**Please answer the following questions and save your answers in a public GitHub repository. You have 24 hours to submit your answer.**

The github repo can be found at:

https://github.com/jdclifton2/DataScienceQuestions

1. Use the table below for problem 1 a - c

a) Based on the following two tables, write a SQL query that returns the name and student ID of all students that have a higher total marks score than the student that has StudentID of 'V002’

|  |
| --- |
| SELECT a.studentid, a.name, b.total\_marks |

|  |
| --- |
| FROM name\_table as a, mark\_table as b |

|  |
| --- |
| WHERE a.studentid = b.studentid AND b.total\_marks > |

|  |
| --- |
| (SELECT total\_marks |

|  |
| --- |
| FROM mark\_table |

where studentid = ‘V002’);

code version: https://github.com/jdclifton2/DataScienceQuestions/blob/main/query.sql

b) Assume that the two tables are pandas data frame variables. Based on those two data frames--utilizing pandas--write a python function that returns a new data frame version of name\_table, where each name containing the letter “e” is uppercased, and lowercased otherwise (e.g. “Edward” → “EDWARD”, “Bob” → “bob”).

def upper\_if\_e(name\_table):

"""

returns a new data frame version of name\_table, where each name containing the letter “e” is uppercased,

and lowercased otherwise.

:param name\_table: A dataframe containing the names of people.

:return: A new dataframe where every name that contained the letter 'e' is uppercased and names that

did contain 'e' are lower cased.

:rtype: Dataframe

Example: “Edward” → “EDWARD”, “Bob” → “bob”

"""

names = name\_table['Name']

new\_df = name\_table.copy()

#if the string does not contain e. ~ is the negation operator.

new\_df.loc[~ new\_df.Name.str.contains('e'), 'Name'] = new\_df.Name.str.lower()

new\_df.loc[new\_df.Name.str.contains('e'), 'Name'] = new\_df.Name.str.upper()

return new\_df

code version : https://github.com/jdclifton2/DataScienceQuestions/blob/main/DataScienceQuestions.ipynb

c) Now write a function that takes in the output of 1) b) and mark\_table and returns a data frame that summarizes the average grade of uppercase names and lowercase names

def summarize\_avg\_grade(new\_name\_table, mark\_table):

"""

returns a new data frame containing the average of the grades contained within mark\_table for uppercased

names and lowercased names contained within new\_name\_table.

:param new\_name\_table: A dataframe containing the names of people. The names should be all uppercased or all

lowercased.

:param mark\_table: A dataframe containing grades.

:return: A new dataframe containing the average score for upper case names and lower case names.

:rtype: Dataframe

"""

#merge the dataframes by their index.

combined\_df = new\_name\_table.merge(mark\_table, left\_index=True, right\_index=True)

upper\_avg = combined\_df[combined\_df.Name.str.isupper()].mean()

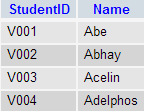
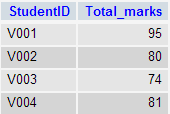
lower\_avg = combined\_df[combined\_df.Name.str.islower()].mean()

averages\_df = pd.DataFrame({"Uppercase Average": upper\_avg, "Lowercase Average": lower\_avg})

return averages\_df

code version: https://github.com/jdclifton2/DataScienceQuestions/blob/main/DataScienceQuestions.ipynb

name\_table mark\_table

1. **Consider the data set below. Write some python code that illustrates some common feature engineering and/or data preparation tasks.**

https://github.com/helloworlddata/white-house-salaries/blob/master/data/converted/2017.csv

answer:

https://github.com/jdclifton2/DataScienceQuestions/blob/main/DataScienceQuestions.ipynb

**Consider the file “data.csv” in the following GitHub repository. What are some descriptive statistics about this set? What can you say about the distribution of this data?**

<https://github.com/fractalbass/data_engineer>

answer:

<https://github.com/jdclifton2/DataScienceQuestions/blob/main/DataScienceQuestions.ipynb>

No code is necessary for the following questions:

1. **If you were asked to impute null values in a column of a file that was 365 Gigabytes, what would you do? What tools would you use? What tools would you NOT use?**

I would not use basic python or R. I would most likely use something like pyspark that would allow me to work with such a large dataset. Through in-memory caching, and optimized query execution, Spark can work with data of any size.

1. **What would you do if you were asked to do the above task every Thursday morning at 2:00am?**

I would write a script to try and automate this task.

1. **Who is your favorite mathematician, statistician or computer scientist and why?**

My favorite mathematician is Sirinivasa Ramanujan. I find his background to be relatable and fascinating. For someone that had no formal education, he pioneered entire branches of mathematics. To this day much of his work is still not fully understood! He was a unique person and perhaps the greatest mathematician of all time.

**Thanks for taking the time to participate in this exercise!**