

Examples

First, we set the seed for reproducibility

```
. log using sampling
. set seed 987654321
```

and then draw a sample of size 100

```
. swor 100
```

We can also keep all `sex` not equal to 0, but sample 100 from `sex` equal to 0

```
. swor 100 if sex == 0
```

or sample 100 from each group of `sex`

```
. swor 100, by(sex)
```

or sample 100 from each group of `sex`, but keep all the observations

```
. swor 100, by(sex) gen(sample) keep
```

Finally, we can sample 100 if `sex` is 0 and keep only those observations.

```
. swor 100 if sex == 0, gen(sample)
. keep if sample
```

References

- Gleason, J. R. 1997. ip18: A command for randomly resampling a dataset. *Stata Technical Bulletin* 37: 17–22. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 77–83.
- . 1999. ip18.1: Update to resample. *Stata Technical Bulletin* 52: 9–10. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, p. 119.
- Weesie, J. 1997. dm46: Enhancement to the sample command. *Stata Technical Bulletin* 37: 6–7. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 37–38.

sbe36.1	Summary statistics for diagnostic tests
---------	---

Paul T. Seed, King's College London, UK, paul.seed@kcl.ac.uk

Aurelio Tobias, Hospital de la Santa Creu i Sant Pau, Barcelona, Spain, atobias@cochrane.es

Abstract: An extensive revision of the `diagtest` command introduced in Tobias (2000) is introduced and illustrated.

Keywords: diagnostic test, sensitivity, specificity, predictive values, confidence intervals, contingency tables.

Introduction

The new command `diagt` is a complete revision of the earlier command `diagtest` (Tobias 2000). It should be regarded as succeeding it and replacing it. Both commands are used to assess a simple diagnostic test in comparison with a reference standard (or “gold standard”), assumed to be completely accurate. The diagnostic test is generally used because it is cheaper, quicker, or less invasive than the reference standard, but may not be as reliable. Results are typically presented in a 2×2 table and summarized as four percentages: sensitivity, specificity, positive predictive values, and negative predictive values, with their respective confidence intervals. `diagti` is an immediate version of `diagt` that does not require data to be entered.

The commands `diagt` and `diagtest` differ in the following ways:

- The exact binomial distribution is used instead of the normal approximation, based on command `ci`.
- An algebraic error that caused `diagtest` to give confidence intervals that were too wide has been corrected. Essentially, `diagtest` used the grand total instead of the row or column totals when calculating the standard errors.
- Only `fweights` are allowed, as with `ci` and `binomial`.
- The command format is changed. The outcome (the true disease status) is placed before the predictor (the diagnostic test). This makes it consistent with commands such as `logistic`, `roctab`, `cs`, and `cc`.

- The table layout is also changed. Rows represent disease status. As with commands `cc` and `cs`, rows represent true disease status, columns represent exposure (or test) result. Positive results are given in the top and left of the table, before negative ones.
- An immediate version is provided.
- The name has been changed. This is partly because the shorter name allows the immediate version to end in `i`, and partly to reduce the risk of people confusing the two command formats and so getting the wrong estimates.

Sensitivity is the proportion of true positives that are correctly identified by the test, and specificity the proportion of true negatives correctly identified. Typically, these are regarded as fixed properties of a particular test, independent of the prevalence of the disease. However, in medical practice, the result of the diagnostic test is all that is known. The positive predictive value (PPV) is the proportion of patients with positive test results who are correctly diagnosed. The negative predictive value (NPV) is the proportion of patients with negative test results who are correctly diagnosed. These values help a clinician trying to make a diagnosis for a particular patient.

True disease status	Test result		
	Positive(+)	Negative(-)	
Abnormal(+)	a	b	\implies Sensitivity = $a/(a + b)$
Normal(-)	c	d	\implies Specificity = $d/(c + d)$
	\Downarrow	\Downarrow	
	PPV = $a/(a + c)$	NPV = $d/(b + d)$	

Table 1. Definition of sensitivity, specificity and predictive values.

The predictive values of a test in clinical practice depend critically on the prevalence of the abnormality in the patients being tested; these values depend on the prevalence of the disease. Sometimes the prior likelihood of the current patient having the disease can be estimated from other signs and symptoms. This can be used instead of the prevalence.

The predictive values (PPV and NPV) can be calculated for any prevalence by using Bayes' theorem, as follows:

$$\text{PPV} = \frac{\text{Sensitivity} \times \text{Prevalence}}{\text{Sensitivity} \times \text{Prevalence} + (1 - \text{Sensitivity}) \times (1 - \text{Prevalence})}$$

$$\text{NPV} = \frac{\text{Specificity} \times (1 - \text{Prevalence})}{\text{Specificity} \times (1 - \text{Prevalence}) + (1 - \text{Specificity}) \times \text{Prevalence}}$$

Syntax

```
diagt diagvar testvar [weight] [if exp] [in range] [, prev(#) level(#) tabulate_options ]
```

```
diagti #a #b #c #d [, prev(#) level(#) tabulate_options ]
```

where *diagvar* is the variable which contains the real status of the patient, and where *testvar* is the variable which identifies the result of the diagnostic test. *testvar* and *diagvar* can have only two nonmissing values. The higher value must identify the positive result of the test or the diseased status of the patient.

For `diagti`, #_a, #_b, #_c, and #_d are, respectively, the numbers of true positives (diseased subjects with correct positive test results), false negatives (diseased, but negative test), false positives (no disease, but positive test) and true negatives (no disease, negative test).

`fweights` are allowed.

Exact binomial confidence intervals are given, as with the command `ci`.

Options

`prev`(#) specifies the estimated prevalence, in percent, of the disease to be used in estimating the positive and negative predicted values using Bayes' theorem. If the `prev` option is used, the confidence interval is only displayed for the sensitivity and specificity values. Otherwise, the prevalence is estimated from the data.

`level`(#) specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`.

All `tabulate` command options are available.

Example

The same example is considered here as in Tobias (2000) describing `diagtest`. Altman and Bland (1994a, 1994b) consider the relationship between the results of a liver scan test and the correct diagnosis (Drum and Christacopoulos 1972). The proportions that were correctly diagnosed by the scan were 89.53% for normal liver scan, and 62.79% for those with abnormal scan. The proportion of correct diagnoses among the patients with abnormal liver scan test was 87.83%, and among the patients with normal liver scans such proportion was 66.67%.

Here are summary results for the liver scan test and the correct diagnosis.

```
. diagt diag test [fw=n]
      |      diagnostic test
      |      result
diagnostic |      Pos.      Neg. |      Total
-----+-----+-----+-----
  Abnormal |      231      27 |      258
   Normal |      32      54 |      86
-----+-----+-----+-----
      Total |      263      81 |      344
True abnormal diagnosis defined as diag = 1 (labelled Abnormal)
                                         [95% Conf. Inter.]
-----+-----+-----+-----
Sensitivity          Pr( +| D)  89.53%   85.14%   92.99%
Specificity          Pr( -| ~D)  62.79%   51.70%   72.98%
Positive predictive value Pr( D| +)  87.83%   83.26%   91.53%
Negative predictive value Pr( ~D| -)  66.67%   55.32%   76.76%
-----+-----+-----+-----
Prevalence           Pr(D)      75.00%   70.08%   79.49%
-----+-----+-----+-----
```

In the liver scan study, the percentage of abnormality was 75%. If the same test was used in a different clinical setting where the prevalence of abnormality was 0.25%, we would have a positive predictive value of 44.51% and a negative predictive value of 94.74%.

Now, we show summary results for the liver scan test and the correct diagnosis for a prevalence of abnormality of 25%.

```
. diagt diag test [fw=n], prev(25)
      |      diagnostic test
      |      result
diagnostic |      Pos.      Neg. |      Total
-----+-----+-----+-----
  Abnormal |      231      27 |      258
   Normal |      32      54 |      86
-----+-----+-----+-----
      Total |      263      81 |      344
True abnormal diagnosis defined as diag = 1 (labelled Abnormal)
                                         [95% Conf. Inter.]
-----+-----+-----+-----
Sensitivity          Pr( +| D)  89.53%   85.14%   92.99%
Specificity          Pr( -| ~D)  62.79%   51.70%   72.98%
Positive predictive value Pr( D| +)  60.15%   .%       .%
Negative predictive value Pr( ~D| -)  40.51%   .%       .%
-----+-----+-----+-----
Prevalence           Pr(D)      25.00%   .%       .%
-----+-----+-----+-----
```

The same results can be achieved without any data in memory using the immediate form of the command:

```
. diagti 231 27 32 54
      True |
disease |      Test result
status |      Pos.      Neg. |      Total
-----+-----+-----+-----
  Abnormal |      231      27 |      258
   Normal |      32      54 |      86
-----+-----+-----+-----
      Total |      263      81 |      344
-----+-----+-----+-----
Sensitivity          Pr( +| D)  89.53%   85.14%   92.99%
Specificity          Pr( -| ~D)  62.79%   51.70%   72.98%
Positive predictive value Pr( D| +)  87.83%   83.26%   91.53%
Negative predictive value Pr( ~D| -)  66.67%   55.32%   76.76%
-----+-----+-----+-----
Prevalence           Pr(D)      75.00%   70.08%   79.49%
-----+-----+-----+-----
```

References

- Altman, D. G. and J. M. Bland. 1994a. Diagnostic tests 1: sensitivity and specificity. *British Medical Journal* 308: 1552.
- . 1994b. Diagnostic tests 2: predictive values. *British Medical Journal* 309: 102.
- Drum, D. E. and J. S. Christapoulos. 1972. Hepatic scintigraphy in clinical decision making. *Journal of Nuclear Medicine* 13: 908–915.
- Tobias, A. 2000. sbe36: Summary statistics report for diagnostic tests. *Stata Technical Bulletin* 56: 16–18.

sbe41	Ordinary case-cohort design and analysis
-------	--

Vincenzo Coviello, Unità di Epidemiologia e Statistica ASL Ba/1, Italy, coviello@mythnet.it

Abstract: The case-cohort design is an efficient alternative to a full cohort analysis. Two new commands for ordinary case-cohort designs are presented. They randomly select a sample from a cohort, prepare the resulting dataset for analysis using a Cox regression model and compute the asymptotically consistent Self-Prentice variance estimator of the parameters.

Keywords: Cohort studies, nested case-control design, survival analysis, Cox regression model, variance estimation.

Introduction

Various designs have been proposed as useful alternatives to the standard full cohort analysis when data collection for any subject may be very expensive. In the case-cohort design (see Barlow et al. 1999, Clayton and Hills 1993, Langholz and Thomas 1990, 1991, and Rothman and Greenland 1998, for example), covariate information is assessed in a sample of subjects selected randomly from the entire cohort, the subcohort, and in all individuals who fail, whether they are in the subcohort or not. The case-cohort design has some advantages with respect to the nested case-control study mainly when the cohort under study is fixed (i.e., without staggered entry times), failures are rare, and there is little loss to follow-up. Unlike the nested case-control study, the case-cohort design allows one to use the same sample of subjects to analyze several failure time outcomes. Furthermore, the subcohort is chosen without regard to any time scale. Recently, simplified methods have appeared for parameter and variance estimation, thus, allowing the analysis of the case-cohort design using a Cox model suitably adapted. Here, we present two new Stata commands that assist the user in the fundamental steps of case-cohort design and analysis: `stcascoh` to sample the full cohort and prepare a dataset for analysis, and `stselpre` to calculate the variance as proposed by Self and Prentice.

Syntax for sampling cohort and preparing dataset

```
stcascoh [varlist] [if exp] [in range] , alpha(#) [ group(varnames) generate(varlist) eps(#)
      seed(#) noshow ]
```

`stcascoh` is for use with survival-time data. You must `stset` the data with an `id()` variable before using this command.

Description

`stcascoh` is used to create an appropriate dataset for analysis as a case-cohort study, drawing an α -fraction random sample of the full cohort and including all failures whether they occur in the random sample or not. To this aim, `stcascoh` expands observations which fall into two parts: 1) time interval $(t_0, t - \epsilon]$, and 2) time interval $(t - \epsilon, t]$. For cases included in the subcohort, both segments are retained in the final dataset, whereas for cases not in the subcohort just the last segment is retained.

The variables in the table below are added to the dataset.

<code>._subco</code>	coded 0 for subcohort member with no failure coded 1 for subcohort member who failed coded 2 for nonsubcohort member who failed (nonsubcohort case)
<code>._wSelPre</code>	log-weights of records as in the Self-Prentice method
<code>._wBarlow</code>	log-weights of records as in the Barlow method

The names of the new variables and the sampling fraction are saved as Stata characteristics as shown in the table below.

<code>._dta[Subco]</code>	subcohort membership variable name
<code>._dta[wSelPre]</code>	Self-Prentice log-weights variable name
<code>._dta[wBarlow]</code>	Barlow log-weights variable name
<code>._dta[Alpha]</code>	sampling fraction

`varlist` defines variables that will be retained in the final dataset. If `varlist` isn't specified, all variables are carried over into the resulting dataset. Observations not meeting `if` and `in` criteria are dropped even if they fail. Randomness in the sampling is obtained using Stata's `uniform()` function.