**Need to complete \*\*one of two\*\* (not both)**

**Before You Begin**

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

2. Clone the new repository to your computer.

3. Inside your local git repository, create a directory for the Pandas Challenge you choose. Use folder names corresponding to the challenges: \*\*HeroesOfPymoli\*\* or \*\*PyCitySchools\*\*.

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

5. Push the above changes to GitHub or GitLab

**Option 2: PyCitySchools**

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 score

as well as various information on the schools they attend

Aggregate the data to and showcase obvious trends in school performance.

Your final report should include each of the following:

District Summary

Create a high level snapshot (in table form) of the district's key metrics, including:

Total Schools

Total Students

Total Budget

Average Math Score

Average Reading Score

% Passing Math (The percentage of students that passed math.)

% Passing Reading (The percentage of students that passed reading.)

% Overall Passing (The percentage of students that passed math \*\*and\*\* reading.)

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 (The percentage of students that passed math.)

% Passing Reading (The percentage of students that passed reading.)

% Overall Passing (The percentage of students that passed math \*\*and\*\* reading.)

Top Performing Schools (By % Overall Passing)

Create a table that highlights the top 5 performing schools based on % Overall Passing. Include:

School Name

School Type

Total Students

Total School Budget

Per Student Budget

Average Math Score

Average Reading Score

% Passing Math (The percentage of students that passed math.)

% Passing Reading (The percentage of students that passed reading.)

% Overall Passing (The percentage of students that passed math \*\*and\*\* reading.)

Bottom Performing Schools (By % Overall Passing)

Create a table that highlights the bottom 5 performing schools based on % Overall Passing

School Name

School Type

Total Students

Total School Budget

Per Student Budget

Average Math Score

Average Reading Score

% Passing Math (The percentage of students that passed math.)

% Passing Reading (The percentage of students that passed reading.)

% Overall Passing (The percentage of students that passed math \*\*and\*\* reading.)

Math Scores by Grade\\*\\*

Create a table 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 table 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 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 (The percentage of students that passed math.)

% Passing Reading (The percentage of students that passed reading.)

% Overall Passing (The percentage of students that passed math \*\*and\*\* reading.)

Scores by School Size

Repeat the above breakdown, but this time group schools based on a reasonable approximation of school size (Small, Medium, Large).

Scores by School Type

Repeat the above breakdown, but this time group schools based on school type (Charter vs. District).

As final considerations:

Use the pandas library and Jupyter Notebook.

submit a link to your Github/Git Lab repo that contains your Jupyter Notebook.

include a written description of at least two observable trends based on the data.

Use Example Solution for a reference on expected format.

Hints and Considerations

Learning to program requires one to constantly tinker, experiment, and learn on the fly.

You are doing exactly the \_right\_ thing, if you find yourself constantly practicing Google-Fu and diving into documentation.

There is just no way (or reason) to try and memorize it all.

Online references are available for you to use when you need them. So use them!

Take each of these tasks one at a time.

Begin your work, answering the basic questions:

"How do I import the data?"

"How do I convert the data into a DataFrame?"

"How do I build the first table?"

Don't get intimidated by the number of asks. Many of them are repetitive in nature with just a few tweaks.

Be persistent and creative!

Expect these exercises to take time!

Don't get discouraged if you find yourself spending hours initially with little progress.

Force yourself to deal with the discomfort of not knowing and forge ahead.

Consider these hours an investment in your future!

As always, feel encouraged to work in groups and get help from your TAs and Instructor.

Just remember, true success comes from mastery and \_not\_ a completed homework assignment. So challenge yourself to truly succeed!

Ensure your repository has regular commits (i.e. 20+ commits) and a thorough README.md file

Rubric

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instructions:**  Evaluate the homework against the outlined criteria in the below rubric, assigning a rating to each criterion. Add points earned across all criteria and convert the total points to a letter grade, assigning a “+” or “-” letter grade designation at your discretion. | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | A (+/-) | 90+ | C (+/-) | 40-64 | F (+/-) | <15 | | B (+/-) | 65-89 | D (+/-) | 15-39 |  |  | |
| **Notes:**  The deployed assignment utilizes the **Pandas** library to analyze 1 of 2 challenges. Only one assignment will be accepted for grading. The source code should also be deployed to **Github** or **Gitlab**. |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instructions:**  Evaluate the homework against the outlined criteria in the below rubric, assigning a rating to each criterion. Add points earned across all criteria and convert the total points to a letter grade, assigning a “+” or “-” letter grade designation at your discretion. | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | A (+/-) | 100-90 | C (+/-) | 79-70 | F (+/-) | < 60 | | B (+/-) | 89-80 | D (+/-) | 69-60 |  |  | |

**Rubric for PyCitySchools:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mastery**  **20 points** | **Approaching Mastery**  **15 points** | **Progressing**  **10 points** | **Emerging**  **5-0 points** | **Incomplete** |
| **Expected output displayed** | ✓ Output for PyCitySchools contains all:  ✓ District Summary  ✓ School Summary  ✓ Top Performing Schools (By % Overall Passing)  ✓ Bottom Performing Schools (By  % Overall Passing)  ✓ Math Score by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type | ✓ Output for PyCitySchools contains at least 7:  ✓ District Summary  ✓ School Summary  ✓ Top Performing Schools (By % Overall Passing)  ✓ Bottom Performing Schools (By  % Overall Passing  ✓ Math Score by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type | ✓ Output for PyCitySchools  contains at least 5:  ✓ District Summary  ✓ School Summary  ✓ Top Performing Schools (By % Overall Passing)  ✓ Bottom Performing Schools ((By % Overall Passing)  ✓ Math Score by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending | ✓ Output for PyCitySchools contains 2 or fewer:  ✓ District Summary  ✓ School Summary  ✓ Top Performing Schools (By % Overall Passing)  ✓ Bottom Performing Schools ((By % Overall Passing)  ✓ Math Score by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending | No submission was received  -OR-  Submission was empty or blank  -OR-  Submission contains evidence of academic dishonesty |
| **Functions used on DataFrames** | The following functions are used on DataFrames and produce correct results:  ✓ Mean  ✓ Sum  ✓ Count | The following functions are used on DataFrames and produce varying results:  ✓ Mean  ✓ Sum  ✓ Count | Two of the following functions are used on DataFrames to produce varying results:  ✓ Mean  ✓ Sum  ✓ Count | One or fewer of the following functions are used on DataFrames to produce varying results:  ✓ Mean  ✓ Sum  ✓ Count |
| **GroupBy used** | GroupBy is used in PyCitySchools in determining the following:  ✓ School Summary  ✓ Math Scores by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type | GroupBy is used for PyCitySchools in determining at least 4 of the following:  ✓ School Summary  ✓ Math Scores by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type | GroupBy is used for PyCitySchools  in determining at least 3 of the following:  ✓ School Summary  ✓ Math Scores by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type | GroupBy is used for PyCitySchools  in determining 1 or fewer of the following:  ✓ School Summary  ✓ Math Scores by Grade  ✓ Reading Score by Grade  ✓ Scores by School Spending  ✓ Scores by School Size  ✓ Scores by School Type |
| **Cut method used to create new series of binned data** | PyCitySchools  data was cut and binned for both correctly:  ✓ Scores by School Spending  ✓ Scores by School Size | PyCitySchools  data was cut and binned for one correctly:  ✓ Scores by School Spending  ✓ Scores by School Size | PyCitySchools data was cut and binned for one with errors:  ✓ Scores by School Spending  ✓ Scores by School Size | PPyCitySchool data was either not attempted or was attempted to cut and bin but produces no results:  ✓ Scores by School Spending  ✓ Scores by School Size |
| **Written Report** | Presents a cohesive written analysis that:  ✓ Draws two correct conclusions from the data for Pyschools | Presents a cohesive written analysis that:  ✓ Draws at least one correct conclusion from the data for Pyschools | Presents a cohesive written analysis that:  ✓ Draws at least one complete but incorrect conclusion from the data for Pyschools | Presents a limited written analysis or no written analysis that:  ✓ Incorrect and incomplete conclusion form the data for Pyschools |