labels = budget_labe

#Display a dataframe
school_summary_df.head()
```

Out[9]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing	Spending Ranges(Per Student)
Rodriguez High School	District	3999	$2,547,363.00	637.0	76.842711	80.744686	66.366592	80.220055	52.988247	$630-644
Figueroa High School	District	2949	$1,884,411.00	639.0	76.711767	81.158020	65.988471	80.739234	53.204476	$630-644
Huang High School	District	2917	$1,910,635.00	655.0	76.629414	81.182722	65.683922	81.316421	53.513884	$645-675
Hernandez High School	District	4635	$3,022,020.00	652.0	77.289752	80.934412	66.752967	80.862999	53.527508	$645-675
Johnson High School	District	4761	$3,094,650.00	650.0	77.072464	80.966394	66.057551	81.222432	53.539172	$645-675

In [10]:

```
# Create groupby object based upon spending ranges
spending_range_grouped = school_summary_df.groupby("Spending Ranges(Per Student)")

# Calculating average of Math, reading score & % math, reading, overall passing
spending_range_avg_math = spending_range_grouped["Average Math Score"].mean()
spending_range_avg_read = spending_range_grouped["Average Reading Score"].mean()
spending_math_percent = spending_range_grouped["% Passing Math"].mean()
spending_read_percent = spending_range_grouped["% Passing Reading"].mean()
spending_overall_passing = spending_range_grouped["% Overall Passing"].mean()

# Create data frame from values obtained above
spending_summary_df = pd.DataFrame({"Average Math Score" : spending_range_avg_math,
                                    "Average Reading Score" : spending_range_avg_read,
                                    "% Passing Math" : spending_math_percent,
                                    "% Passing Reading" : spending_read_percent,
                                    "% Overall Passing" : spending_overall_passing
                                   })

# Format Data Frame
spending_summary_df["Average Math Score"] = spending_summary_df["Average Math Score"].map("{:.2f}".format)
spending_summary_df["Average Reading Score"] = spending_summary_df["Average Reading Score"].map("{:.2f}".format)
spending_summary_df["% Passing Math"] = spending_summary_df["% Passing Math"].map("{:.2f}".format)
spending_summary_df["% Passing Reading"] = spending_summary_df["% Passing Reading"].map("{:.2f}".format)
spending_summary_df["% Overall Passing"] = spending_summary_df["% Overall Passing"].map("{:.2f}".format)

# Display Data Frame
spending_summary_df
```

Out[10]:

Spending Ranges(Per Student)	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
<$585	83.46	83.93	93.46	96.61	90.37
$585-629	81.90	83.16	87.13	92.72	81.42
$630-644	78.52	81.62	73.48	84.39	62.86
$645-675	77.00	81.03	66.16	81.13	53.53

Scores by School Size

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

In [11]:

```
# Create bins and labels
stu_count_bins = [0, 999, 1999, 5000]
stu_bin_label = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]

# Categorized schools as per total student count (size) & add a column as School Size
school_summary_df["School Size"] = pd.cut(school_summary_df["Total Students"], stu_count_bins, labels=stu_bin_label, include_l

# Create a groupby object on School Size
school_size_grouped = school_summary_df.groupby("School Size")

# Calculating average of Math, reading score & % math, reading, overall passing
size_avg_math = school_size_grouped["Average Math Score"].mean()
size_avg_read = school_size_grouped["Average Reading Score"].mean()
size_math_percent = school_size_grouped["% Passing Math"].mean()
size_read_percent = school_size_grouped["% Passing Reading"].mean()
size_overall_passing = school_size_grouped["% Overall Passing"].mean()

# Create data frame with values obtained above
size_score_df = pd.DataFrame({"Average Math Score" : size_avg_math,
                              "Average Reading Score" : size_avg_read,
                              "% Passing Math" : size_math_percent,
                              "% Passing Reading" : size_read_percent,
                              "% Overall Passing" : size_overall_passing})

size_score_df
```

Out[11]:

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Small (<1000)	83.821598	83.929843	93.550225	96.099437	89.883853
Medium (1000-2000)	83.374684	83.864438	93.599695	96.790680	90.621535
Large (2000-5000)	77.746417	81.344493	69.963361	82.766634	58.286003

Scores by School Type

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

In [12]:

```
school_type_grouped = school_summary_df.groupby("School Type")

# Calculating average of Math, reading score & % math, reading, overall passing
type_avg_math = school_type_grouped["Average Math Score"].mean()
type_avg_read = school_type_grouped["Average Reading Score"].mean()
type_math_percent = school_type_grouped["% Passing Math"].mean()
type_read_percent = school_type_grouped["% Passing Reading"].mean()
type_overall_passing = school_type_grouped["% Overall Passing"].mean()

school_type_summary = pd.DataFrame({"Average Math Score" : type_avg_math,
                                    "Average Reading Score" : type_avg_read,
                                    "% Passing Math" : type_math_percent,
                                    "% Passing Reading" : type_read_percent,
                                    "% Overall Passing" : type_overall_passing
                                   })
school_type_summary
```

Out[12]:

School Type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
Charter	83.473852	83.896421	93.620830	96.586489	90.432244
District	76.956733	80.966636	66.548453	80.799062	53.672208