vitC.sas: Explanation of code

Goals of code:

- Reading a contingency table in “count” form
- Contingency table tests, proportions, and measures of association
- Chi-square test
- Reading event data in “individual” form
- Exact tests for general contingency tables (more than 2x2).

Note: There are **many** different methods to analyze contingency table data. Everything we've done previously in the course has had alternative methods. Contingency table data has **a lot** of alternatives. The answers are usually only slightly different, especially with moderate to large sample sizes. I show how to get large sample results and one of the small sample (“exact”) results.

Reading “count form” data: `input row $ col $ n;`
When the counts in a contingency table are pre-tabulated, the easiest way to do stats on that table is to enter the table with one row of data for each cell (combination of row and column) in the table. The key piece of information is the number of observations in that cell. Here we store that in the n variable. The row and column identifiers can be stored as character or numeric variables. n must be numeric.

The table can be stored in a data file or, as here, entered using `cards;` or `datalines;`.

Contingency table computations: `proc freq;`
The basic command in proc freq; is `table`. This counts the numbers of observations for each level of the specified variables. `table var1;` will classify observations by their value of `var1`. `table var1*var2;` will classify observations by each combination of `var1` and `var2`. You can continue adding `*var` terms to classify observations by levels of 3 variables (a 3-way contingency table), 4 variables, $\cdots$. These instructions focus on 2-way contingency tables.

For a 2-way contingency table, the first var specifies the rows; the second specifies the columns. I find it more useful to put “treatments” as rows and “responses” as columns.

For the Vit C data set, we want a 2-way contingency table with cells defined by combinations of `trt` (what treatment a person was assigned to) and `cold` (whether or not they got a cold that winter). Because cold is the response, we want that variable to define columns and trt to define the rows. Hence, `table trt*cold`.

Describing “count” data:`weight n;`
By default, proc freq will count the number of observations in each cell. That is demonstrated at the

end of the example code, where we have "individual" data. When the data are already summarized, we need to tell SAS that each row represents multiple observations. That is what `weight n;` does. Tells SAS that the variable n contains the number of individuals associated with that row of data.

Aside: You can have multiple rows, each representing multiple individuals. For example, if the vit C study had been done in multiple cities, your data might be four rows (one for each combination of trt and cold) for each city. `table trt*cold; weight n;` will then report and analyze the total summed over all cities. This form of data would be very useful because you could use it to analyze the total for all cities, or analyze each city separately (e.g., by adding a `by city;` to the proc freq).

Controlling output from proc freq: `table` statement options
By default, proc freq reports multiple numbers for each group. For a 2-way table, SAS wants to report the count, the percentage of the row total, the percentage of the column total, and the percentage of the overall total. This is too much for me. I usually only want the count.

To suppress the extra output, I add the options `norow nocol nopercent`. All go after a `/` that separates the core table statement from its options. You can mix and match these options. `norow` suppresses the row percentage, `nocol` suppresses the column percentage, and `nopercent` suppresses the overall percentage. Later in the code, the row proportion is useful, so I omit `norow` to see the row percentage but not the other two.

Chi-square statistic: `chisq;` option
The `chisq` option requests the Chi-square test of no association. This is reported in the `Statistics for Ta` piece of output. The Chi-square statistic is the first line. SAS also reports other statistics used in other fields. If you know about the g-statistic, that is called the `Likelihood Ratio Chi-square`, which is a more appropriate name than g. The `Continuity Adj. Chi-square` is the Chi-square statistic with a continuity correction (discussed briefly in class). The other statistics are not relevant (for us, now).

When the 2-way table is a 2x2 table, SAS reports the Fisher exact test automatically. The output is in the `Fisher's exact test` block of output. The line you want is labelled `Two-sided Pr <= P`. This is the two-sided p-value that is the customary result from Fisher's test. The other numbers are intermediate calculations that will be of interest only if you want a non-customary analysis.

Odds ratio and relative risk statistics: `measures` option
When you add the `measures` option to the table statement, you get two more blocks of output: one without a title and one labeled `Estimates of the Relative Risk (Row1/Row2)`. The untitled block contains quite a few statistics that are commonly used in the social sciences. If you recognize and want one, here is where you find it.

We are interested in the `Estimates of the Relative Risk (Row1/Row2)` output. The first row is the Odds ratio. Note this is the odds ratio, not the log odds ratio. The odds are calculated as odds of column 1 in row 1 to odds of column 1 in row 2. If you want odds of the event in column 2, you need the reciprocal of the number reported here.

If you know about relative risk, that is also computed with the measures option. The second and third rows are the relative risk, calculated as the risk for row 1 / risk for row 2. The second row calculates the risk of column 1; the third row calculates risk of column 2.

Confidence intervals for all measures: `cl` option
If you both the `cl` and `measures` options, you get confidence intervals for each measure. These are large sample confidence intervals. Again, the only one needed for this class is the ci for the odds ratio. All three confidence intervals are calculated on the log measure and back transformed.

Contingency tables from individual observations:
SAS can calculate the counts from individual level data, where each row represents one individual. You need their group and their output. Because each individual is adds one to the total count, you do not need any n variable. The vitclong data set is the long version of the vitc data. There is one row of data for each individual in the study. The two variables indicate that persons treatment and response. The `proc freq` code is identical to the previous code with one exception: there is no `weight` statement. SAS will construct and report the table of counts, then do the additional calculations you request.

Exact tests with tables larger than 2x2: `exact chisq`
SAS reports the Fisher exact test by default when the contingency table is 2x2. When the table is larger, the Fisher test isn't defined, but randomization can still be used to get an exact p-value. The exact test would be more appropriate when sample sizes are small and necessary when sample sizes are really small (e.g., most expected counts < 5).

The exact test is obtained by requesting the desired test on the table statement (`chisq` option) and requesting the exact version using `exact chisq;`. If the problem is large and you still want exact intervals, you can get a Monte-Carlo (sampling) estimate of the p-value using `exact chisq / mc;`.

Note: If you want exact ci's for the odds ratio, you can request that by `exact or;`.