Stat 604
Assignment 4 - SAS

OBJECTIVES: This assignment reinforces the following concepts:

- inserting comments into SAS programs
- defining and using SAS libraries
- using the Output Delivery System (ODS)
- displaying the contents of SAS libraries and data sets
- creating, subsetting and printing SAS data sets from other SAS data sets
- creating variables conditionally
- controlling the input and output of rows of data

You should have all of the information you need to complete this assignment by viewing Lectures SAS01 through SAS06.

There will be three PDF files posted to eCampus –one for the program, one for the log, and one containing the SAS output.

Perform each of the exercises listed below. To the extent you have been taught to control it, your output should match that in the PDF file posted on eCampus with certain allowances as indicated in the instructions below. The results of your contents procedures may vary slightly from the sample output depending on whether you are using PC SAS or SAS Studio. There may also be some variation in color and font depending on the version of SAS.

1. Begin your program with a header block. Include a line for the date last submitted at the top of the header block. This will just be a comment that you will manually update in the SAS program each time you submit it. Include a comment box above each data step and proc step in your programs. These boxes should reference the step number from the assignment and contain a brief description of what the step does. In this and all assignments, proper documentation is worth 10 percent of the assignment grade.
2. If you are using SAS Studio Web Editor, create a folder with a name like hwdata under "My Folders" where you can store your original homework data files. If you are using a full copy of PC SAS, create the folder in a convenient place on your computer like "My Documents". Download the **tx_schools** data set from eCampus and save it in the newly created homework folder. Write a libname statement in your program that accesses this folder. Add the statement **access=readonly** to the end of the libname statement so you will not accidentally overwrite the data.

   Familiarize yourself with the data contained in the **tx_schools** data set.
3. Create a second folder with a name of your choosing on your computer or on SAS Studio, if you are using it, so that you can create permanent data sets and store them in this folder. Write a libname statement in your program that uses a name of your choosing to assign a libref to this folder.

4. Specify a fileref that will be used to designate the output file for your PDF output. Use a name for the file like FKincheloe_HW04_output.pdf. (Use your own first initial and last name in place of FKincheloe.) Open a PDF destination using this fileref to capture the output from the procedures that follow. Refer to the document **ODS PDF Tip Sheet** in Course Materials for options to control the output. Add an option to create the bookmarks but hide them so they are not visible by default. You may need to open your output file directly in Adobe Acrobat Reader to see whether the bookmarks are created.
5. Using the **tx_schools** data set as input, create a new revised Texas High School data set in the permanent library you assigned in step 3 above. (If you are using SAS on an operating system other than windows or using SAS Studio, it is normal for you to receive the message, "NOTE: Data file HWDATA.TX_SCHOOLS.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.")

   The new data set will be different from the original data set in the following ways:
   a. The new data set will only contain observations with students in at least one of the high school classes; FR, SO, JR, or SR.
   b. The variables **state**, **type**, **level**, **F16** and **F17** will not be in the new data set.
   c. Supply labels for the variables fte_teachers, ptr, control, gr8, fr, so, jr, and sr. The labels will be Teachers (FTE), Student/Teacher Ratio, School Type, Eighth Graders, Freshmen, Sophomores, Juniors, and Seniors respectively.
   d. Compute the total number of students enrolled in high school in each school. Note: Eighth graders are not considered to be in high school. Give the new variable a label of HS Enrollment. Construct the computation in such a way that there will still be a total even if there is a missing value from one of the high school class variables.
   e. Create a new variable labeled Origin Date that contains the current date as of the time the data set is created. The date should display in the format 05/26/2019. The date should be computed so that it will update automatically whenever the program is run. (This will cause the date in your output to be different than the sample output posted on eCampus.)
6. Print the descriptor portion of your new data set. On this and each subsequent step that produces output to ODS, create a title statement that matches the title in the sample output. Use TITLE or TITLE1 to produce the title because titles are persistent and if you use a number like TITLE2, SAS will create a second line to include with the first title.
7. Print the first 10 observations of the new data set. Ensure that the labels are displayed in the output.
8. In the steps that follow you will be creating temporary data sets that are subsets of the revised recruiting data set created above in your permanent library. The first data set will be a list of academies as indicated by the word ACADEMY in the school name. Exclude ACADEMY H S in Bell County which is named for the small town of Academy. You may find it helpful to use the keyword NOT with the logic that identifies this school. The data set will only contain the variables School, County, Enrollment, and Control.
9. Print the data portion of the academies data set. Ensure the order of the variables in the printed output matches the sample on eCampus.

10. Create a data set where the number of seniors is more than 25% of the value of the enrollment variable but eliminate those schools where all of the enrollment is made up of seniors. This data set will only contain the variables school, county, gr8, fr, so, jr, sr, and enrollment.
11. Print the data portion of the data set created in the previous step. Ensure the order of the variables in the printed output matches the sample on eCampus. Suppress the printing of observation numbers.
12. From the Texas High School data set we want to create three temporary data sets in a single data step as efficiently as possible. All subsetting, variable creation, and keeping or dropping of variables must be positioned for maximum efficiency. The following steps describe how the 3 data sets are to be created:
    a. The variables control, fte_teachers and ptr will not be in any of the new data sets.
    b. Only include schools with students in all 4 of the high school classes FR, SO, JR, and SR.
    c. Use a series of conditional (if) statements to create a new variable named Division that reflects the size classification of each school based on enrollment. Public schools are in Divisions 1A through 6A. Private schools are in Divisions TAPS1 through TAPS3. The value of the variable **Control** indicates whether a school is Public or Private. Divisions are determined by the enrollments shown below:
    6A - 1601 & up
    5A - 801 to 1600
    4A - 401 to 800
    3A - 201 to 400
    2A - 81 to 200
    1A – Below 81
    TAPS3 - 111 & up
    TAPS2 - 56 to 110
    TAPS1 - 55 & below

    Construct the statements so the program executes as efficiently as possible. Process the public school divisions first as a group and then the private schools. Use the distribution shown below to determine the order of the statements within each group.

Division	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1A	406	19.15	406	19.15
2A	355	16.75	761	35.90
3A	301	14.20	1062	50.09
4A	192	9.06	1254	59.15
5A	196	9.25	1450	68.40
6A	339	15.99	1789	84.39
TAPS1	138	6.51	1927	90.90
TAPS2	78	3.68	2005	94.58
TAPS3	115	5.42	2120	100.00

d. Create a **SixA** data set with schools in division **6A** and a **TAPS3** data set containing schools in division **TAPS3**. Neither of these data sets should contain the variable Division. The variable County should not be in the **TAPS3** data set. Create a third data set named **Align19** that contains all rows and all variables except those you were instructed to exclude in steps a and b.

13. From the **Align19** data set we want to create a temporary data set named **GradeCount** that is more conducive to certain types of reporting. This data set will only have 4 variables: School, Division, Grade and Students. Use the appropriate data set option to control which variables are included. There will be up to five rows per school. If the variable for Eighth Grade has an actual value (not missing or 0) then set the value of Grade to Eighth and the value of Students to the number in the Eighth Grade variable. Output the observation whenever there is a value. Your program code must include the word "do". Repeat this process for the Freshman, Sophomore, Junior and Senior variables adjusting the value of Grade and Students accordingly. The step will read the 2120 observations from **Align19** and the resulting data set should have 9283 observations.
14. Add a proc contents step that will show the contents of the work library <u>without</u> displaying the descriptor portion of each dataset. (NOTE: Unless you were told what to name a specific data set, your names may be different.)
15. Use data set options to print 50 observations from **Align19** beginning with B F TERRY HS. Do not show observation numbers in your output.
16. Knowing what you do about the data set **SixA**, print the last 30 observations of the data set. Hard code the number into your print statement.
17. Print the **TAPS3** data set.
18. Print the first 35 observations from the **GradeCount** data set.
19. Copy the proc step below into your program and run it to print a report based on the **GradeCount** data set.

```
proc tabulate data=gradecount;
        class division grade;
        var students;
        table grade='Grade', division*students=' '*sum=' '*f=comma7.;
run;
```

20. When you have your program debugged and running without errors, close SAS. Then reopen SAS and run your program again. Convert your log to PDF and save the log document as a PDF file with a name like FKincheloe_HW04_log.pdf. Use your own first initial and last name instead of FKincheloe in all file names.
21. Save the final version of the program and convert it to a PDF file with a name like FKincheloe_HW04_prog.pdf
22. Submit the program and log PDF files to eCampus along with your SAS output.