First-time Directions:
In addition to completing the tasks given below, all programming assignments should include the following elements. These will not be repeated on future assignments, unless changes are necessary.

- Every program you submit for grading must include a header section that includes the full name of everyone in your group, the last date you edited the program, and a brief description of the intent of the program. (It may be easiest to copy one of my headers and make the necessary updates.)
- Your program must include clear and concise comments to explain the intent of your code. Keep in mind that comments are not just for you - they are also for your colleagues and clients (and TA). There are typically multiple ways to achieve the same result, so use brief comments to explain your reasoning.
- The only file you need to submit is your SAS code. The file must be saved as a .SAS file in the format *MPXGroupX.sas*. For example, if I'm in Group C then my group's submission would be named *MP1TeamC.sas* for the first mini-project. No other submission types will be graded (i.e. you will get a zero on the assignment.) The TA can submit your code to see your output and log. For this reason it is important that you use libraries with the specified names.
- Assignments are graded based on the rubric provided on the Moodle page. Please read through the rubric and make sure you understand it. Because the same rubric is used for every assignment you will always know what is expected of each program. The rubric weights **do** change across assignments as we focus on different concepts. The weights for each assignment are posted with the due date and any assignment documents. (Note this is the same rubric I use in ST 445.)
- Finally, while I can't require you to do the following - I strongly suggest you:
  - Before submitting your code, close and reopen SAS then run your entire program. Doing so ensures that no forgotten options or data sets are affecting your output. This should create a scenario in which you and the TA are looking at the same output.
  - Even after submission (or before, if you prefer) you should take time to experiment with the code. Each assignment will be asking you for a specific set of tasks, but that does not mean it requires mastery of a programming technique. Try changing variables, or formats, or programming the same output a different way and see how it affects your results. Then figure out why! As an example, I won't ask you about the IB and PDV contents or the location of pointers on every assignment, but you should always know what is going on behind the scenes!

- (5%) Coding Budget: You are allowed at most two `DATA` steps and nine `PROC` steps to complete this assignment.

Specifications:

- (5%) Setting up paths: Programmatically change your working directory to point to `L:\ST446\MP1` then set up a *fileref* for the params.txt file that takes advantage of your newly-assigned working directory. Set up a library, named MP1, for all future files you create in this program. Place it on your S drive. Also update your working directory to use this location as well.
- (20%) Simulating: Read in the data set mentioned above into a permanent data set named Sims. The data set contains different parameters for simulating data: Dist, Reps, and Size. The Dist variable simply names the distribution from which you should simulate the data. The Reps variable contains how many new samples need to be generated from the named distribution. The Size variable contains the sample size for each replication. Each line of data in the raw file represents a different combination of Dist and Reps. All the Size values are listed on a single line.

  You'll need to simulate data using the provided parameters. To make sure we get the same results, use a seed of 100 to generate your data. My Sims data set is provided on the shared drive as well. (You might want to compare your simulated data to mine before you worry about the rest of this!)
- (20%) The purpose of these simulations is to explore the sampling distribution of common summary statistics. To facilitate that for this and future mini-projects, you'll need to do the following in this program:
  - Set up a macro variable that names the summary statistic. Valid values of this macro variable are any statistic keyword that you can use in the SUMMARY/MEANS/REPORT procedures.
  - Calculate the specified summary statisic so that you are prepared to plot the sampling distributions as described below.
  - Calculate the minimum and maximum of the summary statistic (across all simulations for that distribution) and save those as macro variables to use in the graphs as described below.
- (50%) To visualize the sampling distributions we'll use histograms overlain with density plots.
  - This assignment requires you to make over 300 graphs (not kidding!) so set up a macro variable to simply control any BY processing. Set up a separate macro variable to define all the options used when setting up the legend for your graphs.
  - Ideally we'd like to compare *all* the combinations of parameters specified by the user in the params.txt file. Begin by using BY-processing that produces graphs for all combinations simultaneously. The name prefix for these images should be histoNoMac, they should be 4 inches wide, and should have a resolution of 300 DPI. All my graphs are in the same location as the data sets I provided above. Your graphs should be ***identical*** to mine and should use ***no hardcoding*** for titles, footnotes, or any other aspect of the graph unless I approve it.
  - Next, set up macro variables that allow the user to specify a value of Dist, Reps, and Size. Use those to generate the graph specified by the combination of macro variables. Use the same resolution and physical size as with the previous graphs, but the graph name should now be styled using the values of the macro parameters. For example, a graph of the normal distribution that uses 25 reps and a sample size of 30 should be named NormalR25N30.

    Use this approach to create graphs for the following scenarios: Cauchy and Normal, both with sample sizes of 2, 30, and 100. For all six use the case where we've taken 25 samples.

    (Note: I know we didn't cover this in every ST 445 class, but to get the left justification in your footer use J=L and use H= to control the size. (I used 4pt font.) Syntax requires those options to come between the keyword token that starts the statement and the literal token.)
  - Be sure to clear your titles and footnotes after each graph.

Bonus If your team finishes up all of the graphics portions, I encourage you to try to also generate the report Frequency Report.pdf that is included with the rest of the files. If you want to tackle it you get an additional three PROC steps added to your coding budget. Hint: they are FORMAT, FREQ, and REPORT. [Note: This is NOT graded. If you attempt it, leave it commented out when you submit your code for grading!]