ST 555 - Programming HW #6

This assignment has two distinct parts. Section 1 will allow you to work with DO loops to generate data. In particular you will investigate how to randomize a designed experiment using the DATA step. Section 2 will ask you to work with real data by employing DO loops and arrays. While the examples are unrelated, the SAS programming techniques *are* related. Be sure that you clearly delineate in your code (via a comment) where Section 1 ends and Section 2 begins. GPP would dictate that we use two separate programs for this - but that makes the grading more difficult for the TAs, so we'll put all our code into a single file. As always, final data sets should be placed in a permanent library - in this case named HW6.

Section 1: These tasks are enumerated on purpose - you need to complete Item 1 before you move on to Item 2 as they build upon the previous items.

1. Write a data step to generate a data set named DESIGN with three variables: Block, Treatment, and Replicate. BLOCK can take on the values 1, 2, 3, 4, and 5. Within each block the treatments are A, B, C, and D. Within each treatment there should be three replicates. You should have 60 observations and three variables in your final data set. This data set represents a designed experiment (specifically a balanced Complete Block Design) with four treatments and three replications per treatment.

2. One of the fundamental principles of a designed experiment is randomization. To randomize your design via the DATA step you can use the RAND() function. Copy the code you used to create the DESIGN data set and use it to create a data set named DESIGN_RAND_0 that includes an additional variable named RANNUM. To create RANNUM include the following assignment statement:

   ```
   rannum = rand('uniform');
   ```

   Your goal is to generate a random number for each of the 60 records. (Note: I'm not telling you *where* the statement goes in your DATA step - that is up to you to figure out.)

3. Once you've created your DESIGN_RAND_0 data set successfully (check the log!) sort it such that all records for a single block remain together but within a block the records are arranged in ascending order by RANNUM. Name this data set RCBD_0. Open this data set and compare it to our original data set, DESIGN. Can you see that DESIGN_RAND_0 has shuffled the order of the records from DESIGN? (Also, if you've learned about designed experiments before, hopefully you can explain why this is so important!)

4. Now that your code for DESIGN_RAND_0 successfully randomizes the data, and since you are so proud of your work, you decide to show it to a colleague at work. Run your code several times (including the sort!) - after each run open the data set and take a look at the records. What stays the same? What changes? Place your answers in a comment immediately after your PROC SORT code for RCBD_0.

5. In a professional setting (such as consulting) your RCBD_0 code presents a problem - it isn't repeatable. The ideal program would generate a random data set in a repeatable way! (This is not nearly as difficult as it sounds!) Let's again copy your code - this time take the code from DESIGN_RAND_0 and use it to create a new data set called DESIGN_RAND_SEED. As before, you will need to update your code with a new statement:

   ```
   call streaminit(12345);
   ```

   Place this statement anywhere in your DATA step - as long as it's before the assignment statement that creates RANNUM (FYI - it makes the most sense after the DATA statement and before your first DO statement). As before, sort your data and store in a data set named RCBD_SEED. With this new statement in place (which you are not expected to learn unless you want to!) what happens when you run your code multiple times (including the PROC SORT)? What stays the same and what changes in your data sets? What happens if you change the number 12345 to a different positive number? What stays the same and what changes? Again, place your answers in a comment. (The positive integer you are changing is called the **seed**, which explains why I asked you to name your data set DESIGN_RAND_SEED.)

Section 2: These tasks are enumerated on purpose - you need to complete Item 1 before you move on to Item 2 as they build upon the previous items. I've included a screenshot of what part of my data set looks like if you find it necessary. Your data set should look similar.

1. Use the provided FISH data set and sort it by lake type and dam status (LT and DAM). You won't need the latitude or longitude variables for this assignment, so you can drop them.

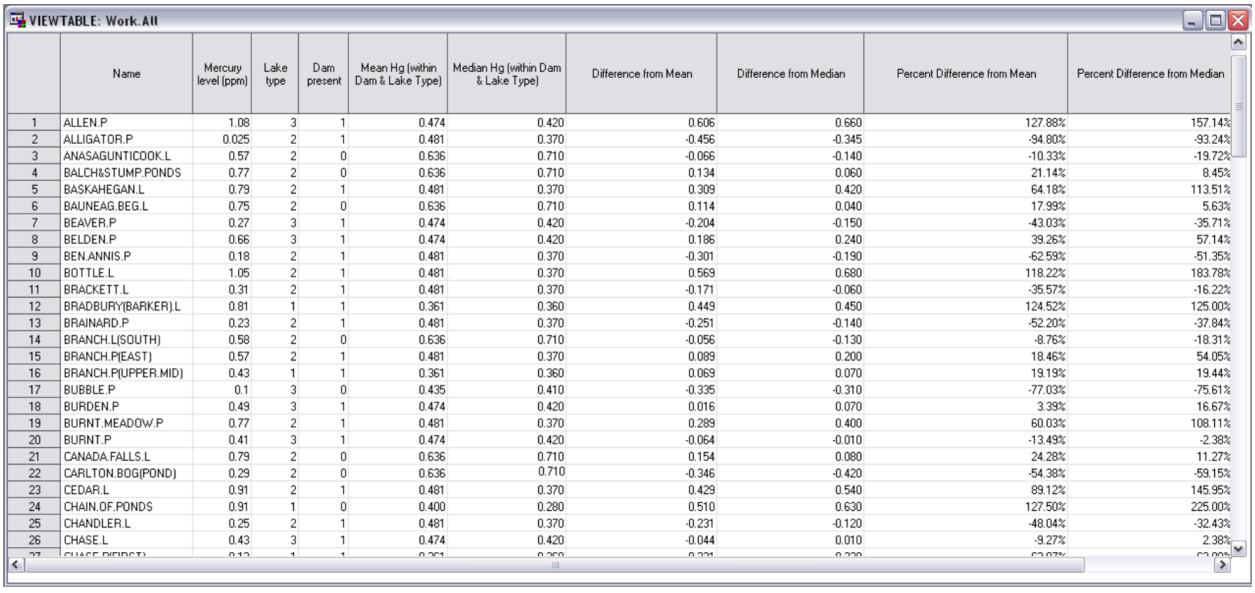
2. Use PROC MEANS to calculate the mean and median for each of the remaining numeric variables (all nine of them) separately for each combination of LT and DAM. (Be sure to decide what to do with missing LT and DAM values - you may want to read ahead to ensure you have a solid plan for how to handle them!) Use ODS to save the results to a data set named STATZ that keeps only the necessary variables.

3. Combine these summary statistics with the original data set using an appropriate join technique to create a data set named ALL. This data set will be required to complete the following items.

4. After the data sets are joined, but in the same data step do the following. You ***must*** use arrays to get credit for this portion of the assignment. (Hopefully you can see why since there are nine variables to work with here...)

    - For each of the nine variables that have a mean and median computed you also need to correctly compute the following for each of the 9 variables. (That means no warnings or errors - syntax or otherwise!)
        - Difference from the mean [variable - mean.of.variable]
        - Percent difference from the mean [(variable - mean.of.variable)/mean.of.variable]
        - Difference from the median [variable - median.of.variable]
        - Percent difference from the median [(variable - median.of.variable)/median.of.variable]
    - Apply reasonable formats and labels to all the variables that weren't originally in the FISH data set.
    - Drop any irrelevant variables.

5. Sort your ALL data set by NAME.



VIEWTABLE: Work.All

	Name	Mercury level (ppm)	Lake type	Dam present	Mean Hg (within Dam & Lake Type)	Median Hg (within Dam & Lake Type)	Difference from Mean	Difference from Median	Percent Difference from Mean	Percent Difference from Median
1	ALLEN.P	1.08	3	1	0.474	0.420	0.606	0.660	127.88%	157.14%
2	ALLIGATOR.P	0.025	2	1	0.481	0.370	-0.456	-0.345	-94.80%	-93.24%
3	ANASAGUNTICOOK.L	0.57	2	0	0.636	0.710	-0.066	-0.140	-10.33%	-19.72%
4	BALCH&STUMP.PONDS	0.77	2	0	0.636	0.710	0.134	0.060	21.14%	8.45%
5	BASKAHEGAN.L	0.79	2	1	0.481	0.370	0.309	0.420	64.18%	113.51%
6	BAUNEAG.BEG.L	0.75	2	0	0.636	0.710	0.114	0.040	17.99%	5.63%
7	BEAVER.P	0.27	3	1	0.474	0.420	-0.204	-0.150	-43.03%	-35.71%
8	BELDEN.P	0.66	3	1	0.474	0.420	0.186	0.240	39.26%	57.14%
9	BEN.ANNIS.P	0.18	2	1	0.481	0.370	-0.301	-0.190	-62.59%	-51.35%
10	BOTTLE.L	1.05	2	1	0.481	0.370	0.569	0.680	118.22%	183.78%
11	BRACKETT.L	0.31	2	1	0.481	0.370	-0.171	-0.060	-35.57%	-16.22%
12	BRADBURY(BARKER).L	0.81	1	1	0.361	0.360	0.449	0.450	124.52%	125.00%
13	BRAINARD.P	0.23	2	1	0.481	0.370	-0.251	-0.140	-52.20%	-37.84%
14	BRANCH.L(SOUTH)	0.58	2	0	0.636	0.710	-0.056	-0.130	-8.76%	-18.31%
15	BRANCH.P(EAST)	0.57	2	1	0.481	0.370	0.089	0.200	18.46%	54.05%
16	BRANCH.P(UPPER.MID)	0.43	1	1	0.361	0.360	0.069	0.070	19.19%	19.44%
17	BUBBLE.P	0.1	3	0	0.435	0.410	-0.335	-0.310	-77.03%	-75.61%
18	BURDEN.P	0.49	3	1	0.474	0.420	0.016	0.070	3.39%	16.67%
19	BURNT.MEADOW.P	0.77	2	1	0.481	0.370	0.289	0.400	60.03%	108.11%
20	BURNT.P	0.41	3	1	0.474	0.420	-0.064	-0.010	-13.49%	-2.38%
21	CANADA.FALLS.L	0.79	2	0	0.636	0.710	0.154	0.080	24.28%	11.27%
22	CARLTON.BOG(POND)	0.29	2	0	0.636	0.710	-0.346	-0.420	-54.38%	-59.15%
23	CEDAR.L	0.91	2	1	0.481	0.370	0.429	0.540	89.12%	145.95%
24	CHAIN.OF.PONDS	0.91	1	0	0.400	0.280	0.510	0.630	127.50%	225.00%
25	CHANDLER.L	0.25	2	1	0.481	0.370	-0.231	-0.120	-48.04%	-32.43%
26	CHASE.L	0.43	3	1	0.474	0.420	-0.044	0.010	-9.27%	2.38%