Title

survival analysis — Introduction to survival analysis & epidemiological tables commands

Description

Stata's survival analysis routines are used to compute sample size, power, and effect size and to declare, convert, manipulate, summarize, and analyze survival data. Survival data is time-to-event data, and survival analysis is full of jargon: truncation, censoring, hazard rates, etc. See the glossary in this manual. For a good Stata-specific introduction to survival analysis, see Cleves et al. (2008).

Stata also has several commands for analyzing contingency tables resulting from various forms of observational studies, such as cohort or matched case–control studies.

This manual documents the following commands, which are described in detail in their respective manual entries.

Declaring and converting count data

`ctset`	[ST] **ctset**	Declare data to be count-time data
`cttost`	[ST] **cttost**	Convert count-time data to survival-time data

Converting snapshot data

`snapspan`	[ST] **snapspan**	Convert snapshot data to time-span data

Declaring and summarizing survival-time data

`stset`	[ST] **stset**	Declare data to be survival-time data
`stdescribe`	[ST] **stdescribe**	Describe survival-time data
`stsum`	[ST] **stsum**	Summarize survival-time data

Manipulating survival-time data

`stvary`	[ST] **stvary**	Report whether variables vary over time
`stfill`	[ST] **stfill**	Fill in by carrying forward values of covariates
`stgen`	[ST] **stgen**	Generate variables reflecting entire histories
`stsplit`	[ST] **stsplit**	Split time-span records
`stjoin`	[ST] **stsplit**	Join time-span records
`stbase`	[ST] **stbase**	Form baseline dataset

Obtaining summary statistics, confidence intervals, tables, etc.

`sts`	[ST] **sts**	Generate, graph, list, and test the survivor and cumulative hazard functions
`stir`	[ST] **stir**	Report incidence-rate comparison
`stci`	[ST] **stci**	Confidence intervals for means and percentiles of survival time
`strate`	[ST] **strate**	Tabulate failure rate
`stptime`	[ST] **stptime**	Calculate person-time, incidence rates, and SMR
`stmh`	[ST] **strate**	Calculate rate ratios with the Mantel–Haenszel method
`stmc`	[ST] **strate**	Calculate rate ratios with the Mantel–Cox method
`ltable`	[ST] **ltable**	Display and graph life tables

Fitting regression models

`stcox`	[ST] **stcox**	Cox proportional hazards model
`estat concordance`	[ST] **stcox postestimation**	Calculate Harrell's C
`estat phtest`	[ST] **stcox PH-assumption tests**	Test Cox proportional-hazards assumption
`stphplot`	[ST] **stcox PH-assumption tests**	Graphically assess the Cox proportional-hazards assumption
`stcoxkm`	[ST] **stcox PH-assumption tests**	Graphically assess the Cox proportional-hazards assumption
`streg`	[ST] **streg**	Parametric survival models
`stcurve`	[ST] **stcurve**	Plot survivor, hazard, cumulative hazard, or cumulative incidence function
`stcrreg`	[ST] **stcrreg**	Competing-risks regression

Sample-size and power determination for survival analysis

`stpower cox`	[ST] **stpower cox**	Sample size, power, and effect size for the Cox proportional hazards model
`stpower exponential`	[ST] **stpower exponential**	Sample size and power for the exponential test
`stpower logrank`	[ST] **stpower logrank**	Sample size, power, and effect size for the log-rank test

Converting survival-time data

`sttocc`	[ST] **sttocc**	Convert survival-time data to case–control data
`sttoct`	[ST] **sttoct**	Convert survival-time data to count-time data

Programmer's utilities

`st_*`	[ST] **st_is**	Survival analysis subroutines for programmers

Epidemiological tables

`ir`	[ST] **epitab**	Incidence rates for cohort studies
`iri`	[ST] **epitab**	Immediate form of `ir`
`cs`	[ST] **epitab**	Risk differences, risk ratios, and odds ratios for cohort studies
`csi`	[ST] **epitab**	Immediate form of `cs`
`cc`	[ST] **epitab**	Odds ratios for case–control data
`cci`	[ST] **epitab**	Immediate form of `cc`
`tabodds`	[ST] **epitab**	Tests of log odds for case–control data
`mhodds`	[ST] **epitab**	Odds ratios controlled for confounding
`mcc`	[ST] **epitab**	Analysis of matched case–control data
`mcci`	[ST] **epitab**	Immediate form of `mcc`

Remarks

Remarks are presented under the following headings:

Introduction

All but one entry in this manual deals with the analysis of survival data, which is used to measure the time to an event of interest such as death or failure. Survival data can be organized in two ways. The first way is as *count data*, which refers to observations on populations, whether people or generators, with observations recording the number of units at a given time that failed or were lost because of censoring. The second way is as *survival-time*, or *time-span*, data. In survival-time data, the observations represent periods and contain three variables that record the start time of the period, the end time, and an indicator of whether failure or right-censoring occurred at the end of the period. The representation of the response of these three variables makes survival data unique in terms of implementing the statistical methods in the software.

Survival data may also be organized as *snapshot data* (a small variation of the survival-time format), in which observations depict an instance in time rather than an interval. When you have snapshot data, you simply use the `snapspan` command to convert it to survival-time data before proceeding.

Stata commands that begin with `ct` are used to convert count data to survival-time data. Survival-time data are analyzed using Stata commands that begin with `st`, known in our terminology as st commands. You can express all the information contained in count data in an equivalent survival-time dataset, but the converse is not true. Thus Stata commands are made to work with survival-time data because it is the more general representation.

The one remaining entry is [ST] **epitab**, which describes epidemiological tables. [ST] **epitab** covers many commands dealing with analyzing contingency tables arising from various observational studies, such as case–control or cohort studies. [ST] **epitab** is included in this manual because the concepts presented there are related to concepts of survival analysis, and both topics use the same terminology and are of equal interest to many researchers.

Declaring and converting count data

Count data must first be converted to survival-time data before Stata's `st` commands can be used. Count data can be thought of as aggregated survival-time data. Rather than having observations that are specific to a subject and a period, you have data that, at each recorded time, record the number lost because of failure and, optionally, the number lost because of right-censoring.

`ctset` is used to tell Stata the names of the variables in your count data that record the time, the number failed, and the number censored. You `ctset` your data before typing `cttost` to convert it to survival-time data. Because you `ctset` your data, you can type `cttost` without any arguments to perform the conversion. Stata remembers how the data are `ctset`.

Converting snapshot data

Snapshot data are data in which each observation records the status of a given subject at a certain point in time. Usually you have multiple observations on each subject that chart the subject's progress through the study.

Before using Stata's survival analysis commands with snapshot data, you must first convert the data to survival-time data; that is, the observations in the data should represent intervals. When you convert snapshot data, the existing time variable in your data is used to record the end of a time span, and a new variable is created to record the beginning. Time spans are created using the recorded snapshot times as breakpoints at which new intervals are to be created. Before converting snapshot data to time-span data, you must understand the distinction between *enduring variables* and *instantaneous variables*. Enduring variables record characteristics of the subject that endure throughout the time span, such as sex or smoking status. Instantaneous variables describe events that occur at the end of a time span, such as failure or censoring. When you convert snapshots to intervals, enduring variables obtain their values from the previous recorded snapshot or are set to missing for the first interval. Instantaneous variables obtain their values from the current recorded snapshot because the existing time variable now records the end of the span.

Stata's `snapspan` makes this whole process easy. You specify an ID variable identifying your subjects, the snapshot time variable, the name of the new variable to hold the beginning times of the spans, and any variables that you want to treat as instantaneous variables. Stata does the rest for you.

Declaring and summarizing survival-time data

Stata does not automatically recognize survival-time data, so you must declare your survival-time data to Stata by using `stset`. Every st command relies on the information that is provided when you `stset` your data. Survival-time data come in different forms. For example, your time variables may be dates, time measured from a fixed date, or time measured from some other point unique to each subject, such as enrollment in the study. You can also consider the following questions. What is the onset of risk for the subjects in your data? Is it time zero? Is it enrollment in the study or some other event, such as a heart transplant? Do you have censoring, and if so, which variable records it? What values does this variable record for censoring/failure? Do you have delayed entry? That is, were some subjects at risk of failure before you actually observed them? Do you have simple data and wish to treat everyone as entering and at risk at time zero?

Whatever the form of your data, you must first `stset` it before analyzing it, and so if you are new to Stata's st commands, we highly recommend that you take the time to learn about `stset`. It is really easy once you get the hang of it, and [ST] **stset** has many examples to help. For more discussion of `stset`, see chapter 6 of Cleves et al. (2008).

Once you `stset` the data, you can use `stdescribe` to describe the aspects of your survival data. For example, you will see the number of subjects you were successful in declaring, the total number of records associated with these subjects, the total time at risk for these subjects, time gaps for any of these subjects, any delayed entry, etc. You can use `stsum` to summarize your survival data, for example, to obtain the total time at risk and the quartiles of time-to-failure in analysis-time units.

Manipulating survival-time data

Once your data have been `stset`, you may want to clean them up a bit before beginning your analysis. Suppose that you had an enduring variable and `snapspan` recorded it as missing for the interval leading up to the first recorded snapshot time. You can use `stfill` to fill in missing values of covariates, either by carrying forward the values from previous periods or by making the covariate

equal to its earliest recorded (nonmissing) value for all time spans. You can use `stvary` to check for time-varying covariates or to confirm that certain variables, such as sex, are not time varying. You can use `stgen` to generate new covariates based on functions of the time spans for each given subject. For example, you can create a new variable called `eversmoked` that equals one for all a subject's observations, if the variable `smoke` in your data is equal to one for *any* of the subject's time spans. Think of `stgen` as just a convenient way to do things that could be done using `by` *subject_id*`:` with survival-time data.

`stsplit` is useful for creating data that have multiple records per subject from data that have one record per subject. Suppose that you have already `stset` your data and wish to introduce a time-varying covariate. You would first need to `stsplit` your data so that separate time spans could be created for each subject, allowing the new covariate to assume different values over time within a subject. `stjoin` is the opposite of `stsplit`. Suppose that you have data with multiple records per subject but then realize that the data could be collapsed into single-subject records with no loss of information. Using `stjoin` would speed up any subsequent analysis using the st commands without changing the results.

`stbase` can be used to set every variable in your multiple-record st data to the value at baseline, defined as the earliest time at which each subject was observed. It can also be used to convert st data to cross-sectional data.

Obtaining summary statistics, confidence intervals, tables, etc.

Stata provides several commands for nonparametric analysis of survival data that can produce a wide array of summary statistics, inference, tables, and graphs. `sts` is a truly powerful command, used to obtain nonparametric estimates, inference, tests, and graphs of the survivor function, the cumulative hazard function, and the hazard function. You can compare estimates across groups, such as smoking versus nonsmoking, and you can adjust these estimates for the effects of other covariates in your data. `sts` can present these estimates as tables and graphs. `sts` can also be used to test the equality of survivor functions across groups.

`stir` is used to estimate incidence rates and to compare incidence rates across groups. `stci` is the survival-time data analog of `ci` and is used to obtain confidence intervals for means and percentiles of time to failure. `strate` is used to tabulate failure rates. `stptime` is used to calculate person-time and standardized mortality/morbidity ratios (SMRs). `stmh` calculates rate ratios by using the Mantel–Haenszel method, and `stmc` calculates rate ratios by using the Mantel–Cox method.

`ltable` displays and graphs life tables for individual-level or aggregate data.

Fitting regression models

Stata has commands for fitting both semiparametric and parametric regression models to survival data. `stcox` fits the Cox proportional hazards model and `predict` after `stcox` can be used to retrieve estimates of the baseline survivor function, the baseline cumulative hazard function, and the baseline hazard contributions. `predict` after `stcox` can also calculate a myriad of Cox regression diagnostic quantities, such as martingale residuals, efficient score residuals, and Schoenfeld residuals. `stcox` has four options for handling tied failures. `stcox` can be used to fit stratified Cox models, where the baseline hazard is allowed to differ over the strata, and it can be used to model multivariate survival data by using a *shared-frailty* model, which can be thought of as a Cox model with random effects. After `stcox`, you can use `estat phtest` to test the proportional-hazards assumption or `estat concordance` to calculate Harrell's C. With `stphplot` and `stcoxkm`, you can graphically assess the proportional-hazards assumption.

Stata offers six parametric regression models for survival data: exponential, Weibull, lognormal, loglogistic, Gompertz, and gamma. All six models are fit using `streg`, and you can specify the model you want with the `distribution()` option. All these models, except for the exponential, have ancillary parameters that are estimated (along with the linear predictor) from the data. By default, these ancillary parameters are treated as constant, but you may optionally model the ancillary parameters as functions of a linear predictor. Stratified models may also be fit using `streg`. You can also fit frailty models with `streg` and specify whether you want the frailties to be treated as spell-specific or shared across groups of observations.

`stcrreg` fits a semiparametric regression model for survival data in the presence of competing risks. Competing risks impede the failure event under study from occurring. An analysis of such competing-risks data focuses on the *cumulative incidence function*, the probability of failure in the presence of competing events that prevent that failure. `stcrreg` provides an analogue to `stcox` for such data. The baseline *subhazard function*—that which generates failures under competing risks—is left unspecified, and covariates act multiplicatively on the baseline subhazard.

`stcurve` is for use after `stcox`, `streg`, and `stcrreg` and will plot the estimated survivor, hazard, cumulative hazard, and cumulative incidence function for the fitted model. Covariates, by default, are held fixed at their mean values, but you can specify other values if you wish. `stcurve` is useful for comparing these functions across different levels of covariates.

Sample size and power determination for survival analysis

Stata has commands for computing sample size, power, and effect size for survival analysis using the log-rank test, the Cox proportional hazards model, and the exponential test comparing exponential hazard rates.

`stpower logrank` estimates required sample size, power, and effect size for survival analysis comparing survivor functions in two groups using the log-rank test. It provides options to account for unequal allocation of subjects between the two groups, possible withdrawal of subjects from the study (loss to follow-up), and uniform accrual of subjects into the study.

`stpower cox` estimates required sample size, power, and effect size for survival analysis using Cox proportional hazards (PH) models with possibly multiple covariates. It provides options to account for possible correlation between the covariate of interest and other predictors and for withdrawal of subjects from the study.

`stpower exponential` estimates required sample size and power for survival analysis comparing two exponential survivor functions using the exponential test (in particular, the Wald test of the difference between hazards or, optionally, of the difference between log hazards). It accommodates unequal allocation between the two groups, flexible accrual of subjects into the study (uniform and truncated exponential), and group-specific losses to follow-up.

The `stpower` commands allow automated production of customizable tables and have options to assist with the creation of graphs of power curves and more.

Converting survival-time data

Stata has commands for converting survival-time data to case–control and count data. These commands are rarely used, because most of the analyses are performed using data in the survival-time format. `sttocc` is useful for converting survival data to case–control data suitable for estimation with `clogit`. `sttoct` is the opposite of `cttost` and will convert survival-time data to count data.

Programmer's utilities

Stata also provides routines for programmers interested in writing their own st commands. These are basically utilities for setting, accessing, and verifying the information saved by `stset`. For example, `st_is` verifies that the data have in fact been `stset` and gives the appropriate error if not. `st_show` is used to preface the output of a program with key information on the st variables used in the analysis. Programmers interested in writing st code should see [ST] **st_is**.

Epidemiological tables

See the *Description* section of [ST] **epitab** for an overview of Stata's commands for calculating statistics and performing lists that are useful for epidemiologists.

Reference

Cleves, M. A., W. W. Gould, R. G. Gutierrez, and Y. Marchenko. 2008. *An Introduction to Survival Analysis Using Stata*. 2nd ed. College Station, TX: Stata Press.

Also see

[ST] **stset** — Declare data to be survival-time data

[ST] **intro** — Introduction to survival analysis manual