Stat 604
Assignment 12 - SAS

This assignment reinforces the concepts covered in lessons SAS01 through SAS11. Specifically, it focuses on the use of arrays and variable lists to process data. You will use these along with other methods to rotate data. It will also allow you to practice using the match merge process to combine data. Perform each of the exercises listed below. Your output should match that in the PDF file posted on eCampus. In many cases you will be given the specific steps necessary to accomplish the task. However, as the assignment progresses, you may not be told all of the preliminary or detailed tasks required to accomplish an objective. In those situations, it is up to you to figure out how to prepare for and accomplish the objective. Download the files **scholarships.sas7bdat** and **fund_data.sas7bdat** from the Assignment Data Files section on eCampus for use in this assignment.

Most undergraduate students taking classes in the STAT department receive some type of financial aid. This financial aid comes from over 250 different aid funds. Currently, each student may have up to 10 different funds in their financial aid package. The **scholarships** data set is already sorted by student_id and contains information about each student along with aid award amounts and the respective fund codes. Even though the available aid funds represent a broad range of categories from Corporate funded scholarships to work study programs, the STAT department is specifically concerned about the amount the University is spending on aid coming from Internal and Athletic scholarship categories. The **fund_data** data set contains the list of available aid funds and the category to which each fund belongs. You need to consolidate this information and restructure it to facilitate analysis of aid from the desired categories. All of your code should be written so that TAMU can increase the number of aid awards in a student's financial aid package without you needing to upgrade your code. (In other words, the number 10 never appears in your program code.)

1. Begin your program with the required header, filename, and libname statements. As always, your program must include comments in the appropriate places. In addition to the comment boxes you have been using above data and proc steps, use single comment lines to identify the sub-steps that are being accomplished within the data step.
2. From the **scholarships** data set, create a narrow data set containing the student id and fund code. Use a data step and an array to accomplish this transformation. For efficiency, only read the variables you need from the input data set. The resulting data set should not contain any missing values for the fund code. You may find it handy later on to have the index value from your array so keep it in the output data set. The first few values and the log from this step are shown below:

```
NOTE: There were 424 observations read from the data set HWDATA.SCHOLARSHIPS.
NOTE: The data set WORK.STUDENT_FUNDS has 2243 observations and 3 variables.
```

Obs	student_id	i	fund_code
1	120101	1	1101
2	120102	1	1011
3	120102	2	1190
4	120102	3	1228
5	120103	1	1130

3. Sort the new data set in place so that it can be merged with the data contained in the **fund_data** data set.
4. In your work library create a sorted copy of the **fund_data** data set for merging with the new data set created in the previous steps.
5. Use match merge to create a new data set that combines the fund type (category) with the student/fund data. Make sure no funds are included in the data if there have been no awards made to students from that scholarship fund. The first few values and the log from this step are shown below. It is essential that your data set has the same number of variables and observations:

```
NOTE: There were 2243 observations read from the data set WORK.STUDENT_FUNDS.
NOTE: There were 255 observations read from the data set WORK.FUNDS_SORTED.
NOTE: The data set WORK.FUND_TYPES has 2243 observations and 4 variables.
```

Obs	student_id	i	fund_code	Category
1	120111	8	1001	Foundations
2	120153	4	1001	Foundations
3	120187	5	1001	Foundations
4	120783	2	1001	Foundations
5	120793	3	1001	Foundations

6. Use the transpose procedure to transform this data set back into a wide data set of fund types for each student that can be merged with the original **scholarships** data set. The first few values and the log from this step are shown below:

```
NOTE: There were 2243 observations read from the data set WORK.FUND_TYPES.
NOTE: The data set WORK.WIDE_TYPES has 424 observations and 11 variables.
```

Obs	student_id	Fund_Type1	Fund_Type2	Fund_Type3	Fund_Type4	Fund_Type5	Fund_Type6	Fund_Type7
1	120101	Foundations						
2	120102	Foundations	Skillset	Athletic				
3	120103	Athletic						
4	120104	Military	Athletic	Corporate	Athletic	Work	Foundations	Foundations
5	120105	Internal	Talent	Foundations	Military	Corporate	Corporate	Corporate

(NOTE: Not all columns are visible due to space limitations.)

7. Use a single data step to perform a match merge that combines the fund types shown above with the original **scholarships** data set and creates some summary variables. Apply the appropriate labels to your new variables. The successful solution will most likely use some arrays. Create a variable that contains the amount of Internal scholarships each student received based on the fund type value loaded from the **fund_data** data set. Likewise, create a variable for the total each student received from Athletic scholarships. Create a third variable that contains the total amount of financial aid the student received from all funds.
8. Print the descriptor portion and then the data portion of the final data set. When you print the descriptor portion, make it print the variables in the order they appear in the data set instead of alphabetically. When you print the data portion, suppress the printing of observation numbers and print only the variables shown in the output posted on eCampus.
9. Convert the program and log to PDF files and submit them to WebAssign along with your SAS output.