

In this assignment, you'll create and manipulate Pandas DataFrames to analyze school and standardized test data.

Background

You are the new Chief Data Scientist for your city's school district. In this capacity, you'll be helping the school board and mayor make strategic decisions regarding future school budgets and priorities.

As a first task, you've been asked to analyze the district-wide standardized test results. You'll be given access to every student's math and reading scores, as well as various information on the schools they attend. Your task is to aggregate the data to showcase obvious trends in school performance.

Before You Begin

Create a new repository for this project called pandas-challenge. **Do not add this homework to an existing repository.**

Clone the new repository to your computer.

Inside your local Git repository, create a folder for this homework assignment and name it PyCitySchools.

Add your Jupyter notebook to this folder. This will be the main script to run for analysis.

Push these changes to GitHub or GitLab.

Files

Download the following files to help you get started:

Module 4 Challenge files

Links to an external site.

Instructions

Using Pandas and Jupyter Notebook, create a report that includes the following data. Your report must include a written description of at least two observable trends based on the data.

Hint: Check out the sample solution called PyCitySchools_starter.ipynb located in the .zip file to review the desired format for this assignment.

District Summary

Perform the necessary calculations and then create a high-level snapshot of the district's key metrics in a DataFrame.

Include the following:

Total number of unique schools

Total students

Total budget

Average math score

Average reading score

% passing math (the percentage of students who passed math)

% passing reading (the percentage of students who passed reading)

% overall passing (the percentage of students who passed math AND reading)

School Summary

Perform the necessary calculations and then create a DataFrame that summarizes key metrics about each school.

Include the following:

School name

School type

Total students

Total school budget

Per student budget

Average math score

Average reading score

% passing math (the percentage of students who passed math)

% passing reading (the percentage of students who passed reading)

% overall passing (the percentage of students who passed math AND reading)

Highest-Performing Schools (by % Overall Passing)

Sort the schools by % Overall Passing in descending order and display the top 5 rows.

Save the results in a DataFrame called "top_schools".

Lowest-Performing Schools (by % Overall Passing)

Sort the schools by % Overall Passing in ascending order and display the top 5 rows.

Save the results in a DataFrame called "bottom_schools".

Math Scores by Grade

Perform the necessary calculations to create a DataFrame that lists the average math score for students of each grade level (9th, 10th, 11th, 12th) at each school.

Reading Scores by Grade

Create a DataFrame that lists the average reading score for students of each grade level (9th, 10th, 11th, 12th) at each school.

Scores by School Spending

Create a table that breaks down school performance based on average spending ranges (per student).

Use the code provided below to create four bins with reasonable cutoff values to group school spending.

```
spending_bins = [0, 585, 630, 645, 680]
```

```
labels = ["<$585", "$585-630", "$630-645", "$645-680"]
```

Use pd.cut to categorize spending based on the bins.

Use the following code to then calculate mean scores per spending range.

```
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()
```

Use the scores above to create a DataFrame called spending_summary.

Include the following metrics in the table:

Average math score

Average reading score

% passing math (the percentage of students who passed math)

% passing reading (the percentage of students who passed reading)

% overall passing (the percentage of students who passed math AND reading)

Scores by School Size

Use the following code to bin the per_school_summary.

```
size_bins = [0, 1000, 2000, 5000]
```

```
labels = ["Small (<1000)", "Medium (1000-2000)", "Large (2000-5000)"]
```

Use pd.cut on the "Total Students" column of the per_school_summary DataFrame.

Create a DataFrame called size_summary that breaks down school performance based on school size (small, medium, or large).

Scores by School Type

Use the per_school_summary DataFrame from the previous step to create a new DataFrame called type_summary.

This new DataFrame should show school performance based on the "School Type".

Requirements

District Summary (20 points)

Calculate the total number of unique schools (2 points)

Calculate the total number of students (2 points)

Calculate the total budget (2 points)

Calculate the average (mean) math score (2 points)

Calculate the average (mean) reading score (2 points)

Use the code provided to calculate the percentage of students who passed math (2 points)

Calculate the percentage of students who passed reading (2 points)

Use the code provided to calculate the percentage of students that passed both math and reading (2 points)

Create a new DataFrame for the above calculations called district_summary (4 points)

School Summary (20 points)

Use the code provided to select the school type (2 points)

Calculate the total student count (2 points)

Use the code provided to calculate the per capita spending (2 points)

Calculate the average test scores (2 points)

Calculate the number of schools with math scores of 70 or higher (2 points)

Calculate the number of schools with reading scores of 70 or higher (2 points)

Use the provided code to calculate the schools that passed both math and reading with scores of 70 or higher (2 points)

Use the provided code to calculate the passing rates (2 points)

Create a new DataFrame for the above calculations called per_school_summary (4 points)

Highest-Performing Schools by Percentage of Overall Passing (5 points)

Sort the schools by % Overall Passing in descending order (2 points)

Save the results to a DataFrame called top_schools (2 points)

Display the first 5 rows (1 point)

Lowest-Performing Schools by Percentage of Overall Passing (5 points)

Sort the schools by % Overall Passing in ascending order (2 points)

Save the results to a DataFrame called bottom_schools (2 points)

Display the first 5 rows (1 point)

Math Scores by Grade (10 points)

Use the code provided to separate the data by grade (1 points)

Group by "school_name" and take the mean of each (4 points)

Use the code to select only the math_score (1 points)

Combine each of the scores above into single DataFrame called math_scores_by_grade (4 points)

Reading Scores by Grade (10 points)

Use the code provided to separate the data by grade (1 points)

Group by "school_name" and take the mean of each (4 points)

Use the code to select only the reading_score (1 points)

Combine each of the scores above into single DataFrame called reading_scores_by_grade (4 points)

Scores by School Spending (5 points)

Use pd.cut with the provided code to bin the data by the spending ranges (2 points)

Use the code provided to calculate the averages (1 points)

Create the spending_summary DataFrame using the binned and averaged spending data (2 points)

Scores by School Size (5 points)

Use pd.cut with the provided code to bin the data by the school sizes (2 points)

Use the code provided to calculate the averages (1 points)

Create the size_summary DataFrame using the binned and averaged size data (2 points)

Scores by School Type (5 points)

Group the per_school_summary DataFrame by "School Type" and average the results (2 points)

Use the code provided to select the new column data (1 point)

Create a new DataFrame called type_summary that uses the new column data (2 points)

Written Report (15 points)

To receive all points, the written report presents a cohesive written analysis that:

Summarizes the analysis (5 points)

Draws two correct conclusions or comparisons from the calculations (10 points)