Aug-2022

Home
Modules
Syllabus
Grades
Zoom
Attendance
Student Support
Career Services
Billing
Internal Tools

Module 4 Challenge

New Attempt

Due Wednesday by 11:59pm   **Points** 100   **Submitting** a text entry box or a website url

Background

Maria has gotten a new version of the student data with several changes. This includes an additional column: "school budget". She wants you to rework part of your analysis by using the new dataset.

What You're Creating

This new assignment consists of five technical analysis deliverables and a written report to deliver the results. You will submit the following deliverables:

- Deliverable 1: Collect the student data into a DataFrame.
- Deliverable 2: Prepare a cleaned version of the DataFrame.
- Deliverable 3: Summarize key pieces of the data.
- Deliverable 4: Drill down into the data to analyze specific subsets.
- Deliverable 5: Compare and contrast the data through grouping and aggregation functions.
- Deliverable 6: A written analysis of your results (`README.md`).

Files

Challenge starter code

Deliverable 1: Collect the Data (5 points)

Deliverable 1 Instructions

1. Import the data from the `new_full_student_data.csv` file into a DataFrame named `student_df` by using the Pandas `read_csv` function and the `os` module.

2. Confirm that Pandas correctly imported the data by using the `head` function, as the following image shows:

	student_id	student_name	grade	school_name	reading_score	math_score	school_type	school_budget
0	103880842	Travis Martin	9th	Sullivan High School	59.0	88.2	Public	961125
1	45069750	Michael Brown	9th	Dixon High School	94.7	73.5	Charter	870334
2	45024902	Gabriela Lucero	9th	Wagner High School	89.0	70.4	Public	846745
3	62582498	Susan Richardson	9th	Silva High School	69.7	80.3	Public	991918
4	16437227	Sherry Davis	11th	Bowers High School	NaN	27.5	Public	848324

REWIND

For this deliverable, you've already done the following in this module:

- Lesson 4.2: Collecting Data

Deliverable 1 Requirements

- The path to the file is built by using `os.path.join`. (2 points)
- The DataFrame is created and named `student_df`. (2 points)
- The first five rows of data are displayed. (1 points)

Deliverable 2: Prepare the Data (25 points)

Deliverable 2 Instructions

1. In the student DataFrame, check for rows that have `NaN` (or missing) values, and remove those rows, as the following image shows:

```
student_id      0
student_name    0
grade           0
school_name     0
reading_score   0
math_score      0
school_type     0
school_budget   0
dtype: int64
```

2. In the student DataFrame, check for duplicate rows, and remove them.

3. Check the data types of the columns by using the `dtypes` property, as the following image shows:

```
student_id      int64
student_name    object
grade           object
school_name     object
reading_score   float64
math_score      float64
school_type     object
school_budget   int64
dtype: object
```

4. In the grade column, remove the "th" suffix from every value by using `str` and `replace`, as the following image shows:

```
0       9
1       9
2       9
3       9
5       9
        ..
19508   10
19509   12
19511   11
19512   11
19513   12
```

```
Name: grade, Length: 14831, dtype: object
```

5. Change the "grade" column to the `int` type, and then verify the column types, as the following image shows:

```
student_id      int64
student_name    object
grade           int64
school_name     object
reading_score   float64
math_score      float64
school_type     object
school_budget   int64
dtype: object
```

Deliverable 2 Requirements

- After the removal of null values, `isna().sum()` displays 0 for all the columns. (5 points)
- After the removal of duplicates, `duplicated().sum()` displays no duplicates. (5 points)
- The column types are displayed. (5 points)
- The "th" suffix is removed from all the values in the "grade" column. (5 points)
- The "grade" column is successfully converted to an `int` type. (5 points)

Deliverable 3: Summarize the Data (20 points)

Deliverable 3 Instructions

1. Generate the summary statistics for the student DataFrame by using the `describe` function, as the following image shows:

	student_id	grade	reading_score	math_score	school_budget
count	1.483100e+04	14831.000000	14831.000000	14831.000000	14831.000000
mean	6.975296e+07	10.355539	72.357865	64.675733	893742.749107
std	3.452909e+07	1.097728	15.224590	15.844093	53938.066467
min	1.000906e+07	9.000000	10.500000	3.700000	817615.000000
25%	3.984433e+07	9.000000	62.200000	54.500000	846745.000000
50%	6.965978e+07	10.000000	73.800000	65.300000	893368.000000
75%	9.927449e+07	11.000000	84.000000	76.000000	956438.000000
max	1.299997e+08	12.000000	100.000000	100.000000	991918.000000

2. Display the mean math score by using the `mean` function.
3. Store the minimum reading score in `min_reading_score`.

Deliverable 3 Requirements

- The summary statistics for the DataFrame are displayed. (6 points)
- The mean of the "math_score" column is displayed. (7 points)
- The minimum of the "reading_score" column is stored in `min_reading_score`. (7 points)

Deliverable 4: Drill Down into the Data (25 points)

Deliverable 4 Instructions

1. Display the grade column by using `loc`, as the following image shows:

```
0        9
1        9
2        9
3        9
5        9
        ..
19508   10
19509   12
19511   11
19512   11
19513   12
Name: grade, Length: 14831, dtype: int64
```

2. Display the first three rows of Columns 3, 4, and 5 by using `iloc`, as the following image shows:

	school_name	reading_score	math_score
0	Sullivan High School	59.0	88.2
1	Dixon High School	94.7	73.5
2	Wagner High School	89.0	70.4

3. Select the rows for Grade 9, and display their summary statistics by using `loc` and `describe`, as the following image shows:

	student_id	grade	reading_score	math_score	school_budget

	count	4.132000e+03	4132.0	4132.000000	4132.000000	4132.000000
mean	6.979441e+07	9.0	69.236713	66.585624	898692.606002
std	3.470565e+07	0.0	15.277354	16.661533	54891.596611
min	1.000906e+07	9.0	17.900000	5.300000	817615.000000
25%	3.953848e+07	9.0	59.000000	56.000000	846745.000000
50%	6.984037e+07	9.0	70.050000	67.800000	893368.000000
75%	9.939504e+07	9.0	80.500000	78.500000	957299.000000
max	1.299997e+08	9.0	99.900000	100.000000	991918.000000

4. Store the row with the minimum overall reading score in `min_reading_row` by using `loc` and the `min_reading_score` variable from Deliverable 3, as the following image shows:

	student_id	student_name	grade	school_name	reading_score	math_score	school_type	school_budget
3706	81758630	Matthew Thomas	10	Dixon High School	10.5	58.4	Charter	870334

5. Select all the reading scores from the 10th graders at Dixon High School by using `loc` with conditionals, as the following image shows:

	school_name	reading_score
45	Dixon High School	71.1
60	Dixon High School	59.5
69	Dixon High School	88.6
94	Dixon High School	81.5
100	Dixon High School	95.3
...	...	...
19283	Dixon High School	52.9
19306	Dixon High School	58.0
19344	Dixon High School	38.0
19368	Dixon High School	84.4
19445	Dixon High School	43.9

569 rows × 2 columns

6. Find the mean reading score for all the students in Grades 11 and 12 combined by using conditional statements and `loc` or `iloc`.

REWIND

For this deliverable, you've already done the following in this module:

- Lesson 4.5: Drilling Down into Data ↗

Deliverable 4 Requirements

- The "grade" column is displayed. (4 points)
- The first three rows of Columns 3, 4, and 5 are displayed. (4 points)
- The summary statistics for 9th graders are displayed. (4 points)
- The row that contains the minimum reading score is displayed. (4 points)
- The reading scores of the 10th graders at Dixon High School is displayed. (4 points)
- The average reading score of all the students in Grades 11 and 12 combined is calculated. (5 points)

Deliverable 5: Compare the Data (20 points)

Deliverable 5 Instructions

1. Display the average budget for each school type by using the `groupby` and `mean` functions, as the following image shows:

	school_budget
school_type	
Charter	872625.656236
Public	911195.558251

2. Find the total number of students at each school, and sort those numbers from largest to smallest by using the `groupby`, `count`, and `sort_values` functions, as the following image shows:

	student_count
school_name	
Montgomery High School	2038
Green High School	1961
Dixon High School	1583
Wagner High School	1541
Silva High School	1109
Woods High School	1052
Sullivan High School	971
Turner High School	846
Bowers High School	803
Fisher High School	798
Richard High School	551
Campos High School	541
Odonnell High School	459
Campbell High School	407
Chang High School	171

3. Find the average math score by grade for each school type by using the `groupby` and `mean` functions, as the following image shows:

		math_score
school_type	grade	
Charter	9	70.0

Charter	9	70.0
	10	66.0
	11	68.0
	12	60.0
Public	9	64.0
	10	64.0
	11	59.0
	12	64.0

REWIND

For this deliverable, you've already done the following in this module:

- Lesson 4.6: Comparing Subsets of Data

Deliverable 5 Requirements

- The average budget for each school type is displayed. (7 points)
- The total number of students per school is displayed in descending order. (6 points)
- The average math scores for each combination of grade and school type are displayed. (7 points)

Deliverable 6: Report Findings (5 points)

Deliverable 6 Instructions

Using the provided cell, write a brief summary of your findings as follows: Write a few sentences to describe any discoveries that you made while performing your analysis. Include any additional analysis that you believe would be worthwhile.

Deliverable 6 Requirements

- In a few sentences, an understanding of the performed analysis is demonstrated. (5 points)

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you meet the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone! Then, commit the deliverables to your PyBer_Analysis GitHub repository.

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

IMPORTANT

Once you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then add this week's Challenge to your professional portfolio.

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 Submit, and then indicate you are skipping by typing "I choose to skip this assignment" in the text box.

◄ Previous | Next ►