```python
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

school_data = "schools_complete.csv"
student_data = "students_complete.csv"

# Read School and Student Data File and store into Pandas DataFrames
s_data = pd.read_csv(school_data)
s1_data = pd.read_csv(student_data)

# Combine the data into a single dataset.
school_data_c = pd.merge(s1_data, s_data, how="left",
on=["school_name", "school_name"])
school_data_c.head()
```

```
   Student ID       student_name gender grade        school_name  \
0           0       Paul Bradley      M   9th  Huang High School
1           1       Victor Smith      M  12th  Huang High School
2           2    Kevin Rodriguez      M  12th  Huang High School
3           3  Dr. Richard Scott      M  12th  Huang High School
4           4         Bonnie Ray      F   9th  Huang High School

   reading_score  math_score  School ID      type  size   budget
0             66          79          0  District  2917  1910635
1             94          61          0  District  2917  1910635
2             90          60          0  District  2917  1910635
3             67          58          0  District  2917  1910635
4             97          84          0  District  2917  1910635
```

District Summary

Calculations

1. **Calculate the total number of schools**
2. **Calculate the total number of students**
3. **Calculate the total budget**
4. **Calculate the average math score**
5. **Calculate the average reading score**
6. **Calculate the percentage of students with a passing math score (70 or greater)**
7. **Calculate the percentage of students with a passing reading score (70 or greater)**

8. **Calculate the percentage of students who passed math and reading (% Overall Passing)** value --> with calculated value -->

with calculated value -->

```
school_data_c.columns

Index(['Student ID', 'student_name', 'gender', 'grade', 'school_name',
       'reading_score', 'math_score', 'School ID', 'type', 'size',
'budget'],
      dtype='object')
```

1. **Calculate the total number of schools**

```
total_school_number=school_data_c['school_name'].nunique()
total_school_number

15
```

1. **Calculate the total number of students**

```
total_students_number=school_data_c['Student ID'].nunique()
total_students_number

39170
```

1. **Calculate the total budget**

```
total_budget=school_data_c['budget'].sum()
total_budget

82932329558
```

1. **Calculate the average math score**

```
average_math_score=school_data_c['math_score'].mean()
average_math_score

78.98537145774827
```

1. **Calculate the average reading score**

```
average_reading_score=school_data_c['reading_score'].mean()
average_reading_score

81.87784018381414
```

1. **Calculate the percentage of students with a passing math score (70 or greater)**

```
percent_passing_math=school_data_c[school_data_c['math_score']>=70]
['math_score'].count()/school_data_c['math_score'].count()*100
percent_passing_math
```

```
74.9808526933878
```

1. **Calculate the percentage of students with a passing reading score (70 or greater)**

```
percent_passing_reading=school_data_c[school_data_c['reading_score']>=
70]['reading_score'].count()/
school_data_c['reading_score'].count()*100
percent_passing_reading

85.80546336482001
```

1. **Calculate the percentage of students who passed math and reading (% Overall Passing)**

```
Overall_passing=school_data_c[(school_data_c['math_score']>=70) &
(school_data_c['reading_score']>=70)].count()['Student ID']
/school_data_c['Student ID'].count()*100
Overall_passing

65.17232575950983

# Create a DataFrame to hold the above results
summary = pd.DataFrame({
    "Total Schools": [total_school_number],
    "Total Students": [f"{total_students_number:,}"],
    "Total Budget": [f"${total_budget:,.2f}"],
    "Average Math Score": [f"{average_math_score:.6f}"],
    "Average Reading Score": [f"{average_reading_score:.5f}"],
    "% Passing Math": [f"{percent_passing_math:.6f}"],
    "% Passing Reading": [f"{percent_passing_reading:.6f}"],
    "% Overall Passing": [f"{Overall_passing:.6f}"]
})

summary

   Total Schools Total Students        Total Budget Average Math Score
\
0             15         39,170  $82,932,329,558.00          78.985371


  Average Reading Score % Passing Math % Passing Reading % Overall
Passing
0              81.87784      74.980853         85.805463
65.172326
```

School Summary

Overview

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.)

```
# Group by school name
school_name =
school_data_c.set_index('school_name').groupby(['school_name'])

# school types by school name
school_type = s_data.set_index('school_name')['type']

#  Calculate total students
total_student = school_name['Student ID'].count()

# Total school budget
total_school_budget = s_data.set_index('school_name')['budget']

# per student budget
budget_per_student = (s_data.set_index('school_name')
['budget']/s_data.set_index('school_name')['size'])

# Average Math Score
average_math_score = school_name['math_score'].mean()

# Average Reading Score
average_reading_score = school_name['reading_score'].mean()

# % Passing Math
pass_math_percent = school_data_c[school_data_c['math_score'] >=
70].groupby('school_name')['Student ID'].count()/total_student*100

# % Passing Reading
pass_read_percent = school_data_c[school_data_c['reading_score'] >=
70].groupby('school_name')['Student ID'].count()/total_student*100

# % Overall Passing (The percentage of students that passed math and
reading.
```

```
overall_pass = school_data_c[(school_data_c['reading_score'] >= 70) &
(school_data_c['math_score'] >= 70)].groupby('school_name')['Student
ID'].count()/total_student*100

school_summary = pd.DataFrame({
    "School Type": school_type,
    "Total Students": total_student,
    "Per Student Budget": budget_per_student,
    "Total School Budget": total_school_budget,
    "Average Math Score": average_math_score,
    "Average Reading Score": average_reading_score,
    '% Passing Math': pass_math_percent,
    '% Passing Reading': pass_read_percent,
    "% Overall Passing": overall_pass
})
#formatting
school_summary.style.format({'Total Students': '{:}',
                          "Total School Budget": "${:,.2f}",
                          "Per Student Budget": "${:.2f}",
                          'Average Math Score': "{:6f}",
                          'Average Reading Score': "{:6f}",
                          "% Passing Math": "{:6f}",
                          "% Passing Reading": "{:6f}"})
```

```
<pandas.io.formats.style.Styler at 0x168a98bd690>
```

Top Performing Schools (By % Overall Passing)

```
# Sort and display the top five schools by passing rate
top_perform = school_summary.sort_values("% Overall Passing",
ascending = False)
top_perform.head().style.format({'Total Students': '{:}',
                           "Total School Budget": "${:,.2f}",
                           "Per Student Budget": "${:.2f}",
                           "% Passing Math": "{:6f}",
                           "% Passing Reading": "{:6f}",
                           "% Overall Passing": "{:6f}"})
```

```
<pandas.io.formats.style.Styler at 0x168aa8d2710>
```

Bottom Performing Schools (By % Overall Passing)

```
# Sort and display the bottom five schools by passing rate
bottom_perform = top_perform.tail()
bottom_perform = bottom_perform.sort_values('% Overall Passing')
bottom_perform.style.format({'Total Students': '{: }',
                       "Total School Budget": "${:,.2f}",
                       "Per Student Budget": "${:.2f}",
                       "% Passing Math": "{:6f}",
                       "% Passing Reading": "{:6f}",
                       "% Overall Passing": "{:6f}"})
```

```
<pandas.io.formats.style.Styler at 0x168aa53ff10>
```

Math Scores by Grade

This notebook creates a table listing the average MATH score for students of each grade level (9th, 10th, 11th, 12th) at each school. We will:

1. Create a Pandas Series for each grade.
2. Group each series by school.
3. Combine the series into a DataFrame.
4. Optionally, format the displayed data for better readability.

```
#creates grade level average math scores for each school
ninth_math = s1_data.loc[s1_data['grade'] ==
'9th'].groupby('school_name')["math_score"].mean()
tenth_math = s1_data.loc[s1_data['grade'] ==
'10th'].groupby('school_name')["math_score"].mean()
eleventh_math = s1_data.loc[s1_data['grade'] ==
'11th'].groupby('school_name')["math_score"].mean()
twelfth_math = s1_data.loc[s1_data['grade'] ==
'12th'].groupby('school_name')["math_score"].mean()

math_scores = pd.DataFrame({
        "9th": ninth_math,
        "10th": tenth_math,
        "11th": eleventh_math,
        "12th": twelfth_math
})
math_scores = math_scores[['9th', '10th', '11th', '12th']]
math_scores.index.name = "School Name"

#show and format
math_scores.style.format({'9th': '{:.6f}',
                          "10th": '{:.6f}',
                          "11th": "{:.6f}",
                          "12th": "{:.6f}"})
```

```
<pandas.io.formats.style.Styler at 0x168aa9a9810>
```

Reading Score by Grade

Perform the same operations as above for reading scores

```
#creates grade level average reading scores for each school
ninth_reading = s1_data.loc[s1_data['grade'] ==
'9th'].groupby('school_name')["reading_score"].mean()
tenth_reading = s1_data.loc[s1_data['grade'] ==
'10th'].groupby('school_name')["reading_score"].mean()
eleventh_reading = s1_data.loc[s1_data['grade'] ==
'11th'].groupby('school_name')["reading_score"].mean()
twelfth_reading = s1_data.loc[s1_data['grade'] ==
'12th'].groupby('school_name')["reading_score"].mean()

#merges the reading score averages by school and grade together
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"

#format
reading_scores.style.format({'9th': '{:.6f}',
                             "10th": '{:.6f}',
                             "11th": "{:.6f}",
                             "12th": "{:.6f}"})
```

```
<pandas.io.formats.style.Styler at 0x168aa99d590>
```

School Performance by Spending Ranges

This notebook creates a table that breaks down school performances based on average Spending Ranges (Per Student). We will use 4 reasonable bins to group school spending. The table will include the following metrics for each spending range:

- Average Math Score
- Average Reading Score
- % Passing Math
- % Passing Reading
- Overall Passing Rate (Average of the above two)

```
# create spending bins


bins = [0, 584, 629, 644, 675]
group_name = ["<$584", "$585-629", "$630-644", "$645-675"]

school_data_c['spending_bins'] =
pd.cut(school_data_c['budget']/school_data_c['size'], bins, labels =
group_name)

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


avg_math = by_spending['math_score'].mean()

avg_read = by_spending['reading_score'].mean()
pass_math = school_data_c[school_data_c['math_score'] >=
70].groupby('spending_bins')['Student
ID'].count()/by_spending['Student ID'].count()*100
pass_read = school_data_c[school_data_c['reading_score'] >=
70].groupby('spending_bins')['Student
ID'].count()/by_spending['Student ID'].count()*100
overall = school_data_c[(school_data_c['reading_score'] >= 70) &
(school_data_c['math_score'] >= 70)].groupby('spending_bins')['Student
ID'].count()/by_spending['Student ID'].count()*100

# df build
scores_by_spend = pd.DataFrame({
    "Average Math Score": avg_math,
    "Average Reading Score": avg_read,
    "% Passing Math": pass_math,
    "% Passing Reading": pass_read,
    "% Overall Passing": overall

})

#reorder columns
scores_by_spend = scores_by_spend[[
    "Average Math Score",
    "Average Reading Score",
    "% Passing Math",
    "% Passing Reading",
    "% Overall Passing"
]]

scores_by_spend.index.name = "Per Student Budget"


#formating
scores_by_spend.style.format({'Average Math Score': '{:.2f}',
                              'Average Reading Score': '{:.2f}',
                              '% Passing Math': '{:.2f}',
                              '% Passing Reading':'{:.2f}',
                              '% Overall Passing': '{:.2f}'})
```

```
<pandas.io.formats.style.Styler at 0x168aaa33f50>
```

Scores by School Size

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

```
# create size bins
bins = [0, 1000, 1999,5000]
group_name = ["Small (<1000)", "Medium (1000-2000)" , "Large (2000-5000)"]
school_data_c['size_bins'] = pd.cut(school_data_c['size'], bins, labels = group_name)

#group by spending
by_size = school_data_c.groupby('size_bins')

#calculations
average_math_score = by_size['math_score'].mean()
average_reading_score = by_size['math_score'].mean()
pass_math_percent = school_data_c[school_data_c['math_score'] >= 70].groupby('size_bins')['Student ID'].count()/by_size['Student ID'].count()*100
pass_read_percent = school_data_c[school_data_c['reading_score'] >= 70].groupby('size_bins')['Student ID'].count()/by_size['Student ID'].count()*100
overall = school_data_c[(school_data_c['reading_score'] >= 70) & (school_data_c['math_score'] >= 70)].groupby('size_bins')['Student ID'].count()/by_size['Student ID'].count()*100


# df build
scores_by_size = pd.DataFrame({
    "Average Math Score": average_math_score,
    "Average Reading Score": average_reading_score,
    '% Passing Math': pass_math_percent,
    '% Passing Reading': pass_read_percent,
    '% Overall Passing': overall

})

#reorder columns
scores_by_size = scores_by_size[[
    "Average Math Score",
    "Average Reading Score",
    '% Passing Math',
    '% Passing Reading',
    '% Overall Passing'
]]

scores_by_size.index.name = "Total Students"
scores_by_size = scores_by_size.reindex(group_name)
```

```
#formating
scores_by_size.style.format({'Average Math Score': '{:.6f}',
                             'Average Reading Score': '{:.6f}',
                             '% Passing Math': '{:.6f}',
                             '% Passing Reading':'{:.6f}',
                             '% Overall Passing': '{:.6f}'})
```

```
<pandas.io.formats.style.Styler at 0x168aaa31050>
```

Scores by School Type

Perform the same operations as above, based on school type

```
# group by type of school
schoo_type = school_data_c.groupby("type")

#calculations
average_math_score = schoo_type['math_score'].mean()
average_reading_score = schoo_type['math_score'].mean()
pass_math_percent = school_data_c[school_data_c['math_score'] >=
70].groupby('type')['Student ID'].count()/schoo_type['Student
ID'].count()*100
pass_read_percent = school_data_c[school_data_c['reading_score'] >=
70].groupby('type')['Student ID'].count()/schoo_type['Student
ID'].count()*100
overall = school_data_c[(school_data_c['reading_score'] >= 70) &
(school_data_c['math_score'] >= 70)].groupby('type')['Student
ID'].count()/schoo_type['Student ID'].count()*100

# df build
scores_schoo_type = pd.DataFrame({
    "Average Math Score": average_math_score,
    "Average Reading Score": average_reading_score,
    '% Passing Math': pass_math_percent,
    '% Passing Reading': pass_read_percent,
    "% Overall Passing": overall})

#reorder columns
scores_schoo_type = scores_schoo_type[[
    "Average Math Score",
    "Average Reading Score",
    '% Passing Math',
    '% Passing Reading',
    "% Overall Passing"
]]
scores_schoo_type.index.name = "Type of School"
```

```
#formating
scores_schoo_type.style.format({'Average Math Score': '{:.6f}',
                              'Average Reading Score': '{:.6f}',
                              '% Passing Math': '{:.6f}',
                              '% Passing Reading':'{:.6f}',
                              '% Overall Passing': '{:.6f}'})
```

```
<pandas.io.formats.style.Styler at 0x168aaa3b910>
```