Specifications:

- You are allowed at most one `DATA` step and ten `PROC` steps to complete this assignment.
- Create the InputDS *libref*, RawData *fileref*, and a *libref* – named HW4 – associated with your HW4 storage location.
- We should be fairly comfortable now using macro variables for repetitive code. When you place macro variables in the code, please define them near the top of your program with the setup code. Specifically, use at least the following macro variables:
  - *Year*: This is used to set the year in which the data was collected. Anytime you need to refer to the year (e.g., when defining your custom format for quarters!) use this macro variable so you don't have to type 1998 a bunch of times!
  - *CompOpts*: This is used to set the options for PROC COMPARE that are consistent throughout the program. Put all the options you don't need to change in here, then use this variable in your independent validation steps.
- This assignment is part of a larger data project for which the data we need is split across five raw text files: LeadProjects, O3Projects, COProjects, SO2Projects, and TSPProjects.
  - This week we will ***only be working with the first file*** - LeadProjects - which contains data on some costs associated with cleaning up lead pollution.
  - *Program defensively!* Because this is part of a larger data analysis project, avoid making unnecessary assumptions/decisions about the data. E.g., don't assume a variable will always have non-missing values.
  - Save formats to your HW4 library so you can use them again. The format you use to place the dates into quarters must be named MyQtr – it will make your life easier on HW5! ☺]
- The data-reading goal for this week is for you to assess the issues present in the LeadProjects file, read the data in, clean the data until all issues have been addressed, and carry out whatever self-validation you need to ensure you have found all the issues.
  - We do not need to see your self-validation code, but it makes sense to leave it in your file - just make sure it is commented out so we do not grade it!
  - Because you will be handling the other files in the future, the kinds of data cleaning you do this week will need to be expanded on later. So, make sure you're happy with it!
- ***After self-validation***, it is time to move on to the independent validation. For independent validation, my data set is named HW4DugginsLead; name yours similarly. (For example, Tony Stark would save his as HW4StarkLead.)

  Electronically validate both the descriptor and content portions of your data set, remembering to validate the metadata first. When carrying out the validation, numeric variables must have values within 1E-15 to be considered equivalent. To ensure your validation lines up as intended, be sure to plan ahead (as always) but also be sure to get rid of the variable called Member from your descriptor portion – our last names aren't the same, so our data set names won't be equal either! For the validation of the metadata, name the resulting data set DiffsA. For the validation of the data, name the resulting data set DiffsB.
- Once you've validated your source data, the analysis goal this week is to produce visualizations of some characteristics of this data. To do this, we are making a few numeric summaries and two bar graphs: HW4Pctile90 and HW4RegionPct. The first bar graph shows the 90^{th} percentile of total job cost for each region and quarter combination and the second graph shows the percent of total costs for a region across quarters. You should produce identical graphs based on your data set while following the guidelines below. (I advise you to make the 90^{th} percentile graph first as it is the simpler of the two!)
  - The numeric summaries and the graphs should appear in a PDF that I've called *HW4 Duggins Lead Report.pdf*. Name yours appropriately.
  - Additionally, both graphs should be saved in your HW4 directory as PNG files without appearing in any additional destinations and the numeric summaries should not appear in any other destination.
  - All data is from 1998 and dates are grouped into quarters using a calendar-year approach: Quarter 1 is January - March, Quarter 2 is April - June, Quarter 3 is July - September, and Quarter 4 is October - December. (This is the custom format I was talking about earlier!)

- For the 90^{th} percentile plot, note the following:
  1. When getting the data for the graph, make sure you do not request more statistics than we need - keeping the computational time low and the data set size small both help improve efficiency when working with large data sets, so it is important for us to practice this on smaller data sets. ***This data set should be 24 rows and 4 columns.***
  2. When computing statistics, obviously sample size is important, so I've printed the sample size along with each bar in the graph. Use the DATALABEL= option to identify the variable you want printed with each bar. I used a font size of 6pt here because it was the largest that would fit without us needing to learn even more options!
  3. No other changes (like sizes, colors, etc.) were made beyond those evident from looking at the graph itself.
- For the remaining graph, there are a number of items to keep in mind:
  1. To emphasize efficiency, the data set you use to make the graphs should be as small as possible. Control this by using only the rows and columns that are absolutely necessary; for this graph, that means 4 variables and 24 records. ***If you think you can do it with only 3 variables, you're overlooking something important!***
  2. You should set the colors for the bar fills. I don't care if you use my color scheme or not, you just need to pick 4 colors! (If you're not great at picking colors, feel free to head over to colorbrewer2.org to pick some colors.)
  3. For both axes, the labels are 16pt. For the x-axis, the values are 14pt and for the y-axis they are 12pt. The grid lines are set to be thicker than normal (3 instead of 1) and are gray. (I used grayCC, but you can use any shade of gray you like.)
  4. The graph is placed in 95% of the vertical space available, with the top 5% not being used. (It is just a buffer to space the graph out from the top of the box.)
- Both graphs use a resolution of 300 DPI and are 6 inches wide in both the listing and PDF destinations.
- If you don't remember how to set the name of the image, the resolution, or the physical size, check back in Chapter 1 – it was covered there.
- To practice validating an object, I've saved the data set *produced* by my second plotting step as HW4DugginsGraph2. Save your data as well, and validate your data set (just the content portion) against mine. (Be sure to note what gets automatically compared here versus what you would still have to manually compare to ensure they really are the same graph!) Name this validation data set DiffsC.

** As with previous assignments, but especially with graphs, there may be differences that occur because you are operating on the VTL when producing output outside of SAS. Before spending countless hours agonizing over what could have gone wrong, please just ask if the discrepancy you're seeing is something we can (and should) control at this point in the class! (Keep in mind the graphs, and the assignment as a whole, can be done entirely with code from this course!)

** If you cannot open the images directly from SAS and didn't get that fixed during your past PPC, make sure you choose something other than Paint as your default program for opening images.