

PyCitySchools

May 25, 2024

1 PyCity Schools Analysis

•

1.1 Data Analysis By Thay Chansy

```
[61]: # Dependencies and Setup
import pandas as pd
from pathlib import Path

# File to Load (Remember to Change These)
school_data_to_load = Path("Resources/schools_complete.csv")
student_data_to_load = Path("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 = pd.merge(student_data, school_data, how="left",
    on=["school_name", "school_name"])
school_data_complete.head()
```

```
[61]: 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
```

1.2 District Summary

```
[62]: # Calculate the total number of unique schools
school_count = school_data_complete['school_name'].nunique()
school_count
```

[62]: 15

```
[63]: # Calculate the total number of students
student_count = school_data_complete['Student ID'].nunique()
student_count
```

[63]: 39170

```
[64]: # Calculate the total budget
total_budget = school_data['budget'].sum()
total_budget
```

[64]: 24649428

```
[65]: # Calculate the average (mean) math score
average_math_score = student_data['math_score'].mean()
average_math_score
```

[65]: 78.98537145774827

```
[66]: # Calculate the average (mean) reading score
average_reading_score = student_data['reading_score'].mean()
average_reading_score
```

[66]: 81.87784018381414

```
[67]: # Use the following to calculate the percentage of students who passed math
      ↪ (math scores greater than or equal to 70)
passing_math_count = school_data_complete[(school_data_complete["math_score"]
      ↪ >= 70)].count()["student_name"]
passing_math_percentage = passing_math_count / float(student_count) * 100
passing_math_percentage
```

[67]: 74.9808526933878

```
[68]: # Calculate the percentage of students who passed reading (hint: look at how
      ↪ the math percentage was calculated)
passing_reading_count =
      ↪ school_data_complete[(school_data_complete["reading_score"] >= 70)].
      ↪ count()["student_name"]
passing_reading_percentage = passing_reading_count / float(student_count) * 100
passing_reading_percentage
```

```
[68]: 85.80546336482001
```

```
[69]: # Use the following to calculate the percentage of students that passed math
      ↪and reading
      passing_math_reading_count = school_data_complete[
          (school_data_complete["math_score"] >= 70) &
          ↪(school_data_complete["reading_score"] >= 70)
      ].count()["student_name"]
      overall_passing_rate = passing_math_reading_count / float(student_count) * 100
      overall_passing_rate
```

```
[69]: 65.17232575950983
```

```
[70]: # Create a high-level snapshot of the district's key metrics in a DataFrame
      district_summary = pd.DataFrame([
          "Total Schools": school_count,
          "Total Students": student_count,
          "Total Budget": total_budget,
          "Average Math Score": average_math_score,
          "Average Reading Score": average_reading_score,
          "% Passing Math": passing_math_percentage,
          "% Passing Reading": passing_reading_percentage,
          "% Overall Passing": overall_passing_rate
      ])

      # Formatting
      district_summary["Total Students"] = district_summary["Total Students"].map("{:,.2f}".format)
      district_summary["Total Budget"] = district_summary["Total Budget"].map("${:,.2f}".format)
      district_summary["Average Math Score"] = district_summary["Average Math Score"].map("{:.2f}".format)
      district_summary["Average Reading Score"] = district_summary["Average Reading Score"].map("{:.2f}".format)
      district_summary["% Passing Math"] = district_summary["% Passing Math"].map("{:,.2f}".format)
      district_summary["% Passing Reading"] = district_summary["% Passing Reading"].map("{:,.2f}".format)
      district_summary["% Overall Passing"] = district_summary["% Overall Passing"].map("{:,.2f}".format)

      # Display the DataFrame
      district_summary
```

```
[70]:   Total Schools Total Students   Total Budget Average Math Score \
      0           15         39,170  $24,649,428.00           78.99
```

	Average Reading Score	% Passing Math	% Passing Reading	% Overall Passing
0	81.88	74.98	85.81	65.17

1.3 School Summary

```
[71]: # Use the code provided to select the type per school from school_data
school_type = school_data.set_index(["school_name"])["type"]

#Display the Dataframe
school_type
```

```
[71]: school_name
Huang High School      District
Figueroa High School   District
Shelton High School     Charter
Hernandez High School  District
Griffin High School    Charter
Wilson High School     Charter
Cabrera High School    Charter
Bailey High School     District
Holden High School     Charter
Pena High School       Charter
Wright High School     Charter
Rodriguez High School  District
Johnson High School   District
Ford High School       District
Thomas High School     Charter
Name: type, dtype: object
```

```
[72]: # Calculate the total student count per school from school_data
per_school_counts = school_data_complete.groupby('school_name')['Student ID'].
    .count()

#Display the Dataframe
per_school_counts
```

```
[72]: school_name
Bailey High School      4976
Cabrera High School     1858
Figueroa High School    2949
Ford High School        2739
Griffin High School     1468
Hernandez High School   4635
Holden High School       427
Huang High School       2917
Johnson High School     4761
Pena High School         962
```

Rodriguez High School	3999
Shelton High School	1761
Thomas High School	1635
Wilson High School	2283
Wright High School	1800

Name: Student ID, dtype: int64

```
[73]: # Calculate the total school budget and per capita spending per school from
      ↪ school_data
per_school_budget = school_data.groupby('school_name')['budget'].sum()

#Display the Dataframe
per_school_budget

per_school_capita = per_school_budget / per_school_counts

#Display the Dataframe
per_school_capita
```

```
[73]: school_name
Bailey High School      628.0
Cabrera High School     582.0
Figueroa High School    639.0
Ford High School        644.0
Griffin High School     625.0
Hernandez High School   652.0
Holden High School      581.0
Huang High School       655.0
Johnson High School    650.0
Pena High School        609.0
Rodriguez High School   637.0
Shelton High School     600.0
Thomas High School      638.0
Wilson High School      578.0
Wright High School      583.0
dtype: float64
```

```
[74]: # Calculate the average test scores per school from school_data_complete
per_school_math = school_data_complete.groupby('school_name')['math_score'].
      ↪ mean()

#Display the Dataframe
per_school_math

per_school_reading = school_data_complete.
      ↪ groupby('school_name')['reading_score'].mean()
```

```
#Display the Dataframe
per_school_reading
```

```
[74]: school_name
      Bailey High School      81.033963
      Cabrera High School    83.975780
      Figueroa High School   81.158020
      Ford High School       80.746258
      Griffin High School    83.816757
      Hernandez High School   80.934412
      Holden High School     83.814988
      Huang High School      81.182722
      Johnson High School    80.966394
      Pena High School       84.044699
      Rodriguez High School  80.744686
      Shelton High School    83.725724
      Thomas High School     83.848930
      Wilson High School     83.989488
      Wright High School     83.955000
      Name: reading_score, dtype: float64
```

```
[75]: # Calculate the number of students per school with math scores of 70 or higher
      ↪ from school_data_complete
      students_passing_math = student_data[student_data['math_score'] >=
      ↪ 70]['student_name'].count()

      school_students_passing_math =
      ↪ (school_data_complete[school_data_complete['math_score'] >= 70].
      ↪ groupby('school_name')['math_score'].count())

      #Display the Dataframe
      school_students_passing_math
```

```
[75]: school_name
      Bailey High School      3318
      Cabrera High School     1749
      Figueroa High School    1946
      Ford High School        1871
      Griffin High School     1371
      Hernandez High School    3094
      Holden High School       395
      Huang High School       1916
      Johnson High School     3145
      Pena High School         910
      Rodriguez High School   2654
      Shelton High School     1653
      Thomas High School      1525
```

```
Wilson High School      2143
Wright High School     1680
Name: math_score, dtype: int64
```

```
[76]: # Calculate the number of students per school with reading scores of 70 or
      ↪ higher from school_data_complete
students_passing_reading = student_data[student_data['reading_score'] >=
      ↪ 70]['student_name'].count()
school_students_passing_reading =
      ↪ (school_data_complete[school_data_complete['reading_score'] >= 70].
      ↪ groupby('school_name')['reading_score'].count())

#Display the Dataframe
school_students_passing_reading
```

```
[76]: school_name
Bailey High School      4077
Cabrera High School     1803
Figueroa High School    2381
Ford High School        2172
Griffin High School     1426
Hernandez High School   3748
Holden High School       411
Huang High School       2372
Johnson High School     3867
Pena High School         923
Rodriguez High School    3208
Shelton High School     1688
Thomas High School      1591
Wilson High School      2204
Wright High School      1739
Name: reading_score, dtype: int64
```

```
[77]: # Use the provided code to calculate the number of students per school that
      ↪ passed both math and reading with scores of 70 or higher
students_passing_math_and_reading = school_data_complete[
      (school_data_complete['reading_score'] >= 70) &
      ↪ (school_data_complete['math_score'] >= 70)
]
school_students_passing_math_and_reading = students_passing_math_and_reading.
      ↪ groupby(['school_name']).size()

students_passing_math_and_reading.groupby('school_name')['Student ID'].count()
```

```
[77]: school_name
Bailey High School      2719
Cabrera High School     1697
```

Figueroa High School	1569
Ford High School	1487
Griffin High School	1330
Hernandez High School	2481
Holden High School	381
Huang High School	1561
Johnson High School	2549
Pena High School	871
Rodriguez High School	2119
Shelton High School	1583
Thomas High School	1487
Wilson High School	2068
Wright High School	1626

Name: Student ID, dtype: int64

```
[78]: # Use the provided code to calculate the passing rates
per_school_passing_math = school_students_passing_math / per_school_counts * 100
per_school_passing_reading = school_students_passing_reading /
    ↪per_school_counts * 100
overall_passing_rate = school_students_passing_math_and_reading /
    ↪per_school_counts * 100

#Display the Dataframe
overall_passing_rate
```

```
[78]: school_name
Bailey High School      54.642283
Cabrera High School     91.334769
Figueroa High School    53.204476
Ford High School        54.289887
Griffin High School     90.599455
Hernandez High School   53.527508
Holden High School      89.227166
Huang High School       53.513884
Johnson High School     53.539172
Pena High School        90.540541
Rodriguez High School   52.988247
Shelton High School     89.892107
Thomas High School      90.948012
Wilson High School      90.582567
Wright High School      90.333333
dtype: float64
```

```
[79]: # Create a DataFrame called `per_school_summary` with columns for the
    ↪calculations above.
per_school_summary = pd.DataFrame({
    "School Type": school_type,
```



```

    "Total Students": per_school_counts,
    "Total School Budget": per_school_budget,
    "Per Student Budget": per_school_capita,
    "Average Math Score": per_school_math,
    "Average Reading Score": per_school_reading,
    "% Passing Math": per_school_passing_math,
    "% Passing Reading": per_school_passing_reading,
    "% Overall Passing": overall_passing_rate})

# Formatting
per_school_summary["Total School Budget"] = per_school_summary["Total School_Budget"].map("${:,.2f}".format)
per_school_summary["Per Student Budget"] = per_school_summary["Per Student_Budget"].map("${:,.2f}".format)

# Display the DataFrame
per_school_summary

```

```

[79]:
      School Type  Total Students  Total School Budget \
school_name
Bailey High School      District      4976      $3,124,928.00
Cabrera High School      Charter      1858      $1,081,356.00
Figueroa High School      District      2949      $1,884,411.00
Ford High School      District      2739      $1,763,916.00
Griffin High School      Charter      1468      $917,500.00
Hernandez High School      District      4635      $3,022,020.00
Holden High School      Charter      427      $248,087.00
Huang High School      District      2917      $1,910,635.00
Johnson High School      District      4761      $3,094,650.00
Pena High School      Charter      962      $585,858.00
Rodriguez High School      District      3999      $2,547,363.00
Shelton High School      Charter      1761      $1,056,600.00
Thomas High School      Charter      1635      $1,043,130.00
Wilson High School      Charter      2283      $1,319,574.00
Wright High School      Charter      1800      $1,049,400.00

```

```

      Per Student Budget  Average Math Score \
school_name
Bailey High School      $628.00      77.048432
Cabrera High School      $582.00      83.061895
Figueroa High School      $639.00      76.711767
Ford High School      $644.00      77.102592
Griffin High School      $625.00      83.351499
Hernandez High School      $652.00      77.289752
Holden High School      $581.00      83.803279
Huang High School      $655.00      76.629414
Johnson High School      $650.00      77.072464

```

Pena High School	\$609.00	83.839917
Rodriguez High School	\$637.00	76.842711
Shelton High School	\$600.00	83.359455
Thomas High School	\$638.00	83.418349
Wilson High School	\$578.00	83.274201
Wright High School	\$583.00	83.682222

school_name	Average Reading Score	% Passing Math \
Bailey High School	81.033963	66.680064
Cabrera High School	83.975780	94.133477
Figueroa High School	81.158020	65.988471
Ford High School	80.746258	68.309602
Griffin High School	83.816757	93.392371
Hernandez High School	80.934412	66.752967
Holden High School	83.814988	92.505855
Huang High School	81.182722	65.683922
Johnson High School	80.966394	66.057551
Pena High School	84.044699	94.594595
Rodriguez High School	80.744686	66.366592
Shelton High School	83.725724	93.867121
Thomas High School	83.848930	93.272171
Wilson High School	83.989488	93.867718
Wright High School	83.955000	93.333333

school_name	% Passing Reading	% Overall Passing
Bailey High School	81.933280	54.642283
Cabrera High School	97.039828	91.334769
Figueroa High School	80.739234	53.204476
Ford High School	79.299014	54.289887
Griffin High School	97.138965	90.599455
Hernandez High School	80.862999	53.527508
Holden High School	96.252927	89.227166
Huang High School	81.316421	53.513884
Johnson High School	81.222432	53.539172
Pena High School	95.945946	90.540541
Rodriguez High School	80.220055	52.988247
Shelton High School	95.854628	89.892107
Thomas High School	97.308869	90.948012
Wilson High School	96.539641	90.582567
Wright High School	96.611111	90.333333

1.4 Highest-Performing Schools (by % Overall Passing)

```
[80]: # Sort the schools by `% Overall Passing` in descending order and display the
      ↪top 5 rows.
      top_schools = per_school_summary.sort_values(['% Overall Passing'],ascending =
      ↪False)
      top_schools.head(5)
```

```
[80]:
```

	School Type	Total Students	Total School Budget	\
school_name				
Cabrera High School	Charter	1858	\$1,081,356.00	
Thomas High School	Charter	1635	\$1,043,130.00	
Griffin High School	Charter	1468	\$917,500.00	
Wilson High School	Charter	2283	\$1,319,574.00	
Pena High School	Charter	962	\$585,858.00	

	Per Student Budget	Average Math Score	\
school_name			
Cabrera High School	\$582.00	83.061895	
Thomas High School	\$638.00	83.418349	
Griffin High School	\$625.00	83.351499	
Wilson High School	\$578.00	83.274201	
Pena High School	\$609.00	83.839917	

	Average Reading Score	% Passing Math	% Passing Reading	\
school_name				
Cabrera High School	83.975780	94.133477	97.039828	
Thomas High School	83.848930	93.272171	97.308869	
Griffin High School	83.816757	93.392371	97.138965	
Wilson High School	83.989488	93.867718	96.539641	
Pena High School	84.044699	94.594595	95.945946	

	% Overall Passing
school_name	
Cabrera High School	91.334769
Thomas High School	90.948012
Griffin High School	90.599455
Wilson High School	90.582567
Pena High School	90.540541

1.5 Bottom Performing Schools (By % Overall Passing)

```
[81]: # Sort the schools by `% Overall Passing` in ascending order and display the
      ↪top 5 rows.
      bottom_schools = per_school_summary.sort_values(['% Overall Passing'],ascending
      ↪= True)
      bottom_schools.head(5)
```

```
[81]:
```

	School Type	Total Students	Total School Budget \
school_name			
Rodriguez High School	District	3999	\$2,547,363.00
Figueroa High School	District	2949	\$1,884,411.00
Huang High School	District	2917	\$1,910,635.00
Hernandez High School	District	4635	\$3,022,020.00
Johnson High School	District	4761	\$3,094,650.00

	Per Student Budget	Average Math Score \
school_name		
Rodriguez High School	\$637.00	76.842711
Figueroa High School	\$639.00	76.711767
Huang High School	\$655.00	76.629414
Hernandez High School	\$652.00	77.289752
Johnson High School	\$650.00	77.072464

	Average Reading Score	% Passing Math \
school_name		
Rodriguez High School	80.744686	66.366592
Figueroa High School	81.158020	65.988471
Huang High School	81.182722	65.683922
Hernandez High School	80.934412	66.752967
Johnson High School	80.966394	66.057551

	% Passing Reading	% Overall Passing
school_name		
Rodriguez High School	80.220055	52.988247
Figueroa High School	80.739234	53.204476
Huang High School	81.316421	53.513884
Hernandez High School	80.862999	53.527508
Johnson High School	81.222432	53.539172

1.6 Math Scores by Grade

```
[82]: # Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] ==
↪ "11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] ==
↪ "12th")]

# Group by `school_name` and take the mean of the `math_score` column for each.
ninth_grade_math_scores = ninth_graders.groupby(["school_name"])["math_score"].
↪ mean()
tenth_grade_math_scores = tenth_graders.groupby(["school_name"])["math_score"].
↪ mean()
```

```

eleventh_grade_math_scores = eleventh_graders.
    ↪groupby(["school_name"])["math_score"].mean()
twelfth_grade_math_scores = twelfth_graders.
    ↪groupby(["school_name"])["math_score"].mean()

# Combine each of the scores above into single DataFrame called ↪
    ↪`math_scores_by_grade`
math_scores_by_grade = pd.DataFrame({
    "9th": ninth_grade_math_scores,
    "10th": tenth_grade_math_scores,
    "11th": eleventh_grade_math_scores,
    "12th": twelfth_grade_math_scores})

# Minor data wrangling
math_scores_by_grade.index.name = None

# Display the DataFrame
math_scores_by_grade

```

```
[82]:
```

	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

1.7 Reading Score by Grade

```

[83]: # Use the code provided to separate the data by grade
ninth_graders = school_data_complete[(school_data_complete["grade"] == "9th")]
tenth_graders = school_data_complete[(school_data_complete["grade"] == "10th")]
eleventh_graders = school_data_complete[(school_data_complete["grade"] == ↪
    ↪"11th")]
twelfth_graders = school_data_complete[(school_data_complete["grade"] == ↪
    ↪"12th")]

```

```

# Group by `school_name` and take the mean of the the `reading_score` column
↳ for each.
ninth_grade_reading_scores = ninth_graders.
    ↳ groupby(["school_name"])["reading_score"].mean()
tenth_grade_reading_scores = tenth_graders.
    ↳ groupby(["school_name"])["reading_score"].mean()
eleventh_grade_reading_scores = eleventh_graders.
    ↳ groupby(["school_name"])["reading_score"].mean()
twelfth_grade_reading_scores = twelfth_graders.
    ↳ groupby(["school_name"])["reading_score"].mean()

# Combine each of the scores above into single DataFrame called
↳ `reading_scores_by_grade`
reading_scores_by_grade = pd.DataFrame({
    "9th": ninth_grade_reading_scores,
    "10th": tenth_grade_reading_scores,
    "11th": eleventh_grade_reading_scores,
    "12th": twelfth_grade_reading_scores})

# Minor data wrangling
reading_scores_by_grade = reading_scores_by_grade[["9th", "10th", "11th",
↳ "12th"]]
reading_scores_by_grade.index.name = None

# Display the DataFrame
reading_scores_by_grade

```

```

[83]:

```

	9th	10th	11th	12th
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

1.8 Scores by School Spending

```
[84]: # Establish the bins
      spending_bins = [0, 585, 630, 645, 680]
      labels = ["<$585", "$585-630", "$630-645", "$645-680"]

[85]: # Create a copy of the school summary since it has the "Per Student Budget"
      school_spending_df = per_school_summary.copy()

[86]: # Use `pd.cut` to categorize spending based on the bins.
      school_spending_df["Spending Ranges (Per Student)"] = pd.cut(per_school_capita,
      ↪spending_bins, labels=labels)

      #Display the Dataframe
      school_spending_df
```

```
[86]:
```

	School Type	Total Students	Total School Budget \
school_name			
Bailey High School	District	4976	\$3,124,928.00
Cabrera High School	Charter	1858	\$1,081,356.00
Figueroa High School	District	2949	\$1,884,411.00
Ford High School	District	2739	\$1,763,916.00
Griffin High School	Charter	1468	\$917,500.00
Hernandez High School	District	4635	\$3,022,020.00
Holden High School	Charter	427	\$248,087.00
Huang High School	District	2917	\$1,910,635.00
Johnson High School	District	4761	\$3,094,650.00
Pena High School	Charter	962	\$585,858.00
Rodriguez High School	District	3999	\$2,547,363.00
Shelton High School	Charter	1761	\$1,056,600.00
Thomas High School	Charter	1635	\$1,043,130.00
Wilson High School	Charter	2283	\$1,319,574.00
Wright High School	Charter	1800	\$1,049,400.00

	Per Student Budget	Average Math Score \
school_name		
Bailey High School	\$628.00	77.048432
Cabrera High School	\$582.00	83.061895
Figueroa High School	\$639.00	76.711767
Ford High School	\$644.00	77.102592
Griffin High School	\$625.00	83.351499
Hernandez High School	\$652.00	77.289752
Holden High School	\$581.00	83.803279
Huang High School	\$655.00	76.629414
Johnson High School	\$650.00	77.072464
Pena High School	\$609.00	83.839917
Rodriguez High School	\$637.00	76.842711

Shelton High School	\$600.00	83.359455
Thomas High School	\$638.00	83.418349
Wilson High School	\$578.00	83.274201
Wright High School	\$583.00	83.682222

	Average Reading Score	% Passing Math \
school_name		
Bailey High School	81.033963	66.680064
Cabrera High School	83.975780	94.133477
Figueroa High School	81.158020	65.988471
Ford High School	80.746258	68.309602
Griffin High School	83.816757	93.392371
Hernandez High School	80.934412	66.752967
Holden High School	83.814988	92.505855
Huang High School	81.182722	65.683922
Johnson High School	80.966394	66.057551
Pena High School	84.044699	94.594595
Rodriguez High School	80.744686	66.366592
Shelton High School	83.725724	93.867121
Thomas High School	83.848930	93.272171
Wilson High School	83.989488	93.867718
Wright High School	83.955000	93.333333

	% Passing Reading	% Overall Passing \
school_name		
Bailey High School	81.933280	54.642283
Cabrera High School	97.039828	91.334769
Figueroa High School	80.739234	53.204476
Ford High School	79.299014	54.289887
Griffin High School	97.138965	90.599455
Hernandez High School	80.862999	53.527508
Holden High School	96.252927	89.227166
Huang High School	81.316421	53.513884
Johnson High School	81.222432	53.539172
Pena High School	95.945946	90.540541
Rodriguez High School	80.220055	52.988247
Shelton High School	95.854628	89.892107
Thomas High School	97.308869	90.948012
Wilson High School	96.539641	90.582567
Wright High School	96.611111	90.333333

	Spending Ranges (Per Student)
school_name	
Bailey High School	\$585-630
Cabrera High School	<\$585
Figueroa High School	\$630-645
Ford High School	\$630-645

Griffin High School	\$585-630
Hernandez High School	\$645-680
Holden High School	<\$585
Huang High School	\$645-680
Johnson High School	\$645-680
Pena High School	\$585-630
Rodriguez High School	\$630-645
Shelton High School	\$585-630
Thomas High School	\$630-645
Wilson High School	<\$585
Wright High School	<\$585

```
[87]: # Calculate averages for the desired columns.
spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per_
↪Student)"])[["Average Math Score"]].mean()
spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per_
↪Student)"])[["Average Reading Score"]].mean()
spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per_
↪Student)"])[["% Passing Math"]].mean()
spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per_
↪Student)"])[["% Passing Reading"]].mean()
overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per_
↪Student)"])[["% Overall Passing"]].mean()
```

```
C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2524176425.py:2:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
    spending_math_scores = school_spending_df.groupby(["Spending Ranges (Per
Student)"])[["Average Math Score"]].mean()
C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2524176425.py:3:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
    spending_reading_scores = school_spending_df.groupby(["Spending Ranges (Per
Student)"])[["Average Reading Score"]].mean()
C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2524176425.py:4:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
    spending_passing_math = school_spending_df.groupby(["Spending Ranges (Per
Student)"])[["% Passing Math"]].mean()
C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2524176425.py:5:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
    spending_passing_reading = school_spending_df.groupby(["Spending Ranges (Per
```

```
Student)"])[ "% Passing Reading"].mean()
C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2524176425.py:6:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
    overall_passing_spending = school_spending_df.groupby(["Spending Ranges (Per
Student)"])[ "% Overall Passing"].mean()
```

```
[88]: # Assemble into DataFrame
spending_summary = pd.DataFrame({
    "Average Math Score": spending_math_scores,
    "Average Reading Score": spending_reading_scores,
    "% Passing Math": spending_passing_math,
    "% Passing Reading": spending_passing_reading,
    "% Overall Passing": overall_passing_spending
})

# Display results
spending_summary
```

```
[88]:
```

Spending Ranges (Per Student)	Average Math Score	Average Reading Score \
<\$585	83.455399	83.933814
\$585-630	81.899826	83.155286
\$630-645	78.518855	81.624473
\$645-680	76.997210	81.027843

Spending Ranges (Per Student)	% Passing Math	% Passing Reading \
<\$585	93.460096	96.610877
\$585-630	87.133538	92.718205
\$630-645	73.484209	84.391793
\$645-680	66.164813	81.133951

Spending Ranges (Per Student)	% Overall Passing
<\$585	90.369459
\$585-630	81.418596
\$630-645	62.857656
\$645-680	53.526855

1.9 Scores by School Size

```
[89]: # Establish the bins.
size_bins = [0, 1000, 2000, 5000]
labels = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

```
[90]: # Categorize the spending based on the bins
# Use `pd.cut` on the "Total Students" column of the `per_school_summary`
↳DataFrame.

per_school_summary["School Size"] = pd.cut(per_school_summary['Total_
↳Students'], size_bins, labels=labels)

#Display results
per_school_summary
```

```
[90]:
```

	School Type	Total Students	Total School Budget \
school_name			
Bailey High School	District	4976	\$3,124,928.00
Cabrera High School	Charter	1858	\$1,081,356.00
Figueroa High School	District	2949	\$1,884,411.00
Ford High School	District	2739	\$1,763,916.00
Griffin High School	Charter	1468	\$917,500.00
Hernandez High School	District	4635	\$3,022,020.00
Holden High School	Charter	427	\$248,087.00
Huang High School	District	2917	\$1,910,635.00
Johnson High School	District	4761	\$3,094,650.00
Pena High School	Charter	962	\$585,858.00
Rodriguez High School	District	3999	\$2,547,363.00
Shelton High School	Charter	1761	\$1,056,600.00
Thomas High School	Charter	1635	\$1,043,130.00
Wilson High School	Charter	2283	\$1,319,574.00
Wright High School	Charter	1800	\$1,049,400.00

	Per Student Budget	Average Math Score \
school_name		
Bailey High School	\$628.00	77.048432
Cabrera High School	\$582.00	83.061895
Figueroa High School	\$639.00	76.711767
Ford High School	\$644.00	77.102592
Griffin High School	\$625.00	83.351499
Hernandez High School	\$652.00	77.289752
Holden High School	\$581.00	83.803279
Huang High School	\$655.00	76.629414
Johnson High School	\$650.00	77.072464
Pena High School	\$609.00	83.839917
Rodriguez High School	\$637.00	76.842711
Shelton High School	\$600.00	83.359455
Thomas High School	\$638.00	83.418349
Wilson High School	\$578.00	83.274201
Wright High School	\$583.00	83.682222

Average Reading Score % Passing Math \

school_name		
Bailey High School	81.033963	66.680064
Cabrera High School	83.975780	94.133477
Figueroa High School	81.158020	65.988471
Ford High School	80.746258	68.309602
Griffin High School	83.816757	93.392371
Hernandez High School	80.934412	66.752967
Holden High School	83.814988	92.505855
Huang High School	81.182722	65.683922
Johnson High School	80.966394	66.057551
Pena High School	84.044699	94.594595
Rodriguez High School	80.744686	66.366592
Shelton High School	83.725724	93.867121
Thomas High School	83.848930	93.272171
Wilson High School	83.989488	93.867718
Wright High School	83.955000	93.333333

	% Passing Reading	% Overall Passing \
school_name		
Bailey High School	81.933280	54.642283
Cabrera High School	97.039828	91.334769
Figueroa High School	80.739234	53.204476
Ford High School	79.299014	54.289887
Griffin High School	97.138965	90.599455
Hernandez High School	80.862999	53.527508
Holden High School	96.252927	89.227166
Huang High School	81.316421	53.513884
Johnson High School	81.222432	53.539172
Pena High School	95.945946	90.540541
Rodriguez High School	80.220055	52.988247
Shelton High School	95.854628	89.892107
Thomas High School	97.308869	90.948012
Wilson High School	96.539641	90.582567
Wright High School	96.611111	90.333333

	School Size
school_name	
Bailey High School	Large (2000-5000)
Cabrera High School	Medium (1000-2000)
Figueroa High School	Large (2000-5000)
Ford High School	Large (2000-5000)
Griffin High School	Medium (1000-2000)
Hernandez High School	Large (2000-5000)
Holden High School	Small (<1000)
Huang High School	Large (2000-5000)
Johnson High School	Large (2000-5000)
Pena High School	Small (<1000)

Rodriguez High School	Large (2000-5000)
Shelton High School	Medium (1000-2000)
Thomas High School	Medium (1000-2000)
Wilson High School	Large (2000-5000)
Wright High School	Medium (1000-2000)

```
[91]: # Calculate averages for the desired columns.
size_math_scores = per_school_summary.groupby(["School Size"])["Average Math_
↳Score"].mean()
size_reading_scores = per_school_summary.groupby(["School Size"])["Average_
↳Reading Score"].mean()
size_passing_math = per_school_summary.groupby(["School Size"])["% Passing_
↳Math"].mean()
size_passing_reading = per_school_summary.groupby(["School Size"])["% Passing_
↳Reading"].mean()
size_overall_passing = per_school_summary.groupby(["School Size"])["% Overall_
↳Passing"].mean()
```

C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2944522441.py:2:

FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_math_scores = per_school_summary.groupby(["School Size"])["Average Math
Score"].mean()
```

C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2944522441.py:3:

FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_reading_scores = per_school_summary.groupby(["School Size"])["Average
Reading Score"].mean()
```

C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2944522441.py:4:

FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_passing_math = per_school_summary.groupby(["School Size"])["% Passing
Math"].mean()
```

C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2944522441.py:5:

FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_passing_reading = per_school_summary.groupby(["School Size"])["% Passing
Reading"].mean()
```

C:\Users\thayc\AppData\Local\Temp\ipykernel_25004\2944522441.py:6:

FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
size_overall_passing = per_school_summary.groupby(["School Size"])["% Overall
```

```
Passing"].mean()
```

```
[92]: # Create a DataFrame called `size_summary` that breaks down school performance
      ↪ based on school size (small, medium, or large).
      # Use the scores above to create a new DataFrame called `size_summary`
      size_summary = pd.DataFrame({
          "Average Math Score": size_math_scores,
          "Average Reading Score": size_reading_scores,
          "% Passing Math": size_passing_math,
          "% Passing Reading": size_passing_reading,
          "% Overall Passing": size_overall_passing
      })

      # Display results
      size_summary
```

```
[92]:
```

	Average Math Score	Average Reading Score	% Passing Math	\
School Size				
Small (<1000)	83.821598	83.929843	93.550225	
Medium (1000-2000)	83.374684	83.864438	93.599695	
Large (2000-5000)	77.746417	81.344493	69.963361	

	% Passing Reading	% Overall Passing
School Size		
Small (<1000)	96.099437	89.883853
Medium (1000-2000)	96.790680	90.621535
Large (2000-5000)	82.766634	58.286003

1.10 Scores by School Type

```
[93]: # Group the per_school_summary DataFrame by "School Type" and average the
      ↪ results.
      average_math_score_by_type = per_school_summary.groupby(["School_
      ↪ Type"])["Average Math Score"].mean()
      average_reading_score_by_type = per_school_summary.groupby(["School_
      ↪ Type"])["Average Reading Score"].mean()
      average_percent_passing_math_by_type = per_school_summary.groupby(["School_
      ↪ Type"])["% Passing Math"].mean()
      average_percent_passing_reading_by_type = per_school_summary.groupby(["School_
      ↪ Type"])["% Passing Reading"].mean()
      average_percent_overall_passing_by_type = per_school_summary.groupby(["School_
      ↪ Type"])["% Overall Passing"].mean()

[94]: # Assemble the new data by type into a DataFrame called `type_summary`
      type_summary = pd.DataFrame({
          "Average Math Score": average_math_score_by_type,
          "Average Reading Score": average_reading_score_by_type,
```

```

"% Passing Math": average_percent_passing_math_by_type,
"% Passing Reading": average_percent_passing_reading_by_type,
"% Overall Passing": average_percent_overall_passing_by_type
})

```

```

# Display results
type_summary

```

```

[94]:
      Average Math Score  Average Reading Score  % Passing Math  \
School Type
Charter                83.473852             83.896421         93.620830
District              76.956733             80.966636         66.548453

      % Passing Reading  % Overall Passing
School Type
Charter                96.586489          90.432244
District              80.799062          53.672208

```