SAS Optimization Challenge: Case Study

Title: HELP YOUR OPTIMIZATION PROFESSOR!!!!!

TIER 2 (of 5)

Tier 2: Modeling Details

The model you will build in Tier 2 builds off of the model you recently completed in Tier 1, so use the SAS program that you submitted for Tier 1 as the starting point for Tier 2.

The Tier 2 objective is to incorporate the **Frenemy_Group** column into the optimization model.

Recall from the Case Study Overview that questions in the same frenemy group are highly similar to one another. For example, suppose you're asked the following two questions:

I. Which Hall of Fame baseball player wore jersey No. 3 for the New York Yankees?
II. Which Hall of Fame baseball player wore jersey No. 7 for the New York Yankees?

There's a high likelihood that you either know the answers to both of these questions, or you don't know the answer to either question. As a result, there's no need for both questions to appear on the final exam.

These two questions are examples of "frenemies", and in the context of the question bank, would be contained within the same frenemy group.

The answers, by the way, are Babe Ruth and Mickey Mantle, respectively.

Your objective for Tier 2 is to choose the fifty (50) questions that produce the most difficult exam possible, while ensuring that each *topic objective* contains the correct number of questions, as well as **ensuring that no more than one question in each frenemy group is assigned to the exam**.

Use the `create data` statement to generate one (1) output SAS data set containing only the fifty (50) questions that your model chose for the final exam. Include the following columns in

the output data set: **Question**, **Frenemy_Group**, **Topic**, **Objective**, and **Correct**. Questions that were not chosen should not appear in your output data set.

Additionally, report the average percentage correct (i.e., take the average of the **Correct** column) across all fifty (50) questions chosen by your model. This value can be calculated by any means necessary (e.g., in SAS, Excel, calculator, etc.).

To do this in SAS, copy and paste the following SQL procedure code at the end of your program, and replace *<insert output data set name>* with the name of your output data set.

```
proc sql;
 select avg(Correct) as Avg_Correct
  from <insert output data set name>;
quit;
```

Tier 2: Getting Started

Since Tier 2 builds off of the solution developed in Tier 1, copy and paste your Tier 1 solution (with DATA step and datalines code creating the two data sets) into a new programming editor to begin the process of reformulating your optimization model to accommodate the new requirement.

Modifications to the two data sets are not needed.

Tier 2: Hints

The hints below are provided to help overcome relatively nuanced programming syntax and model formulation barriers that may arise in Tier 2. Hints provided in previous tiers may be helpful for Tier 2, however only new hints specifically for Tier 2 are given below.

I. The set of QUESTIONS should be indexed by **Question**, **Frenemy_Group**, **Topic**, and **Objective** to read the **work.question_bank** data set into the OPTMODEL procedure.

   (e.g., `set <str,num,str,str> QUESTIONS;`)

II. To avoid confusion, consider using local dummy parameters `q`, `f`, `t`, and `o`, respectively.

III. Use the `setof{}` operator to create a new set called FRENEMYGROUPS from the existing set of QUESTIONS

IV. When creating the FRENEMYGROUPS set, use the colon operator within the `setof{}` operator to remove the missing group, since only groups 1-40 are valid frenemy groups.

V. When creating a new set (i.e., FRENEMYGROUPS) from an existing set (i.e., QUESTIONS), it's a good practice to write the new set to the log to verify it was created correctly. You can do this using the `put` statement.

(e.g., `put FRENEMYGROUPS=;`)

VI. To accommodate the new frenemy group requirement, the reformulated model should contain one new constraint for each frenemy group. When summing across all questions, since both the FRENEMYGROUPS set and the QUESTIONS set are indexed by frenemy groups, use parenthesis ( ) on the local dummy parameter `f` in the QUESTIONS set to mask the implicit declaration.

(e.g., `sum{<q,(f),t,o> in QUESTIONS}`)