

Creativity3.sas: Explanation of code

These explanations describe what is new. You should already know how to read the data. That is necessary for this to be a standalone program. Reading the data explained in the creativity1 and creativity2 documentation, not again here.

Goals of code:

- Clearing log and results windows
- T-test with equal variances

Clearing log and results windows:`dm log 'clear; dm odsresults 'clear';`

Note: log and odsresults are keywords (not in quotes); clear is a text string, so it must be in quotes. If you get a message like options ignored you got quotes in the wrong place(s).

Normally, SAS appends information to the log and results windows. This is exactly what you want when you write to extend an analysis. It is not helpful when you are trying to get code to run; the log window gets filled with incomplete attempts. It is possible to clear the log (and/or) the results windows so the only contents are the last execution of your code.

dm commands control the Display Manager, which manages all the SAS windows. If you want to do something to the log window, you use `dm log` followed by some command. If you want to do something to the results window, you use `dm odsresults` followed by some command.

The command to clear all contents of a window is `'clear'`. Note that the command is in quotes.

Two sample t-test: `proc ttest;`

Proc ttest requests a two-sample t-test. Like with proc npar1way used last week, the class statement names the groups and the var statement names the response.

Proc ttest requires that there be only two groups and will stop if it finds three. Some ways of reading data create a dummy record from the header line in the data file (containing the variable names). The response variable has a missing value, so it won't be included in the computations (a good thing). However, proc ttest checks the number of groups before deleting observations with missing values, so the data set appears to have three groups. Fix reading the data, e.g. by starting reading at the second observation (`infile ... firstobs=2;`) and the problem will go away.

SAS gives you lots of numbers. Some are more useful than the others. How to find quantities that you might want:

- Averages for each group: First box, Mean column
- Group-specific sd: First box, Std Dev column

- Pooled sd: First box, Diff row, Std Dev column, or Second box, Pooled row, Std Dev column.
- df for pooled sd: Third box, Method Pooled row, DF column.
- se for each group using group-specific sd: First box, Std Err column
- Estimated difference in means: First box, Diff row, Mean column
- se of estimated mean, using pooled sd: First box, Diff row, Std Err column
- 95% confidence interval for difference: Second box, Diff / Pooled row, 95% CL Mean columns
- two-sided p-value for test of difference = 0: Third box, Method Pooled, Pr > —t— column.

The lines labelled Satterthwaite are for the “unequal” variance (Welch) version of the t-test.

Ignore the columns labeled 95% CL Std Dev. These are confidence intervals for the standard deviation. Any estimate can have a confidence interval, so it is possible to compute a confidence interval for the variability. We will not use or discuss these in 587.

The numeric results are followed by plots showing the distribution of values and a Q-Q plot to assess the assumption of normality. We’ll talk about the Q-Q plot in a week or so.

non-95% confidence intervals: `proc ttest alpha=0.10;`

Proc ttest reports 95% intervals by default. To compute intervals with other coverages, specify `alpha =` on the proc ttest line. This is an option to proc ttest, so `alpha =` must come **before** the `;`. You specify `alpha = 1 - your desired coverage`. `alpha=0.01` gives you 99% intervals, `alpha = 0.10` gives you 90% intervals.

None of the other output changes - just the confidence intervals.