The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$115	1-year subscription	$145
2-year subscription	$210	2-year subscription	$270
3-year subscription	$285	3-year subscription	$375
1-year student subscription	$ 85	1-year student subscription	$115
1-year institutional subscription	$345	1-year institutional subscription	$375
2-year institutional subscription	$625	2-year institutional subscription	$685
3-year institutional subscription	$875	3-year institutional subscription	$965
Electronic only		**Electronic only**	
1-year subscription	$ 85	1-year subscription	$ 85
2-year subscription	$155	2-year subscription	$155
3-year subscription	$215	3-year subscription	$215
1-year student subscription	$ 55	1-year student subscription	$ 55

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

Tools for checking calibration of a Cox model in external validation: Prediction of population-averaged survival curves based on risk groups

Patrick Royston
MRC Clinical Trials Unit at University College London
London, UK
j.royston@ucl.ac.uk

Abstract. Royston (2014, *Stata Journal* 14: 738–755) explained how a popular application of the Cox proportional hazards model "is to develop a multivariable prediction model, often a prognostic model to predict the future clinical outcome of patients with a particular disorder from 'baseline' factors measured at some initial time point. For such a model to be useful in practice, it must be 'validated'; that is, it must perform satisfactorily in an external sample of patients independent of the one on which the model was originally developed. One key aspect of performance is calibration, which is the accuracy of prediction, particularly of survival (or equivalently, failure or event) probabilities at any time after the time origin". In this article, I suggest an approach to assess calibration by comparing observed (Kaplan–Meier) and predicted survival probabilities in several prognostic groups derived by placing cutpoints on the prognostic index. I distinguish between full validation, where all relevant quantities are estimated on the derivation dataset and predicted on the validation dataset, and partial validation, where the prognostic index and prognostic groups are derived from published information and the baseline distribution function is estimated in the validation dataset. Partial validation is more feasible in practice because it is uncommon to have access to individual patient values in both datasets. I exemplify the method by detailed analysis of two datasets in the disease primary biliary cirrhosis; the datasets comprise a derivation and a validation dataset. I describe a new ado-file, `stcoxgrp`, that performs the necessary calculations. Results for `stcoxgrp` are displayed graphically, which makes it easier for users to picture calibration (or lack thereof) according to follow-up time.

Keywords: st0380, stcoxgrp, Cox proportional hazards model, multivariable model, prognostic factors, prognostic groups, external validation, calibration, survival probabilities

1 Introduction

The Cox proportional hazards model has long been a standard tool in developing multivariable models for time-to-event data. However, the ability to check the fit of a Cox model by comparing predicted survival functions with observed (Kaplan–Meier) curves in subgroups is, I believe, a tool that many researchers want and that currently available commands do not comprehensively provide. Such predictions are often known as population-averaged survival curves. I describe a new command, `stcoxgrp`, that calculates population-averaged survival curves. Prognostic subgroups are often defined on the data. For example, risk groups may be derived by categorizing a prognostic index (PI) (linear predictor from the Cox model) or may be expressed as the levels of an important categorical variable (such as disease severity). Predicted curves in such subgroups are easier to interpret than those calculated at specific covariate values, for example, by using the Stata postestimation command `stcurve`. If one categorical variable has been fit, `stcoxgrp` gives the same results for the subgroups as `stcoxkm`. However, if the model includes finer structure—for example, if it includes several covariates, some of which may be continuous—`stcoxgrp` averages predicted survival curves across the structure within a given subgroup as defined by the covariate patterns in the data. I provide further details of these calculations in section 4.

An important extension to assessing model fit on a given dataset is external validation. This entails evaluating the performance (discrimination and calibration or predictive accuracy) of a model in a sample independent of that used to develop the model. Little has been published on techniques for external validation of Cox models. In the context of medical prognosis, successful validation means that a model discriminates between good and bad outcomes in patients whose data were not involved in the development of the model. Calibration refers to the predictive accuracy of survival probabilities.

Assessing discrimination of a model in the development ("derivation") or independent ("validation") datasets is rather straightforward. Tools such as Harrell et al.'s (1982) c-index of concordance or Royston and Sauerbrei's (2004) D measure of discrimination are applied to the PI (linear predictor, $\mathbf{x}\widehat{\boldsymbol{\beta}}$) from the Cox model. Assessing calibration of Cox models is trickier, mainly because the Cox model estimates event probabilities indirectly and only relative to an unspecified baseline survival function. The challenges in validating a Cox model are further discussed by Royston and Altman (2013).

In this article, I describe `stcoxgrp` and how it may help the analyst to assess the calibration of a Cox model developed on one dataset and, if relevant, one externally validated on others. As indicated above, the approach to external validation is based on the prediction of survival (or event) probabilities for groups of individuals in the validation dataset. I make a graphical comparison between the predicted population-averaged survival probabilities in risk groups and the nonparametric (Kaplan–Meier) estimates in the same groups.

As an example, I apply the methods to a pair of datasets in primary cirrhosis of the liver. I treat the larger of the pair as the derivation sample and the smaller as the validation sample.

2 Data

Derivation dataset Primary biliary cirrhosis (PBC) is a serious liver disease that usually results in liver failure and death. Here I use a dataset originally in Fleming and Harrington (1991) to illustrate certain aspects of survival analysis. Observations on the first 312 patients (125 deaths) in the dataset were obtained from a randomized controlled trial of 2 treatments for PBC performed at the Mayo Clinic between 1974 and 1984. An additional 106 patients (36 deaths) did not take part in the trial but consented to have 6 variables measured and to be followed up with for survival (the "cohort study"). Randomized treatment and 16 prognostic factors were recorded in the trial, including the subset of 6 recorded in the cohort study. For our purposes, we use only the three prognostic variables that are common to the cohort study, the trial, and our chosen validation dataset (see below): `x1` (age), `x2` (log bilirubin), and `x3` (albumin). The derivation dataset comprises the cohort and trial datasets (418 patients). The outcome is time to death from any cause.

Validation dataset The effect of the drug azathioprine on the survival of patients with PBC was compared with that of a placebo in a multinational, double-blind, randomized clinical trial (Christensen et al. 1985). Between 1971 and 1977, 248 patients were randomized to receive either azathioprine or a placebo, with follow-up until 1983. After 41 (17%) cases with missing values or no patient follow-up were removed, data on 207 patients (105 deaths) were available for analysis. As with the derivation dataset, we used only the prognostic factors age, log bilirubin, and albumin in studying validation.

Please note that because different drugs were used in the two studies, the possible effect of the drug has not been included in the prognostic model. However, analysis of the drug effect in each dataset separately shows only a weak, nonsignificant effect. By contrast, the influence of the three prognostic variables is far stronger and will overwhelm small drug effects.

The combined dataset (`pbc.dta`) is supplied with the software described in this article. It includes a variable called `val`, which is 0 for the derivation dataset and 1 for the validation dataset.

3 Assessing calibration of survival probabilities

Suppose we have a vector of explanatory variables, $\mathbf{x} = (x_1, \ldots, x_k)$. A Cox model with parameter vector $\boldsymbol{\beta}$ incorporates multiplicative effects of $\mathbf{x}$ on the hazard function and is usually written as

$$h\left(t; \mathbf{x}\right) = h_0\left(t\right) \exp\left(\mathbf{x}\boldsymbol{\beta}\right)$$

where $h\left(t; \mathbf{x}\right)$ is the hazard function, $h_0\left(t\right) = h\left(t; \mathbf{0}\right)$ is the baseline hazard function, and t is the follow-up time. The corresponding survival function $S\left(t; \mathbf{x}\right)$ is given by

$$S\left(t; \mathbf{x}\right) = S_0\left(t\right)^{\exp\left(\mathbf{x}\boldsymbol{\beta}\right)} \tag{1}$$

where the baseline survival function is $S_0\left(t\right) = \exp\{-H_0\left(t\right)\}$, and $H_0\left(t\right) = \int_0^t h_0\left(u\right) du$ is the baseline cumulative hazard function. In Stata 11 and later, estimates of $S_0\left(t\right)$

and $H_0(t)$ are available following model fitting with `stcox` through the `basesurv` and `basechazard` options of `predict`, respectively.

To check the calibration of a model, we need to compare the observed event probabilities with the event probabilities predicted by the model. For the Cox model, the most relevant outputs are cumulative hazards and survival (or event) probabilities. Because these are functions of each other (see above), we need to check only one of them. We focus on survival probabilities. Although the Cox model does not directly estimate baseline survival probabilities, we can use postestimation features implemented in the `basesurv` and `basechazard` options of `predict` following model fitting by `stcox`. From these, we can use (1) to estimate entire survival curves for individuals. We can determine nonparametric estimates of survival probabilities in groups of patients by using the Kaplan–Meier method. The appropriate Stata command for this is `sts generate` *newvar* = `s, by(`*groupvar*`)`.

It is commonplace to create only a few (say, three or four) "risk groups" by imposing cutpoints on the PI, $\mathbf{x}\widehat{\boldsymbol{\beta}}$. A plot of Kaplan–Meier curves by group indicates the discrimination available with the model and the appearance of the survival curves. An example from the derivation PBC dataset is shown in figure 1.

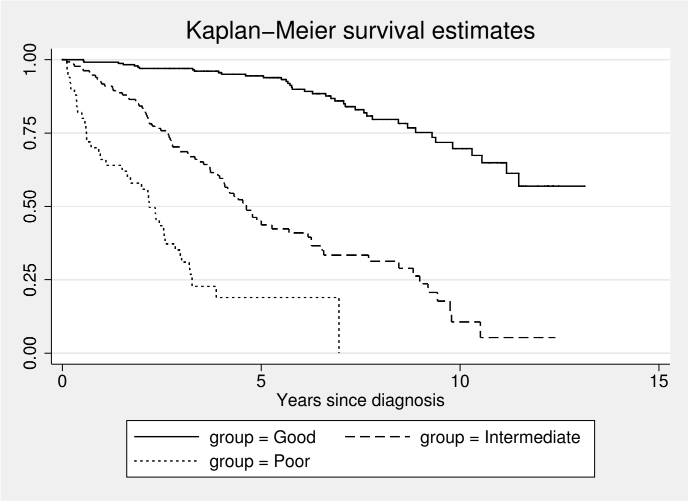

Figure 1. Kaplan–Meier survival curves for three prognostic groups in the derivation dataset

We adopt a similar approach here, but to assess calibration, we must go further. We need to compute survival probabilities predicted from a Cox model to compare them with the nonparametric Kaplan–Meier estimates. We can do this in both the derivation dataset and the validation dataset.

4 Steps to predict survival probabilities for groups

Suppose that the baseline survival function (estimated at PI = 0) in the derivation dataset is $S_0(t)$. If the Cox model is correctly specified in both datasets, the datasets should have similar baseline survival functions. If the model is flawed, the baseline survival functions will differ. Let's assume that $S_0(t)$ in the derivation dataset is available in some convenient form, for example, as an approximating mathematical function. (We will return to the question of finding such an estimate of $S_0(t)$ later.) A procedure for checking calibration in the derivation and validation datasets is as follows:

1. Let $x_i\widehat{\boldsymbol{\beta}}$ denote the PI of the ith individual. According to the Cox model on the derivation dataset, the individual predicted survival function is $\widehat{S}(t;x_i) = \widehat{S}_0(t)^{\exp\left(x_i\widehat{\boldsymbol{\beta}}\right)}$.

2. For a given risk group with patient indices belonging to a set G, we average the individual survival functions, $\{\widehat{S}(t;x_i)\}_{i\in G}$, at the observed event or censoring times to obtain the survival curves predicted by the model; that is, we determine the population-average predictions for G within the risk group according to the fitted Cox model.

3. For a graphical comparison, we plot the predicted and observed (Kaplan–Meier) survival curves against follow-up time within each risk group.

We also could present the expected and observed survival probabilities in tabular form at specific time points. Using 95% confidence intervals for the Kaplan–Meier estimates helps us to gauge informally how well the predictions agree with the observed probabilities.

5 Worked example

I now give an example to show how the above procedure can be performed with real data in Stata. As a preliminary, we assume the following:

1. The data file comprises the derivation and validation datasets, and it includes a variable (say, `val`) that equals 0 for the derivation dataset and 1 for the validation dataset.

2. Each dataset has the same outcome variable and the same covariates defined in the same way (for example, same units of measurement, same categories for categorical variables).

3. A Cox model has been fit to the derivation dataset, and a PI (`xb`) has been predicted on both datasets by using the command `predict xb, xb`. To ensure numerical stability, we center the PI by subtracting its mean over the derivation dataset.

Below we present code to perform the model fitting and calibration:

1. Fit the model on the derivation data, and predict the PI on both datasets.

   ```
   use pbc
   stcox x1 x2 x3 if val==0
   predict xb, xb
   ```

2. Center the PI on the derivation dataset mean.

   ```
   summarize xb if val==0
   replace xb = xb - r(mean)
   ```

3. Define 3 prognostic groups from the 25th and 75th centiles of `xb` in the derivation dataset. (This is done on events because the number of events gets very small in the "Good" prognostic group.)

   ```
   centile xb if val==0 & _d==1, centile(25 75)
   generate byte group = cond(xb <= r(c_1), 1, cond(xb >= r(c_2), 3, 2))
   ```

4. Get the baseline log cumulative hazard, `lnH0`, in the derivation data.

   ```
   stcox xb if val==0
   predict H0, basechazard
   generate lnH0 = ln(H0)
   ```

5. Compute the smoothed baseline log cumulative-hazard function on `_t`.

   ```
   fracpoly: regress lnH0 _t if val==0
   ```

6. Compute mean survival probabilities in each dataset at `t` = 0(1)10 years.

   ```
   range t 0 10 11
   fraceval var t
   generate S0 = cond(t==0, 1, exp(-exp(_fp)))
   stcoxgrp xb S0 t, mean(s) km(km) by(val group)
   ```

7. For the derivation data, compare observed with predicted survival.

   ```
   twoway (scatter km1 km2 km3 t, mcolor(gs5 gs8 gs10))                     ///
    (rcap km_lb1 km_ub1 t, lcolor(gs5 ..))                                  ///
    (rcap km_lb2 km_ub2 t, lcolor(gs8 ..))                                  ///
    (rcap km_lb3 km_ub3 t, lcolor(gs10 ..))                                 ///
    (line s1 s2 s3 t, sort lwid(medthick ..) lcolor(gs5 gs8 gs10)),         ///
    legend(off) title("Derivation data")                                    ///
    xlabel(0(2)10) ylabel(0(.25)1, angle(h) format(%4.2f)) ytitle("")       ///
    xtitle("") name(g1, replace)
   ```

8. For the validation data, compare observed with predicted survival.

```
twoway (scatter km4 km5 km6 t, mcolor(gs5 gs8 gs10))                       ///
 (rcap km_lb4 km_ub4 t, lcolor(gs5 ..))                                     ///
 (rcap km_lb5 km_ub5 t, lcolor(gs8 ..))                                     ///
 (rcap km_lb6 km_ub6 t, lcolor(gs10 ..))                                    ///
 (line s4 s5 s6 t, sort lwid(medthick ..) lcolor(gs5 gs8 gs10)),            ///
 legend(off) title("Validation data")                                       ///
 xlabel(0(2)10) ylabel(0(.25)1, angle(h) format(%4.2f)) ytitle("")         ///
 xtitle("") name(g2, replace)
```

The following explanations of these steps may be helpful:

Step 1. We fit a Cox model to the derivation data.

Step 2. Centering the PI is helpful for two reasons: numerical stability and interpretation of the ensuing baseline function. According to the Stata reference manual (StataCorp 2013),

> [w]hen predicting with `basesurv` or `basechazard`, for numerical accuracy reasons, the baseline functions must correspond to something reasonable in your data. Remember, the baseline functions correspond to all covariates equal to 0 in your Cox model.

And further:

> For these reasons [...], covariate values of 0 must be meaningful if you are going to specify the `basechazard` or `basesurv` option. As the baseline values move to absurdity, the first problem you will encounter is a baseline survivor function that is too hard to interpret, even though the baseline hazard contributions are estimated accurately. Further out, the procedure Stata uses to estimate the baseline hazard contributions will break down—it will produce results that are exactly 1.

We chose to center the PI on its mean, but any reasonable value within the range of the PI would be acceptable.

Step 3. Deciding the number and location of cutpoints for creating prognostic (risk) groups is more an art than a science. As Royston and Altman (2013) state,

> [S]tatistical common sense dictates that a modest number of risk groups (say, 5 or fewer) is preferable to a large number. Two groups is probably too few to satisfy the needs of clinical practice and research applications. With a large number, the survival curves may be unstable and the discrimination between neighboring groups is likely to be poor. Unequal group sizes seem preferable to equal groups, because they enable identification of patients with more extreme prognoses and group together patients with largely similar prognoses.

We categorized the PI at the 25th and 75th centiles of the failure times, that is, not counting censored observations. We restricted centiles to the failure times because the number of events in the Good prognosis group would otherwise be too small to compute Kaplan–Meier curves with reasonable precision. The corresponding centiles of all the observed times were the 56th and 88th, so the Good prognostic group contains more than half of the patients but only a quarter of the events.

Step 4. We estimated the baseline log cumulative-hazard function (`lnH0`) in the derivation dataset following Cox regression on the centered PI. We did this by using `predict H0, basechazard` followed by `generate lnH0 = ln(H0)`. We used $\ln H_0(t)$ because its functional form tends to be easier to smooth on t than $H_0(t)$ or $S_0(t)$.

Step 5. We approximated the log cumulative hazard, `lnH0`, as a function of t using a second-degree fractional polynomial (FP2). The `fracpoly` command determined that the best-fitting powers were $(-0.5, 0.5)$, so the approximating model was $\ln H_0(t) = a_0 + a_1 t^{-0.5} + a_2 t^{0.5}$. The coefficients (a_0, a_1, a_2) were estimated by least squares. The statement `predict lnH0f` predicted the log baseline cumulative-hazard function across both datasets. Note that as of Stata 13, we could use `fp` instead of `fracpoly` (although the latter command is still available). We further discuss this step in the next section.

Step 6. Note that `fraceval` and `stcoxgrp`, both programs provided with this article, must be installed before performing this step. (See `help net` for further information on installing packages.) We decided to display the observed and fitted survival probabilities, with pointwise 95% confidence intervals for the latter, at times $0, 1, \ldots, 10$ years after diagnosis. First, we used the `range` command to create a new variable, `t`, taking these values. This occupied only the first 11 rows of the dataset. Second, we then used the user-written command `fraceval` to predict the fitted values, here the baseline log cumulative-hazard function, from the FP2 function at `t`. Here we used out-of-sample prediction. We computed the corresponding baseline survival function in `S0`. Third, we applied `stcoxgrp` (see description below) to the PI (`xb`) and to `S0` and `t` to obtain the survival probabilities based on the Cox model at t for each group (`group` = 1, 2, 3) and dataset (`val` = 0, 1). The command `stcoxgrp` used the information from `stset` to compute the corresponding probabilities and confidence intervals based on the Kaplan–Meier estimator.

Steps 7 and 8. Finally, we plotted the results for the derivation and validation datasets separately. We could also have tabulated the results. The resulting plots are shown in figure 2.

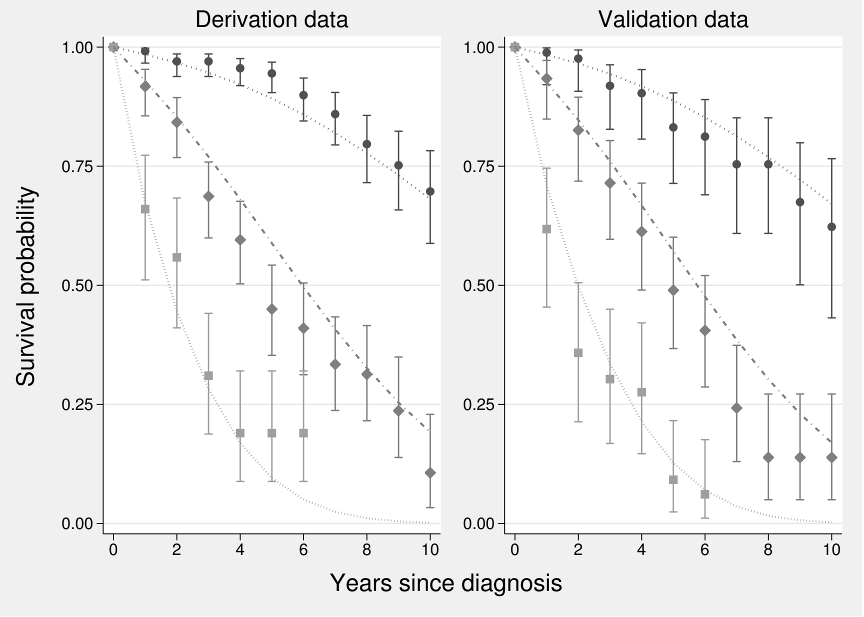

Figure 2. Calibration of a Cox model in the derivation and validation datasets. Smooth dashed lines represent predicted survival probabilities, and vertical capped lines denote Kaplan–Meier estimates with 95% confidence intervals. Three prognosis groups are plotted: the "Good" group (darkest lines), the "Intermediate" group (medium-dark lines), and the "Poor" group (paler lines).

The calibration of the model is imperfect in each dataset. In the derivation data, the actual survival is slightly better than predicted in the Good prognosis group and slightly worse than predicted in the Intermediate group. The predictions in the validation dataset are somewhat too high; that is, survival tends to be worse than predicted. Because there are fewer events in the validation data, the confidence intervals tend to be wider than in the derivation data.

6 Approximating the baseline survival function

Let's assume that steps 1, 2, and 3 of the previous section have been done. We wish to approximate the baseline survival function in the derivation dataset. We approach this via the log cumulative-hazard function. We refit the Cox model to the centered PI, predict the baseline cumulative hazard, transform it to logs, and then predict the baseline survival function.

```
stcox xb if val==0
predict H0, basechazard
generate lnH0 = ln(H0)
predict S0, basesurv
```

Figure 3 illustrates the results and shows plots of the baseline log cumulative-hazard function, `lnH0`, and baseline survival function, `S0`, for the derivation dataset against time.

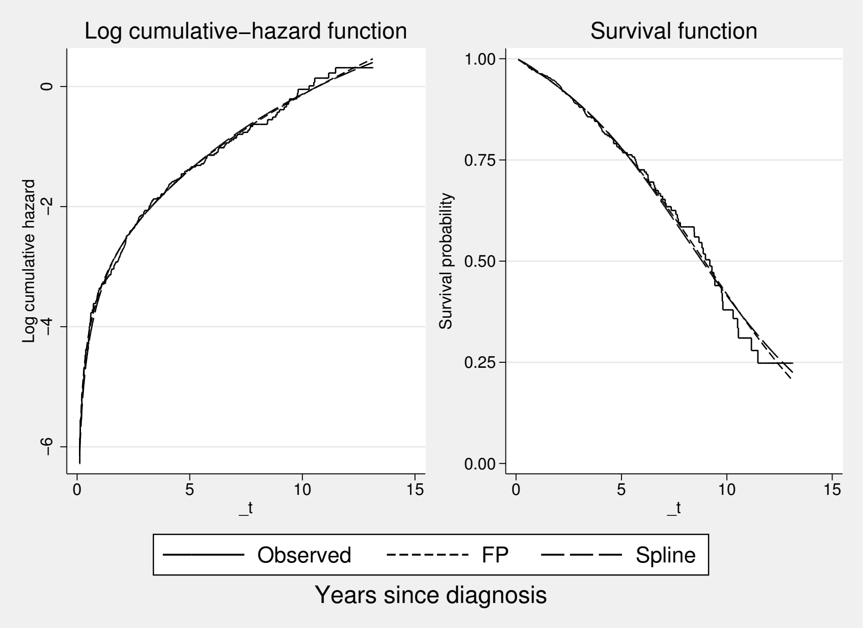

Figure 3. Baseline log cumulative-hazard function and baseline survival function in the derivation dataset. Jagged line, Kaplan−Meier-like, given by `predict, basechazard` and `predict, basesurv` ; smooth lines, FP, or spline approximations.

The observed log cumulative-hazard values typically form a simple curve, which fractional polynomials (FP) often do a good job of approximating as a function of t. My experience suggests that the crude approach of applying ordinary least-squares regression to FP functions of time is quite satisfactory, despite the high autocorrelation among the values of the dependent variable. It is possible to estimate the same FP function "properly" by using maximum likelihood, but the extra effort and complexity makes this a less attractive proposition.

We could also approximate $\ln H_0(t)$ by using restricted cubic spline regression on $\ln t$. Full maximum-likelihood estimation of such models has been implemented by Royston (2001) (the stpm command) and Lambert and Royston (2009) (the stpm2 command). The postestimation predict command for stpm2 allows out-of-sample prediction of fitted values, which is also available with stcoxgrp (see step 6 of the previous section). The following commands illustrate how to approximate $S_0(t)$ using stpm2 and stcoxgrp:

```
stpm2 xb if val==0, scale(hazard) df(3)
range t 0 10 11
predict S0, zeros survival timevar(t)
stcoxgrp xb S0 t, mean(s) km(km) by(val group)
```

We need only to select the required complexity of the spline function, as determined by the df() option of stpm2. Here we used df(3), which, in my experience, is a good general choice. See Royston and Lambert (2011, 121–133) for how to choose the spline degrees of freedom. The predict command handles the change of scale from log cumulative hazard to survival.

Figure 3 shows that in this case, the resulting smooth curves from the two methods (an FP2 curve estimated by ordinary least squares and a spline curve estimated by maximum likelihood) are almost indistinguishable.

7 Partial validation

In reality, it is unusual for a researcher to have access to individual patient values for both the derivation and the validation datasets. More commonly, the regression coefficients (or hazard ratios) for the terms of the Cox model of interest are published. This allows one to reconstruct the PI in an independent dataset.

The methods described above are applicable to what may be called a “partial validation” approach. The PI and prognostic groups are calculated from the published information, and the tools described here are applied within the validation dataset. The resulting graph of observed and predicted survival probabilities, akin to the right-hand panel of figure 2, shows how well the “external” PI combined with the “internal” smooth baseline cumulative survival function is calibrated on the validation dataset.

The steps of the procedure are very similar to those for “proper” external validation. The steps are as follows:

1. Fit the model on the derivation data, and predict the PI on both datasets. Keep only the validation data together with the predicted PI.

    ```
    use pbc, clear
    stcox x1 x2 x3 if val==0
    predict xb, xb
    ```

2. Define 3 prognostic groups from the 25th and 75th centiles of `xb` in the derivation dataset. (This is done on events because the number of events gets very small in the Good prognostic group.)

   ```
   centile xb if val==0 & _d==1, centile(25 75)
   generate byte group = cond(xb <= r(c_1), 1, cond(xb >= r(c_2), 3, 2))
   ```

3. Retain the validation data, and center the PI on the validation dataset mean.

   ```
   keep if val==1
   summarize xb
   replace xb = xb - r(mean)
   ```

4. Get the baseline log cumulative hazard in the validation data.

   ```
   stcox xb
   predict H0, basechazard
   generate lnH0 = ln(H0)
   ```

5. Compute the smoothed baseline cumulative-hazard function.

   ```
   fracpoly: regress lnH0 _t
   ```

6. Compute mean survival probabilities in the validation dataset at `t` $= 0(1)10$ years.

   ```
   range t 0 10 11
   fraceval var t
   generate S0 = cond(t==0, 1, exp(-exp(_fp)))
   stcoxgrp xb S0 t, mean(s) km(km) by(group)
   ```

7. For the validation data, compare observed with predicted survival.

   ```
   twoway (scatter km1 km2 km3 t, mcolor(gs5 gs8 gs10))                   ///
    (rcap km_lb1 km_ub1 t, lcolor(gs5 ..))                                 ///
    (rcap km_lb2 km_ub2 t, lcolor(gs8 ..))                                 ///
    (rcap km_lb3 km_ub3 t, lcolor(gs10 ..))                                ///
    (line s1 s2 s3 t, sort lwid(medthick ..) lcolor(gs5 gs8 gs10)),        ///
    legend(off) title("Validation data with reestimated baseline")         ///
    xlabel(0(2)10) ylabel(0(.25)1, angle(h) format(%4.2f)) ytitle("")      ///
    xtitle("Years since diagnosis") ytitle("Survival probability")         ///
    name(g1, replace)
   ```

In steps 1 and 2, we estimate the PI (`xb`) on the derivation data and compute the predicted PI and the prognostic groups in all the data. Steps 1 and 2 could also be done with published regression coefficients and cutpoints on the PI without using the individual derivation data. All subsequent steps are performed using the validation data, mimicking application of the published information to a validation dataset.

Steps 3 through 7 are similar to the ones before, except that the baseline cumulative-hazard function is estimated on the validation data. The call to `stcoxgrp` produces estimated survival curves for the three original prognostic groups. The result is one graph (see figure 4) showing the calibration of the model's original PI on the validation data with the baseline reestimated.

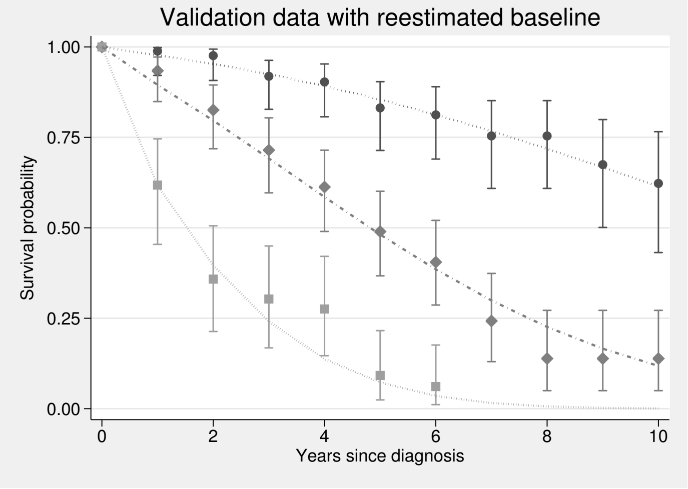

Figure 4. Partial calibration of a Cox model with the PI estimated from the derivation dataset and evaluated on the validation dataset with reestimation of the baseline cumulative-hazard function. Smooth dashed lines represent predicted survival probabilities, and vertical capped lines represent Kaplan–Meier estimates with 95% confidence intervals. Three prognosis groups are plotted: the "Good" group (darkest lines), the "Intermediate" group (medium-dark lines), and the "Poor" group (paler lines).

Because of reestimation of the baseline, the predicted curves match the Kaplan–Meier estimates even better than as seen in figure 2.

8 The stcoxgrp command

8.1 Syntax

The syntax of `stcoxgrp` is as follows:

`stcoxgrp` *xbetavar s0var* [*timevar*] [*if*] [*in*], `mean(`*mean_stub*`)` [`by(`*by_varlist*`)` `km(`*km_stub*`)`]

8.2 Description

`stcoxgrp` calculates population-averaged survival curves. A predicted survival curve is obtained for each subject in the dataset. The survival curves are averaged within subsamples defined by *by_varlist* or across the entire dataset if `by()` is not specified. Survival curves are estimated using the user-supplied baseline survival probabilities in *s0var* and the PI from a proportional hazards model supplied in *xbetavar*.

If the optional variable *timevar* is supplied, the results are calculated for the time-to-event values in *timevar*. It is assumed that the baseline survival probabilities in *s0var* correspond (1:1) to the times in *timevar*. This feature conveniently provides an out-of-sample prediction of population-averaged survival probabilities at user-specified time points. Because the averaging process can be computationally intensive, it is recommended that the *timevar* approach be used to reduce the number of survival times at which the survival curves are averaged.

Note that the population-averaged survival curve differs from the survival curve predicted at the mean of the covariates in the model.

8.3 Options

`mean(`*mean_stub*`)` stores Cox model-based estimates of population-averaged survival probabilities in new variables called *mean_stub*`1`, *mean_stub*`2`, ... The numbering `1`, `2`, ... corresponds to the enumeration of subsets defined by *by_varlist* or is `1` if the `by()` option has not been used. `mean()` is required.

`by(`*by_varlist*`)` provides estimates in subsets representing all possible combinations of values of variables in *by_varlist*.

`km(`*km_stub*`)` stores Kaplan–Meier estimates of survival probabilities in new variables called *km_stub*`1`, *km_stub*`2`, ... It stores lower and upper bounds of 95% confidence intervals in variables *km_stub*`_lb1`, *km_stub*`_lb2`, ... and *km_stub*`_ub1`, *km_stub*`_ub2`, ..., respectively. The numbering `1`, `2`, ... corresponds to the enumeration of subsets defined by *by_varlist* or is `1` if the `by()` option has not been used.

8.4 Examples

For this example, we `use` and `stset` the German breast cancer dataset:

```
. webuse brcancer, clear
(German breast cancer data)
. stset rectime, failure(censrec) scale(365.24)
  (output omitted)
```

Here we have a simple example on a single dataset, in which we compare predicted and Kaplan–Meier survival curves for a Cox model including only the variable x4 (tumor grade). We plot the population-averaged and Kaplan–Meier survival curves against _t.

```
. stcox i.x4
  (output omitted)
. predict xb, xb
. predict s0, basesurv
. stcoxgrp xb s0, mean(m) km(k) by(x4)
Processing 3 distinct values of xb ... ... done.
. line m1 m2 m3 k1 k2 k3 _t, sort lpattern(l l l - ..) connect(l l l J ..)
  (output omitted)
```

The pattern of survival curves suggests nonproportional hazards may be present for x4.

In a second example on the same dataset, we examine predictions within subgroups defined by a multivariable PI. Here the model is based on fractional polynomial transformations of the covariates.

```
. fracpoly: stcox x1 -2 -0.5 x4a x5e x6 0.5 hormon
  (output omitted)
. predict xb, xb
. summarize xb
  (output omitted)
. replace xb = xb - r(mean)
(686 real changes made)
. stcox xb
  (output omitted)
. predict s0, basesurv
. xtile group = xb, nquantiles(2)
. stcoxgrp xb s0, mean(s) by(group) km(km)
Processing 674 distinct values of xb ... 100 200 300 400 500 600 ... done.
. line s1 km1 km_lb1 km_ub1 s2 km2 km_lb2 km_ub2 _t, sort connect(J ..)
> lpattern(l - - - l - - -) legend(off)
  (output omitted)
```

We extend the same example and create a smooth baseline survival curve using stpm2 and plot it with a user-defined time variable, t, in yearly intervals over the range (0, 7) years.

```
. range t 0 7 8
(678 missing values generated)
. stpm2 xb, scale(hazard) df(3)
  (output omitted)
. predict s0a, zeros survival timevar(t)
. stcoxgrp xb s0a t, mean(sa) by(group) km(kma)
Processing 674 distinct values of xb ... 100 200 300 400 500 600 ... done.
```

```
. twoway (scatter kma1 kma2 t, mcolor(navy red)) (rcap kma_lb1 kma_ub1 t,
> lcolor(navy ..)) (rcap kma_lb2 kma_ub2 t, lcolor(red ..))
> (line sa1 sa2 t, sort lcolor(navy red)),
> legend(off) ylabel(0(.25)1, angle(h) format(%4.2f))
  (output omitted)
```

9 Limitations

I believe that the comparison of observed (Kaplan–Meier) and predicted population-averaged survival curves in the validation dataset is intuitive, natural, and simple to understand and that it represents an advance on current practice. Against that, the proposed calibration approach of Royston and Altman (2013), implemented here for Stata, has two main limitations:

1. It relies on defining risk groups, and the results of the exercise may depend on precisely which groups are chosen. Many equally valid selections are possible, and a sensitivity analysis is advisable.
2. The method is purely graphical (see figure 2). For example, no statistical inference is available to determine whether calibration is "significantly" worse in the validation dataset or whether it changes over time.

10 Comments

The methods outlined above can be used to check the calibration (fit) of categorical covariates in a Cox model on a given dataset. This may be helpful when a covariate exhibits statistically significant nonproportional hazards that may or may not be of clinical importance. I wish to analyze the lack of fit in terms of predicted event or survival probabilities at the levels of the covariate. As with the PI, continuous covariates must first be categorized into a small number of subgroups.

Graphical and analytical methods together with corresponding Stata tools to assess time-related calibration of a Cox model, not relying on grouping of continuous variables, are described in Royston (2014).

11 Acknowledgment

I am most grateful to an anonymous reviewer whose detailed critique helped me considerably to improve the article.

12 References

Christensen, E., J. Neuberger, J. Crowe, D. G. Altman, H. Popper, B. Portmann, D. Doniach, L. Ranek, N. Tygstrup, and R. Williams. 1985. Beneficial effect of azathioprine and prediction of prognosis in primary biliary cirrhosis: Final results of an international trial. *Gastroenterology* 89: 1084–1091.

Fleming, T. R., and D. P. Harrington. 1991. *Counting Processes and Survival Analysis.* New York: Wiley.

Harrell, F. E., Jr., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. 1982. Evaluating the yield of medical tests. *Journal of the American Medical Association* 247: 2543–2546.

Lambert, P. C., and P. Royston. 2009. Further development of flexible parametric models for survival analysis. *Stata Journal* 9: 265–290.

Royston, P. 2001. Flexible parametric alternatives to the Cox model, and more. *Stata Journal* 1: 1–28.

———. 2014. Tools for checking calibration of a Cox model in external validation: Approach based on individual event probabilities. *Stata Journal* 14: 738–755.

Royston, P., and D. G. Altman. 2013. External validation of a Cox prognostic model: Principles and methods. *BMC Medical Research Methodology* 13: 33.

Royston, P., and P. C. Lambert. 2011. *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model.* College Station, TX: Stata Press.

Royston, P., and W. Sauerbrei. 2004. A new measure of prognostic separation in survival data. *Statistics in Medicine* 23: 723–748.

StataCorp. 2013. *Stata 13 Survival Analysis and Epidemiological Tables Reference Manual.* College Station, TX: Stata Press.

About the author

Patrick Royston is a medical statistician with more than 30 years of experience and a strong interest in biostatistical methods and in statistical computing and algorithms. He works largely in methodological issues in the design and analysis of clinical trials and observational studies. He is currently focusing on alternative outcome measures in trials with a time-to-event outcome; on problems of model building and validation with survival data, including prognostic factor studies and treatment-covariate interactions; on parametric modeling of survival data; and on novel clinical trial designs.