# Stat 604
# Assignment 7 - SAS

This assignment reinforces the concepts covered in lectures SAS01 through SAS13. Specifically, it focuses on combining data, sorting data, and producing reports. You will get some additional practice using the match merge process.
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.

Download the two data sets named **ok_mid.sas7bdat** and **ok_high.sas7bdat** to your homework data library from the Assignment Files section on eCampus. The first contains Oklahoma Middle Schools with students in some of the grades 7-9. The second contains Oklahoma High Schools with students in at least grades 10-12. Familiarize yourself with the structure and data in these data sets before writing your program. One of our objectives is to create a data set that combines these two based on observations where the MapCity and School name are the same in both data sets. Do not alter these two original data sets in any way throughout the course of this assignment. If you need to sort either of these data sets, created sorted copies in the work library. Use variable lists in your code any time you can to minimize typing.

1. Begin your program with the required header, filename, and libname statements. Make your homework data library readonly. As always, your program must include comments in the appropriate places.
2. Using a single merge step, create three data sets as described in steps a, b, and c below:
    a. Create a temporary data set as mentioned above that combines Middle (ok_mid) and High Schools (ok_high) based on a match of MapCity and School. Only include observations where there is a match in both source data sets. One of the drawbacks of this process is that all of the variables have the same names in both data sets. This will cause values, including missing values, from the second data set to overwrite data in the first data set. Another problem with this data is that some 9th graders are in middle schools and some are in high schools. To overcome these issues, we are going to rename some of the variables from each data set and then combine them to form new variables in our data step. Set up the merge so that High School MailCity and County will overwrite the values from the Middle School data set when they are different. All of the numeric variables that are kept will be renamed. Rename all of the "grade" variables so that they are unique for each of the two data sets. One type of variable list can be used with the rename option so you do not need to list out each of these variables individually. If you will use the same prefixes such as Mid and HS for all of your numeric variables, you can drop them all with a single variable list definition at the appropriate time. Drop the Teachers variable. Based on the position of the variables in the data sets, use a variable list to drop variables Ungraded through HSTotal.
        i. Create two arrays that can be used to access the number of students in grades 7 through 12 from each of your two input data sets. Create a third array that will create and access new variables named Grade7 through Grade12.
        ii. Use a loop to process these arrays. For each grade, add the number of Middle School students to the number of High School students from that grade. Since,

in almost every case, one of the two values will be missing, your arithmetic must be able to tolerate the missing values.

iii. We are going to "impute" the number of teachers based on the PTRatio from each of our respective data sets as follows: For each input grade that has students in it, divide the number of students by the PTRatio from that data set and add the result to a running total of teachers for that school.

iv. Even though we are looking at full time equivalent (FTE) teachers for which a decimal value would be acceptable, we want a whole number for this assignment. When all classes have been considered, adjust the number of teachers for that school up to the next whole number.

v. Create a variable that contains the total number of students in grades 7 through 12 in each school.

vi. Compute the revised Pupil/Teacher ratio using the total number of students divided by the adjusted Teachers variable. Format the value so that it shows 2 decimal places. Include a condition in your calculation statement to prevent a divide by zero data error if the number of teachers is 0 or missing.

b. Create a temporary data set of High Schools (from ok_high) with no matching record in ok_mid based on MapCity and School name. Keep the school, MapCity, MailCity, and county in this data set.

c. Create a temporary data set of Middle Schools (from ok_mid) with no matching record in ok_high. Keep the same variables as in the data set in step b.

The following notes were generated by step 2 using SAS Studio:

```
NOTE: Missing values were generated as a result of performing an operation on
missing values.
       Each place is given by: (Number of times) at (Line):(Column).
       1138 at 130:9   4 at 134:40     12 at 135:42
 NOTE: There were 610 observations read from the data set WORK.MID.
 NOTE: There were 459 observations read from the data set WORK.HIGH.
 NOTE: The data set WORK.MATCHED_SCHOOLS has 375 observations and 13
variables.
 NOTE: The data set WORK.MID_NOMATCH has 235 observations and 4 variables.
 NOTE: The data set WORK.HIGH_NOMATCH has 87 observations and 4 variables.
```

3. Set up your program to ensure that the output is printed with a landscape layout (wider than it is tall). Ensure the date is printed but that the date/time resets to the current time instead of using the SAS session's system time. Suppress the printing of page numbers on the page. Open the PDF destination using the default settings.
4. Without creating a new dataset, change the order of the observations in the "Matched Schools" data set (from step 2a) by descending Pupil/Teacher Ratio and descending number of Students.
5. Print the "top 25" schools based on highest Pupil/Teacher Ratios from this dataset with the variables in the order shown in the sample output on eCampus. Use the same titles and footnotes as shown on eCampus. Use MapCity as the City variable. Note that there is a blank line between the two titles. Add temporary labels to match the eCampus output.
6. Suppress the printing of the date and time on the remaining pages. Note: There is also no footnote on the remaining pages.
7. Use the "Matched Schools" data set to print a frequency count and percentage of schools from each county based on the number of times each county appears in the data set. Make sure the

counties are listed in the order shown. Include only the statistics shown on eCampus. There are two titles with no blank line between them for this report.

8. Use the MEANS procedure to reproduce the report beginning on the sixth page of output as posted on eCampus based on the data in the "Matched Schools" data set created above.
    a. There are two titles. There is a blank line between the two titles.
    b. Analysis statistics are only carried out to two decimal places.
9. Use the TABULATE procedure to create essentially the same report as you did in the previous step as shown beginning on page ten of the posted output. Add a line showing the overall statistics at the bottom of the table.
10. Use a single proc step to print the descriptor portion of all data sets in the WORK library.
11. Convert the program and log to PDF files and submit them to eCampus along with your SAS output.