**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)A screenshot of a computer

  Description automatically generated
* 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) A screenshot of a computer code

  Description automatically generated
* Create a new DataFrame for the above calculations called district\_summary (4 points)
* A screenshot of a computer

  Description automatically generated

**School Summary (20 points)**

* Use the code provided to select the school type (2 points)
* Calculate the total student count (2 points)
* A screenshot of a computer

  Description automatically generated
* Use the code provided to calculate the per capita spending (2 points)
* A screenshot of a computer

  Description automatically generated
* Calculate the average test scores (2 points) A screenshot of a computer

  Description automatically generated
* Calculate the number of schools with math scores of 70 or higher (2 points)
* A screenshot of a computer

  Description automatically generated
* Calculate the number of schools with reading scores of 70 or higher (2 points)
* A screenshot of a computer

  Description automatically generated
* Use the provided code to calculate the schools that passed both math and reading with scores of 70 or higher (2 points)
* A screenshot of a computer

  Description automatically generated
* Use the provided code to calculate the passing rates (2 points) A computer screen shot of a computer code

  Description automatically generated
* Create a new DataFrame for the above calculations called per\_school\_summary (4 points)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer screen

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**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)
* A screenshot of a computer

  Description automatically generated

**Written Report (15 points)**

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

* Summarizes the analysis (5 points)
* In PyCity School Analysis we created a couple summary tables for students, which be utilized estimating following

1. How much money is spent per student was analyzed in **Scores by School Spending data frame**
2. School sizes small, medium or large are created in **Scores by School Size**
3. **School type and performance are analyzed in Scores by School Type**
4. **Best performing schools and bad performing schools are analyzed by creating data frames top schools and bottom schools**
5. **All the above data frames created on basic total students, total school, Average math and reading scores…….**

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

After creating multiple data frames and observing the data my 2 conclusions are

1. Charter Schools are performing better than District school in both math and reading
2. Medium sized schools are performing better than large and small sized schools
3. spending per student is low and overall performance better as per analysis
4. Johnson High School, which is a district type of school is the least overall performing school than rest of the schools
5. **Cabrera High School,** which is a charter type of school, is the best overall performing school than rest of the schools

**Grading**

This assignment will be evaluated against the requirements and assigned a grade according to the following table:

| **Grade** | **Points** |
| --- | --- |
| A (+/-) | 90+ |
| B (+/-) | 80–89 |
| C (+/-) | 70–79 |
| D (+/-) | 60–69 |
| F (+/-) | < 60 |

**Submission**

To submit your Challenge assignment, click Submit, and then provide the URL of your GitHub repository for grading.

**note**

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next module.

Comments are disabled for graded submissions in Bootcamp Spot. If you have questions about your feedback, please notify your instructional staff or your Student Success Advisor. If you would like to resubmit your work for an additional review, you can use the Resubmit Assignment button to upload new links. You may resubmit up to three times for a total of four submissions.

**important**

**It is your responsibility to include a note in the README section of your repo specifying code source and its location within your repo**. This applies if you have worked with a peer on an assignment, used code in which you did not author or create sourced from a forum such as Stack Overflow, or you received code outside curriculum content from support staff such as an Instructor, TA, Tutor, or Learning Assistant. This will provide visibility to grading staff of your circumstance in order to avoid flagging your work as plagiarized.

If you are struggling with a challenge assignment or any aspect of the academic curriculum, please remember that there are student support services available for you:

1. Ask the class Slack channel/peer support.
2. AskBCS Learning Assistants exists in your class Slack application.
3. Office hours facilitated by your instructional staff before and after each class session.
4. [Tutoring GuidelinesLinks to an external site.](https://docs.google.com/document/d/1hTldEfWhX21B_Vz9ZentkPeziu4pPfnwiZbwQB27E90/edit?usp=sharing) - schedule a tutor session in the Tutor Sessions section of Bootcampspot - Canvas
5. If the above resources are not applicable and you have a need, please reach out to a member of your instructional team, your Student Success Advisor, or submit a support ticket in the Student Support section of your BCS application.

**References**

Data generated by [Mockaroo, LLCLinks to an external site.](https://mockaroo.com/" \t "_blank), (2022). Realistic Data Generator. Data for this dataset was generated by edX Boot Camps LLC, and is intended for educational purposes only.

[Previous](https://bootcampspot.instructure.com/courses/6264/modules/items/1318938)[Next](https://bootcampspot.instructure.com/courses/6264/modules/items/1318949)