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 [179]:
```
# 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_df = pd.merge(student_data, school_data, how="left", on=["sc

school_data_complete_df.columns
```

Out[179]:
```
Index(['Student ID', 'student_name', 'gender', 'grade', 'school_name',
       'reading_score', 'math_score', 'School ID', 'type', 'size', 'budget'],
      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 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 [190]:
```
# calculate the total number of unique schools and unique student IDs

Total_Schools = len(school_data_complete_df["school_name"].unique())
Total_Students = len(school_data_complete_df["Student ID"].unique())

# calculate the total budget across schools
Total_Budget = school_data["budget"].sum()

# calculate the average math and reading scores
Average_Math = school_data_complete_df["math_score"].mean()
Average_Reading = school_data_complete_df["reading_score"].mean()

# filter to the number of students with >70% on math and then calculate unique st
Passing_Math = school_data_complete_df[school_data_complete_df["math_score"] >= 7
Passing_Math_Total = len(Passing_Math["Student ID"].unique())
Passing_Math_Percent = ((Passing_Math_Total/ Total_Students) * 100)

Passing_Reading = school_data_complete_df[school_data_complete_df["reading_score'
Passing_Reading_Total = len(Passing_Reading["Student ID"].unique())
Passing_Reading_Percent = ((Passing_Reading_Total/ Total_Students) * 100)

# calculate students who passed reading with math scores >70
Passing_Both = Passing_Reading[Passing_Reading["math_score"] >= 70]
Passing_Both_Total = len(Passing_Both["Student ID"].unique())
Passing_Both_Percent = ((Passing_Both_Total/ Total_Students) * 100)

# create data summary
data = {'Total Schools': [Total_Schools],
       'Total Students': [Total_Students],
       'Total Budget': [Total_Budget],
       'Average Math Score': [Average_Math],
       'Average Reading Score': [Average_Reading],
       'Percent of students passing reading': [Passing_Reading_Percent],
       'Percent of students passing math': [Passing_Math_Percent],
       'Percent of students passing both': [Passing_Both_Percent]}

# create and print dataframe
df = pd.DataFrame(data, columns = ['Total Schools', 'Total Students', 'Total Budg

df.head()
```

Out[190]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	Percent of students passing reading	Percent of students passing math	Percent of students passing both
0	15	39170	24649428	78.985371	81.87784	85.805463	74.980853	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 [181]:
```
# find the number of unique schools and group by name and type of school
unique_schools = school_data_complete_df["school_name"].unique()
grouped_schools_df = school_data_complete_df.groupby(["school_name", "type"])

# find the number of students and the average math and reading scores
student_count_per_school = grouped_schools_df["Student ID"].count()
avg_math_per_school = grouped_schools_df["math_score"].mean()
avg_read_per_school = grouped_schools_df["reading_score"].mean()

# make budget data an integer and divide the budget by the number of students
converted_school_data_budget_df = school_data_complete_df.copy()
converted_school_data_budget_df["budget"] = converted_school_data_budget_df.loc[:
budget_per_school = grouped_schools_df["budget"].mean()
budget_per_student = (grouped_schools_df["budget"].mean() / grouped_schools_df["S

# make math data an integer, find how many students passed, and divide the passir
converted_school_data_df = school_data_complete_df.copy()
converted_school_data_df["math_score"] = converted_school_data_df.loc[:, "math_sc
math_pass_df = converted_school_data_df.loc[converted_school_data_df["math_score"
math_pass_per_school = math_pass_df["Student ID"].value_counts()
grouped_school_math_pass_df = math_pass_df.groupby(['school_name'])
school_math_passes = grouped_school_math_pass_df["Student ID"].count()
percent_school_math_passes = ((school_math_passes / student_count_per_school) * 1

# make reading data an integer, find how many students passed, and divide the pas
converted_school_data_df["reading_score"] = converted_school_data_df.loc[:, "read
reading_pass_df = converted_school_data_df.loc[converted_school_data_df["reading_
reading_pass_per_school = reading_pass_df["Student ID"].value_counts()
grouped_school_reading_pass_df = reading_pass_df.groupby(['school_name'])
grouped_school_reading_pass_df["Student ID"].count()
school_reading_passes = grouped_school_reading_pass_df["Student ID"].count()
percent_school_reading_passes = ((school_reading_passes / student_count_per_schoo

# using the reading passes find how many students also passed math, and divide th
both_pass_df = reading_pass_df.loc[converted_school_data_df["math_score"] >= 70,
both_pass_per_school = both_pass_df["Student ID"].value_counts()
grouped_school_both_pass_df = both_pass_df.groupby(['school_name'])
grouped_school_both_pass_df["Student ID"].count()
school_both_passes = grouped_school_both_pass_df["Student ID"].count()
percent_school_both_passes = ((school_both_passes / student_count_per_school) * 1

# create data summary
school_summary_df = pd.DataFrame({"Number of Students": student_count_per_school,
                                  "Total school budget": budget_per_school,
                                  "Total budget per student": budget_per_student,
                                 "Average Math Per School": avg_math_per_school,
                                 "Average Reading Per School": avg_read_per_schoo
                                 "% of math passes": percent_school_math_passes,
                                 "% of reading passes": percent_school_reading_pa
                                 "Overall passing": percent_school_both_passes,})
# create and print dataframe
school_summary_df.head(100, )
```

Out[181]:

school_name	type	Number of Students	Total school budget	Total budget per student	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes
Bailey High School	District	4976	3124928	628.0	77.048432	81.033963	66.680064	81.933280
Cabrera High School	Charter	1858	1081356	582.0	83.061895	83.975780	94.133477	97.039828
Figueroa High School	District	2949	1884411	639.0	76.711767	81.158020	65.988471	80.739234
Ford High School	District	2739	1763916	644.0	77.102592	80.746258	68.309602	79.299014
Griffin High School	Charter	1468	917500	625.0	83.351499	83.816757	93.392371	97.138965
Hernandez High School	District	4635	3022020	652.0	77.289752	80.934412	66.752967	80.862999
Holden High School	Charter	427	248087	581.0	83.803279	83.814988	92.505855	96.252927
Huang High School	District	2917	1910635	655.0	76.629414	81.182722	65.683922	81.316421
Johnson High School	District	4761	3094650	650.0	77.072464	80.966394	66.057551	81.222432
Pena High School	Charter	962	585858	609.0	83.839917	84.044699	94.594595	95.945946
Rodriguez High School	District	3999	2547363	637.0	76.842711	80.744686	66.366592	80.220055
Shelton High School	Charter	1761	1056600	600.0	83.359455	83.725724	93.867121	95.854628
Thomas High School	Charter	1635	1043130	638.0	83.418349	83.848930	93.272171	97.308869
Wilson High School	Charter	2283	1319574	578.0	83.274201	83.989488	93.867718	96.539641
Wright High School	Charter	1800	1049400	583.0	83.682222	83.955000	93.333333	96.611111

Top Performing Schools (By % Overall Passing)

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

In [182]:
```
# Sort the DataFrame by the values in the "Passing" column descending to find the
school_summary_df = school_summary_df.sort_values("Overall passing", ascending=Fa

# create and print dataframe of 5
school_summary_df.head(5,)
```

Out[182]:

school_name	type	Number of Students	Total school budget	Total budget per student	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes	
Cabrera High School	Charter	1858	1081356	582.0	83.061895	83.975780	94.133477	97.039828	91
Thomas High School	Charter	1635	1043130	638.0	83.418349	83.848930	93.272171	97.308869	90
Griffin High School	Charter	1468	917500	625.0	83.351499	83.816757	93.392371	97.138965	90
Wilson High School	Charter	2283	1319574	578.0	83.274201	83.989488	93.867718	96.539641	90
Pena High School	Charter	962	585858	609.0	83.839917	84.044699	94.594595	95.945946	90

Bottom Performing Schools (By % Overall Passing)

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

In [183]:
```
# Sort the DataFrame by the values in the "Passing" column ascending to find the
school_summary_df = school_summary_df.sort_values("Overall passing", ascending=Tr

# create and print dataframe of 5
school_summary_df.head(5,)
```

Out[183]:

school_name	type	Number of Students	Total school budget	Total budget per student	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes	p
Rodriguez High School	District	3999	2547363	637.0	76.842711	80.744686	66.366592	80.220055	52.
Figueroa High School	District	2949	1884411	639.0	76.711767	81.158020	65.988471	80.739234	53.
Huang High School	District	2917	1910635	655.0	76.629414	81.182722	65.683922	81.316421	53.
Hernandez High School	District	4635	3022020	652.0	77.289752	80.934412	66.752967	80.862999	53.
Johnson High School	District	4761	3094650	650.0	77.072464	80.966394	66.057551	81.222432	53.

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 [184]:
```
# filter the dataframe for each grade
Ninth_Grade_df = converted_school_data_df.loc[converted_school_data_df["grade"] =
Tenth_Grade_df = converted_school_data_df.loc[converted_school_data_df["grade"] =
Eleventh_Grade_df = converted_school_data_df.loc[converted_school_data_df["grade"
Twelve_Grade_df = converted_school_data_df.loc[converted_school_data_df["grade"]

# group the students from each grade by their school and type
Ninth_grade_grouped_schools_df = Ninth_Grade_df.groupby(["school_name", "type"])
Tenth_grade_grouped_schools_df = Tenth_Grade_df.groupby(["school_name", "type"])
Eleventh_grade_grouped_schools_df = Eleventh_Grade_df.groupby(["school_name", "ty
Twelve_grade_grouped_schools_df = Twelve_Grade_df.groupby(["school_name", "type"]

# calculate the avergae math scores for each school's grade
Ninth_Average_Math = Ninth_grade_grouped_schools_df["math_score"].mean()
Tenth_Average_Math = Tenth_grade_grouped_schools_df["math_score"].mean()
Eleventh_Average_Math = Eleventh_grade_grouped_schools_df["math_score"].mean()
Twelve_Average_Math = Twelve_grade_grouped_schools_df["math_score"].mean()

# create data summary
grade_summary_df = pd.DataFrame({"9th Grade Math Average": Ninth_Average_Math,
                                 "10th Grade Math Average": Tenth_Average_Math,
                                 "11th Grade Math Average": Eleventh_Average_Mat
                                "12th Grade Math Average": Twelve_Average_Math,}

# print dataframe
grade_summary_df.head(100, )
```

Out[184]:

school_name	type	9th Grade Math Average	10th Grade Math Average	11th Grade Math Average	12th Grade Math Average
Bailey High School	District	77.083676	76.996772	77.515588	76.492218
Cabrera High School	Charter	83.094697	83.154506	82.765560	83.277487
Figueroa High School	District	76.403037	76.539974	76.884344	77.151369
Ford High School	District	77.361345	77.672316	76.918058	76.179963
Griffin High School	Charter	82.044010	84.229064	83.842105	83.356164
Hernandez High School	District	77.438495	77.337408	77.136029	77.186567
Holden High School	Charter	83.787402	83.429825	85.000000	82.855422
Huang High School	District	77.027251	75.908735	76.446602	77.225641
Johnson High School	District	77.187857	76.691117	77.491653	76.863248
Pena High School	Charter	83.625455	83.372000	84.328125	84.121547

school_name	type	9th Grade Math Average	10th Grade Math Average	11th Grade Math Average	12th Grade Math Average
Rodriguez High School	**District**	76.859966	76.612500	76.395626	77.690748
Shelton High School	**Charter**	83.420755	82.917411	83.383495	83.778976
Thomas High School	**Charter**	83.590022	83.087886	83.498795	83.497041
Wilson High School	**Charter**	83.085578	83.724422	83.195326	83.035794
Wright High School	**Charter**	83.264706	84.010288	83.836782	83.644986

Reading Score by Grade

- Perform the same operations as above for reading scores

In [185]:
```
# calculate the average reading scores for each school's grade
Ninth_Average_Reading = Ninth_grade_grouped_schools_df["reading_score"].mean()
Tenth_Average_Reading = Tenth_grade_grouped_schools_df["reading_score"].mean()
Eleventh_Average_Reading = Eleventh_grade_grouped_schools_df["reading_score"].mea
Twelve_Average_Reading = Twelve_grade_grouped_schools_df["reading_score"].mean()

# create data summary
grade_reading_summary_df = pd.DataFrame({"9th Grade Reading Average": Ninth_Avera
                                         "10th Grade Reading Average": Tenth_Average_Rea
                                         "11th Grade Reading Average": Eleventh_Average_
                                        "12th Grade Reading Average": Twelve_Average_Rea

# print dataframe
grade_reading_summary_df.head(100, )
```

Out[185]:

		9th Grade Reading Average	10th Grade Reading Average	11th Grade Reading Average	12th Grade Reading Average
school_name	**type**				
Bailey High School	**District**	81.303155	80.907183	80.945643	80.912451
Cabrera High School	**Charter**	83.676136	84.253219	83.788382	84.287958
Figueroa High School	**District**	81.198598	81.408912	80.640339	81.384863
Ford High School	**District**	80.632653	81.262712	80.403642	80.662338
Griffin High School	**Charter**	83.369193	83.706897	84.288089	84.013699
Hernandez High School	**District**	80.866860	80.660147	81.396140	80.857143
Holden High School	**Charter**	83.677165	83.324561	83.815534	84.698795
Huang High School	**District**	81.290284	81.512386	81.417476	80.305983
Johnson High School	**District**	81.260714	80.773431	80.616027	81.227564
Pena High School	**Charter**	83.807273	83.612000	84.335938	84.591160
Rodriguez High School	**District**	80.993127	80.629808	80.864811	80.376426
Shelton High School	**Charter**	84.122642	83.441964	84.373786	82.781671
Thomas High School	**Charter**	83.728850	84.254157	83.585542	83.831361
Wilson High School	**Charter**	83.939778	84.021452	83.764608	84.317673
Wright High School	**Charter**	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 [186]:
```
# Create the bins in which Data will be held
bins = [1, 585, 630, 645, 675]

# Create the names for the four bins
group_names = ["0 to 584","585 to 629", "630 to 644", "645 to 675"]

# add a column of the bins to the table
pd.cut(school_summary_df["Total budget per student"], bins, labels=group_names).h
school_summary_df["Budget Range"] = pd.cut(school_summary_df["Total budget per st

# Create a GroupBy object based upon budget
budget_group = school_summary_df.groupby("Budget Range")

# Find how many rows fall into each bin
print(budget_group["Number of Students"].count())

# Get the average of each column within the GroupBy object
budget_group[["Average Math Per School", "Average Reading Per School", "% of math
```

```
Budget Range
0 to 584      4
585 to 629    4
630 to 644    4
645 to 675    3
Name: Number of Students, dtype: int64
```

Out[186]:

Budget Range	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes	Overall passing
0 to 584	83.455399	83.933814	93.460096	96.610877	90.369459
585 to 629	81.899826	83.155286	87.133538	92.718205	81.418596
630 to 644	78.518855	81.624473	73.484209	84.391793	62.857656
645 to 675	76.997210	81.027843	66.164813	81.133951	53.526855

Scores by School Size

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

In [187]:
```
# Create the bins in which Data will be held
bins_2 = [1, 1000, 2000, 3000, 5000]

# Create the names for the four bins
group_names_2 = ["0 to 999","1000 to 1999", "2000 to 2999", "3000 to 5000"]

# add a column of the bins to the table
pd.cut(school_summary_df["Number of Students"], bins, labels=group_names).head()
school_summary_df["Number of Students Range"] = pd.cut(school_summary_df["Number

# Create a GroupBy object based upon "size of school"
students_group = school_summary_df.groupby("Number of Students Range")

# Find how many rows fall into each bin
print(students_group["Number of Students"].count())

# Get the average of each column within the GroupBy object
students_group[["Average Math Per School", "Average Reading Per School", "% of ma
```

```
Number of Students Range
0 to 999        2
1000 to 1999    5
2000 to 2999    4
3000 to 5000    4
Name: Number of Students, dtype: int64
```

Out[187]:

Number of Students Range	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes	Overall passing
0 to 999	83.821598	83.929843	93.550225	96.099437	89.883853
1000 to 1999	83.374684	83.864438	93.599695	96.790680	90.621535
2000 to 2999	78.429493	81.769122	73.462428	84.473577	62.897703
3000 to 5000	77.063340	80.919864	66.464293	81.059691	53.674303

Scores by School Type

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

In [188]:
```
# Create a GroupBy object based upon "View Group"
school_type_group = school_summary_df.groupby("type")

# Find how many rows fall into each bin
print(school_type_group["Number of Students"].count())

# Get the average of each column within the GroupBy object
school_type_group[["Average Math Per School", "Average Reading Per School", "% of
```

```
type
Charter     8
District    7
Name: Number of Students, dtype: int64
```

Out[188]:

type	Average Math Per School	Average Reading Per School	% of math passes	% of reading passes	Overall passing
Charter	83.473852	83.896421	93.620830	96.586489	90.432244
District	76.956733	80.966636	66.548453	80.799062	53.672208

In [ ]: