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

# File to Load
school_data_load = "Resources/schools_complete.csv"
student_data_load = "Resources/students_complete.csv"

# Read School and Student Data File and store into Pandas Data Frames
school_data = pd.read_csv(school_data_load)
student_data = pd.read_csv(student_data_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.head()
```

Out[3]:

	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

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 of schools
total_schools=(school_data_complete["school_name"]).nunique()
total_schools
```

Out[4]:

```
15
```

In [5]:

```
#total number of students
total_students=(school_data_complete["student_name"]).count()
total_students
```

Out[5]:

```
39170
```

In [6]:

```
#total budget
total_budget=sum(school_data_complete["budget"])
total_budget
```

Out[6]:

```
82932329558
```

In [7]:

```
#average math score
avg_math= sum(school_data_complete["math_score"])/total_students
avg_math
```

Out[7]:

```
78.98537145774827
```

In [8]:

```
#average reading score
avg_reading=sum(school_data_complete["reading_score"])/total_students
avg_reading
```

Out[8]:

```
81.87784018381414
```

In [9]:

```
#overall passing (math score + reading score)/2
overall_passing=(avg_math+avg_reading)/2
overall_passing
```

Out[9]:

```
80.43160582078121
```

In [10]:

```
pass_math=(school_data_complete["math_score"]>= 70)
pass_math=(pass_math).sum()
total_pass_math=(pass_math/total_students)*100
total_pass_math
```

Out[10]:

```
74.9808526933878
```

In [11]:

```
pass_reading=(school_data_complete["reading_score"]>= 70)
pass_reading=(pass_reading).sum()
total_pass_reading=(pass_reading/total_students)*100
total_pass_reading
```

Out[11]:

```
85.80546336482001
```

In [12]:

```
dis_sum={'Total Number of Schools': total_schools
```

```
dis_sum={'Total Number of Schools': total_schools,
           'Total Number of Students': total_students,
           'Total Budget': total_budget,
           'Average Math Score': avg_math,
           'Average Reading score': avg_reading,
           'Total Pass Math %':total_pass_math,
           'Total Pass Reading %': total_pass_reading,
         'Overall Passing': overall_passing
        }
dis_sum
```

Out[12]:

```
{'Total Number of Schools': 15,
 'Total Number of Students': 39170,
 'Total Budget': 82932329558,
 'Average Math Score': 78.98537145774827,
 'Average Reading score': 81.87784018381414,
 'Total Pass Math %': 74.9808526933878,
 'Total Pass Reading %': 85.80546336482001,
 'Overall Passing': 80.43160582078121}
```

In [ ]:

```

```

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 [13]:

```
#create a copy of the master data in order not to alter master file then print
school_complete_copy= school_data_complete.copy()
school_copy=school_data.copy()
#school_copy
```

In [14]:

```
#remove unncessary column 'School ID' then print to verify
#del school_copy['School ID']
#school_copy
```

In [15]:

```
#number of total students per school
school_data_complete['size'].unique()
```

Out[15]:

```
array([2917, 2949, 1761, 4635, 1468, 2283, 1858, 4976,  427,  962, 1800,
       3999, 4761, 2739, 1635])
```

In [16]:

```
#Per Student Budget: 'budget' divided by 'size' then print
```

```
school_copy['Per Student Budget']=school_copy['budget']/school_copy['size']
#school_copy
```

In [17]:

```
#Average Math Score: Use groupby function for 'school_name' and 'math_score'
#Reset index and print
avg_math_score= school_complete_copy.groupby(['school_name'])['math_score'].mean().reset_index()
#avg_math_score
```

In [18]:

```
#Average Reading Score: Use groupby function for 'school_name' and 'reading_score'
#Reset index and print
avg_reading_score= school_complete_copy.groupby(['school_name'])
['reading_score'].mean().reset_index()
#avg_reading_score
```

In [19]:

```
#Merge 'avg_math_score' with 'school_copy' then print
school_copy=school_copy.merge(avg_math_score, on='school_name',how='outer')
#school_copy
```

In [20]:

```
#Merge 'avg_reading_score' with 'school_copy' then print
school_copy=school_copy.merge(avg_reading_score, on='school_name',how='outer')
#school_copy
```

In [21]:

```
#Use a conditional function to find %Passing Math (passing >= 70%) then print
pass_math_total=school_complete_copy[school_complete_copy['math_score']>=70]
#pass_math_total
```

In [22]:

```
#Use a conditional function to find %Passing Reading (passing >= 70%) then print
pass_reading_total=school_complete_copy[school_complete_copy['reading_score']>=70]
#pass_reading_total
```

In [23]:

```
#Count students passing math then reset index followed by print
pass_math_group=pass_math_total.groupby(['school_name'])['math_score'].count().reset_index()
#Rename column 'math_score' to 'math pass count'
pass_math_group.rename({'math_score' : 'math pass count'}, axis=1, inplace=True)
#pass_math_group
```

In [24]:

```
#Count students passing reading then reset index followed by print
pass_reading_group=pass_reading_total.groupby(['school_name'])
['reading_score'].count().reset_index()
#Rename column 'reading_score' to 'reading pass count'
pass_reading_group.rename({'reading_score' : 'reading pass count'}, axis=1, inplace=True)
#pass_reading_group
```

In [25]:

```
#merge the grouped math pass and reading pass counts by school then print
pass_math_reading=pass_math_group.merge(pass_reading_group, on='school_name',how='inner')
#pass_math_reading
```

In [26]:

```
#merge the pass_math_reading dataset with the copy of master file (school_copy) then print
school_copy=school_copy.merge(pass_math_reading, on='school_name',how='outer')
#school_copy
```

In [27]:

```
#math pass count divided by school size*100 to find %passing math then print
school_copy['% Passing Math']=(school_copy['math pass count']/school_copy['size'])*100
#school_copy
```

In [28]:

```
#reading pass count divided by school size*100 to find %passing reading then print
school_copy['% Passing Reading']=(school_copy['reading pass count']/school_copy['size'])*100
#school_copy
```

In [29]:

```
#%overall passing = adding %passing math and %passing reading and dividing by 2 then print
school_copy['% Overall Passing'] = (school_copy['% Passing Math'] + school_copy['% Passing Reading'
])/2
#school_copy
```

In [30]:

```
#delete extra copy of average math score and print
#del school_copy['math_score_x']
#school_copy
```

In [31]:

```
#delete math pass count and reading pass count columns
#del school_copy['math pass count']
#del school_copy['reading pass count']
#school_copy
```

In [160]:

```
#rename each of the columns and print
school_copy.rename({'school_name': 'School Name',
                   'type':'School Type',
                   'size':'Total Students',
                    'budget':'Total Budget',
                   'math_score':'Average Math Score',
                   'reading_score':'Average Reading Score'}, axis=1, inplace=True)
school_copy
```

Out[160]:

	School Name	School Type	Total Students	Total Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
0	Huang High School	District	2917	1910635	655.0	76.629414	81.182722	65.683922	81.316421	73.500171
1	Figueroa High School	District	2949	1884411	639.0	76.711767	81.158020	65.988471	80.739234	73.363852
2	Shelton High School	Charter	1761	1056600	600.0	83.359455	83.725724	93.867121	95.854628	94.860875
3	Hernandez High School	District	4635	3022020	652.0	77.289752	80.934412	66.752967	80.862999	73.807983
4	Griffin High School	Charter	1468	917500	625.0	83.351499	83.816757	93.392371	97.138965	95.265668
5	Wilson High School	Charter	2283	1319574	578.0	83.274201	83.989488	93.867718	96.539641	95.203679
6	Cabrera High School	Charter	1858	1081356	582.0	83.061895	83.975780	94.133477	97.039828	95.586652
7	Bailey High School	District	4976	3124928	628.0	77.048432	81.033963	66.680064	81.933280	74.306672

	School Name	School Type	Total Students	Total Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
8	Holden High School	Charter	427	248087	581.0	83.803279	83.814988	92.505855	96.252927	94.379391
9	Pena High School	Charter	962	585858	609.0	83.839917	84.044699	94.594595	95.945946	95.270270
10	Wright High School	Charter	1800	1049400	583.0	83.682222	83.955000	93.333333	96.611111	94.972222
11	Rodriguez High School	District	3999	2547363	637.0	76.842711	80.744686	66.366592	80.220055	73.293323
12	Johnson High School	District	4761	3094650	650.0	77.072464	80.966394	66.057551	81.222432	73.639992
13	Ford High School	District	2739	1763916	644.0	77.102592	80.746258	68.309602	79.299014	73.804308
14	Thomas High School	Charter	1635	1043130	638.0	83.418349	83.848930	93.272171	97.308869	95.290520

Top Performing Schools (By Passing Rate)

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

In [161]:

```
#sort schools by % overall passing to determine top performing schools then print
best_schools = school_copy.sort_values(by=['% Overall Passing'], ascending=False).head()
best_schools
```

Out[161]:

	School Name	School Type	Total Students	Total Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
6	Cabrera High School	Charter	1858	1081356	582.0	83.061895	83.975780	94.133477	97.039828	95.586652
14	Thomas High School	Charter	1635	1043130	638.0	83.418349	83.848930	93.272171	97.308869	95.290520
9	Pena High School	Charter	962	585858	609.0	83.839917	84.044699	94.594595	95.945946	95.270270
4	Griffin High School	Charter	1468	917500	625.0	83.351499	83.816757	93.392371	97.138965	95.265668
5	Wilson High School	Charter	2283	1319574	578.0	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 [162]:

```
#sort schools by % overall passing to determine worst performing schools then print
worst_schools = school_copy.sort_values(by=['% Overall Passing'], ascending=True).head()
worst_schools
```

Out[162]:

	School Name	School Type	Total Students	Total Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
11	Rodriguez High School	District	3999	2547363	637.0	76.842711	80.744686	66.366592	80.220055	73.293323
1	Figueroa High School	District	2949	1884411	639.0	76.711767	81.158020	65.988471	80.739234	73.363852
0	Huang High School	District	2917	1910635	655.0	76.629414	81.182722	65.683922	81.316421	73.500171
12	Johnson High	District	4761	3094650	650.0	77.072464	80.966394	66.057551	81.222432	73.639992

	School Name	School Type	Total Students	Total Budget	Per Student Budget	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
13	Ford High School	District	2739	1763916	644.0	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 [163]:

```
#pandas pivot table to sort math scores by grade then print
math_grade_scores = pd.pivot_table(school_complete_copy, values=['math_score'],
index=['school_name'],
                                         columns=['grade'])
math_grade_scores = math_grade_scores.reindex(labels=['9th',
                                                                    '10th',
                                                                    '11th',
                                                                    '12th'], axis=1, level=1)
#math_grade_scores
```

In [164]:

```
#rename columns and header for cleaner dataset then print
math_grade_scores.rename({'math_score': 'Math Score',
                   '9th':'9th Grade',
                   '10th':'10th Grade',
                    '11th':'11th Grade',
                   'school_name':'School Name',
                   'grade':'Grade'}, axis=1, inplace=True)
math_grade_scores
```

Out[164]:

	Math Score			
grade	9th Grade	10th Grade	11th Grade	12th
school_name				
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

In [165]:

```
#pandas pivot table to sort reading scores by grade then print
reading_grade_scores = pd.pivot_table(school_complete_copy, values=['reading_score'], index=['school_name'],
                                      columns=['grade'])
reading_grade_scores = reading_grade_scores.reindex(labels=['9th',
                                                            '10th',
                                                            '11th',
                                                            '12th'], axis=1, level=1)
#reading_grade_scores
```

In [166]:

```
#rename columns and header for cleaner dataset then print
reading_grade_scores.rename({'reading_score': 'Reading Score',
                   '9th':'9th Grade',
                   '10th':'10th Grade',
                    '11th':'11th Grade',
                   'school_name':'School Name',
                   'grade':'Grade'}, axis=1, inplace=True)
reading_grade_scores
```

Out[166]:

	Reading Score			
grade	9th Grade	10th Grade	11th Grade	12th
school_name				
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

- Perform the same operations as above for reading scores

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 [167]:

```
# Sample bins. Feel free to create your own bins.
spending_bins = [0, 585, 615, 645, 675]
spending_bins = [0, 585, 615, 645, 675]
group_labels = ["<$585", "$585-615", "$615-645", "$645-675"]
```

In [168]:

```
#create copy of master school data (school_copy) to manipulate new data then print
school_spend=school_copy.copy()
#school_spend
```

In [169]:

```
#Per Student Budget binned (with previously established bins) and labeled by group labels then pri
nt
school_spend_binned=pd.cut(school_spend['Per Student Budget'],spending_bins,labels=group_labels)
#school_spend_binned
#create dataset from binned data
school_spend_binned=pd.DataFrame(school_spend_binned)
#add column 'Spending Per Student' to binned dataset then print
school_spend['Spending Per Student']=school_spend_binned
#school_spend_binned
```

In [170]:

```
#use groupby function to add columns from master data set then print
school_spend = school_spend.groupby(['Spending Per Student'])['Average Reading Score','Average Mat
h Score','% Passing Reading','% Passing Math','% Overall Passing'].max()
school_spend
```

Out[170]:

	Average Reading Score	Average Math Score	% Passing Reading	% Passing Math	% Overall Passing
Spending Per Student					
$585-615	84.044699	83.839917	95.945946	94.594595	95.270270
$615-645	83.848930	83.418349	97.308869	93.392371	95.290520
$645-675	81.182722	77.289752	81.316421	66.752967	73.807983
<$585	83.989488	83.803279	97.039828	94.133477	95.586652

Scores by School Size

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

In [190]:

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

In [191]:

```
#create copy of master school data (school_copy) to manipulate new data then print
school_size=school_copy.copy()
#school_size
```

In [192]:

```
#Total Students binned (with previously established bins) and labeled by group labels then print
school_size_binned=pd.cut(school_size['Total Students'],size_bins,labels=size_names)
#school_size_binned
#create dataset from binned data
school_size_binned=pd.DataFrame(school_size_binned)
#add column 'Size of Population' to binned dataset then print
school_size['Size of Population']=school_size_binned
#school_size_binned
```

In [193]:

```
#use groupby function to add columns from master data set then print
school_size = school_size.groupby(['Size of Population'])['Average Reading Score','Average Math Sc
ore','% Passing Reading','% Passing Math','% Overall Passing'].max()
school_size
```

Out[193]:

	Average Reading Score	Average Math Score	% Passing Reading	% Passing Math	% Overall Passing
Size of Population					
Large (2000-5000)	83.989488	83.274201	96.539641	93.867718	95.203679
Medium (1000-2000)	83.975780	83.682222	97.308869	94.133477	95.586652
Small (<1000)	84.044699	83.839917	96.252927	94.594595	95.270270

Scores by School Type

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

In [210]:

```
#create copy of master school data (school_copy) to manipulate new data then print
school_type=school_copy.copy()
#school_type
```

In [212]:

```
#use groupby function to add columns from master data set then print
school_type = school_type.groupby(['School Type'])['Average Reading Score','Average Math Score','%
Passing Reading','% Passing Math','% Overall Passing'].max()
school_type
```

Out[212]:

	Average Reading Score	Average Math Score	% Passing Reading	% Passing Math	% Overall Passing
School Type					
Charter	84.044699	83.839917	97.308869	94.594595	95.586652
District	81.182722	77.289752	81.933280	68.309602	74.306672