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 [3]:

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

In [2]:

```
# File to Load (Remember to Change These)
school_data =  "Resources/schools_complete.csv"
student_data = "Resources/students_complete.csv"

#Read school and Student Data File and store into Pandas Data Frames
school_df = pd.read_csv(school_data)
student_df = pd.read_csv(student_data)

#Combine the data into a single dataset
merge_df = pd.merge(student_df, school_df, how = "left", on = ["school_name", "school_name"])

# merge_df = pd.merge(school_df, student_df, on = ["school_name", "school_name"])
merge_df
```

Out[2]:

	Student ID	student_name	gender	grade	school_name	reading_score	math_score	School ID	type	size	budget
0	0	Paul Bradley	M	9th	Huang High School	66	79	0	District	2917	1910635
1	1	Victor Smith	M	12th	Huang High School	94	61	0	District	2917	1910635
2	2	Kevin Rodriguez	M	12th	Huang High School	90	60	0	District	2917	1910635
3	3	Dr. Richard Scott	M	12th	Huang High School	67	58	0	District	2917	1910635
4	4	Bonnie Ray	F	9th	Huang High School	97	84	0	District	2917	1910635
...	...	...	...	...	...	...	...	...	...	...	...
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

In [3]:

```
merge_df.dtypes
```

Out[3]:

```
Student ID        int64
student_name     object
gender           object
grade            object
school_name      object
reading_score     int64
math_score        int64
School ID         int64
type             object
size              int64
budget            int64
dtype: object
```

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 overall passing rate (overall average score), i.e. (avg. math score + avg. reading score)/2
- 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)
- Create a dataframe to hold the above results
- Optional: give the displayed data cleaner formatting

In [4]:

```
total_number_schools = school_df["school_name"].count()
total_number_students = merge_df["student_name"].count()

# total budget calculates incorrectly, need to talk to TA or Travis
total_budget = school_df["budget"].sum()

average_math_score = merge_df["math_score"].mean()
average_reading_score = merge_df["reading_score"].mean()

passing_math_score = merge_df.loc[merge_df["math_score"] >= 70]["student_name"].count()
passing_math_rate = (passing_math_score / total_number_students) * 100

passing_reading_score = merge_df.loc[merge_df["reading_score"] >= 70]["student_name"].count()
passing_reading_rate = (passing_reading_score / total_number_students) * 100

overall_passing_rate = (passing_math_rate + passing_reading_rate) / 2

results_df = pd.DataFrame({"Total Schools" : total_number_schools, "Total Students" :
total_number_students,
                           "Total Budget" : total_budget,
                           "Average Math Score" : average_math_score, "Average Reading Score" : ave
age_reading_score,
                          "% Passing Math" : passing_math_rate, "% Passing Reading" : passing_readi
g_rate,
                          "% Overall Passing Rate" : overall_passing_rate}, index = [0])

# using map to format displaying $ and two decimal points
results_df["Total Budget"] = results_df["Total Budget"].map("${:,.2f}".format)
results_df
```

Out[4]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
0	15	39170	$24,649,428.00	78.985371	81.87784	74.980853	85.805463	80.393158

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 Rate (Average of the above two)

- Create a dataframe to hold the above results

In [5]:

```
school_summary_df = pd.merge(student_df, school_df, on = "school_name")
school_summary_df
```

Out[5]:

	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

In [6]:

```
# School Summary
# school_summary = school_df
school_summary_df = pd.merge(student_df, school_df, on = "school_name")

school_name = school_df["school_name"].unique()
# school_name = school_df["school_name"].count()
# school_name = school_summary_df.groupby("school_name")["school_name"].unique()
school_type = school_summary_df.groupby("school_name")["type"].unique()

# total_school_students = school_summary_df["size"].count()
total_school_students = school_summary_df["school_name"].value_counts()

# total school budget and per student budget
total_school_budget = school_summary_df.groupby("school_name")["budget"].mean()
per_student_budget = total_school_budget / total_school_students

# average school math and reading score to get passing rate and overall passing rate
average_school_math_score = school_summary_df.groupby("school_name")["math_score"].mean()
average_school_reading_score = school_summary_df.groupby("school_name")["reading_score"].mean()


passing_school_math_score = school_summary_df[(school_summary_df["math_score"] >= 70)]
passing_school_reading_score = school_summary_df[(school_summary_df["reading_score"] >= 70)]

passing_school_math_rate = (passing_school_math_score.groupby(["school_name"])
["student_name"].count() / total_school_students) * 100
passing_school_reading_rate = (passing_school_reading_score.groupby(["school_name"])
["student_name"].count() / total_school_students) * 100


overall_school_passing_rate = (passing_school_math_rate + passing_school_reading_rate) / 2

# create dataframe
results_school_summary_df = pd.DataFrame({"School Name" : school_name, "School Type" : school_type,
"Total Students" : total_school_students,
                                "Total School Budget" : total_school_budget, "Per Student Budget"
: per_student_budget,
                                "Average Math Score" : average_school_math_score, "Average Readir
Score" : average_school_reading_score,
                               "% Passing Math" : passing_school_math_rate, "% Passing Reading" :
passing_school_reading_rate,
```

```
                                    "% Overall Passing Rate" : overall_school_passing_rate
                                   })
del results_school_summary_df["School Name"]

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

results_school_summary_df
```

Out[6]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Bailey High School	[District]	4976	$3,124,928.00	$628.00	77.048432	81.033963	66.680064	81.933280	74.306672
Cabrera High School	[Charter]	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	95.586652
Figueroa High School	[District]	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	73.363852
Ford High School	[District]	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602	79.299014	73.804308
Griffin High School	[Charter]	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	95.265668
Hernandez High School	[District]	4635	$3,022,020.00	$652.00	77.289752	80.934412	66.752967	80.862999	73.807983
Holden High School	[Charter]	427	$248,087.00	$581.00	83.803279	83.814988	92.505855	96.252927	94.379391
Huang High School	[District]	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	73.500171
Johnson High School	[District]	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	73.639992
Pena High School	[Charter]	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	95.270270
Rodriguez High School	[District]	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	73.293323
Shelton High School	[Charter]	1761	$1,056,600.00	$600.00	83.359455	83.725724	93.867121	95.854628	94.860875
Thomas High School	[Charter]	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	95.290520
Wilson High School	[Charter]	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	95.203679
Wright High School	[Charter]	1800	$1,049,400.00	$583.00	83.682222	83.955000	93.333333	96.611111	94.972222

Top Performing Schools (By Passing Rate)

- Sort and display the top five schools in overall passing rate

In [7]:

```
# Top Five Schools by Passing Rate
top_five = results_school_summary_df.sort_values("% Overall Passing Rate", ascending = False)
top_five.head()
```

Out[7]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Cabrera High School	[Charter]	1858	$1,081,356.00	$582.00	83.061895	83.975780	94.133477	97.039828	95.586652
Thomas High									

[illegible] School	[Charter]	1635	$1,043,130.00	$638.00	83.418349	83.848930	93.272171	97.308869	95.290520
	School Type	**Total Students**	**Total School Budget**	**Per Student Budget**	**Average Math Score**	**Average Reading Score**	**% Passing Math**	**% Passing Reading**	**% Overall Passing Rate**
Pena High School	[Charter]	962	$585,858.00	$609.00	83.839917	84.044699	94.594595	95.945946	95.270270
Griffin High School	[Charter]	1468	$917,500.00	$625.00	83.351499	83.816757	93.392371	97.138965	95.265668
Wilson High School	[Charter]	2283	$1,319,574.00	$578.00	83.274201	83.989488	93.867718	96.539641	95.203679

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

In [8]:

```
bottom_five = results_school_summary_df.sort_values("% Overall Passing Rate", ascending = True)
bottom_five.head()
```

Out[8]:

	School Type	Total Students	Total School Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Rodriguez High School	[District]	3999	$2,547,363.00	$637.00	76.842711	80.744686	66.366592	80.220055	73.293323
Figueroa High School	[District]	2949	$1,884,411.00	$639.00	76.711767	81.158020	65.988471	80.739234	73.363852
Huang High School	[District]	2917	$1,910,635.00	$655.00	76.629414	81.182722	65.683922	81.316421	73.500171
Johnson High School	[District]	4761	$3,094,650.00	$650.00	77.072464	80.966394	66.057551	81.222432	73.639992
Ford High School	[District]	2739	$1,763,916.00	$644.00	77.102592	80.746258	68.309602	79.299014	73.804308

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 [9]:

```
ninth_math = student_df.loc[student_df["grade"] == "9th"].groupby("school_name")["math_score"].mean
()
tenth_math = student_df.loc[student_df["grade"] == "10th"].groupby("school_name")["math_score"].mea
n()
eleventh_math = student_df.loc[student_df["grade"] == "11th"].groupby("school_name")["math_score"].
mean()
twelfth_math = student_df.loc[student_df["grade"] == "12th"].groupby("school_name")["math_score"].m
ean()

student_df.rename(columns = {"school_name" : " "})

grade_df = pd.DataFrame({"School Name" : school_name, "9th" : ninth_math, "10th" : tenth_math,
                         "11th" : eleventh_math, "12th" : twelfth_math})

# .set_index("School Name").rename_axis(None)

del grade_df["School Name"]

# grade_df.columns

grade_df
```

Out[9]:

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	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 [10]:

```
ninth_reading = student_df.loc[student_df["grade"] == "9th"].groupby("school_name")["math_score"].m
ean()
tenth_reading = student_df.loc[student_df["grade"] == "10th"].groupby("school_name")["math_score"].
mean()
eleventh_reading = student_df.loc[student_df["grade"] == "11th"].groupby("school_name")["math_score
"].mean()
twelfth_reading = student_df.loc[student_df["grade"] == "12th"].groupby("school_name")["math_score"
].mean()

student_df.rename(columns = {"school_name" : " "})

grade_df = pd.DataFrame({"School Name" : school_name, "9th" : ninth_reading, "10th" : tenth_reading
,
                         "11th" : eleventh_reading, "12th" : twelfth_reading})

del grade_df["School Name"]

# del grade_df.index.name

grade_df
```

Out[10]:

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

school_name	9th	10th	11th	12th
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

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 [11]:

```
#create bins
school_spending = results_school_summary_df
spending_bins = [0, 585, 615, 645, 675]
group_names = ["<$585", "$585-615", "$615-645", "$645-675"]
student_spending = school_spending["Per Student Budget"].replace("[\$,)]", "" , regex=True)

#change data type to float
per_student_spending = student_spending.astype(float)
per_student_spending

school_spending["Spending Ranges (Per Student)"] = pd.cut(per_student_spending, spending_bins,
labels = group_names)
school_spending["Average Math Score"] = school_spending["Average Math Score"].astype(float)
school_spending["Average Reading Score"] = school_spending["Average Reading Score"].astype(float)


school_spending_math = school_spending.groupby("Spending Ranges (Per Student)")["Average Math Scor
e"].mean()
school_spending_reading = school_spending.groupby("Spending Ranges (Per Student)")["Average
Reading Score"].mean()

#reset data types to float
school_spending["% Passing Math"] = school_spending["% Passing Math"].astype(float)
school_spending["% Passing Reading"] = school_spending["% Passing Reading"].astype(float)
school_spending["% Overall Passing Rate"] = school_spending["% Overall Passing Rate"].astype(float)

#calculate percentages
spending_pass_math = school_spending.groupby("Spending Ranges (Per Student)")["% Passing Math"].me
an()
spending_pass_reading = school_spending.groupby("Spending Ranges (Per Student)")["% Passing
Reading"].mean()
spending_overall_reading = school_spending.groupby("Spending Ranges (Per Student)")["% Overall Pas
sing Rate"].mean()

#set dataframe
results_spending = pd.DataFrame({"Average Math Score": school_spending_math, "Average Reading
Score": school_spending_reading,
                                 "% Passing Math": spending_pass_math, "% Passing Reading": spendin
_pass_reading,
                                 "% Overall Passing Rate": spending_overall_reading})

results_spending = results_spending[["Average Math Score", "Average Reading Score", "% Passing
Math", "% Passing Reading",
                                     "% Overall Passing Rate"]]
results_spending
```

Out[11]:

Spending Ranges (Per Student)	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
<$585	83.455399	83.933814	93.460096	96.610877	95.035486
$585-615	83.599686	83.885211	94.230858	95.900287	95.065572
$615-645	79.079225	81.891436	75.668212	86.106569	80.887391
$645-675	76.997210	81.027843	66.164813	81.133951	73.649382

Scores by School Size

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

In [12]:

```
# Sample bins. Feel free to create your own bins.
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

In [13]:

```
#create bins
school_size = results_school_summary_df
size_bins = [0, 1000, 2000, 5000]
group_names = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
student_size = school_size["Total Students"]

#change data type to float
per_student_spending = student_size.astype(float)
per_student_spending

school_size["School Size"] = pd.cut(per_student_spending, size_bins, labels = group_names)
school_size["Average Math Score"] = school_size["Average Math Score"].astype(float)
school_size["Average Reading Score"] = school_size["Average Reading Score"].astype(float)


school_spending_math = school_size.groupby("School Size")["Average Math Score"].mean()
school_spending_reading = school_size.groupby("School Size")["Average Reading Score"].mean()

#reset data types to float
school_size["% Passing Math"] = school_size["% Passing Math"].astype(float)
school_size["% Passing Reading"] = school_size["% Passing Reading"].astype(float)
school_size["% Overall Passing Rate"] = school_size["% Overall Passing Rate"].astype(float)

#calculate percentages
spending_pass_math = school_size.groupby("School Size")["% Passing Math"].mean()
spending_pass_reading = school_size.groupby("School Size")["% Passing Reading"].mean()
spending_overall_reading = school_size.groupby("School Size")["% Overall Passing Rate"].mean()

#set dataframe
results_size = pd.DataFrame({"Average Math Score": school_spending_math, "Average Reading Score":
school_spending_reading,
                             "% Passing Math": spending_pass_math, "% Passing Reading": spendin
_pass_reading,
                             "% Overall Passing Rate": spending_overall_reading})

results_size = results_size[["Average Math Score", "Average Reading Score", "% Passing Math", "% P
assing Reading",
                             "% Overall Passing Rate"]]
results_size
```

Out[13]:

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate

School Size	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
Small (<1000)	83.821598	83.929843	93.550225	96.099437	94.824831
School Size Medium (1000-2000)	83.374684	83.864438	93.599695	96.790680	95.195187
Large (2000-5000)	77.746417	81.344493	69.963361	82.766634	76.364998

Scores by School Type

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

In [14]:

```
school_type = results_school_summary_df

# school_type["School Size"] = pd.cut(per_student_spending, size_bins, labels = group_names)
school_type["Average Math Score"] = school_type["Average Math Score"].astype(float)
school_type["Average Reading Score"] = school_type["Average Reading Score"].astype(float)

# school_type.dtypes
# school_type["School Type"].value_counts()
# help from Travis
# this is a subtle difference: because lists are mutable,
# they can't be "hash"ed, which is what groupby is using to do the grouping.
# However, tuples are immutable, and can be hashed
school_type['School Type'] = school_type['School Type'].map(tuple)

school_type_math = school_type.groupby("School Type")["Average Math Score"].mean()
school_type_reading = school_type.groupby("School Type")["Average Reading Score"].mean()

#reset data types to float
school_type["% Passing Math"] = school_type["% Passing Math"].astype(float)
school_type["% Passing Reading"] = school_type["% Passing Reading"].astype(float)
school_type["% Overall Passing Rate"] = school_type["% Overall Passing Rate"].astype(float)

#calculate percentages
type_pass_math = school_type.groupby("School Type")["% Passing Math"].mean()
type_pass_reading = school_type.groupby("School Type")["% Passing Reading"].mean()
type_overall_reading = school_type.groupby("School Type")["% Overall Passing Rate"].mean()

#set dataframe
results_type = pd.DataFrame({"Average Math Score": school_type_math, "Average Reading Score":
school_type_reading,
                                "% Passing Math": type_pass_math, "% Passing Reading": type_pass_r
ading,
                                "% Overall Passing Rate": type_overall_reading})

results_type = results_type[["Average Math Score", "Average Reading Score", "% Passing Math", "% P
assing Reading",
                                     "% Overall Passing Rate"]]
results_type
```

Out[14]:

School Type	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
(Charter,)	83.473852	83.896421	93.620830	96.586489	95.103660
(District,)	76.956733	80.966636	66.548453	80.799062	73.673757

In [ ]:

```
# Description of observable trends

# 1. Based on the data, we can analyze that the Chareter schools performed better than the Distric
t schools.
# 2. Schools with smaller number of students performed better in both math and reading
```

In [ ]: