**Module 4 Challenge Report**

**(Francis Odo)**

**Background**

This project is a continuation of School Data Analysis in this Module. We are presented with the task of re-evaluating the School and Student data following an academic dishonesty with student grades at Thomas High School within the ninth graders.

**Objective**

The objective is to replace the ninth-grade math and reading scores from Thomas High School with NaN and maintain the integrity of all other associated data with the school.

We will identify any change in the overall data parameters as a result, if any.

**Code Plan**

1. Import all the necessary libraries and dependencies.
2. Load and read the CSV data files for the project
3. Apply the data cleaning process to remove empty spaces, prefixes and suffixes and any unnecessary aliases from student names.
4. Select Reading Scores and Math Scores from the student\_data\_df DataFrame.

Use the loc() method to select School Name (Thomas High School), Grade (9th) and Reading Score in student\_data\_df.

Use the loc() method to select School Name (Thomas High School), Grade (9th) and Math Score in student\_data\_df.

1. Use Numpy np .where() and np.nan() methods to process the replacement of elements depending on conditions.
2. Verify that the replacement or changes in Reading Scores and Math Scores for the 9th Graders actually takes effect.

**Approach**

The technique applied in this project is to implement the change in the school data for Thomas High School Ninth grade student data. Then, recalculate essential parameters and DataFrames for the analysis. Among these parameters are:

* + District and school summary DataFrames
  + District Summary
  + School Summary
  + High and Low-performing schools
  + Thomas High School performance relative to other schools
  + Math and Reading Scores by Grade
  + Scores by School Spending
  + Scores by School Size
  + Scores by School Type

**Conclusion / Observation**

The changes to Thomas High School 9th Grade student scores to NaN are reflected in the output below.

student\_data\_df.loc[student\_data\_df['school\_name'] == 'Thomas High School']

| **Student ID** | **student\_name** | **gender** | **grade** | **school\_name** | **reading\_score** | **math\_score** |
| --- | --- | --- | --- | --- | --- | --- |
| **37535** | 37535 | Norma Mata | F | 10th | Thomas High School | 76.0 | 76.0 |
| **37536** | 37536 | Cody Miller | M | 11th | Thomas High School | 84.0 | 82.0 |
| **37537** | 37537 | Erik Snyder | M | 9th | Thomas High School | NaN | NaN |
| **37538** | 37538 | Tanya Martinez | F | 9th | Thomas High School | NaN | NaN |
| **37539** | 37539 | Noah Erickson | M | 9th | Thomas High School | NaN | NaN |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **39165** | 39165 | Donna Howard | F | 12th | Thomas High School | 99.0 | 90.0 |
| **39166** | 39166 | Dawn Bell | F | 10th | Thomas High School | 95.0 | 70.0 |
| **39167** | 39167 | Rebecca Tanner | F | 9th | Thomas High School | NaN | NaN |
| **39168** | 39168 | Desiree Kidd | F | 10th | Thomas High School | 99.0 | 90.0 |
| **39169** | 39169 | Carolyn Jackson | F | 11th | Thomas High School | 95.0 | 75.0 |

1635 rows × 7 columns

After replacing the Reading and Math scores:

District and school summary DataFrames output Barely to Not affected

School summary Barely to Not affected

District Summary Barely to Not affected

High/Low performing Barely to Not affected

Thomas High School’s performance/others Barely to Not affected

Math and Reading scores by grade Very low change/Insignificant

Scores by School Spending Not affected

Scores by School Size Not affected

Scores by school Type Not affected

The main reason why there is no significant changes is attributed to the number of Ninth Grade students at Thomas High School compared to total number of students.

Out of about 1635 rows of Thomas High School student data 461 appeared to be the number of 9th Graders.

37537 NaN

37538 NaN

37539 NaN

37540 NaN

37543 NaN

39152 NaN

39153 NaN

39157 NaN

39164 NaN

39167 NaN

Name: reading\_score, Length: 461, dtype: float64