

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 [46]: # 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 Data Frames
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"])
```

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 [47]: school_data_complete.head()
```

```
Out[47]:
```

	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

```
In [67]: school_data_complete.head()
total_schools=len(school_data_complete["school_name"].unique())
total_budget=school_data_complete["budget"].unique().sum()
average_math_score=school_data_complete["math_score"].mean()
average_reading_score=school_data_complete["reading_score"].mean()
overall_pass_rate=((average_math_score+average_reading_score)/2)
total_students = school_data_complete["Student ID"].sum()
passingmath = school_data_complete.query('math_score >70')["School ID"].count()/total_students*100
passingreading = school_data_complete.query('reading_score>70')["School ID"].count()/total_students*100

data=[{'Total Schools':total_schools,'Total Students':total_students,'Total Budget':total_budget,'Average Math Score':average_math_score,'Average Reading Score':average_reading_score,'Passing Math':passingmath,'Passing Reading':passingreading,'Overall Passing Rate':overall_pass_rate}]
df = pd.DataFrame(data)
```

```
In [41]: df
```

```
Out[41]:
```

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	Passing Math	Passing Reading	Overall Passing Rate
0	[15]	[767124865]	[24649428]	[78.98537145774827]	[81.87784018381414]	[0.0036963995424656193]	[0.004236598431729885]	[80.43160582078121]

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 [51]: school_data
```

```
Out[51]:
```

	School ID	school_name	type	size	budget
0	0	Huang High School	District	2917	1910635
1	1	Figueroa High School	District	2949	1884411
2	2	Shelton High School	Charter	1761	1056600
3	3	Hernandez High School	District	4635	3022020
4	4	Griffin High School	Charter	1468	917500
5	5	Wilson High School	Charter	2283	1319574
6	6	Cabrera High School	Charter	1858	1081356
7	7	Bailey High School	District	4976	3124928
8	8	Holden High School	Charter	427	248087
9	9	Pena High School	Charter	962	585858
10	10	Wright High School	Charter	1800	1049400
11	11	Rodriguez High School	District	3999	2547363
12	12	Johnson High School	District	4761	3094650
13	13	Ford High School	District	2739	1763916
14	14	Thomas High School	Charter	1635	1043130

```
In [68]: school_names=school_data_complete["school_name"].unique()
```

```

... [61]: school_names=school_data_complete['School Name'].unique()
school_type=school_data_complete['type'].unique()
total_students=school_data_complete['Student ID'].unique().sum()
total_budget=school_data_complete['budget'].unique().sum()

data2=[{'Total Schools':total_schools,'Total Students':total_students,'Total Budget':total_budget,'Average Math Score':[average_math_score],'Average Reading Score':[average_reading_score],'Passing Math':[passingmath],'Passing Reading':[passingreading],'Overall Passing Rate':[overall_pass_rate],'School Names':[school_names],'School Type':[school_type]}]

df = pd.DataFrame(data2)

```

In [62]: df

Out[62]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	Passing Math	Passing Reading	Overall Passing Rate	School Names
0	[15]	[767124865]	[24649428]	[78.98537145774827]	[81.87784018381414]	[0.0036963995424656193]	[0.004236598431729885]	[80.43160582078121]	[Huang High School, Figueroa High School, She...]

Top Performing Schools (By Passing Rate)

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

```
top_5=df.sort_values("Overall Passing Rate", ascending = False).head().style.format(["Total Students":'{:,}', "Total School Budget":'$', ", ", "Per Student Budget ":"{:,.0f}", "% Passing Math":'{:.1%}', "% Passing Reading":'{:.1%}', "Overall Passing Rate":'{:.1%}'})
```

In [80]: top_5

Out[80]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	Passing Math	Passing Reading	Overall Passing Rate	School Names
0	[15]	[767124865]	[24649428]	[78.98537145774827]	[81.87784018381414]	[0.0036963995424656193]	[0.004236598431729885]	[80.43160582078121]	[Huang High School, Figueroa High School, She...]

Bottom Performing Schools (By Passing Rate)

- Sort and display the five worst-performing schools

```
In [85]: bottom_5 = top_5.tail()
bottom_5 = bottom_5.sort_values('Overall Passing Rate')
```

In [83]: bottom_5

Out[83]:

	Total Schools	Total Students	Total Budget	Average Math Score	Average Reading Score	Passing Math	Passing Reading	Overall Passing Rate	School Names
0	[15]	[767124865]	[24649428]	[78.98537145774827]	[81.87784018381414]	[0.0036963995424656193]	[0.004236598431729885]	[80.43160582078121]	[Huang High School, Figueroa High School, She...]

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 [4]: ninth_math=school_data_complete.loc[school_data['grade'] == '9th'].groupby('School Names')[["math_score"]].mean()
tenth_math=school_data_complete.loc[school_data['grade'] == '10th'].groupby('School Names')[["math_score"]].mean()
eleventh_math=school_data_complete.loc[school_data['grade'] == '11th'].groupby('School Names')[["math_score"]].mean()
twelfth_math=school_data_complete.loc[school_data['grade'] == '12th'].groupby('School Names')[["math_score"]].mean()
```

```
NameError: name 'school_data_complete' is not defined
Traceback (most recent call last)
<ipython-input-4-8c2a1f86d363> in <module>
----> 1 ninth_math=school_data_complete.loc[school_data['grade'] == '9th'].groupby('School Names')[["math_score"]].mea
n()
      2 tenth_math=school_data_complete.loc[school_data['grade'] == '10th'].groupby('School Names')[["math_score"]].me
an()
      3 eleventh_math=school_data_complete.loc[school_data['grade'] == '11th'].groupby('School Names')[["math_scor
e"]].mean()
      4 twelfth_math=school_data_complete.loc[school_data['grade'] == '12th'].groupby('School Names')[["math_score"]].
mean()

NameError: name 'school_data_complete' is not defined
```

Reading Score by Grade

- Perform the same operations as above for reading scores

```
In [3]: ninth_reading=school_data_complete.loc[school_data['grade'] == '9th'].groupby('School Name')[["reading_score"]].mean()
tenth_reading=school_data_complete.loc[school_data['grade'] == '10th'].groupby('School Name')[["reading_score"]].mean()
eleventh_reading=school_data_complete.loc[school_data['grade'] == '11th'].groupby('School Name')[["reading_score"]].mean()
twelfth_reading=school_data_complete.loc[school_data['grade'] == '12th'].groupby('School Name')[["reading_score"]].mean()

reading_scores = pd.DataFrame({
    "9th": ninth_reading,
    "10th": tenth_reading,
    "11th": eleventh_reading,
    "12th": twelfth_reading})
reading_scores = reading_scores[['9th', '10th', '11th', '12th']]
reading_scores.index.name = "School Name"
```

```
NameError: name 'school_data_complete' is not defined
Traceback (most recent call last)
<ipython-input-3-23576e45e6c0> in <module>
----> 1 ninth_reading = student_data.loc[student_data['grade'] == '9th'].groupby('School Name')[["reading_score"]].mean()
      2 tenth_reading = student_data.loc[student_data['grade'] == '10th'].groupby('School Name')[["reading_score"]].mean()
      3 eleventh_reading = student_data.loc[student_data['grade'] == '11th'].groupby('School Name')[["reading_score"]].mean()
      4 twelfth_reading = student_data.loc[student_data['grade'] == '12th'].groupby('School Name')[["reading_score"]].mean()
```

```

3 eleventh_reading = student_data.loc[students_df['grade'] == '11th'].groupby('school')['reading_score'].mean()
4 twelfth_reading = student_data.loc[students_df['grade'] == '12th'].groupby('school')['reading_score'].mean()
5
NameError: name 'student_data' is not defined

```

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

In [6]: # create spending bins
bins = [0, 584.999, 614.999, 644.999, 999999]
group_name = ['< $585', '$585 - 614', '$615 - 644', "> $644"]
merged_df['spending_bins'] = pd.cut(merged_df['budget']/merged_df['size'], bins, labels = group_name)

#group by spending
by_spending = merged_df.groupby('spending_bins')

-----
NameError                                 Traceback (most recent call last)
<ipython-input-6-86672bea6f7c> in <module>
      2 bins = [0, 584.999, 614.999, 644.999, 999999]
      3 group_name = ['< $585', '$585 - 614', '$615 - 644', "> $644"]
----> 4 merged_df['spending_bins'] = pd.cut(merged_df['budget']/merged_df['size'], bins, labels = group_name)
      5
      6 #group by spending

NameError: name 'pd' is not defined

```

Scores by School Size

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

```

In [ ]: # 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 [19]: 
Out[19]:
```

	Average Math Score	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing Rate
School Size					
Small (<1000)	83.821598	83.929843	93.550225	96.099437	94.824831
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 [20]: 
Out[20]:
```

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

In []: