

# Stat 657

## Assignment 13 - SAS

### Scope:

This assignment focuses on the techniques taught in the last few lectures of the semester. You should have all of the information you need by the end of Lecture 22. You will be using the **ncaam06** and **ncaam04** datasets that were used in previous assignments. Be sure to set the system option that will cause additional index information to be written to the log. If you are not already in the habit of doing this, I recommend that you send your output to the output window first until you have finished writing and debugging the entire program so you can see how each one of the steps behaves, especially in part 6.

### Specific Instructions for this Assignment:

1. You will use the **ncaam04** dataset as the source data from which you will create a format.
  - a. Familiarize yourself with the first data step in the sample program **p306d02**. You will convert these commands to SQL to create a similar dataset from **ncaam04**. We are using SQL instead of the data step because we need to have distinct values in the data set for our format.
  - b. The Start value of your format will come from the Team column.
  - c. Use `seed_` as the label in your format. Note, labels are text but the `seed_` column is numeric. Use a `put` function to convert the number to a text. Don't worry that this looks backwards because the range values are usually codes or short values and the label is usually the longer part. We are intentionally switching it around because that suits our purpose.
  - d. Give the new format a name of your choosing.
  - e. After you have created the new dataset, use SQL to insert a row at the end. This row will have values of `start='other'` and `label='NA'` along with the name of your format.
  - f. Use the format procedure to create from this dataset a user defined format in the work library.
  - g. Use the format procedure to write the contents of the new format out to your output document. Don't panic if it looks like your start values are truncated. This output only displays the first 16 characters.
2. Create a picture format that can be applied to PPG column. You can see what the results look like in the output document. Allow space for PPG to be displayed as two digits along with a single decimal place (00.0).
  - a. Values of 15 or more will be in parenthesis preceded by the word High.
  - b. Values of 7.7 or more but less than 15 preceded by the word Medium.
  - c. Values under 7.7 are preceded by the word Low.

3. Use a SAS procedure to place a copy of the ncaam06 dataset in the WORK library. All future references to ncaam06 will be to the WORK copy.
4. Use SQL statements to make the following modifications to the ncaam06 dataset.
  - a. Create a character column labeled PPG Rating. This column needs to be wide enough to accept all values of the picture format label created above.
  - b. Create a character column with a length of 2 labeled 2004 Seeding.
  - c. Change the length and format of the School column so that it is wide enough to fully display Wisconsin Milwaukee.
  - d. Populate the PPG Rating column using the put function to create values by applying the picture format to the PPG column.
  - e. Correct the following school names:  
'Indiana' to 'Indiana'  
'Boston Coll' to 'Boston College'  
'George mason' to 'George Mason'  
'Oral Robt-16' to 'Oral Roberts'  
'Wisc. Milwaukee' to 'Wisconsin Milwaukee'
  - f. Populate the 2004 Seeding column using the put function to create values by applying the format from step 1 to the school column.
5. Create a composite index for ncaam06 using the variables player, school, and region in that order. Print the descriptor portion of the ncaam06 data set after the index has been created.
6. Create a print procedure that will print the columns shown in the sample output from the ncaam06 dataset. Remember that you need the label option in order for labels to appear in your output. You will make 5 copies of the print procedure in your program. Each copy will be modified with different titles, dataset options, and where clauses to demonstrate the use of indexes. Two of these will generate errors in the log instead of producing output. You are not to try to fix those errors but to explain what happened. Create a comment block above each of these sections. In the comment block, write your answers to the questions on each step or an explanation of what happened as directed in the bold text at the bottom of each step below.
  - a. 

```
title '6a. IDXWHERE on Player';  
dataset option idxwhere=yes  
where player in ('Steve Burtt', 'Jared Dudley', 'Stanley  
Burrell');
```

**Is an index used on this step? Why or why not? Do you think the most efficient action was used? Why or why not?**
  - b. 

```
title '6b. IDXWHERE on School';  
dataset option idxwhere=yes  
where school='Texas';
```

**Explain the results of this step.**
  - c. 

```
title '6c. IDXNAME on School';
```

Use the same where clause as step b but change the dataset option to specify your index using idxname.

**Explain the results of this step.**

d. `title '6d. IDXWHERE on Player or School';`  
`dataset option idxwhere=yes`  
`where player in ('Steve Burtt', 'Jared Dudley', 'Stanley Burrell') or school='Indiana';`

**Explain the usage of an index on this step.**

e. `title '6e. IDXWHERE on Player and School';`  
`dataset option idxwhere=yes`  
`where substr(player,1,1)='S' and`  
`school in ('Duke', 'Oral Roberts', 'Iona', 'Boston College',`  
`'Gonzaga');`

**Explain the difference in index usage between this step and the previous step.**

7. After you have completed your program along with the comment blocks in section 6, restart your SAS session and run the program so the log and output will be from a continuous fresh run. Convert your program and the SAS log to PDF files and upload them to WebAssign along with the output from ODS PDF.