pairedt.sas: Explanation of code

Goals of code:

- useful tricks working with SAS
- reading observations from a pre-existing SAS data set
- paired t-test by creating a variable with the difference
- paired t-test without calculating differences
- Wilcoxon signed rank test

Some useful tricks when working with SAS:
clearing the log window: `dm 'log; clear';`
You may have to fix multiple errors in your code. If so, the log window will accumulate SAS's responses to all the erroneous code. It is often useful to clear the contents of the log window. There are various ways to do this. The "code" way is the command `dm 'log; clear';`
Note the quotes (single or double) and the semicolon between log (identifying the window) and clear (what you want to do to that window).

clearing the results window: `dm 'odsresults; clear';`
Done in a similar way. The window is called odsresults. The results window disappears, but it reappears the next time you request output.

keyboard shortcuts:
There are shortcut keys for many SAS mouse choices.
clearing a window: ctrl-e
Click in a window to make it active, then hold down the Ctrl key and type e. Mac uses may have a different key combination; look for the equivalent of the "control" key.

You will need to click in the editor window to make it active again. If you don't, the Run button will not do what you expect.

If you accidently clear the editor window, type ctrl-z to undo the clear.
If a new "Program editor" window opens up when you click the Run button, you forget to make the editor window active before hitting the Run button. Close the program editor window, make the editor window active, and you'll be able to run your code.

submitting SAS code: F3 key
The F3 key submits code and runs it. A laptop with a small keyboard may need a combination of keys.

Schizophrenia data:
The example this week is the schizophrenia data discussed in lecture. The data are in case0202.txt. This is a space delimited file, so we use a data step to read the file. Because these are paired data, all values for a pair of individuals are on one line. The value for the unaffected individual is first; that for the affected individual is second.

This is often called "wide" format data. Two variables for the two pieces of information. "long" format data has one response on a line, two additional variables indicate which pair it is from and which treatment it is from. SAS can convert between formats. We'll talk about that later.

Note: There is a way to use proc import to read a space delimited file. But, there is a subtle and sometimes important difference between between .csv and .txt files. Two successive commas in a .csv file indicate a blank field in between, so 1,,2 provides values for three variables: a 1, a missing value, then a 2. Multiple spaces in a space delimited file are often used to line up columns for ease of reading. So 1 2 (two spaces in between) and 1 2 (one space in between) are (usually) the same information: values for two variables, a 1 and a 2. The data step treats one space and multiple spaces identically. Proc import treats them as if it were a csv file with spaces instead of commas, so 1 2 becomes 1, missing, 2. Would I like proc import to be smarter: definitely. But, it isn't.

Takehome message: I recommend using a data step to read txt files.

diff = unaff - aff; Creating a new variable in a data step
Once a line of data has been read in, you add code to do any sort of computation. These are additional commands, each ending in a ;. We want to compute the difference between the unaffected and affected values. We can easily do this because the two values we need are on the same line of data.

Note: This line, to compute the difference, must go **after** the input command. That's because we have to read in the values before we can do the subtraction.

Reading observations from a pre-existing SAS data set: `data schiz; set schiz0;`
The above code, using a data step to read lines of data then compute the difference, works when the data file is space delimited and so is read using a data step.

When you need to read .csv or .xlsx data files, then compute the difference, you need to do something different. That's because proc import only reads data to create a SAS data set. It doesn't allow you to do any computations. With a few advanced exceptions, computations can only be done in a data step.

The `set` command in a data step reads observations from a pre-existing data set. You can then follow that with any necessary computations.

The proc import ... out=schiz0; step reads the schiz.csv file and creates the schiz0 data set.
The data step that follows creates the schiz data set (`data schiz;`). `set schiz0;` reads observations from the schiz0 data set (the one created by the proc import). After reading in an observation,

`diff = unaff - aff;` calculates the difference.

Paired t-tests using a variable containing the difference: `Proc means`
If you have created a variable with the difference (see previous section), you can use `proc means` or `proc univariate` to do the paired t-test. Remember, this is a one-sample T test of whether the mean difference = 0. Both proc means and proc univariate provide summary statistics for a sample of data. They differ in what they do and how they organize the output.

Using `proc means`: The default output from proc means includes the average, standard deviation, minimum and maximum values. You can request quite a bit more by adding the appropriate keywords to the proc means statement. These keywords are part of the proc means statement, so they go between proc means and the semicolon that ends that statement. The options in this code are useful for a paired t-test.

mean: sample average	std: sample standard deviation
stderr: standard error of the average	t: T statistic testing mean = 0
prt: p-value for the test of mean = 0	clm: 95% confidence interval for the mean difference

As with other procs, the `var` statement names the variable (or variables) to analyze.

Using `proc univariate`: We've already used proc univariate to compute summary statistics. There are many more useful numbers in the output:
In the output box labeled "Tests for Location: Mu0=0", you have 3 tests of location = 0
Student's t: (first row) The paired t-test. The Statistic column is the value of T, the two-sided p-value is in the p-value column.
Signed Rank: (last row) The Wilcoxon signed-rank test. The two-sided p-value is in the p-value column.
Sign: (middle row) Another non-parametric test of location = 0. Not emphasized in this class.

paired t-test without computing differences first: `proc ttest; paired;`
SAS provides a way to compute the paired t-test without first computing the differences. This is the `paired` command in proc ttest. Without `paired`, proc ttest does a two-sample test. With `paired`, it does the paired t-test. The paired statement names the two variables containing the paired responses. These are separated by an asterisk.

The output includes:
which way around the difference is being taken, here unaff - aff.
a summary of the differences: average, standard deviation, standard error, smallest and largest value
the average and a 95% ci for the mean. That box includes a ci for the sd; we will ignore that.
the paired t-test: degrees of freedom, T statistic, and two-sided p-value.

The numeric output is followed by four plots, which may be of interest:
A histogram of the differences, with a horizontal box plot, and two smoothed histograms
A profile plot. This plots the two variables (Y = value, X = variable) and connects the pair from

each subject with a line. If these lines are all parallel, all subjects have about the same difference.
An XY plot with X = one variable and Y = the other. This allows you to visualize the correlation between the two responses (we'll talk about correlations in a few weeks).
A quantile-quantile plot of the differences. This assesses normality of the differences. We'll talk more about QQ plots in a few weeks.

Wilcoxon signed rank test: proc univariate
Use proc univariate to summarize the differences. As described above, the output box labeled "Tests for Location: Mu0=0" gives you Wilcoxon signed rank test results. Look at the last row labeled Signed Rank. The two-sided p-value is in the p-value column.