

**streg postestimation** — Postestimation tools for streg

Postestimation commands  
Methods and formulas

predict  
References

margins  
Also see

Remarks and examples

## Postestimation commands

The following postestimation command is of special interest after `streg`:

| Command                      | Description   |
|------------------------------|---|
| * <code>estat gofplot</code> | produce goodness-of-fit plot                                      |
| <code>stcurve</code>         | plot the survivor, failure, hazard, or cumulative hazard function |

\* `estat gofplot` is not appropriate with `svy` estimation results.

The following standard postestimation commands are also available:

| Command                      | Description   |
|------------------------------|---|
| <code>contrast</code>        | contrasts and ANOVA-style joint tests of estimates  |
| <code>estat ic</code>        | Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC) |
| <code>estat summarize</code> | summary statistics for the estimation sample  |
| <code>estat vce</code>       | variance–covariance matrix of the estimators (VCE)  |
| <code>estat (svy)</code>     | postestimation statistics for survey data   |
| <code>estimates</code>       | cataloging estimation results   |
| <code>etable</code>          | table of estimation results   |
| * <code>hausman</code>       | Hausman's specification test  |
| <code>lincom</code>          | point estimates, standard errors, testing, and inference for linear combinations of coefficients                          |
| <code>linktest</code>        | link test for model specification   |
| * <code>lrtest</code>        | likelihood-ratio test   |
| <code>margins</code>         | marginal means, predictive margins, marginal effects, and average marginal effects  |
| <code>marginsplot</code>     | graph the results from margins (profile plots, interaction plots, etc.)   |
| <code>nlcom</code>           | point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients                       |
| <code>predict</code>         | hazard ratios, survivor functions, influence statistics, residuals, etc.  |
| <code>predictnl</code>       | point estimates, standard errors, testing, and inference for generalized predictions                                      |
| <code>pwcompare</code>       | pairwise comparisons of estimates   |
| <code>suest</code>           | seemingly unrelated estimation  |
| <code>test</code>            | Wald tests of simple and composite linear hypotheses  |
| <code>testnl</code>          | Wald tests of nonlinear hypotheses  |

\* `hausman` and `lrtest` are not appropriate with `svy` estimation results.

# predict

## Description for predict

`predict` creates a new variable containing predictions such as median and mean survival times; hazards; hazard ratios; linear predictions; standard errors; probabilities; Cox–Snell, martingale-like, and deviance residuals.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] newvar [if] [in] [, statistic options]
```

```
predict [type] stub* [if] [in], scores
```

| <i>statistic</i> | Description |
|------------------|-------------|
|------------------|-------------|

Main

|                             |   |
|-----------------------------|---|
| <u>median</u> time          | median survival time; the default   |
| <u>median</u> <u>lntime</u> | median ln(survival time)  |
| <u>mean</u> time            | mean survival time  |
| <u>mean</u> <u>lntime</u>   | mean ln(survival time)  |
| <u>hazard</u>               | hazard  |
| <u>hr</u>                   | hazard ratio, also known as the relative hazard   |
| <u>xb</u>                   | linear prediction $\mathbf{x}_j\widehat{\boldsymbol{\beta}}$                            |
| <u>stdp</u>                 | standard error of the linear prediction; $SE(\mathbf{x}_j\widehat{\boldsymbol{\beta}})$ |
| <u>surv</u>                 | $S(t t_0)$  |
| * <u>csurv</u>              | $S(t   \text{earliest } t_0 \text{ for subject})$                                       |
| * <u>csnell</u>             | Cox–Snell residuals   |
| * <u>mgale</u>              | martingale-like residuals   |
| * <u>deviance</u>           | deviance residuals  |

| <i>options</i> | Description |
|----------------|-------------|
|----------------|-------------|

|                      |   |
|----------------------|---|
| <u>oos</u>           | make <i>statistic</i> available in and out of sample                      |
| <u>nooffset</u>      | ignore the <code>offset()</code> variable specified in <code>streg</code> |
| <u>alpha1</u>        | predict <i>statistic</i> conditional on frailty value equal to one        |
| <u>unconditional</u> | predict <i>statistic</i> unconditionally on the frailty                   |
| <u>marginal</u>      | synonym for <code>unconditional</code>                                    |
| <u>partial</u>       | produce observation-level results   |

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated for the estimation sample by default, but the `oos` option makes them available both in and out of sample.

When no option is specified, the predicted median survival time is calculated for all models. The predicted hazard ratio, option `hr`, is available only for the exponential, Weibull, and Gompertz models. The mean time and mean `lntime` options are not available for the Gompertz model. Unconditional estimates of mean time and mean `lntime` are not available if `frailty()` was specified with `streg`; see [ST] `streg`.

`csnell`, `mgale`, and `deviance` are not allowed with `svy` estimation results.

## Options for predict

### Main

`median time` calculates the predicted median survival time in analysis-time units. This is the prediction from time 0 conditional on constant covariates. When no options are specified with `predict`, the predicted median survival time is calculated for all models.

`median lntime` calculates the natural logarithm of what `median time` produces.

`mean time` calculates the predicted mean survival time in analysis-time units. This is the prediction from time 0 conditional on constant covariates. This option is not available for Gompertz regressions and is available for frailty models only if `alpha1` is specified, in which case what you obtain is an estimate of the mean survival time conditional on a frailty effect of one.

`mean lntime` predicts the mean of the natural logarithm of `time`. This option is not available for Gompertz regression and is available for frailty models only if `alpha1` is specified, in which case what you obtain is an estimate of the mean log survival-time conditional on a frailty effect of one.

`hazard` calculates the predicted hazard.

`hr` calculates the hazard ratio. This option is valid only for models having a proportional-hazards parameterization.

`xb` calculates the linear prediction from the fitted model. That is, you fit the model by estimating a set of parameters,  $\beta_0, \beta_1, \beta_2, \dots, \beta_k$ , and the linear prediction is  $\hat{y}_j = \hat{\beta}_0 + \hat{\beta}_1 x_{1j} + \hat{\beta}_2 x_{2j} + \dots + \hat{\beta}_k x_{kj}$ , often written in matrix notation as  $\hat{y}_j = \mathbf{x}_j \hat{\beta}$ .

The  $x_{1j}, x_{2j}, \dots, x_{kj}$  used in the calculation are obtained from the data currently in memory and need not correspond to the data on the independent variables used in estimating  $\hat{\beta}$ .

`stdp` calculates the standard error of the prediction, that is, the standard error of  $\hat{y}_j$ .

`surv` calculates each observation's predicted survivor probability,  $S(t|t_0)$ , where  $t_0$  is `_t0`, the analysis time at which each record became at risk. For multiple-record data, see the `csurv` option below.

`csurv` calculates the predicted  $S(t|\text{earliest } t_0)$  for each subject in multiple-record data by calculating the conditional survivor values,  $S(t|t_0)$  (see the `surv` option above), and then multiplying them.

What you obtain from `surv` will differ from what you obtain from `csurv` only if you have multiple records for that subject.

In the presence of gaps or delayed entry, the estimates obtained from `csurv` can be different for subjects with gaps from those without gaps, having the same covariate values, because the probability of survival over gaps is assumed to be 1. Thus the predicted cumulative conditional survivor function is not a smooth function of time `_t` for constant values of the covariates. Use `stcurve`, `survival` to obtain a smooth estimate of the cumulative survivor function  $S(t|x)$ .

`csnell` calculates the Cox–Snell generalized residuals. For multiple-record-per-subject data, by default only one value per subject is calculated and it is placed on the last record for the subject.

Adding the `partial` option will produce partial Cox–Snell residuals, one for each record within subject; see [partial](#) below. Partial Cox–Snell residuals are the additive contributions to a subject’s overall Cox–Snell residual. In single-record-per-subject data, the partial Cox–Snell residuals are the Cox–Snell residuals.

`mgale` calculates the martingale-like residuals. For multiple-record data, by default only one value per subject is calculated and it is placed on the last record for the subject.

Adding the `partial` option will produce partial martingale residuals, one for each record within subject; see [partial](#) below. Partial martingale residuals are the additive contributions to a subject’s overall martingale residual. In single-record data, the partial martingale residuals are the martingale residuals.

`deviance` calculates the deviance residuals. Deviance residuals are martingale residuals that have been transformed to be more symmetric about zero. For multiple-record data, by default only one value per subject is calculated and it is placed on the last record for the subject.

Adding the `partial` option will produce partial deviance residuals, one for each record within subject; see [partial](#) below. Partial deviance residuals are transformed partial martingale residuals. In single-record data, the partial deviance residuals are the deviance residuals.

`oos` makes `csurv`, `csnell`, `mgale`, and `deviance` available both in and out of sample. `oos` also dictates that summations and other accumulations take place over the sample as defined by `if` and `in`. By default, the summations are taken over the estimation sample, with `if` and `in` merely determining which values of `newvar` are to be filled in once the calculation is finished.

`nooffset` is relevant only if you specified `offset(varname)` with `streg`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as  $\mathbf{x}\beta$  rather than  $\mathbf{x}\beta + \text{offset}$ .

`alpha1`, when used after fitting a frailty model, specifies that *statistic* be predicted conditional on a frailty value equal to one. This is the default for shared-frailty models.

`unconditional` and `marginal`, when used after fitting a frailty model, specify that *statistic* be predicted unconditional on the frailty. That is, the prediction is averaged over the frailty distribution. This is the default for unshared-frailty models.

`partial` is relevant only for multiple-record data and is valid with `csnell`, `mgale`, and `deviance`. Specifying `partial` will produce “partial” versions of these statistics, where one value is calculated for each record instead of one for each subject. The subjects are determined by the `id()` option of `stset`.

Specify `partial` if you wish to perform diagnostics on individual records rather than on individual subjects. For example, a partial deviance can be used to diagnose the fitted characteristics of an individual record rather than those of the set of records for a given subject.

`scores` calculates equation-level score variables. The number of score variables created depends upon the chosen distribution.

The first new variable will always contain  $\partial \ln L / \partial (\mathbf{x}_j \beta)$ .

The subsequent new variables will contain the partial derivative of the log likelihood with respect to the ancillary parameters.

## margins

### Description for margins

`margins` estimates margins of response for median and mean survival times, hazard ratios, and linear predictions.

### Menu for margins

Statistics > Postestimation

### Syntax for margins

```
margins [marginlist] [, options]
```

```
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

| <i>statistic</i>                             | Description                                     |
|--|---|
| <code>median time</code>                     | median survival time; the default               |
| <code>median <u>l</u>n<sub>time</sub></code> | median ln(survival time)                        |
| <code>mean time</code>                       | mean survival time                              |
| <code>mean <u>l</u>n<sub>time</sub></code>   | mean ln(survival time)                          |
| <code>hr</code>                              | hazard ratio, also known as the relative hazard |
| <code>xb</code>                              | linear prediction $x_j\hat{\beta}$              |
| <code><u>h</u>azard</code>                   | not allowed with <code>margins</code>           |
| <code><u>s</u>tdp</code>                     | not allowed with <code>margins</code>           |
| <code><u>s</u>urv</code>                     | not allowed with <code>margins</code>           |
| <code><u>c</u>surv</code>                    | not allowed with <code>margins</code>           |
| <code><u>c</u>snell</code>                   | not allowed with <code>margins</code>           |
| <code><u>m</u>gale</code>                    | not allowed with <code>margins</code>           |
| <code><u>d</u>eviance</code>                 | not allowed with <code>margins</code>           |

Statistics not allowed with `margins` are functions of stochastic quantities other than  $e(b)$ .

For the full syntax, see [R] [margins](#).

### Remarks and examples

[stata.com](http://www.stata.com)

`predict` after `streg` is used to generate a variable containing predicted values or residuals.

For a more detailed discussion on residuals, read *Residuals and diagnostic measures* in the [ST] [stcox postestimation](#) entry. Many of the concepts and ideas presented there also apply to `streg` models.

Regardless of the metric used, `predict` can generate predicted median survival times and median log survival-times for all models, and predicted mean times and mean log survival-times where available. Predicted survival, hazard, and residuals are also available for all models. The predicted hazard ratio can be calculated only for models with a proportional-hazards parameterization, that is, the Weibull, exponential, and Gompertz models. However, the estimation need not take place in the log-hazard metric. You can perform, for example, a Weibull regression specifying the `time` option and then ask that hazard ratios be predicted.

After fitting a frailty model, you can use `predict` with the `alpha1` option to generate predicted values based on  $S(t)$  or use the `unconditional` option to generate predictions based on  $S_\theta(t)$ ; see [ST] `streg`.

### ► Example 1

Let's return to [example 1](#) of [ST] `streg` concerning the ability of emergency generators with new-style bearings to withstand overloads. Assume that, as before, we fit a proportional hazards Weibull model:

```
. use https://www.stata-press.com/data/r18/kva
(Generator experiment)
. stset failtime
(output omitted)
. streg load bearings, distribution(weibull) nolog
      Failure _d: 1 (meaning all fail)
      Analysis time _t: failtime
Weibull PH regression
No. of subjects = 12                Number of obs = 12
No. of failures = 12
Time at risk   = 896
Log likelihood = 5.6934189          LR chi2(2)    = 30.27
                                      Prob > chi2    = 0.0000
```

| _t       | Haz. ratio | Std. err. | z     | P> z  | [95% conf. interval] |          |
|----------|------------|-----------|-------|-------|----------------------|----------|
| load     | 1.599315   | .1883807  | 3.99  | 0.000 | 1.269616             | 2.014631 |
| bearings | .1887995   | .1312109  | -2.40 | 0.016 | .0483546             | .7371644 |
| _cons    | 2.51e-20   | 2.66e-19  | -4.26 | 0.000 | 2.35e-29             | 2.68e-11 |
| /ln_p    | 2.051552   | .2317074  | 8.85  | 0.000 | 1.597414             | 2.505691 |
| p        | 7.779969   | 1.802677  |       |       | 4.940241             | 12.25202 |
| 1/p      | .1285352   | .0297826  |       |       | .0816192             | .2024193 |

Note: `_cons` estimates baseline hazard.

Now we can predict both the median survival time and the log-median survival time for each observation:

```
. predict time, time
(option median time assumed; predicted median time)
. predict lntime, lntime
(option median lntime assumed; predicted median log time)
. format time lntime %9.4f
```

```
. list failtime load bearings time lntime
```

|     | failtime | load | bearings | time     | lntime |
|-----|----------|------|----------|----------|--------|
| 1.  | 100      | 15   | 0        | 127.5586 | 4.8486 |
| 2.  | 140      | 15   | 1        | 158.0407 | 5.0629 |
| 3.  | 97       | 20   | 0        | 94.3292  | 4.5468 |
| 4.  | 122      | 20   | 1        | 116.8707 | 4.7611 |
| 5.  | 84       | 25   | 0        | 69.7562  | 4.2450 |
| 6.  | 100      | 25   | 1        | 86.4255  | 4.4593 |
| 7.  | 54       | 30   | 0        | 51.5845  | 3.9432 |
| 8.  | 52       | 30   | 1        | 63.9114  | 4.1575 |
| 9.  | 40       | 35   | 0        | 38.1466  | 3.6414 |
| 10. | 55       | 35   | 1        | 47.2623  | 3.8557 |
| 11. | 22       | 40   | 0        | 28.2093  | 3.3397 |
| 12. | 30       | 40   | 1        | 34.9504  | 3.5539 |

◀

## ▶ Example 2

Using the cancer data of [example 6](#) in [ST] [streg](#), again with drug remapped into a drug-treatment indicator, we can examine the various residuals that Stata produces. For a more detailed discussion on residuals, read *Residuals and diagnostic measures* in [ST] [stcox postestimation](#). Many of the concepts and ideas presented there also apply to [streg](#) models. For a more technical presentation of these residuals, see *Methods and formulas*.

We will begin by evaluating goodness of fit based on the generalized Cox–Snell residuals. If the model fits the data, these residuals should have a standard exponential distribution with  $\lambda = 1$ . One way to verify the fit is to calculate an empirical estimate of the cumulative hazard function—based, for example, on the Kaplan–Meier survival estimates or the Aalen–Nelson estimator, taking the Cox–Snell residuals as the time variable and the censoring variable as before—and plot it against the Cox–Snell residuals. If the model fits the data, the plot should be a straight line with a slope of 1.

After fitting the model, we can use the `estat gofplot` command to create this graph. Here the `km` option specifies that the cumulative hazard function be based on the minus log of the Kaplan–Meier estimator. By default, the cumulative hazard function is based on the Nelson–Aalen estimator.

```
. use https://www.stata-press.com/data/r18/cancer, clear
(Patient survival in drug trial)
. replace drug = drug==2 | drug==3           // 0, placebo : 1, nonplacebo
(48 real changes made)
. quietly stset studytime, failure(died)
. quietly streg age drug, distribution(exp)
. estat gofplot, km
```

We did this separately for each of four distributions. Results are plotted in figure 1:

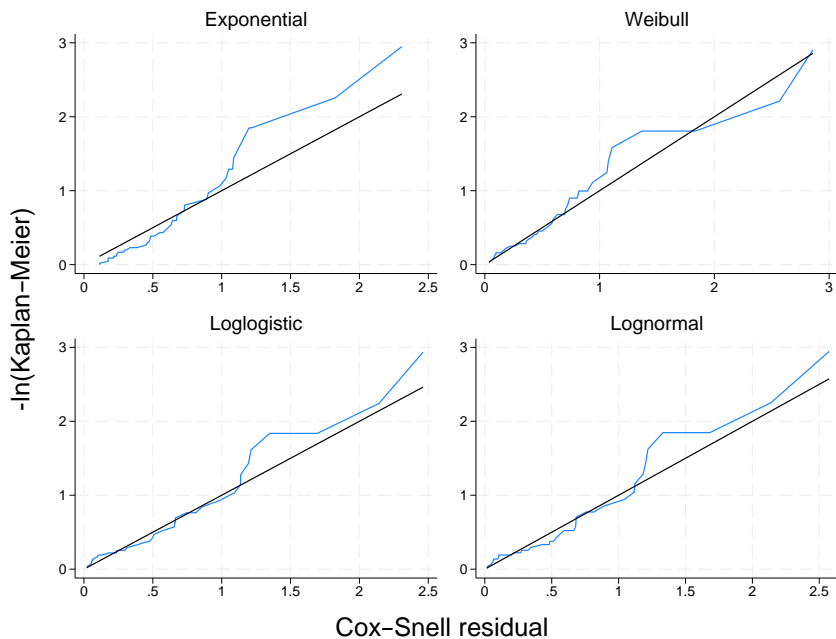


Figure 1. Cox-Snell residuals to evaluate model fit of four regression models

The plots indicate that the Weibull and lognormal models fit the data best and that the exponential model fits poorly. These results are consistent with our previous results (in [ST] **streg**) based on Akaike's information criterion.



### ► Example 3

Let’s now look at the martingale-like and deviance residuals. We use the term “martingale-like” because, although these residuals do not arise naturally from martingale theory for parametric survival models as they do for the Cox proportional hazards model, they do share similar form. We can generate these residuals by using `predict`’s `mgale` option. Martingale residuals take values between  $-\infty$  and 1 and therefore are difficult to interpret. The deviance residuals are a rescaling of the martingale-like residuals so that they are symmetric about zero and thus more like residuals obtained from linear regression. Plots of either deviance residuals against the linear predictor (that is, the log relative hazard in PH models) or of deviance residuals against individual predictors can be useful in identifying aberrant observations and in assessing model fit. Continuing with our modified cancer data, we plot the deviance residual obtained after fitting a lognormal model:

```
. quietly streg age drug, distribution(lnormal)
. predict dev, deviance
. scatter dev studytime, yline(0) m(o)
```

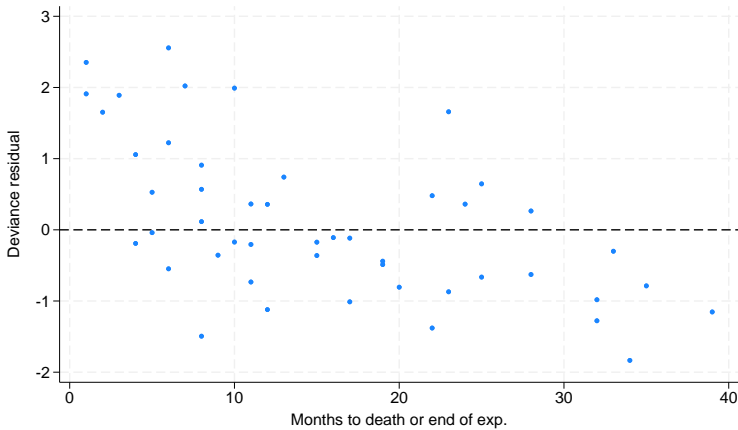


Figure 2. Deviance residuals to evaluate model fit of lognormal model

Figure 2 shows the deviance residuals to be relatively well behaved, with a few minor early exceptions. ◀

## Methods and formulas

`predict newvar`, *options* may be used after `streg` to predict various quantities, according to the following *options*:

**median time:**

$$\text{newvar}_j = \{t : \widehat{S}_j(t) = 1/2\}$$

where  $\widehat{S}_j(t)$  is  $S_j(t)$  with the parameter estimates “plugged in”.

**median lntime:**

$$\text{newvar}_j = \left\{y : \widehat{S}_j(e^y) = 1/2\right\}$$

**mean time:**

$$\text{newvar}_j = \int_0^{\infty} \widehat{S}_j(t) dt$$

mean lntime:

$$newvar_j = \int_{-\infty}^{\infty} ye^y \hat{f}_j(e^y) dy$$

where  $\hat{f}_j(t)$  is  $f_j(t)$  with the parameter estimates plugged in.

hazard:

$$newvar_j = \hat{f}_j(t_j) / \hat{S}_j(t_j)$$

hr (PH models only):

$$newvar_j = \exp(\mathbf{x}_j^* \hat{\boldsymbol{\beta}}^*)$$

where  $\hat{\boldsymbol{\beta}}^*$  does not contain the constant and  $\mathbf{x}_j^*$  does not contain the coefficient of 1 corresponding to the constant.

xb:

$$newvar_j = \mathbf{x}_j \hat{\boldsymbol{\beta}}$$

stdp:

$$newvar_j = \widehat{\text{se}}(\mathbf{x}_j \hat{\boldsymbol{\beta}})$$

surv and csurv:

$$newvar_j = \hat{S}_j(t_j) / \hat{S}_j(t_{0j})$$

The above represents the probability of survival past time  $t_j$  given survival up until  $t_{0j}$  and represents what you obtain when you specify `surv`. If `csurv` is specified, these probabilities are multiplied (in time order) over a subject's multiple observations. What is obtained is then equal to the probability of survival past time  $t_j$ , given survival through the earliest observed  $t_{0j}$ , and given the subject's (possibly time-varying) covariate history. In single-record-per-subject data, `surv` and `csurv` are identical.

csnell:

$$newvar_j = -\log \hat{S}_j(t_j)$$

The Cox–Snell (1968) residual,  $CS_j$ , for observation  $j$  at time  $t_j$  is defined as  $\hat{H}_j(t_j) = -\log \hat{S}_j(t_j)$ , which is the estimated cumulative hazard function obtained from the fitted model (Collett 2003, 111–112). Cox and Snell argued that if the correct model has been fit to the data, these residuals are  $n$  observations from an exponential distribution with unit mean. Thus a plot of the cumulative hazard rate of the residuals against the residuals themselves should result in a straight line of slope 1. Cox–Snell residuals can never be negative and therefore are not symmetric about zero. The options `csnell` and `partial` store in each observation that observation's contribution to the subject's Cox–Snell residual, which we refer to as a partial Cox–Snell residual. If only `csnell` is specified, partial residuals are summed within each subject to obtain one overall Cox–Snell residual for that subject. If there is only 1 observation per subject, `partial` has no effect.

mgale:

$$newvar_j = d_j - CS_j$$

Martingale residuals follow naturally from martingale theory for Cox proportional hazards, but their development does not carry over for parametric survival models. However, martingale-like residuals similar to those obtained for Cox can be derived from the Cox–Snell residuals:  $M_j = d_j - CS_j$ , where  $CS_j$  are the Cox–Snell residuals, as previously described.

Because martingale-like residuals are calculated from the Cox–Snell residuals, they also could be partial or not. Partial martingale residuals are generated with the `mgale` and `partial` options, and overall martingale residuals are generated with the `mgale` option.

Martingale residuals can be interpreted as the difference over time between the number of deaths in the data and the expected number from the fitted model. These residuals take values between  $-\infty$  and 1 and have an expected value of zero, although, like the Cox–Snell residuals, they are not symmetric about zero, making them difficult to interpret.

deviance:

$$newvar_j = \text{sign}(M_j) [-2 \{M_j + d_j \log(d_j - M_j)\}]^{1/2}$$

Deviance residuals are a scaling of the martingale-like residuals in an attempt to make them symmetric about zero. When the model fits the data, these residuals are symmetric about zero and thus can be more readily used to examine the data for outliers. If you also specify the `partial` option, you obtain partial deviance residuals, one for each observation.

`predict` also allows two options for use after fitting frailty models: `alpha1` and `unconditional`. If `unconditional` is specified, the above predictions are modified to be based on  $S_\theta(t)$  and  $f_\theta(t)$ , rather than  $S(t)$  and  $f(t)$ ; see [ST] `streg`. If `alpha1` is specified, the predictions are as described above.

## References

- Boswell, T. M., and R. G. Gutierrez. 2011. *Stata tip 94: Manipulation of prediction parameters for parametric survival regression models*. *Stata Journal* 11: 143–144.
- Collett, D. 2003. *Modelling Binary Data*. 2nd ed. London: Chapman and Hall/CRC.
- Cox, D. R., and E. J. Snell. 1968. A general definition of residuals (with discussion). *Journal of the Royal Statistical Society, Series B* 30: 248–275. <https://doi.org/10.1111/j.2517-6161.1968.tb00724.x>.

## Also see

- [ST] `streg` — Parametric survival models
- [ST] `estat gofplot` — Goodness-of-fit plots after `streg`, `stcox`, `stintreg`, or `stintcox`
- [ST] `stcurve` — Plot the survivor or related function after `streg`, `stcox`, and more
- [U] 20 **Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).