The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–19

Comparing the predictive power of survival models using Harrell's c or Somers' D

Roger B. Newson
National Heart and Lung Institute, Imperial College London
London, United Kingdom
r.newson@imperial.ac.uk

Abstract. Medical researchers frequently make statements that one model predicts survival better than another, and are frequently challenged to provide rigorous statistical justification for these statements. Stata provides the command `estat concordance` to calculate the rank parameters Harrell's c and Somers' D as a measure of the ordinal predictive power of a model. However, no confidence limits or P–values are provided to compare the predictive power of distinct models. The `somersd` package, downloadable from SSC, can provide such confidence intervals, but these should not be taken seriously, if they are calculated in the dataset in which the model was fitted. Methods are demonstrated for fitting alternative models to a training set of data, and then measuring and comparing their predictive power by using out–of–sample prediction, and `somersd`, in a test set, to produce statistically sensible confidence intervals and P–values for the differences between the predictive powers of different models.

Keywords: st0001, `somersd`, `stcox`, `estat concordance`, `streg`, `predict`, survival, model validation, prediction, concordance, rank methods, Harrell's c, Somers' D

1 Introduction

Harrell's c and the equivalent parameter Somers' D were proposed as measures of the general predictive power of a general regression model by Harrell et al. (1982) and Harrell et al. (1996), who focussed attention on the case of a survival model with a possibly right–censored outcome, interpreted as a lifetime. In the case of a Cox proportional–hazards regression model, both parameters are output by the Stata post–estimation command `estat concordance` (see [ST] **stcox postestimation**). However, as Harrell's c and Somers' D are rank parameters, they are equally valid as a measure of the predictive power of any model in which the scalar outcome Y is at least ordinal (with or without censorship), and in which the conditional distribution of the outcome, given the predictor variables, is governed by a scalar function of the predictor variables and the parameters, such as the hazard ratio in a Cox regression, or the linear predictor in a generalized linear model. If the assumptions of the model are true, then such a scalar predictive score plays the role of a balancing score, as defined by Rosenbaum and Rubin (1983).

Harrell's c and Somers' D are members of the "Kendall family" of rank parameters, whose family history can be summarized as "Kendall's τ_a begat Somers' D begat Theil–Sen percentile slopes". This family is implemented in Stata using the `somersd`

package, which can be downloaded from SSC. An overview of the parameter family is given in Newson (2002), and the methods and formulas are given in detail in Newson (2006a), Newson (2006b), and Newson (2006c). Parameters in this family are defined by assuming the existence of a population of bivariate data pairs of the form (X_i, Y_i), and a sampling scheme for sampling pairs of pairs $[(X_i, Y_i), (X_j, Y_j)]$ from that population. A pair of pairs is said to be concordant if the larger of the X–values is paired with the larger of the Y–values, and is said to be discordant if the larger of the X–values is paired with the smaller of the Y–values. Kendall's τ_a is the difference between the probability of concordance and the probability of discordance, and Somers'$D(X|Y)$ is the difference between the corresponding *conditional* probabilities, assuming that the two Y–values can be ordered. Harrell's $c(X|Y)$ is defined as $[D(X|Y) + 1]/2$, and is equal to the conditional probability of concordance plus half the conditional probability that the data pairs are neither concordant nor discordant, assuming that the two Y–values can be ordered. In the case where Y is an outcome to be predicted by a multivariate model with a scalar predictive score, there is an underlying population of multivariate data points $(Y_i, V_{i1}, \ldots, V_{ik})$, where the V_{ih} are predictive covariates, and the role of the X_i is played by the scalar predictive score $\eta(V_{i1}, \ldots, V_{ik})$. In this case, the Somers' D and Harrell's c parameters can be denoted as $D[\eta(V_1, \ldots, V_k)|Y]$ and $c[\eta(V_1, \ldots, V_k)|Y]$, respectively. If the model is a survival model, then the Y–values are lifetimes, and there is the possibility that one or both of a pair of Y–values may be censored, which sometimes implies that they cannot be ordered.

We often want to compare the predictive power of alternative predictors of the same outcome Y. In Newson (2002) and Newson (2006b), it is argued that, if there is an underlying population of trivariate data points (W_i, X_i, Y_i), and any positive association between the Y_i and the X_i is caused by a positive association of both of these variables with the W_i, then we must have the inequality $D(X|Y) - D(W|Y) \leq 0$, or (equivalently) $c(X|Y) - c(W|Y) = [D(X|Y) - D(W|Y)]/2 \leq 0$. (This inequality still holds if the Y–variable may be censored, but not if the W– and/or X–variables may be censored.) This implies that, if we have multiple alternative positive predictors of the same outcome, such as alternative predictive scores from alternative multivariate models, then it may be useful to calculate confidence intervals for the differences between the Somers' D or Harrell's c parameters of these predictors, with respect to the outcome, and then make statements regarding which predictors may or may not be secondary to which other predictors. In Stata, this can be done by using `lincom` after the `somersd` command, as demonstrated in Section 4.1 of Newson (2002).

Medical researchers frequently make statements that one model predicts survival better than another, and are frequently challenged, by statistical referees acting for medical journals, to provide rigorous statistical justification for these statements. The Stata post–estimation command `estat concordance` provides estimates of Harrell's c and Somers' D, but provides no confidence limits for these, and no confidence limits or P–values for the differences between the values of these rank parameters from different models. There are good reasons why this is the case, because confidence interval formulas do not cover the user for finding a model in the same data in which its parameters are then estimated. The `somersd` and `lincom` command provides confidence limits and

P–values for differences between the Somers' *D* or Harrell's *c* parameters between different predictors. However, not all medical researchers know how to do this when the predictors are scalar predictive scores from models, and fewer still know how to do so in such a way that the confidence limits can be taken seriously.. This article aims to explain how medical researchers can do this, and to pre–empt possible queries that may arise in the process.

The remainder of this article is divided into 4 further Sections. Section 2 addresses the queries that commonly arise when users try to duplicate the results of `estat concordance` using `somersd`. Section 3 describes the method of splitting the data into a training set (to which models are fitted) and a test set (in which their predictive power is measured). Section 4 describes the extension to non–Cox survival models, such as those described in [ST] **streg**. Finally, Section 5 discusses briefly how the methods might be extended even further.

2 The Cox model: `somersd` versus `estat concordance`

We will demonstrate the principles using the Cox proportional hazards model, implemented in Stata using the `stcox` command (see [ST] **stcox**), and the Stanford drug trial data, used for the examples in [ST] **stcox postestimation**.

Before we raise the issue of confidence limits, we need to show how `somersd` can produce the same estimates as `estat concordance`. This is done using `predict` after the survival command to define the predictive score, and then measuring the association of the predictive score with the lifetime, using `somersd`. There are 3 issues waiting to cause confusion for users who attempt to use `somersd` to duplicate the estimates of `estat concordance`:

1. The `predict` command, used after `stcox`, produces a negative prediction score by default, in contrast to the positive prediction score produced by using `predict` after most estimation commands.
2. The default coding of a censorship status variable for `stcox` is different from the coding of a censorship status variable for `somersd`.
3. The treatment of tied failure times by `estat concordance` is different from that used by `somersd`.

There are solutions to all of these problems, and we will demonstrate these, enabling users to use `somersd` and `estat concordance` as checks on each other.

We will start our demonstration by inputting the Stanford drug trial data, fitting a Cox model, and calling `estat concordance`:

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stset
```

```
-> stset studytime, failure(died)
     failure event:  died != 0 & died < .
obs. time interval:  (0, studytime]
 exit on or before:  failure

------------------------------------------------------------------------------
       48  total obs.
        0  exclusions
------------------------------------------------------------------------------

       48  obs. remaining, representing
       31  failures in single record/single failure data
      744  total analysis time at risk, at risk from t =         0
                            earliest observed entry t =         0
                                 last observed exit t =        39

. stcox drug age
         failure _d:  died
   analysis time _t:  studytime

Iteration 0:   log likelihood = -99.911448
Iteration 1:   log likelihood = -83.551879
Iteration 2:   log likelihood = -83.324009
Iteration 3:   log likelihood = -83.323546
Refining estimates:
Iteration 0:   log likelihood = -83.323546

Cox regression -- Breslow method for ties

No. of subjects =           48                     Number of obs   =        48
No. of failures =           31
Time at risk    =          744
                                                   LR chi2(2)      =     33.18
Log likelihood  =   -83.323546                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1048772   .0477017    -4.96   0.000     .0430057    .2557622
         age |   1.120325   .0417711     3.05   0.002     1.041375     1.20526
------------------------------------------------------------------------------

. estat concordance

  Harrell´s C concordance statistic

         failure _d:  died
   analysis time _t:  studytime

  Number of subjects (N)              =       48
  Number of comparison pairs (P)      =      849
  Number of orderings as expected (E) =      679
  Number of tied predictions (T)      =       15

          Harrell´s C = (E + T/2) / P =    .8086
                        Somers´ D     =    .6172
```

The `stset` command shows us that the input dataset has already been set up as a survival time dataset, with 1 observation per drug trial subject, and data on survival time and termination modes, among other things (see [ST] **stset**). The Cox model contains two predictive covariates, `age` (subject age in years) and `drug` (indicating treatment group, with the values 0 for placebo and 1 for the drug being tested). We then show that, according to `estat concordance`, Harrell's c is .8086, and Somers' D is 0.6172. The Somers' D implies that, when one of two subjects is observed to survive another, it is 61.72% more likely that the survivor has the lower of the two hazard ratios,

predicted by the model, than that the survivor has the higher of the two predicted hazard ratios. The Harrell's c is the probability that the survivor has the lower hazard ratio plus half the (possibly negligible) probability that the two subjects have equal hazard ratios, and this sum is 80.86% on a percentage scale.

We will now show how to duplicate these estimates using `predict` and `somersd`. We start by defining a negative predictor of lifetime by using `predict` to calculate a hazard ratio, and then deriving an inverse hazard ratio, which we expect to be a positive predictor of lifetime:

```
. predict hr
(option hr assumed; relative hazard)
. gene invhr=1/hr
```

This addresses the first of the 3 sources of confusion mentioned above. We now address the second. We need to define a censorship indicator for input to the `somersd` command. The `somersd` command has a `cenind()` option, requiring a list of censorship indicators. These censorship indicators are allocated to the corresponding variables of the variable list input to `somersd`, and must be either variable names or zeros (implying a censorship indicator variable whose values are all zero), and which are matched one–to–one with the variables in the input variable list. Censorship indicator variables for `somersd` are positive in observations where the corresponding input variable value is right–censored (or known to be equal to or greater than its stated value), negative in observations where the corresponding input variable value is left–censored (or known to be equal to or less than its stated value), and zero in observations where the corresponding input variable value is uncensored (or known to be equal to its stated value). If the list of censorship indicators is shorter than the input variable list, then it is extended on the right with zeros, implying that the variables without censorship indicators are uncensored. This coding is not the same as that for the censorship indicator variable `_d`, created by the `svset` command, which is 1 in observations where the corresponding lifetime is uncensored, and 0 in observations where the corresponding lifetime is right–censored. To convert a `stset` censorship indicator variable to a `somersd` censorship indicator variable, we use the command:

```
. gene censind=1-_d if _st==1
```

This creates a new variable `censind`, which is missing in observations excluded from the survival sample indicated by the variable `_st` created by `svset`, 1 in observations with right–censored lifetimes (where `_d` is 0), and 0 in observations with uncensored lifetimes (where `_d` is 1).

We can now use `somersd` to calculate Harrell's c and Somers' D, using the option `transf(c)` for Harrell's c, and the option `transf(z)` (indicating the Normalizing and variance–stabilizing Fisher's z or hyperbolic arctangent transformation) for Somers' D:

```
. somersd _t invhr if _st==1, cenind(censind) tdist transf(c)
Somers´ D with variable: _t
Transformation: Harrell´s c
Valid observations: 48
```

```
Degrees of freedom: 47
Symmetric 95% CI for Harrell´s c
```

_t	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
invhr	.8106332	.0423076	19.16	0.000	.7255213	.8957451

```
. somersd _t invhr if _st==1, cenind(censind) tdist transf(z)
Somers´ D with variable: _t
Transformation: Fisher´s z
Valid observations: 48
Degrees of freedom: 47
Symmetric 95% CI for transformed Somers´ D
```

_t	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
invhr	.7270649	.1378034	5.28	0.000	.4498402	1.00429

```
Asymmetric 95% CI for untransformed Somers´ D
               Somers_D     Minimum     Maximum
       invhr   .62126643   .42176765   .76338983
```

In both cases, we use the survival time variable `_t` and the survival sample indicator `_st`, created by `stset`, together with the inverse hazard rate `invhr` created using `predict`, to estimate rank parameters of inverse hazard ratio with respect to survival time (censored by censorship status). In the case of Harrell's c, the estimated parameter is on a scale from 0 to 1, and is expected to be at least 0.5 for a positive predictor of lifetime, such as an inverse hazard ratio. In the case of Somers' D, the untransformed parameter is on a scale from -1 to 1, and is expected to be at least 0 for a positive predictor of lifetime.

However, we now encounter the third source of confusion. If we compare the estimates here to those produced earlier by `estat concordance`, then we find that the estimates for Harrell's c and Somers'D are similar, but not exactly the same. The estimates are .8106 and .6213, respectively, when computed by `somersd`, and .8086 and .6172, respectively, when computed by `estat concordance`. The reason for this is that `somersd` and `estat concordance` have different policies for comparing two lifetimes that terminate simultaneously, of which one is right–censored and the other is uncensored. The `estat concordance` policy assumes that the owner of the right–censored lifetime survived the owner of the uncensored lifetime, whereas the `somersd` policy assumes that neither of the two owners can be said to have survived the other. In the case of a drug trial, one subject might be known to have died in a certain month, whereas another might be known to have left the country in the same month, and therefore become lost to follow–up. The `estat concordance` policy assumes that the second subject must have survived the first, which might be probable, given that this second subject seems to have been in a fit state to travel out of the country. The `somersd` policy, more cautiously, allows the possibility that the second subject may have left the country early in the month, and died unexpectedly of a venous thromboembolism on

the outbound plane, whereas the first subject may have died, under observation by the trial organizers, later in the same month.

Whatever the merits of the two policies, we might still like to show that `somersd` and `estat concordance` can be made to duplicate each other's estimates. This can easily be done, if lifetimes are expressed as whole numbers of time units, as they are in the Stanford drug trial data, where lifetimes are expressed in months. In this case, we can add half a unit to right–censored lifetimes only, and this will cause right–censored lifetimes to become greater than uncensored lifetimes terminating in the same time unit, without affecting any other orderings of lifetimes. In our example, we do this by generating a new lifetime variable `studytime2`, equal to the modified survival time, and using `stset` to reset the various survival–time variables and characteristics so that the modified survival time is now used. (This is done after using the `assert` command to check that the old study time variable is indeed integer–valued; see [D] **assert** and [D] **functions**.) We then proceed as in the previous example:

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. assert studytime==int(studytime)
. gene studytime2=studytime+0.5*(died==0)
. stset studytime2, failure(died)
     failure event:  died != 0 & died < .
obs. time interval:  (0, studytime2]
 exit on or before:  failure
------------------------------------------------------------------------------
       48  total obs.
        0  exclusions
------------------------------------------------------------------------------
       48  obs. remaining, representing
       31  failures in single record/single failure data
    752.5  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      39.5
. stcox drug age
         failure _d:  died
   analysis time _t:  studytime2
Iteration 0:   log likelihood = -99.911448
Iteration 1:   log likelihood = -83.551879
Iteration 2:   log likelihood = -83.324009
Iteration 3:   log likelihood = -83.323546
Refining estimates:
Iteration 0:   log likelihood = -83.323546
Cox regression -- Breslow method for ties
No. of subjects =           48                     Number of obs   =        48
No. of failures =           31
Time at risk    =        752.5
                                                   LR chi2(2)      =     33.18
Log likelihood  =   -83.323546                     Prob > chi2     =    0.0000
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1048772   .0477017    -4.96   0.000     .0430057    .2557622
```

```
------------------------------------------------------------------------------
         age |   1.120325   .0417711     3.05   0.002     1.041375     1.20526
------------------------------------------------------------------------------

. estat concordance

  Harrell´s C concordance statistic

        failure _d:  died
  analysis time _t:  studytime2

  Number of subjects (N)              =       48
  Number of comparison pairs (P)      =      849
  Number of orderings as expected (E) =      679
  Number of tied predictions (T)      =       15

         Harrell´s C = (E + T/2) / P =    .8086
                       Somers´ D =    .6172

. predict hr
(option hr assumed; relative hazard)

. gene invhr=1/hr

. gene censind=1-_d if _st==1

. somersd _t invhr if _st==1, cenind(censind) tdist transf(c)
Somers´ D with variable: _t
Transformation: Harrell´s c
Valid observations: 48
Degrees of freedom: 47

Symmetric 95% CI for Harrell´s c

------------------------------------------------------------------------------
             |                    Jackknife
          _t |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       invhr |   .8085984   .0425074    19.02   0.000     .7230845    .8941122
------------------------------------------------------------------------------

. somersd _t invhr if _st==1, cenind(censind) tdist transf(z)
Somers´ D with variable: _t
Transformation: Fisher´s z
Valid observations: 48
Degrees of freedom: 47

Symmetric 95% CI for transformed Somers´ D

------------------------------------------------------------------------------
             |                    Jackknife
          _t |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       invhr |   .7204641   .1373271     5.25   0.000     .4441976    .9967306
------------------------------------------------------------------------------

Asymmetric 95% CI for untransformed Somers´ D
               Somers_D     Minimum     Maximum
       invhr    .6171967   .41711782   .76021766
```

This time, the model fit produces the same output as before, and `estat concordance` produces the same estimates of .8086 and .6172 for Harrell's c and Somers' D, respectively, but the same estimates are now also produced by `somersd`, at least after rounding to 4 decimal places.

It should be stressed that Harrell's c and Somers' D, computed as above either by `somersd` or by `estat concordance`, are only valid measures of the predictive power of a survival model if there are no time–dependent covariates or lifetimes with delayed entries. However, if `somersd` (instead of `estat concordance`) is used, then sensible

estimates can still be produced with weighted data, as long as these weights are explicitly supplied to `somersd`.

3 Comparing predictive power with training and test sets

Another caution about the results of the previous section is that the confidence intervals generated by `somersd` should not really be taken seriously. This is because, in general, confidence intervals *do not* cover the user against the consequences of finding a model in a dataset, and then estimating its parameters in the same dataset. In the case of Harrell's c and Somers' D of inverse hazard ratios with respect to lifetime, we would expect this incorrect practice to lead to over–optimistic estimates of predictive power, because we are measuring the "predictive power" of a model optimized for the dataset in which the predictive power is measured.

We should really be finding models in a training set of data, and testing their predictive power, both absolutely and relatively to each other, in a test set of data, independent of the training set. If we only have one set of data, then we might divide its primary sampling units (PSUs) randomly, or semi-randomly, into two subsets, and use the first subset as the training set and the second subset as the test set. The next two subsections demonstrate this practice by splitting the Stanford drug trial data into a training set and a test set of similar size, using random subsets and semi–random stratified subsets, respectively. We will use the `somersd` policy, rather than the `estat concordance` policy, regarding tied censored and non–censored lifetimes.

3.1 Completely–random training and test sets

We will first demonstrate the relatively simple practice of splitting the sampling units, completely at random, into a training set and a test set. We will fit 3 models to the training set, namely "Model 1" (containing the variables `drug` and `age`), "Model 2" (containing `drug` only), and "Model 3" (containing `age` only). We will then use out–of–sample prediction, and `somersd`, to estimate the predictive power of these 3 models in the test set, and then use `lincom` to compare their predictive power, in the manner of Section 5.2 of Newson (2006b).

We start by inputting the data, and then split the data, completely at random, into a training set and a test set, using the `runiform()` function to create a uniformly–distributed pseudo–random variable, `sort` to sort the dataset by this variable, and the `mod()` function to allocate alternate observations to the training and test sets. (See [D] **sort** and [D] **functions**.) We then re–sort the data back to their old order, using the generated variable `oldord`.

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. set seed 987654321
. gene ranord=runiform()
. gene long oldord=_n
```

```
. sort ranord, stable
. gene testset=mod(_n,2)
. sort oldord
. tab testset, m
```

testset	Freq.	Percent	Cum.
0	24	50.00	50.00
1	24	50.00	100.00
Total	48	100.00	

We see that there are 24 patient lifetimes in the training set (where `testset==0`), and 24 in the test set (where `testset==1`). We then fit the 3 Cox models to the training set, and create inverse hazard rate variables `invhr1`, `invhr2`, and `invhr3`, for Models 1, 2 and 3, respectively:

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. set seed 987654321
. gene ranord=runiform()
. gene long oldord=_n
. sort ranord, stable
. gene testset=mod(_n,2)
. sort oldord
. tab testset, m
```

testset	Freq.	Percent	Cum.
0	24	50.00	50.00
1	24	50.00	100.00
Total	48	100.00	

```
. stcox drug age if testset==0
        failure _d:  died
  analysis time _t:  studytime
Iteration 0:   log likelihood = -36.900079
Iteration 1:   log likelihood = -30.207704
Iteration 2:   log likelihood = -30.075862
Iteration 3:   log likelihood = -30.075741
Refining estimates:
Iteration 0:   log likelihood = -30.075741
Cox regression -- Breslow method for ties
No. of subjects =           24                     Number of obs   =        24
No. of failures =           14
Time at risk    =          370
                                                   LR chi2(2)      =     13.65
Log likelihood  =   -30.075741                     Prob > chi2     =    0.0011
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
drug	.1302894	.085747	-3.10	0.002	.0358683	.473269
age	1.139011	.0678588	2.18	0.029	1.013482	1.280089

```
. predict hr1
(option hr assumed; relative hazard)
. gene invhr1=1/hr1
. stcox drug if testset==0
        failure _d:  died
  analysis time _t:  studytime
Iteration 0:   log likelihood = -36.900079
Iteration 1:   log likelihood = -32.692209
Iteration 2:   log likelihood = -32.647379
Iteration 3:   log likelihood = -32.647309
Refining estimates:
Iteration 0:   log likelihood = -32.647309
Cox regression -- Breslow method for ties
No. of subjects =           24                     Number of obs   =        24
No. of failures =           14
Time at risk    =          370
                                                   LR chi2(1)      =      8.51
Log likelihood  =   -32.647309                     Prob > chi2     =    0.0035
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1843768    .112761    -2.76   0.006     .0556069      .611341
------------------------------------------------------------------------------
. predict hr2
(option hr assumed; relative hazard)
. gene invhr2=1/hr2
. stcox age if testset==0
        failure _d:  died
  analysis time _t:  studytime
Iteration 0:   log likelihood = -36.900079
Iteration 1:   log likelihood = -35.587135
Iteration 2:   log likelihood =  -35.58462
Refining estimates:
Iteration 0:   log likelihood =  -35.58462
Cox regression -- Breslow method for ties
No. of subjects =           24                     Number of obs   =        24
No. of failures =           14
Time at risk    =          370
                                                   LR chi2(1)      =      2.63
Log likelihood  =    -35.58462                     Prob > chi2     =    0.1048
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.082178   .0526849     1.62   0.105     .9836912    1.190526
------------------------------------------------------------------------------
. predict hr3
(option hr assumed; relative hazard)
. gene invhr3=1/hr3
```

Note that the variables `invhr1`, `invhr2` and `invhr3` are defined for all observations, both in the training set and in the test set. We then define the censorship indicator as before, and estimate the Harrell's c indices in the test set, for all 3 models fitted to the

training set:

```
. gene censind=1-_d if _st==1
. somersd _t invhr1 invhr2 invhr3 if _st==1 & testset==1, cenind(censind) tdist
>  transf(c)
Somers´ D with variable: _t
Transformation: Harrell´s c
Valid observations: 24
Degrees of freedom: 23
Symmetric 95% CI for Harrell´s c
```

_t	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
invhr1	.8819444	.0490633	17.98	0.000	.7804493	.9834396
invhr2	.7916667	.0330999	23.92	0.000	.7231944	.860139
invhr3	.6365741	.0831046	7.66	0.000	.4646592	.808489

We see that Harrell's c of inverse hazard ratio with respect to lifetime is .8819 for Model 1 (using both drug treatment and age), .7917 for Model 2 (using drug treatment only), and .6366 for Model 3 (using age only), and all of these estimates have confidence limits, which are probably less unreliable than the ones we saw in the previous Section. However, the sample Harrell's c is likely to have a skewed distribution in the presence of such strong positive associations, for the same reasons as Kendall's τ_a (see Daniels and Kendall (1947)). Differences between Harrell's c indices are likely to have a less skewed sampling distribution, and are also what we probably really wanted to know. We estimate these with `lincom`, as follows:

```
. lincom invhr1-invhr2
 ( 1)  invhr1 - invhr2 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.0902778	.0350965	2.57	0.017	.0176751	.1628804

```
. lincom invhr1-invhr3
 ( 1)  invhr1 - invhr3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.2453704	.0736766	3.33	0.003	.0929586	.3977821

```
. lincom invhr2-invhr3
 ( 1)  invhr2 - invhr3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.1550926	.0823647	1.88	0.072	-.0152917	.3254769

We note that Model 1 seems to have a slightly higher predictive power than Model 2 or (especially) Model 3, while the difference between Model 2 and Model 3 is slightly

less convincing. We can also do the same comparison using Somers' D rather than Harrell's c, using the Normalizing and variance–stabilizing z–transform, recommended by Edwardes (1995), and implemented using the `somersd` option `transf(z)`. In that case, the differences between the predictive power of the different models will be expressed in z–units (not shown).

3.2 Stratified semi–random training and test sets

Completely–random training and test sets may have the disadvantage that, by chance, important predictor variables may have different sample distributions in the training and test sets, making both the training set and the test set less representative of the sample as a whole, and of the total population from which the training and test sets were sampled. We might feel safer if we chose the training and test sets semi–randomly, with the constraint that the two sets have similar distributions of key predictor variables in the various models. In our case, we might want to ensure that both the training set and the test set contain their "fair share" of drug–treated older subjects, drug–treated younger subjects, placebo–treated older subjects, and placebo–treated younger subjects. To do this, we might start by defining sampling strata which are combinations of treatment status and age group, and split each of these strata as evenly as possible between the training set and the test set. Again, this requires the dataset to be sorted, and we will sort it back to its old order. This is done as follows, using `xtile` to define age groups (see [D] **pctile**):

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. set seed 987654321
. gene ranord=runiform()
. gene long oldord=_n
. xtile agegp=age, nquantiles(2)
. tab drug agegp, m
```

Drug type (0=placebo)	2 quantiles of age 1	2	Total
0	11	9	20
1	16	12	28
Total	27	21	48

```
. sort drug agegp ranord, stable
. by drug agegp: gene testset=mod(_n,2)
. sort oldord
. table testset drug agegp, row col scol
```

	2 quantiles of age and Drug type (0=placebo)								
	1			2			Total		
testset	0	1	Total	0	1	Total	0	1	Total
0	5	8	13	4	6	10	9	14	23

1	6	8	14	5	6	11	11	14	25
Total	11	16	27	9	12	21	20	28	48

This time, the training set is slightly smaller than the test set, because of odd total numbers of subjects in sampling strata. We then carry out the model fitting in the training set, and the calculation of inverse hazard ratios in both sets, using the same command sequence as with the completely–random training and test sets, producing mostly similar results (not shown). Finally, we estimate the Harrell's c indices in the test set:

```
. gene censind=1-_d if _st==1
. somersd _t invhr1 invhr2 invhr3 if _st==1 & testset==1, cenind(censind) tdist
>  transf(c)
Somers´ D with variable: _t
Transformation: Harrell´s c
Valid observations: 25
Degrees of freedom: 24
Symmetric 95% CI for Harrell´s c
```

_t	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
invhr1	.7911392	.0674598	11.73	0.000	.6519091	.9303694
invhr2	.7257384	.049801	14.57	0.000	.6229542	.8285226
invhr3	.5780591	.0972101	5.95	0.000	.3774274	.7786908

The c–estimates for the three models are not dissimilar to the previous ones, with completely–random training and test sets. Their pairwise differences are as follows:

```
. lincom invhr1-invhr2
 ( 1)  invhr1 - invhr2 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.0654008	.0491405	1.33	0.196	-.0360202	.1668219

```
. lincom invhr1-invhr3
 ( 1)  invhr1 - invhr3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.2130802	.0763467	2.79	0.010	.0555084	.3706519

```
. lincom invhr2-invhr3
 ( 1)  invhr2 - invhr3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.1476793	.1080388	1.37	0.184	-.0753017	.3706603

Model 1 (with drug treatment and age) still seems to predict better than Model 3 (with age alone). This conclusion is similar if we compare the z–transformed Somers' D values (not shown).

4 Extensions to non–Cox survival models

Measuring predictive power using Harrell's c and Somers' D is not restricted to Cox models, but can be applied to any model with a positive or negative ordinal predictor. The `streg` command (see [ST] **streg**) fits a wide range of survival models, each of which has a wide choice of predictive output variables, which can be computed using `predict` (see [ST] **streg postestimation**). These output variables may predict survival times positively or negatively on an ordinal scale, and include median survival times, mean survival times, median log survival times, mean log survival times, hazards, hazard ratios, or linear predictors.

We will briefly demonstrate the principles involved by fitting Gompertz models to the survival dataset that we used in previous sections. The Gompertz model assumes an exponentially–increasing (or decreasing) hazard rate, and the linear predictor is the log of the zero–time baseline hazard rate, whereas the rate of increase (or decrease) in hazard rate, after time zero, is a nuisance parameter. Therefore, if the Gompertz model is true, then so is the Cox model. However, the argument of Fisher (1935) presumably implies that, *if* the Gompertz model is true, *then* we can be no less efficient, asymptotically, by fitting a Gompertz model instead of a Cox model. We will use the predicted median lifetime as the positive predictor, whose predictive power will be assessed using `somersd`.

We start by inputting the cancer trial dataset, and defining the stratified, semi–random training and test sets, exactly as in Section 3.2. We then fit, to the training set, Gompertz models 1, 2 and 3, containing, respectively, both drug treatment and age, drug treatment only, and age only. After fitting each of the 3 models, we use `predict` to compute the predicted median survival time for the whole sample, deriving the alternative positive lifetime predictors `medsurv1`, `medsurv2`, and `medsurv3` for Models 1, 2, and 3, respectively:

```
. streg drug age if testset==0, distribution(gompertz) nolog
        failure _d:  died
  analysis time _t:  studytime
Gompertz regression -- log relative-hazard form
No. of subjects =           23                     Number of obs   =        23
No. of failures =           15
Time at risk    =          338
                                                   LR chi2(2)      =     20.62
Log likelihood  =   -14.076214                     Prob > chi2     =    0.0000
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .0948331   .0594575    -3.76   0.000     .0277512    .3240694
         age |   1.172588   .0616365     3.03   0.002     1.057798    1.299836
------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
      /gamma |   .1553139   .0430892     3.60   0.000     .0708605    .2397672
------------------------------------------------------------------------------
. predict medsurv1
(option median time assumed; predicted median time)
. streg drug if testset==0, distribution(gompertz) nolog
        failure _d:  died
  analysis time _t:  studytime
Gompertz regression -- log relative-hazard form
No. of subjects =           23                     Number of obs   =        23
No. of failures =           15
Time at risk    =          338
                                                   LR chi2(1)      =     11.02
Log likelihood  =   -18.873214                     Prob > chi2     =    0.0009
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |    .153411   .0877048    -3.28   0.001     .0500295    .4704213
-------------+----------------------------------------------------------------
      /gamma |   .1063648   .0361612     2.94   0.003     .0354901    .1772394
------------------------------------------------------------------------------
. predict medsurv2
(option median time assumed; predicted median time)
. streg age if testset==0, distribution(gompertz) nolog
        failure _d:  died
  analysis time _t:  studytime
Gompertz regression -- log relative-hazard form
No. of subjects =           23                     Number of obs   =        23
No. of failures =           15
Time at risk    =          338
                                                   LR chi2(1)      =      5.56
Log likelihood  =   -21.606438                     Prob > chi2     =    0.0184
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.117255   .0516156     2.40   0.016     1.020536    1.223142
-------------+----------------------------------------------------------------
      /gamma |    .088458   .0341184     2.59   0.010     .0215871    .1553288
------------------------------------------------------------------------------
. predict medsurv3
(option median time assumed; predicted median time)
```

Unsurprisingly, the fitted parameters are not dissimilar to the corresponding parameters for the Cox regression. We then compute the censorship indicator `censind`, and then the Harrell's c indices, for the test set:

```
. gene censind=1-_d if _st==1
. somersd _t medsurv1 medsurv2 medsurv3 if _st==1 & testset==1, cenind(censind)
>  tdist transf(c)
Somers´ D with variable: _t
Transformation: Harrell´s c
Valid observations: 25
Degrees of freedom: 24
Symmetric 95% CI for Harrell´s c
------------------------------------------------------------------------------
```

_t	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
medsurv1	.7911392	.0674598	11.73	0.000	.6519091	.9303694
medsurv2	.7257384	.049801	14.57	0.000	.6229542	.8285226
medsurv3	.5780591	.0972101	5.95	0.000	.3774274	.7786908

We then compare the Harrell's c parameters for the alternative median survival functions, using `lincom`, just as before:

```
. lincom medsurv1-medsurv2
 ( 1)  medsurv1 - medsurv2 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.0654008	.0491405	1.33	0.196	-.0360202	.1668219

```
. lincom medsurv1-medsurv3
 ( 1)  medsurv1 - medsurv3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.2130802	.0763467	2.79	0.010	.0555084	.3706519

```
. lincom medsurv2-medsurv3
 ( 1)  medsurv2 - medsurv3 = 0
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	.1476793	.1080388	1.37	0.184	-.0753017	.3706603

Unsurprisingly, the conclusions for the Gompertz model are essentially the same as those for the Cox model.

5 Further extensions

The use of Harrell's c and Somers' D in test sets, to compare the power of models fitted to training sets, can be extended further, to non–survival regression models. In this case, life is even simpler, as we do not have to define a censorship indicator, such as `censind`, for input to `somersd`. The predictive score is still computed using out–of–sample prediction, and can be *either* the fitted regression value *or* the linear predictor (if one exists in the model).

The methods presented so far have the limitation that the Harrell's c and Somers' D parameters calculated estimate only the ordinal predictive power, in the population from which the training and test sets were sampled, of the precise model fitted to the training set. We might prefer to estimate the mean predictive power that we can expect, in the whole universe of possible training and test sets, using the same set of alternative models. Bootstrap–like methods for doing this, involving repeated splitting of the same

sample into training and test sets, are described in Harrell et al. (1982) and Harrell et al. (1996).

Another limitation of the methods presented here, mentioned at the end of Section 2, is that they should not (usually) be used with models with time–dependent covariates. This is because the predicted variable input to `somersd`, which the alternative predictive scores are competing to predict, is the length of a lifetime, rather than an event of survival or non–survival through a minimal time interval (such as a day). A predictor variable for such a lifetime must therefore stay constant, at least through that lifetime, and this rules out functions of continuously–varying time–dependent covariates. In Stata, survival–time datasets may have multiple observations for each subject with a lifetime, representing multiple sub–lifetimes. Discretely–varying time–dependent covariates, which remain constant through a sub–lifetime, can also be included in such datasets. `somersd` can therefore be used in the case where the model is a Cox regression, the time-dependent covariates vary only discretely, the multiple sub–lifetimes are the times spent by a subject in an age group, and each subject becomes at risk at the start of each age group to which s/he survives. If the subject identifier variable is named `subid`, and the age group for each sub–lifetime is represented by a discrete variable `agegp`, then the user may use `somersd`, with the options `cluster(subid) funtype(bcluster) wstrata(agegp)`, to calculate Somers' D or Harrell's c estimates restricted to comparisons between sub–lifetimes of different subjects in the same age group. (See Newson (2006b) for details of the options for `somersd`, and [ST] **stset** for details on survival–time datasets.) If the user has access to sufficient data storage space, then the age groups might be defined finely (as subject–years or even subject–days), and the discretely time–dependent covariates might therefore be very nearly continuously time–dependent. Any training sets or test sets in this case should, of course, be sets of subjects, rather than sets of lifetimes.

6 Acknowledgements

I would like to thank Samia Mora, MD, of Partners HealthCare, for sending me the query that prompted me to write this article, and the many other Stata users who have also contacted me with essentially similar queries, over the past few years, on how to use `somersd` to compare the predictive power of survival models.

7 References

Daniels, H. E., and M. G. Kendall. 1947. The significance of rank correlation where parental correlation exists. *Biometrika* 34: 197–208.

Edwardes, M. D. 1995. A confidence interval for $\Pr(X < Y) - \Pr(X > Y)$ estimated from simple cluster samples. *Biometrics* 51: 571–578.

Fisher, R. A. 1935. The logic of inductive inference. *Journal of the Royal Statistical Society* 98: 39–82.

Harrell, F. E., 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(18): 2543–2546.

Harrell, F. E., K. L. Lee, and D. B. Mark. 1996. Multivariate prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine* 15: 361–387.

Newson, R. 2002. Parameters behind "nonparametric" statistics: Kendall's tau, Somers' *D* and median differences. *Stata Journal* 2(1): 45–64.

———. 2006a. Efficient calculation of jackknife confidence intervals for rank statistics. *Journal of Statistical Software* 15(1): 1–10. Downloadable from *http://www.jstatsoft.org/* as of 5 May 2010.

———. 2006b. Confidence intervals for rank statistics: Somers' *D* and extensions. *Stata Journal* 6(3): 308–334.

———. 2006c. Confidence intervals for rank statistics: Percentile slopes, differences, and ratios. *Stata Journal* 6(4): 497–20.

Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70(1): 41–55.