estat gofplot — Goodness-of-fit plots after streg, stcox, stintreg, stintcox, or stmgintcox

Description | Quick start | Menu for estat | Syntax
Options | Remarks and examples | Methods and formulas | References
Also see

Description

`estat gofplot` plots the estimated cumulative hazard function for the Cox–Snell residuals versus the residuals themselves to assess the goodness of fit of the model visually after `streg`, `stcox`, `stintreg`, `stintcox`, or `stmgintcox`.

Quick start

Plot the default cumulative hazard function for the Cox–Snell residuals versus the residuals themselves

```
estat gofplot
```

Plot separate cumulative hazard functions for each group of `x` on the same graph

```
estat gofplot, by(x)
```

Plot separate cumulative hazard functions for each group of `x` on different graphs

```
estat gofplot, by(x) separate
```

After fitting a stratified model, plot separate cumulative hazard functions for each stratum

```
estat gofplot, stratify
```

Plot the minus log of the Kaplan–Meier estimator for the Cox–Snell residuals versus the residuals themselves after `streg` or `stcox`

```
estat gofplot, km
```

Plot the cumulative hazard function for the Cox–Snell-like residuals versus the residuals themselves, for all events after `stmgintcox`

```
estat gofplot
```

Same as above, but plot only the functions for the first two events, and display the function for each event on a separate graph

```
estat gofplot, events(#1 #2) sepevents
```

Menu for estat

Statistics > Postestimation

Syntax

`estat gofplot` [, *options*]

options	Description
`na`	calculate the cumulative hazard function of the Cox–Snell residuals using the Nelson–Aalen estimator; the default for and available only after `streg` and `stcox`
`km`	calculate the cumulative hazard function of the Cox–Snell residuals using the minus log of the Kaplan–Meier estimator; available only after `streg` and `stcox`
`turnbull`	calculate the cumulative hazard function of the Cox–Snell-like residuals using the Turnbull estimator; the estimator for `stintreg`, `stintcox`, and `stmgintcox`
`events(`*evlist*`)`	estimate and graph functions for specified events; default is all events; available only after `stmgintcox`
`sepevents`	show event-specific curves on separate graphs; default is to show event-specific curves as subgraphs on one graph; available only after `stmgintcox`
`by(`*varlist*`)`	estimate and graph separate functions for each group formed by *varlist*
`stratify`	estimate and graph separate functions for each stratum defined by `strata()` in estimation
`separate`	show group- and stratum-specific plots as subgraphs on one graph; default is to overlay these plots on one graph
`outfile(`*filename* [`, replace`]`)`	save values used to plot the goodness-of-fit graph
Options	
`name(`*namespec*`, ...)`	specify names for graphs
`saving(`*filespec*`, ...)`	save graphs in files
Plot	
connect_options	affect rendition of all plotted cumulative hazard functions
`plot#opts(`*connect_options*`)`	affect rendition of the #th plot
`byplot#opts(`*connect_options*`)`	affect rendition of the #th by-plot or stratum-specific plot
`event#opts(`*connect_options*`)`	affect rendition of plots for the #th event; available only after `stmgintcox`
`graph#opts(`*twoway_options*`)`	control the look of the #th graph; allowed only with `sepevents` after `stmgintcox`
`by#opts(`*byopts*`)`	how subgraphs are combined, labeled, etc. on the #th graph; allowed only with `sepevents` after `stmgintcox`
Reference line	
`rlopts(`*cline_options*`)`	affect rendition of reference lines
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
twoway_options	control the look of all graphs; any options other than `by()`, `name()`, or `saving()` documented in [G-3] ***twoway_options***
By options	
`byopts(`*byopts*`)`	how all subgraphs created by `by()`, `stratify`, or `events()` are combined, labeled, etc.

`estat gofplot` is not appropriate with `svy` estimation results and is not supported after estimation with `stcox, tvc()`.

Options

na specifies that the cumulative hazard function of the Cox–Snell residuals be calculated using the Nelson–Aalen estimator, which is the default after streg and stcox. na is not available after stintreg, stintcox, and stmgintcox.

km specifies that the cumulative hazard function of the Cox–Snell residuals be calculated using the minus log of the Kaplan–Meier estimator instead of the default Nelson–Aalen estimator after streg and stcox. km is not available after stintreg, stintcox, and stmgintcox.

turnbull specifies that the cumulative hazard function of the Cox–Snell-like residuals be calculated using the Turnbull estimator. turnbull is the only estimator after stintreg, stintcox, and stmgintcox, and it is not available after streg and stcox.

events(*evlist*) specifies that only the functions for the specified events be plotted. This option is available only after stmgintcox. The default is events(_all), which means estat gofplot will plot the cumulative hazard function of the Cox–Snell-like residuals for all events.

evlist may be _all (indicating all events), a numlist with values of the event variable, a list of labels from the value label for the event variable, or a list such as #1 #2 ..., with #1 meaning the first event, #2 meaning the second event, etc. For example, suppose the event variable contains values 1, 2, 3 with corresponding labels "event1", "event2", and "event3" defined in its value label. If we would like to plot the cumulative hazard functions for the first two events, we can specify estat gofplot with one of the following options: events(1 2), events("event1" "event2"), or events(#1 #2).

sepevents is meaningful only after stmgintcox. By default, estat gofplot creates a single graph with subgraphs for each event. sepevents specifies that the plots for each event be placed on separate graphs.

by(*varlist*) estimates a separate function for each by-group and overlays all the functions on one graph. By-groups are identified by equal values of the variables in *varlist*. Up to five variables are allowed. by() may not be combined with stratify. Additionally, by() is not allowed after estimation was performed with the strata() option.

stratify requires that a stratified model has been previously fit using the strata() option; it estimates a separate function for each stratum and overlays all the functions on one graph. stratify may not be combined with by().

separate is meaningful only with by() or stratify; it requests that each plot be placed as a separate subgraph rather than overlaid on top of other plots.

outfile(*filename*[, replace]) saves in *filename*.dta the values used to plot the goodness-of-fit graphs.

Options

`name(`*namespec*`[ , replace ])` specifies the name of the graph or multiple graphs. For a single graph, see [G-3] ***name_option***. If multiple graphs are produced, then the argument of `name()` is either a list of names or *stub*, in which case graphs are named *stub*`1`, *stub*`2`, and so on. `replace` causes existing graphs with the specified name or names to be replaced.

`saving(`*filespec*`[ , replace ])` specifies the filename or filenames to use to save the graph or multiple graphs to disk. For a single graph, see [G-3] ***saving_option***. If multiple graphs are produced, then the argument of `saving()` is either a list of filenames or a *stub*, in which case graphs are saved with filenames *stub*`1`, *stub*`2`, and so on. `replace` specifies that the file (or files) be replaced if it already exists.

Plot

connect_options control the rendition of all plotted cumulative hazard functions; see [G-3] ***connect_options***. They may be overridden for specific plots by using `plot#opts()`, `byplot#opts()`, or `event#opts()`.

`plot#opts(`*connect_options*`)` affects the rendition of the #th plotted cumulative hazard function. When multiple options apply to the same plot, the *connect_options* specified with `plot#opts()` will override those specified with `byplot#opts()`, and the options specified with `byplot#opts()` will override those specified with `event#opts()`.

`byplot#opts(`*connect_options*`)` affects the rendition of the #th by-plot created by `by()` or the #th stratum-specific plot created by `stratify`. When multiple options apply to the same plot, the *connect_options* specified with `plot#opts()` will override those specified with `byplot#opts()`, and the options specified with `byplot#opts()` will override those specified with `event#opts()`.

`event#opts(`*connect_options*`)` affects the rendition of the plotted cumulative hazard functions for the #th event after `stmgintcox`. When multiple options apply to the same plot, the *connect_options* specified with `plot#opts()` will override those specified with `byplot#opts()`, and the options specified with `byplot#opts()` will override those specified with `event#opts()`.

`graph#opts(`*twoway_options*`)` affects the appearance of the #th graph when `sepevents` is specified after `stmgintcox`. *twoway_opts* are any of the options documented in [G-3] ***twoway_options***, excluding `by()`, `name()`, and `saving()`.

`by#opts(`*byopts*`)` affects the appearance of the combined subgraphs on the #th graph when `sepevents` is specified after `stmgintcox`.

byopts may be any of the suboptions of `by()` documented in [G-3] ***by_option***, except for `total`, `missing`, and *legend_options*.

Reference line

`rlopts(`*cline_options*`)` affects the rendition of reference lines; see [G-3] ***cline_options***.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph; see [G-3] ***addplot_option***. `addplot()` is not allowed when the graph contains subgraphs.

Y axis, X axis, Titles, Legend, Overall

twoway_options control the appearance of all graphs; they are any of the options documented in [G-3] ***twoway_options***, excluding `by()`, `name()`, or `saving()`. These options include titling the graph (see [G-3] ***title_options***) and specifying legends (see [G-3] ***legend_options***). They may be overridden for specific graphs by using the `graph#opts()` option.

By options

`byopts(`*byopts*`)` affects the appearance of the combined subgraphs on all graphs. This option is applicable only when a graph contains subgraphs, and it may be overridden for specific graphs by using the `by#opts()` option.

byopts may be any of the suboptions of `by()` documented in [G-3] ***by_option***, except for `total`, `missing`, and *legend_options*.

After you fit a model with `streg`, `stcox`, `stintreg`, or `stintcox`, either the `by()` or `stratify` option must be specified along with the `separate` option for `byopts()` to be applicable. By default, plots for each group of the by variable are overlaid in a single graph, unless you specify the `separate` option. Combining the `separate` option with `by()` or `stratify` results in subgraphs for group- and stratum-specific plots; option `byopts()` allows you to modify the appearance of the combined subgraphs.

After you fit a model with `stmgintcox`, the simplest specification of `estat gofplot` will create a graph with subgraphs for each event. The `separate` option is not needed when using `byopts()` after estimation with `stmgintcox`.

Remarks and examples

To assess the overall model fit, we can use the Cox–Snell residuals. If the survival regression model fits the data, these residuals should have a censored standard exponential distribution for right-censored data. Therefore, when we consider these residuals as failure (or censoring) times together with the original censoring variable, the hazard function should be constant and equal to 1, and the cumulative hazard should be a straight line with slope 1. `estat gofplot` allows us to verify the model's fit visually by calculating an empirical estimate of such a cumulative hazard function, which is based on either the Nelson–Aalen estimator or the Kaplan–Meier estimator for `streg` or `stcox` or is based on the Turnbull estimator for `stintreg`, `stintcox`, or `stmgintcox`. If the model fits the data, a plot of the cumulative hazard versus the residuals themselves should approximate a straight line with slope 1. See example 2 in [ST] **streg postestimation**, example 4 in [ST] **stcox postestimation**, example 4 in [ST] **stintreg postestimation**, example 2 in [ST] **stintcox postestimation**, and example 3 in [ST] **stmgintcox postestimation** for more detailed discussions. For interval-censored data, the Cox–Snell-like residuals are defined, and, under the correct model assumption, they are expected to approximate an interval-censored sample from the standard exponential distribution; see [ST] **stintcox postestimation**.

Example 1: Goodness-of-fit plots for a stratified model

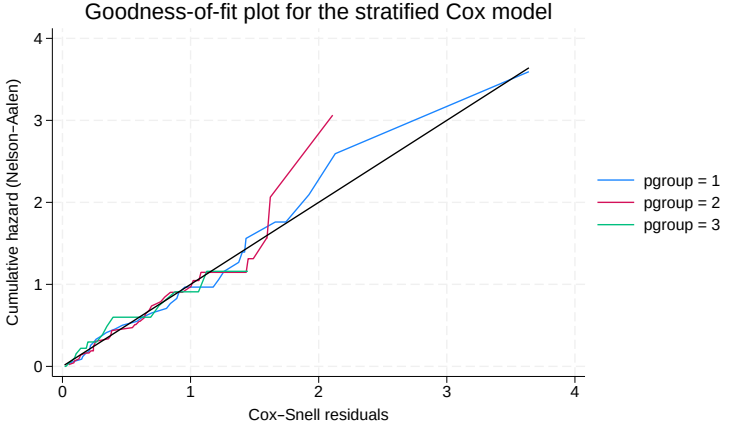
Returning to the Stanford heart experiment data from example 8 in [ST] **stcox**, we refit our model stratified by year of diagnosis group.

```
. use https://www.stata-press.com/data/r19/stan3
(Heart transplant data)
. generate pgroup = year
. recode pgroup min/69=1 70/72=2 73/max=3
(172 changes made to pgroup)
. stcox age posttran surg year, strata(pgroup) nolog
        Failure _d: died
  Analysis time _t: t1
       ID variable: id
Stratified Cox regression with Breslow method for ties
Strata variable: pgroup
No. of subjects =      103                     Number of obs =    172
No. of failures =       75
Time at risk    = 31,938.1
                                               LR chi2(4)    =  20.67
Log likelihood = -213.35033                    Prob > chi2   = 0.0004
```

_t	Haz. ratio	Std. err.	z	P>|z|	[95% conf.	interval]
age	1.027406	.0150188	1.85	0.064	.9983874	1.057268
posttran	1.075476	.3354669	0.23	0.816	.583567	1.982034
surgery	.2222415	.1218386	-2.74	0.006	.0758882	.6508429
year	.5523966	.1132688	-2.89	0.004	.3695832	.825638

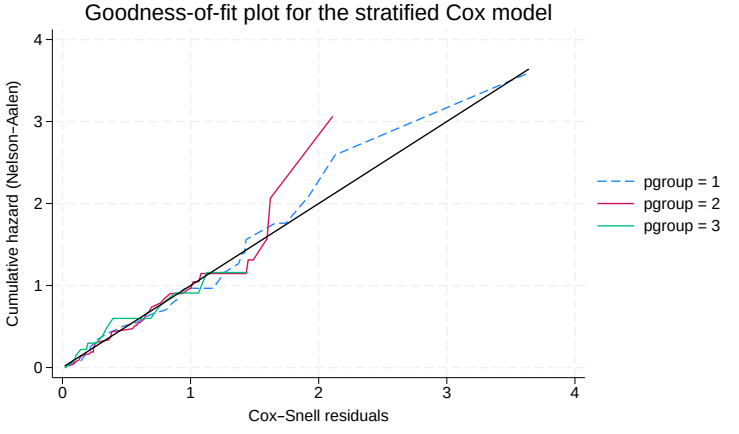
We can visually explore the goodness of fit for the stratified model by using `estat gofplot`. Here we will use the `stratify` option to plot the residuals separately for each stratum.

```
. estat gofplot, stratify
```



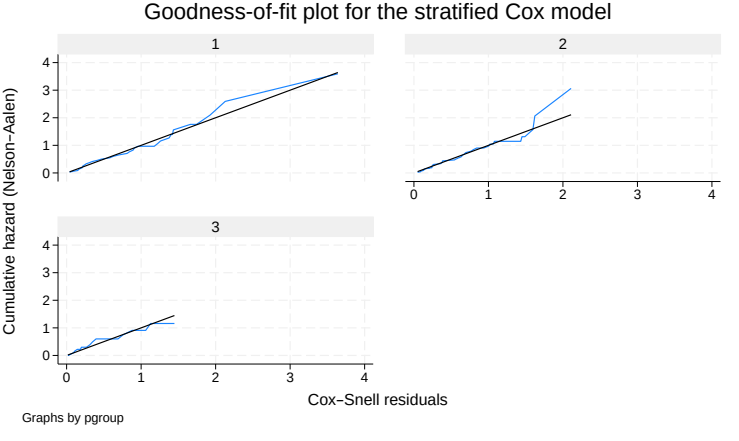
We can use the `plot#opts()` option to control the rendition of individually plotted cumulative hazard functions. Suppose that we want to change the first plotted cumulative hazard function to be a dashed line; to do that, we specify the `lpattern(dash)` suboption in the `plot1opts()` option.

```
. estat gofplot, stratify plot1opts(lpattern(dash))
```



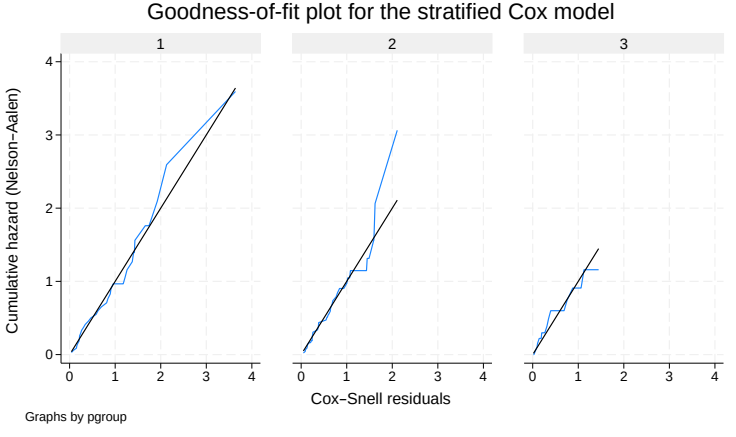
Even with perfect (artificially simulated) data, we expect to see departures from the diagonal at the right end of the curve, where values are based on only a few observations and greater fluctuation is observed. The above plot indicates the model fits well in all strata. To aid visual inspection of the plot, we can also add the `separate` option to produce separate graphs for each stratum.

```
. estat gofplot, stratify separate
```



By default, `estat gofplot` uses the rules described in [G-3] ***by_option*** to place those by-plots or stratum-specific plots. You can change the look of those plots using the `byopts()` option. Suppose that we would like to create those stratum-specific plots side by side; to do that, we specify the `rows(1)` suboption in the `byopts()` option.

```
. estat gofplot, stratify separate byopts(rows(1))
```



Example 2: Goodness-of-fit plots for interval-censored multiple-event data

To visually assess the overall model fit for interval-censored multiple-event data, we can use the event-specific Cox–Snell-like residuals. `estat gofplot` calculates an empirical estimate of the cumulative hazard function based on the Cox–Snell-like residuals for each event and plots the resulting cumulative hazard rate against the residuals themselves. If the model fits the data, those plots are expected to approximate a straight line with slope 1.

Continuing with example 1 in [ST] **stmgintcox postestimation**, we first refit our model but also suppress the log with the `nolog` option and use the `favorspeed` option to speed up command execution for demonstration.

```
. use https://www.stata-press.com/data/r19/aric
(Simulated ARIC data)
. stmgintcox age i.male i.community i.race bmi glucose sysbp diabp,
> id(id) event(event) interval(ltime rtime) nolog favorspeed
note: using fixed step size with a multiplier of 5 to compute derivatives.
note: using EM and VCE tolerances of 0.0001.
note: option noemhsgtolerance assumed.
```

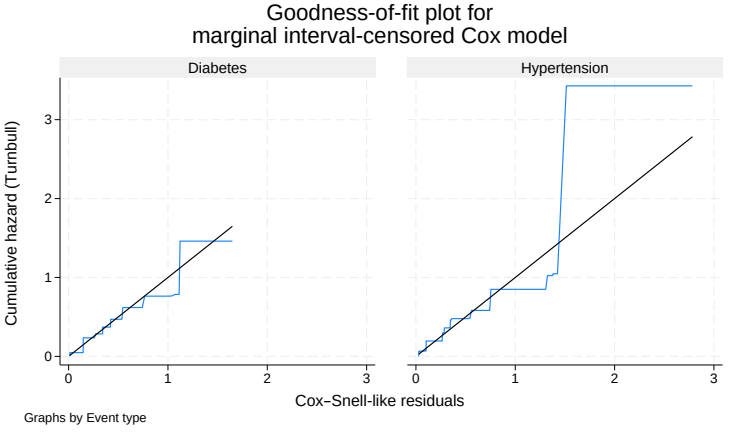
```
Marginal interval-censored Cox regression          Number of events   =      2
Baseline hazard: Reduced intervals                 Number of subjects =    200
                                                   Number of obs      =    400
ID variable: id                                           Uncensored  =      0
Event variable: event                                Left-censored    =     47
Event-time interval:                                 Right-censored   =    240
  Lower endpoint: ltime                              Interval-cens.   =    113
  Upper endpoint: rtime
                                                   Wald chi2(20)      =  84.36
Log pseudolikelihood = -270.83984                  Prob > chi2        = 0.0000
```

	Haz. ratio	Robust std. err.	z	P>|z|	[95% conf.	interval]
Diabetes						
age	.9552606	.0295589	-1.48	0.139	.8990481	1.014988
male						
Yes	.8084224	.2400335	-0.72	0.474	.451755	1.446684
community						
Jackson	1.597828	.6069935	1.23	0.217	.7588748	3.364265
Minneapolis	1.028054	.342976	0.08	0.934	.5346148	1.976929
Washington	1.407869	.5192024	0.93	0.354	.6833627	2.900504
race						
White	.4289702	.1273669	-2.85	0.004	.2397145	.7676444
bmi	1.116579	.034187	3.60	0.000	1.051545	1.185636
glucose	1.139753	.0303702	4.91	0.000	1.081756	1.200859
sysbp	1.020295	.0122308	1.68	0.094	.9966021	1.04455
diabp	.9928634	.0127512	-0.56	0.577	.9681835	1.018172
Hypertension						
age	.9950085	.0225503	-0.22	0.825	.9517779	1.040203
male						
Yes	.6671401	.1599892	-1.69	0.091	.4169533	1.067448
community						
Jackson	.6085406	.1953944	-1.55	0.122	.3243246	1.141824
Minneapolis	.9040647	.2719638	-0.34	0.737	.5013468	1.630275
Washington	.674088	.2085739	-1.27	0.202	.3675707	1.23621
race						
White	1.261355	.425064	0.69	0.491	.6516152	2.441652
bmi	1.012196	.0195117	0.63	0.529	.9746672	1.05117
glucose	.989899	.0101396	-0.99	0.322	.9702238	1.009973
sysbp	1.075011	.0162901	4.77	0.000	1.043553	1.107418
diabp	1.025533	.0134835	1.92	0.055	.9994433	1.052303

Note: Standard error estimates may be more variable for small datasets and datasets with low proportions of interval-censored observations.

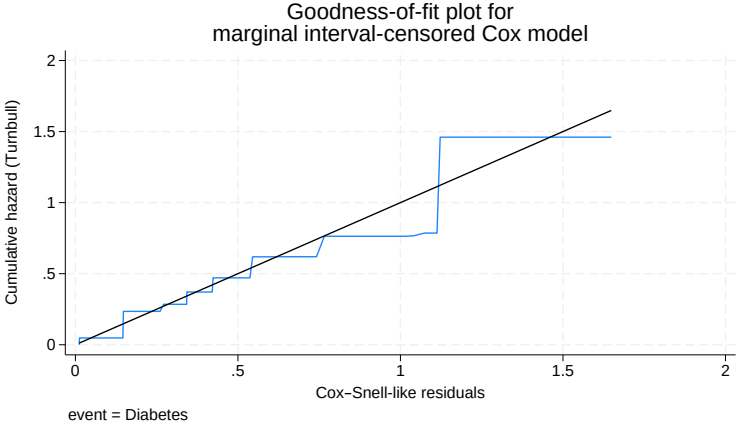
Now, let's produce the goodness-of-fit plots for all events. By default, `estat gofplot` creates a single graph with subgraphs for the cumulative hazard function for each event.

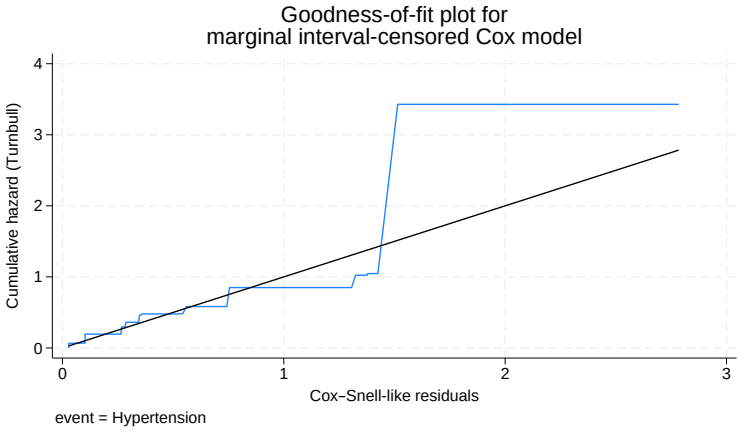
```
. estat gofplot
```



You can add the `sepevents` option to request that the plot for each event be placed on a separate graph.

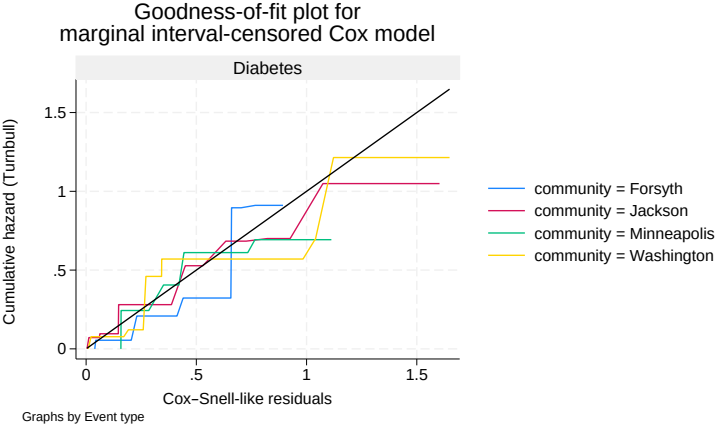
```
. estat gofplot, sepevents
```





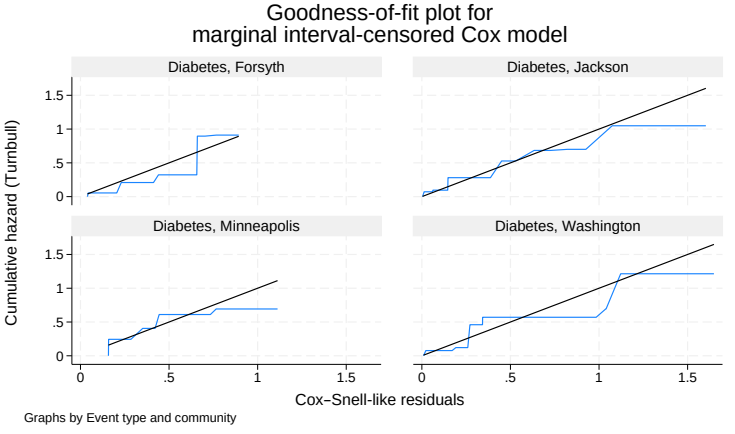
If we want to examine the goodness-of-fit plots for diabetes across different communities, we can use the `by(community)` option along with `events("Diabetes")`. The `estat gofplot` command will display these community-specific plots for diabetes overlaid on a single graph.

```
. estat gofplot, events("Diabetes") by(community)
```



To aid visual inspection, we can also add the `separate` option to produce separate subgraphs for each community.

```
. estat gofplot, events("Diabetes") by(community) separate
```



Methods and formulas

The Cox and Snell (1968) residual for the jth observation at time t_j is defined as the estimated cumulative hazard function, $\widehat{H}_j(t_j) = -\log\hat{S}_j(t_j)$, from the fitted model (Klein and Moeschberger 2003). Cox and Snell argued that if the correct model has been fit to the data, these residuals are n observations from a censored standard exponential distribution for right-censored data. 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 0. In practice, we can calculate an empirical estimate of the cumulative hazard rate of the residuals.

The default method of calculating the cumulative hazard rate of the residuals after `streg` and `stcox` is to use the Nelson–Aalen estimator (Nelson 1972; Aalen 1978). Alternatively, we may use the minus log of the Kaplan and Meier (1958) estimator by specifying the `km` option. For multiple-record data, the overall Cox–Snell residual is used, and hence, the cumulative hazard function is evaluated at the subject level defined by `id()` in the `stset` command.

For interval-censored data, Cox–Snell-like residuals are intervals themselves for single-record-per-subject data. Farrington (2000) proposed to calculate the Cox–Snell-like residuals for both lower and upper endpoints of the time intervals, then to use those predicted Cox–Snell-like residual intervals as the new time intervals and to compute the cumulative hazard function using the Turnbull estimator (Turnbull 1976). For multiple-record-per-subject data fit by `stintcox`, the overall Cox–Snell-like residual is used, and the cumulative hazard function is evaluated at the subject level defined by the `id()` option of `stintcox`. For interval-censored multiple-event data fit by `stmgintcox`, the cumulative hazard function for each event is evaluated and plotted.

References

Aalen, O. O. 1978. Nonparametric inference for a family of counting processes. *Annals of Statistics* 6: 701–726. https://doi.org/10.1214/aos/1176344247.

Cox, D. R., and E. J. Snell. 1968. A general definition of residuals (with discussion). *Journal of the Royal Statistical Society*, B ser., 30: 248–275. https://doi.org/10.1111/j.2517-6161.1968.tb00724.x.

Farrington, C. P. 2000. Residuals for proportional hazards models with interval-censored survival data. *Biometrics* 56: 473–482. https://doi.org/10.1111/j.0006-341X.2000.00473.x.

Kaplan, E. L., and P. Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association* 53: 457–481. https://doi.org/10.2307/2281868.

Klein, J. P., and M. L. Moeschberger. 2003. *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd ed. New York: Springer.

Nelson, W. 1972. Theory and applications of hazard plotting for censored failure data. *Technometrics* 14: 945–966. https://doi.org/10.2307/1267144.

Turnbull, B. W. 1976. The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society*, B ser., 38: 290–295. https://doi.org/10.1111/j.2517-6161.1976.tb01597.x.

Also see

[ST] **stcox postestimation** — Postestimation tools for stcox

[ST] **stintcox postestimation** — Postestimation tools for stintcox

[ST] **stintreg postestimation** — Postestimation tools for stintreg

[ST] **stmgintcox postestimation** — Postestimation tools for stmgintcox

[ST] **streg postestimation** — Postestimation tools for streg

[U] **20 Estimation and postestimation commands**