Concordance

Terry Therneau, Elizabeth Atkinson

November 23, 2026

1 The concordance statistic

1.1 Overview

Let (x_i, y_i) be paired data values, say two measurements on a set of subjects, or observed data y and predictions x for those values from a statistical model. A pair of observations $(i,\ j)$ is considered concordant if the prediction and the data go in the same direction, i.e., $(x_i > x_j, y_i > y_j)$ or $(x_i < x_j, y_i < y_j)$. The concordance statistic C is defined as the fraction of concordant pairs, and is an estimate of $P(x_i > x_j | y_i > y_j)$, or equivalently $P(y_i > y_j | x_i > x_j)$. The first form might make more sense if, for instance, one took a list of elected officials, say, and looked back at local newspapers to see how many of the election contests had been predicted correctly (probablility the prediction is correct, given outcome). The second conforms to our though process if we started with the predictions, before election day, and then counted up the fraction correct after the contest. When looking at a statistical model's predictions either form can be argued; they give the same result.

One wrinkle is what to do with ties in either x or y. Such pairs can be ignored in the count (treated as incomparable), treated as discordant, or given a score of 1/2. Let c, d, t_x, t_y and t_{xy} be a count of the pairs that are concordant, discordant, tied on the predictor x (but not y), tied on y (but not x), and tied on both. Then

$$\tau_a = \frac{c-d}{c+d+t_x+t_y+t_{xy}} \tag{1}$$

$$\tau_b = \frac{c-d}{\sqrt{(c+d+t_x)(c+d+t_y)}} \tag{2}$$

$$\gamma = \frac{c-d}{c+d} \tag{3}$$

$$D = \frac{c-d}{c+d+t_x} \tag{4}$$

$$C = (D+1)/2 = \frac{c+t_x/2}{c+d+t_x} \tag{5}$$

- Kendall's tau-a (1) is the most conservative, ties shrink the value towards zero.
- The Goodman-Kruskal γ statistic (3) ignores ties in either y or x.

- Somers' D (4) treats ties in y as incomparable; pairs that are tied in x (but not y) score as $1/2$, as we can see from equation (5) which is the concordance statistic.
- Kendall's tau-b (2) can be viewed as a version of Somers' D that is symmetric in x and y.

The first 4 statistics range from -1 to 1, similar to the correlation coefficient r. The concordance (5) ranges from 0–1, which matches the scale for a probability.

Why is C defined using Somers' D rather than one of the other three?

- If y is a 0/1 variable, then C = AUROC, the area under the receiver operating curve, which is well established for binary outcomes. (Proving this simple theorem is harder than it looks, but the result is well known.)
- For survival data, this choice will agree with Harrell's C. More importantly, as we will see below, it has strong connections to standard tests for equality of survival curves.

The concordance has a natural interpretation as an experiment: present pairs of subjects one at a time to the physician, statistical model, or some other oracle, and count the number of correct predictions. Pairs that have the same outcome $y_i = y_j$ are not put forward for scoring, since they do not help discriminate a good oracle from a bad one. If the oracle cannot decide then a random choice is made. This leads to $c + t_x/2$ correct selections out of $c + d + t_x$ choices.

This hypothetical experiment gives a baseline insight into the concordance. A value of $1/2$ corresponds to using a random guess for each subject. Values of .5–.55 are not very impressive, since the ordering for some pairs of subjects will be obvious, and someone with almost no medical knowledge could do that well by marking these easy pairs and using a coin flip for the rest. Values of less than $1/2$ are possible — some stock market analysts come to mind.

1.2 Simple examples

The `concordance` function accepts simple data or models as input. For the latter, it assesses the concordance between y and the model prediction $\hat{y}$. Here is a set of simple examples.

Direct

```
> # x1 and y2 are both continuous variables
> concordance(y2 ~ x1, data= anscombe)
Call:
concordance.formula(object = y2 ~ x1, data = anscombe)

n= 11
Concordance= 0.7818 se= 0.1255
concordant discordant     tied.x     tied.y    tied.xy
        43         12          0          0          0
```

Logistic regression

```
> # Fisher's iris data
> fit1 <- glm(Species=="versicolor" ~ ., family=binomial, data=iris)
> concordance(fit1)  # equivalent to an AUC
Call:
concordance.lm(object = fit1)

n= 150
Concordance= 0.8258 se= 0.03279
concordant discordant     tied.x     tied.y    tied.xy
      4129        871          0       6174          1
```

Linear regression

```
> # Anscombe data (all variables are continuous)
> fit2 <- lm(y2 ~ x1 + x4, data= anscombe)
> concordance(fit2)  # C
Call:
concordance.lm(object = fit2)

n= 11
Concordance= 0.7818 se= 0.1255
concordant discordant     tied.x     tied.y    tied.xy
        43         12          0          0          0
> sqrt(summary(fit2)$r.squared) # R
[1] 0.891425
```

Parametric survival

```
> # parametric survival
> fit3 <- survreg(Surv(time, status) ~ karno + age + trt, data=veteran)
> concordance(fit3)
Call:
concordance.survreg(object = fit3)

n= 137
Concordance= 0.7122 se= 0.02232
concordant discordant     tied.x     tied.y    tied.xy
      6263       2527         14         39          0
```

Cox regression

```
> # 3 Cox models
> fit4 <- coxph(Surv(time, status) ~ karno + age + trt, data=veteran)
> fit5 <- update(fit4, . ~ . + celltype)
> fit6 <- update(fit5, . ~ . + prior)
> ctest <- concordance(fit4, fit5, fit6)
> ctest
Call:
concordance.coxph(object = fit4, fit5, fit6)

n= 137
     concordance     se
fit4      0.7119 0.0224
fit5      0.7384 0.0210
fit6      0.7359 0.0212

     concordant discordant tied.x tied.y tied.xy
fit4       6261       2529     14     39       0
fit5       6499       2301      4     39       0
fit6       6478       2324      2     39       0
```

As shown in the last example, the concordance for multiple fits can be obtained from a single call. The variance-covariance matrix for all three concordance values is available using `vcov(ctest)`; this is used in Section 3.1 to formally test the equality of two concordance values. The above also shows that addition of another variable to a fitted model can decrease the concordance. The larger model will have higher correlation between the linear predictor $X\beta$ and the response y, by definition, but this does not guarantee a greater association between rank($X\beta$) and rank(y).

1.3 Efficient compuatation

For continuous data without ties, the concordance involves a comparison between all $n(n-1)/2$ pairs. This $O(n^2)$ computation will become painfully slow for large datasets. To improve this, first order the data by increasing y values, leading to

$$c - d = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \text{sign}(y_i - y_j)\text{sign}(x_i - x_j) \tag{6}$$

$$= \sum_{i=n}^{1} \left[\sum_{y_j > y_i} \text{sign}(x_i - x_j) \right] \tag{7}$$

The first equation is the simple definition of concordance as a sum over all n^2 possible pairs, where sign is the R `sign` function. Equation (7) makes the obvious simplification of counting each pair only once, by taking advantage of the fact that y is sorted. The key coding insight is

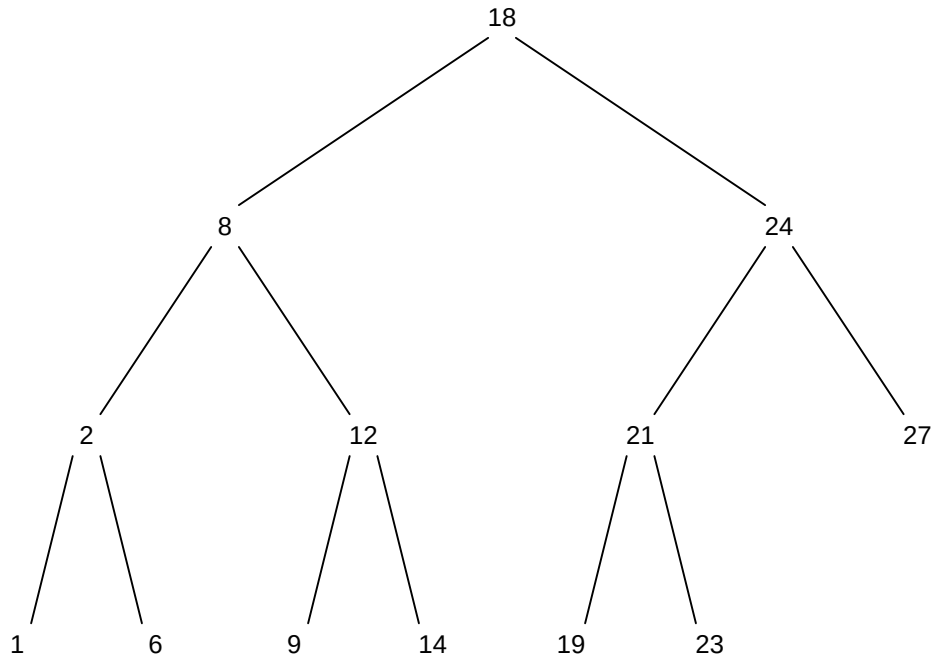


Figure 1: A balanced binary tree with 13 elements.

to store the x values as part of a balanced binary tree; an example of such a tree is shown in Figure 1. The basic algorithm is to:

1. Create a balanced binary tree for all n x_i values. This can be done in $O(n \log_2(n))$ steps. The final tree will have a node for each unique x value. Each node contains the value, along with counts for the number of observations at that value, for left hand children, and for right hand children. Initialize all the counts to 0.
2. Go through the data from largest y to smallest y.
   (a) For this new (x_i, y_i) pair, count the number of x_j values in the tree that are smaller or larger than this element, and increment the overall counts of concordant, discordant, and tied. This can be done in $\log_2(n)$ steps (proceed down from the top).
   (b) Add this observation to the tree. Each addition will update the count for its node, then "walk up" the tree updating child counts of the parent, grandparent, etc.

If there are tied y values, do all the counts for a set of ties first, and then add their x values to the tree.

2 Concordance and survival data

As stated above, for continuous data without ties, the concordance involves a comparison between all $n(n-1)/2$ pairs. If there are tied y values, however, those pairs are ignored. For example, in the iris data above $C = 4129/(4129 + 871)$, the 6175 pairs with tied response values play no role. For survival data, this set of incomparables is extended to include those pairs for which the time ordering is ambiguous. For instance, assume that y_i is censored at time 10 and y_j is an event (or censor) at time 20. Subject i may or may not survive longer than subject j, and so it is not possible to tell if a rule has ranked them correctly or not. Note that if y_i is censored at time 10 and y_j is an event at time 10 then $y_i > y_j$. This same convention is followed for all the survival models, and agrees with common clinical data, i.e., a patient censored on day 100 was observed to be still alive on day 100, their death time must be strictly greater than 100.

A second, smaller issue for survival data is to recognize that we desire to assess the concordance between an observed survival time y_i and a predicted survival time $\hat{y}_i$. This can be done without creating an explicit survival curve from the model: for a `survreg` fit, for instance, $\eta_1 > \eta_2$ implies $S(t; x_1) > S(t; x_2)$ for all t, where η_1 and η_2 are the respective linear predictors $x_i\beta$ and $x_2\beta$. Not creating the survival curves saves considerable computation time. For a Cox model $\eta_1 > \eta_2$ implies $S(t; x_1) < S(t; x_2)$; the order is reversed. This is because the Cox model is a hazard model, and higher hazard implies a shorter survival. When a `coxph` object is used as the argument to `concordance` the reversal is handled automatically. However, when a concordance is done "by hand" the user needs to be aware of this as seen in the example below. In this case the concordance routine does not know that the prediction came from a Cox model, resulting in a swap of the count for concordant and discordant pairs and $C < .5$. The user is responsible for adding the `reverse = TRUE` argument in this case.

```
> # Concordance for a coxph object
> concordance(fit4)
```

```
Call:
concordance.coxph(object = fit4)

n= 137
Concordance= 0.7119 se= 0.02235
concordant discordant     tied.x     tied.y    tied.xy
      6261       2529         14         39          0
> # Concordance using predictions from a Cox model
> concordance(Surv(time, status) ~ predict(fit4), data = veteran, reverse = TRUE)
Call:
concordance.formula(object = Surv(time, status) ~ predict(fit4),
    data = veteran, reverse = TRUE)

n= 137
Concordance= 0.7119 se= 0.02235
concordant discordant     tied.x     tied.y    tied.xy
      6261       2529         14         39          0
```

Stratified models

Stratified models present a further variation: if observations i and j are in different strata, the survival curves for those strata might cross; $S(t; x_i)$ and $S(t; x_j)$ no longer have a simple ordering. A solution is to use a stratified concordance, which compares all pairs *within* each stratum, and then adds up the result. In the example below there is a separate count for each stratum, the final concordance is based on the column sums. The same issue, and solution, applies to stratified `survreg` models. (The fact that strata names are not retained as labels for the counts matrix is a deficiency in the routine.)

```
> fit4b <- coxph(formula = Surv(time, status) ~ karno + age + trt +
                      strata(celltype), data = veteran)
> concordance(fit4b)
Call:
concordance.coxph(object = fit4b)

n= 137
Concordance= 0.6986 se= 0.02679
          concordant discordant tied.x tied.y tied.xy
squamous         357        161      0      1       0
smallcell        728        361      3      9       0
adeno            275         65      1      1       0
large            240        102      0      0       0
> table(veteran$celltype)
 squamous smallcell     adeno     large
       35        48        27        27
```

2.1 Time-weighted concordance

Look again at equation (7), rewriting it for survival with t_i as the response in order to more closely match standard notation for survival. Watson and Therneau [10] show that this can be further rewritten as

$$\begin{aligned} c - d &= \sum_{i=1}^{n} \delta_i \left[\sum_{t_j > t_i} \text{sign}(x_i - x_j) \right] \\ &= \sum_{i=1}^{n} \delta_i \left[\sum_{t_j \geq t_i} \text{sign}(x_i - x_j) \right] \qquad (8) \\ &= 2 \sum_i \delta_i n(t_i) \left[r_i(t_i) - \overline{r} \right] \qquad (9) \end{aligned}$$

Here δ is 0 = censored, 1 = uncensored; if t_i is censored then all other observations with $t_j \geq t_i$ are not comparable. Equation (8) rewrites the inner term is a sum over all subjects in the risk set at time t_i, a familiar concept in survival models. In equation (9) the inner terms has been rewritten with $n(t)$ as the number of subjects still at risk at time t, and $r_i(t)$ the rank of x_i among all those still at risk at time t, where ranks are defined such that $0 \leq r \leq 1$, and $\overline{r}$ the mean of those ranks. (Proofs for (8) and (9) have been omitted.) It turns out that equation (9) is exactly the score statistic for a Cox model with a single time-dependent covariate $n(t)r(t)$. One immediate consequence of this connection is a straightforward definition of concordance for a risk score containing time dependent covariates. Since the Cox model score statistic is well defined for time dependent covariates, the concordance is also well defined for a time-dependent risk score: at each event time the current risk score of the subject who failed is compared to the current (time dependent) scores of all those still at risk. A deeper consequence of the equivalence between the concordance and the Cox model is a link to alternate weightings of the risk scores. If the original Cox model has a single 0/1 treatment covariate then equation (9) exactly matches the numerator of the Gehan-Wilcoxon statistic; replacing $n(t)$ with weights of 1 will yield the log-rank statistic.

There is a deep literature with respect to the "best" weight for survival tests, and we can apply the same historical arguments to the concordance as well. We will point out four of interest:

- Peto and Peto [7] point out that $n(t) \approx n(0)S(t-)G(t-)$, where S is the survival distribution and G the censoring distribution. They argue that $S(t-)$ would be a better weight since G may have features that are irrelevant to the question being tested. For a particular dataset, Prentice [8] later showed that these concerns were indeed justified, and most software now uses the Peto-Wilcoxon variant.

- Tarone and Ware point out that weights of $n(t)$ and 1 give the Gehan-Wilcoxon and log-rank tests, respectively, and suggest $\sqrt{n(t)}$ as an intermediate value.

- Schemper et al [9] argue for a weight of $S(t)/G(t)$ in the Cox model. When proportional hazards does not hold the coefficient from the Cox model is an "average" hazard ratio, and they show that using S/G leads to a value that remains interpretable in terms of an underlying population model. The same argument would also apply to the concordance, since our goal is an "assumption free" assessment of association.

- Uno et al [11] recommend the use of n/G^2 as a weight based on a consistency argument. If we assume that the concordance value that would be obtained after full follow-up of all subjects (no censoring) is the "right" one, and proportional hazards does not hold, then the standard concordance will not consistently estimate this target quantity when there is censoring.

In practice, weights need to be based on left continuous versions of the survival curves $S(t-)$ and $G(t-)$, and extra care needs to be exercised in computation of G. Consider the `aml` dataset as an example; the first few lines of the relevant survival curve are shown below.

```
> afit <- survfit(Surv(time, status) ~1, aml, se = FALSE)
> summary(afit, times = afit$time[1:6], censor = TRUE)
Call: survfit(formula = Surv(time, status) ~ 1, data = aml, se = FALSE)

 time n.risk n.event survival
    5     23       2    0.913
    8     21       2    0.826
    9     19       1    0.783
   12     18       1    0.739
   13     17       1    0.696
   16     15       0    0.696
```

The first censor is at time 13, when there is both a death and a censor. For the event at time 13, however, no one has yet been censored, and no "censoring correction" should be done at this point. Weights at any time t, like the number at risk at time t, must be the values in effect just *before* the event. For a weight based on S; at time 13 the proper value would be .739, dropping to .696 just after time 13. The value of $G(t-)$ at 13 will be 1. Just after time 13 will be 15/16, not the value of 16/17 that will be obtained from `survfit(Surv(time, 1-status) 1)`. Censors happen after deaths, thus there are only 16 at risk for the censoring event.

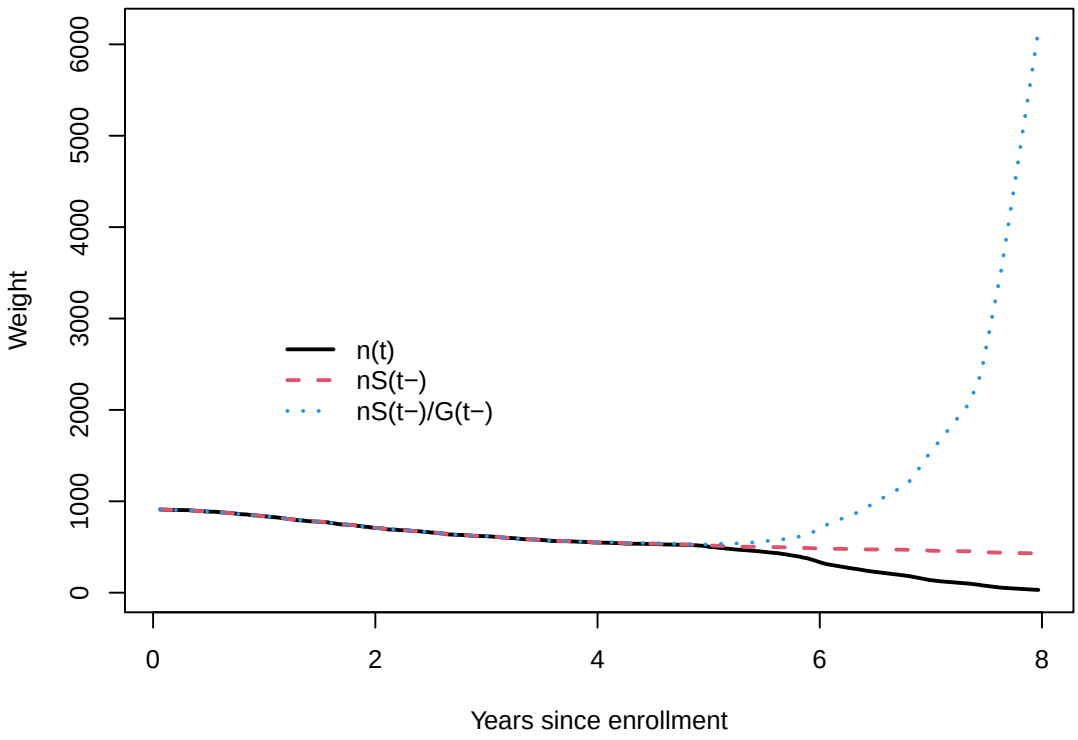
Based on the Peto and Peto argument that $n(t) \approx n(0)S(t-)G(t-)$, we might expect the Schemper and Uno weights to be similar. In testing, we discovered to our surprise that if computations are done carefully as per above, then $n(t) = n(0)S(t-)G(t-)$ exactly. As a consequence, the Schemper and Uno weights are also identical.

The `timewt` option in the `concordance` function allows you to modify weights for the concordance. The options are: `n, S, S/G, n/G2` and `I`, the last of which giving equal weight to each event time; this would correspond to a log-rank test. (We do not recommend this nor are we aware of any literature which does so, but included it for completeness). For non-survival data all the weights become identical.

The figure below shows the first three weights for the colon cancer dataset. This data is from a clinical trial of 929 subjects, with 3 years of enrollment followed by 5 years of follow. Since there is almost no one lost to follow-up in the first 5 years, the choices are nearly identical over that time. From 5 to 8 years, $S(t)$ continues its steady decline, $n(t)$ plummets due to administrative censoring, and S/G explodes. Even with these changes in weights, the concordance values are all very similar.

```
> colonfit <- coxph(Surv(time, status) ~ rx + nodes + extent, data = colon,
                    subset = (etype == 2))   # death only
```

```
> cord1 <- concordance(colonfit, timewt="n",    ranks=TRUE)
> cord2 <- concordance(colonfit, timewt="S",    ranks=TRUE)
> cord3 <- concordance(colonfit, timewt="S/G",  ranks=TRUE)
> cord4 <- concordance(colonfit, timewt="n/G2", ranks=TRUE)
> temp <- c("n(t)"= coef(cord1), S=coef(cord2), "S/G"= coef(cord3),
            "n/G2"= coef(cord4))
> round(temp,5) # 4 different concordance estimates
   n(t)       S     S/G    n/G2
0.65559 0.65437 0.65357 0.65357
> # Plot the weights over time using the first 3 approaches
> matplot(cord1$ranks$time/365.25, cbind(cord1$ranks$timewt,
                                         cord2$ranks$timewt,
                                         cord3$ranks$timewt),
          type= "l", lwd=2, col=c(1,2,4),
          xlab="Years since enrollment", ylab="Weight")
> legend(1, 3000, c("n(t)", "nS(t-)", "nS(t-)/G(t-)"), lwd=2,
         col=c(1,2,4), lty=1:3, bty="n")
> # Note that n/G2 and S/G are identical
> all.equal(cord3$ranks$timewt,cord4$ranks$timewt)
[1] TRUE
```



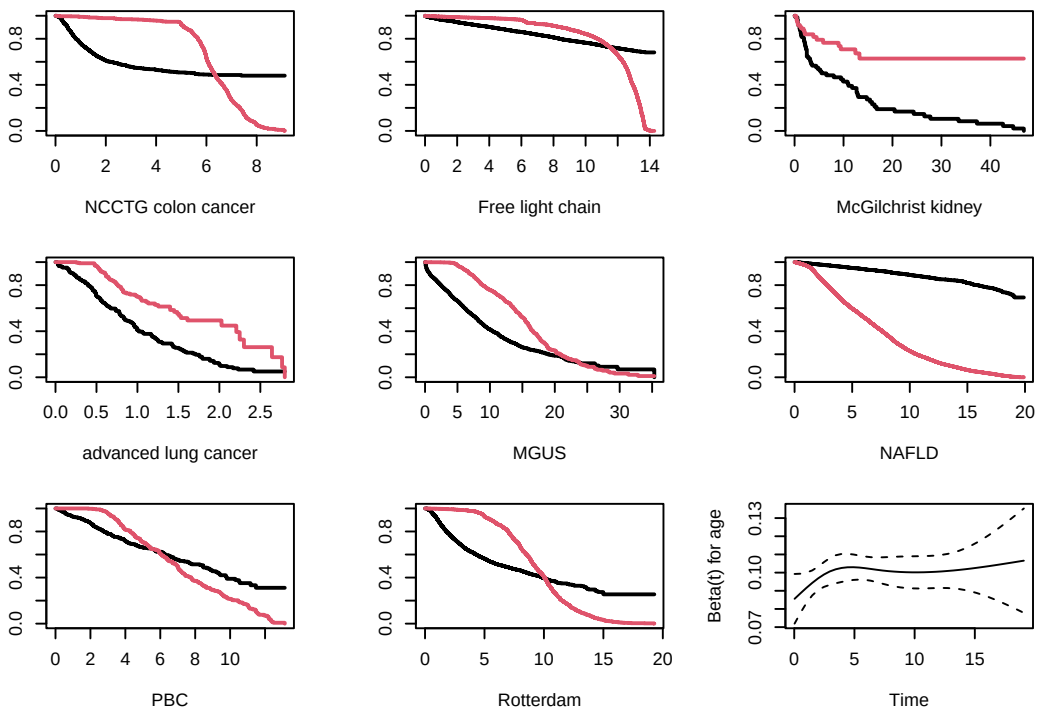


Figure 2: Survival (black) and censoring (red) curves for 8 datasets found in the survival package. The final panel shows a proportional hazards evaluation for the age variable, in a fit of age + male to the NAFLD data.

When do weights matter?

When might the concordance differ between weight functions? In order for S/G weights to show an important difference we need to have two conditions.

- First, sufficient censoring that the two weights differ for a reasonable fraction of the data, that is, $G(t)$ is low and $S(t)$ has not flattened (deaths are still occurring).
- Second, per the arguments in Schemper and in Uno, is the potential presence of non-proportional hazards in the fit.

Figure 2 shows survival and censoring curves for 8 different datasets found in the survival package. Based on these, the dataset with the greatest potential for an S/G difference is the NAFLD data; the dataset also has some early non-proportionality for age. The code below shows the calculation of the S/G difference. Surprisingly, the four weightings still yield very similar concordance values; the Harrell (n) and Uno (n/G2) weighting differ in only the second decimal place.

```
> nfit <- coxph(Surv(futime/365.25, status) ~ age + male, data = nafld1)
> ncord1 <- concordance(nfit, timewt = "n")
> ncord2 <- concordance(nfit, timewt = "S")
> ncord3 <- concordance(nfit, timewt = "S/G")
> ncord4 <- concordance(nfit, timewt = "n/G2")
> temp <- c(n = coef(ncord1), S = coef(ncord2),
            "S/G" = coef(ncord3), "n/G2" = coef(ncord4))
> round(temp,6)
       n        S      S/G     n/G2
0.823254 0.821457 0.805438 0.805438
```

The `concordance` function provides a `ranks=TRUE` argument which can be used to further exploration of the weights. I f set, the output will include a dataframe that contains one row for each event, containing the time point, its relative rank in the risk set ($r_i - \overline{r}$), the case weight for the observation and the time weight at that time point. The relative ranks are comparable to Schoenfeld residuals, and their weighted sum will equal $c - d$. We can plot them over time and apply a smooth. Figure 3, for the veteran dataset, shows a precipitous drop to 0. The Veteran's cancer data is perhaps an extreme of this pattern. Due to the very rapid progression of disease, baseline measurements soon lose their meaning.

```
> # code for Figure 3
> # pick a dataset with a smaller number of points, and non PH
> vfit <- coxph(Surv(time/365.25, status) ~ age + karno, data = veteran)
> temp <- concordance(vfit, ranks=TRUE)$rank
> # Two outliers at 999 days = 2.7 years stretch the axis too far
> plot(rank ~ time, data=temp, xlim=c(0,1.6),
       xlab="Years", ylab="Rank residual")
> lines(lowess(temp$time, temp$rank, iter=1), lwd=2, col=2)
> abline(0, 0, lty=3)
```

2.2 Restricted time range

A first consideration in any model assessment is to stop and ponder what a "good fit" means. This is far too often ignored in the rush to computation. Two wonderful papers in this regard are Korn and Simon [4] and Altman and Royston [1]. The title of the second makes its purpose clear: "What do we mean by validating a prognostic model?"

Korn and Simon, for instance, point out the importance of restricting the range of comparison, a point echoed by Altman and Royston. Though a risk model can be used for long-range prediction, in actual patient practice this will very often not be the need; the model should be evaluated over the time range in which it will actually be used. For example, for a patient who returns every 4 years, a 10 year prediction will never actually be put to the test; predictions will be updated every 4 years based on the most current information. If the model is to be used for ongoing patient management we will want to restrict attention to a 4 year horizon, with respect to evaluating its utility. (A 10 year Framingham heart risk might still be useful in *convincing* a subject to stop smoking, however.)

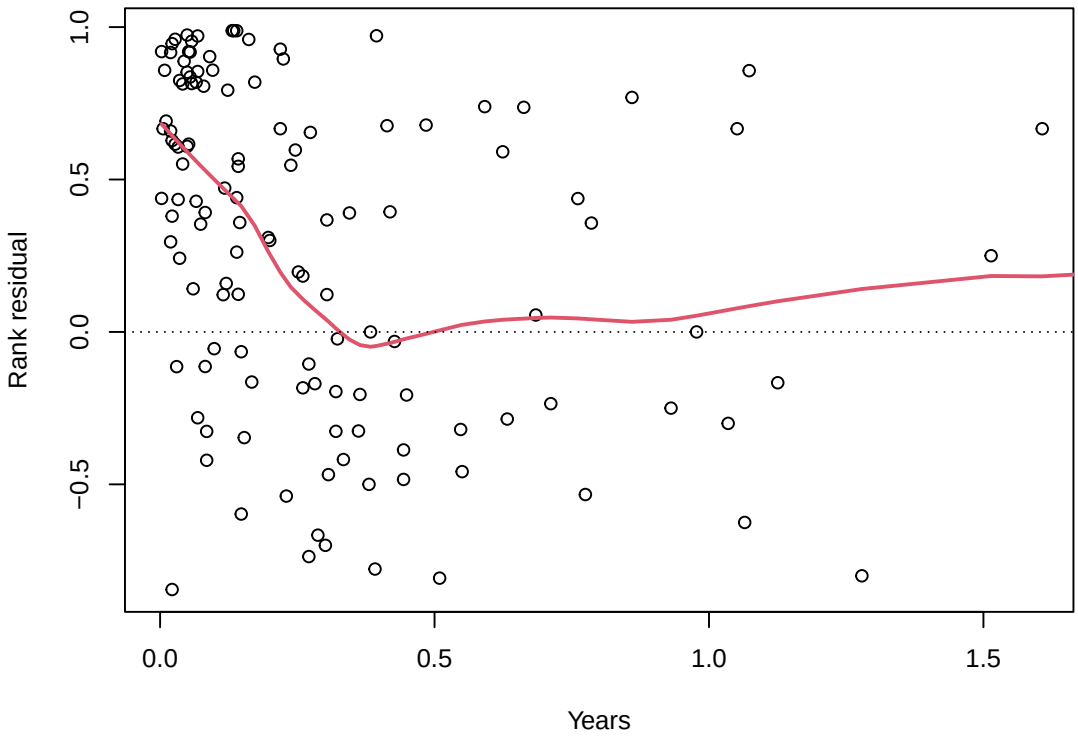


Figure 3: Schoenfeld residuals for the scaled ranks, from a fit to the veteran dataset.

We sometimes receive push-back on this, with the argument that one should use all the data. We disagree, and think that the target of the validation is critical. Predictions become less accurate the further out we reach in time; this is true for everything from weather forecasts to the stock market, and survival models are not immune. Reaching too far into the future may return an overly pessimistic value of C.

Another reason for using an upper limit is that $1/G$ can become unstable as the sample size becomes small (large jumps in the KM), or unreasonably large as G approaches 0. Most authors suggest an upper limit for this purely technical reason.

```
> # should this be included?
> # recurrence free survival = earlier of recurrence and death
> rdata <- rotterdam
> rdata$rfs <-     with(rdata, ifelse(recur==1, 1, death))
> rdata$rfstime <- with(rdata, ifelse(recur==1, rtime, dtime))/ 365.25
> rfit <- coxph(Surv(rfstime, rfs) ~ age + meno + grade + pspline(nodes), rdata)
> ctemp <- matrix(0, 100, 2)  # concordance and std err
> ctime <- seq(.1, 10, length=100)
> for (i in 1:100) {
      temp <- concordance(rfit, ymax=ctime[i])
      ctemp[i,] <- c(temp$concordance, sqrt(temp$var))
  }
> yhat <- ctemp[,1] + outer(ctemp[,2], c(0, -1.96, 1.96), '*')
> matplot(ctime, yhat, type='l', lty=c(1,2,2), lwd=c(2,1,1), col=1,
          xlab="Upper cutoff", ylab="C", ylim=c(0.5,1))
```

Argument could also be made for a lower limit, though this would be uncommon for censored data. Many laboratory values, for instance, treat all results less than some threshold as identical. However, the ability to implement a lower limit correctly is constrained by censoring. Say for instance that there were values of 5, 8+, and 9, and a lower limit of 10 were chosen. The approach used for non-censored data is to treat 5 and 9 as tied values, but this logic does not correctly extend to the censored value. This issue and possible solutions will be discussed more fully in the external validation vignette.

2.3 Synthetic C

As another method of addressing censoring, Göen and Heller [3] show that if the statistical model is correct, and if proportional hazards holds, then for any pair of covariate vectors

$$P(y_i > y_j) = \frac{1}{1 + e^{\eta_j - \eta_i}}$$

They then order the η values from a fitted model, and take an average over all $n(n-1)/2$ ordered pairs. The authors argue that this is an estimate that is independent of censoring, and therefore preferable to Harrell's C. (The estimate can be obtained by using the `royston` function.)

The biggest problem with this approach is that it gives an estimate of concordance under an assumption that the model is *exactly correct*. Our goal, rather, is to assess how *well* the model performs, for our needs, knowing that it will be imperfect. The G and H formula answers a

question that we did not ask, over a time range $(0, \infty)$ which is not of interest. The calculation is also $O(n^2)$ so will be slow for large sample sizes.

2.4 Picking a weight

So, which weight should we use? As shown in the examples above, it may not matter that much. (The fact that we have never found an example where the effect is large does not mean there are no such datasets, however.) Time weights play no role for uncensored data.

Further issues to consider:

1. An important issue that has not been sorted out is how to extend $1/G$ weighting arguments to datasets that are subject to delayed entry, e.g., when using age as the time scale instead of time since enrollment. There is, in this case, no natural estimate available for G. It is also not possible for the `coxph` routine to reliably tell the difference between such left truncation and simple time-dependent covariates or strata. The default action of the routine is to use the safe choice of $n(t)$.

2. Consider setting a time (y) restriction using the `ymax` option, based on careful thought about the proper range of interest. This often has a larger practical effect than the choice of time weight.

3. Safety. If using the usual Gehan-Wilcoxon weights of $n(t)$, the Peto-Wilcoxon variant $S(t)$ would appear advantageous, particularly if there is differential censoring for some subjects.

4. Equality vs. efficiency. On one hand we would like to treat each data pair equally, but in our quest for ever sharper p-values we want to be efficient. The first argues for $n(t)$ as the weight and the second for using equal weights, since the variances of each ranking term are nearly identical. This is exactly the argument between the Gehan-Wilcoxon and the log-rank tests.

5. For uncensored data n, S and S/G weights are all identical.

Our current opinion is that the point of the concordance is to evaluate the model in a more non-parametric way, so a log-rank type of focus on ideal p-values is misplaced. This suggests using either S or S/G as weights. Both give more prominence to the later time points as compared to the default $n(t)$ choice, but if time limits have been thought through carefully the difference between these three will almost always be ignorable.

We most definitely disagree with Uno's unstated assumption that the C statistic one would obtain with infinite follow-up and no censoring is the proper target of estimation, and the ordinary concordance is therefore biased. That target will never be attainable, and we would argue that it is largely irrelevant if it was. Proportional hazards never is true over the long term, simply because it is almost impossible to predict events that are a decade or more away and thus the rank residuals shown above will eventually tend to 0. The starting point should always be to think through exactly *what* one wants to estimate. As stated by Yogi Berra "If you don't know where you are going, you'll end up someplace else."

3 Variance

The variance of the statistic is estimated in two ways. The first is to use the variance of the equivalent Cox model score statistic. As pointed out by Watson, this estimate is both correct and efficient under $H_0 : C = .5$, and so it forms a valid test of H_0. However, when the concordance is over .7 or so, this estimator systematically overestimates the true variance. An alternative that remains unbiased is the infinitesimal jackknife (IJ) variance

$$V = \sum_{i=1}^{n} w_i U_i^2$$
$$U_i = \frac{\partial C}{\partial w_i}$$

The concordance routine calculates an influence matrix U with one row per subject and columns that contain derivatives for the 5 individual counts: concordant, discordant, tied on x, tied on y, and tied on xy pairs. From this it is straightforward to derive the influence of each subject on the concordance, or on any other of the other possible association measures such as τ-a mentioned earlier. The IJ variance is printed by default but the PH variance is also returned; an earlier `survConcordance` function only computed the PH variance.

The `condordance` function does not compute Kendall's τ-a or τ-b, nor Goodman's gamma. However, since all of the necessary components for those values are returned, along with IJ influence for each, it can be used as the computational engine for those measures and their variance, should someone wish to do so.

The variance computation accounts for the influence of each subject on both the numberator and denominator of C; this agrees with parallel development used by Newson [6] and implemented in STATA. An alternate approach is to treat the total number of comparable pairs is an ancillary statistic, thus $\text{var}(C) = \text{var}(c-d)/(4m)$ where m is the number of comparable pairs ($n(n-1)/2$ for uncensored data). Arguments about what aspects of a dataset can or should be treated as ancillary are as old as statistics, e.g., treating the margins of a 2x2 table as ancillary leads to Fisher's exact test. In this case we anticipate that the differences will be quite small, but have done no formal exploration.

3.1 Multiple concordances

One useful property of using a jackknife variance estimate is that the variance of the difference in concordance between two separately fitted models is also easily obtained. If c_a and c_b are the two concordance statistics and U_{ia} and U_{ib} the corresponding influence values, the influence vector for $c_a - c_b$ is $U_a - U_b$. (If subject i increases c_a by .03 and c_b by .01, then he/she raises the difference between them by .02.) It is not necessary that the models be nested. However, it is crucial that they be computed on the exact same set of observations. Here is a comparison of concordance values from previous models.

```
> ctest <- concordance(fit4, fit5, fit6)
> ctest
Call:
concordance.coxph(object = fit4, fit5, fit6)
```

```
n= 137
     concordance     se
fit4      0.7119 0.0224
fit5      0.7384 0.0210
fit6      0.7359 0.0212

     concordant discordant tied.x tied.y tied.xy
fit4       6261       2529     14     39       0
fit5       6499       2301      4     39       0
fit6       6478       2324      2     39       0
> # compare concordance values of fit4 and fit5
> contr <- c(-1, 1, 0)
> dtest <- contr %*% coef(ctest)
> dvar  <- contr %*% vcov(ctest) %*% contr
> c(contrast=dtest, sd=sqrt(dvar), z=dtest/sqrt(dvar))
  contrast         sd          z
0.02646524 0.01662275 1.59211003
```

To do a similar comparison for models which do not have a `concordance` method, use a set of dummy linear model fits as a container. For instance, assume that `y` was continuous, and we have 3 different predicted values `phat1`, `phat2`, `phat3` from three different machine learning models, say, and we want to compute and compare concordance. The one could use the following code:

```
dummy1 <- lm(y ~ phat1)
dummy2 <- lm(y ~ phat2)
dummy3 <- lm(y ~ phat3)
cfit <- concordance(dummy1, dummy2, dummy3)
print(cfit)
etc.
```

In order to produce correct answers, it is necessary that y, phat1, phat2, and phat3 be results for exactly the same observations, in exactly the same order. If one model has some subject removed for missing values, say, then those subjects can not be present in any of the ML fits.

3.2 Asymmetric confidence intervals

The infinitesimal jackknife (IJ) has provided us with an honest estimate of the standard deviation of C. A natural confidence interval for the concordance is then $C \pm z_\alpha sd(C)$. As with confidence intervals for an ordinary proportion $\hat{p}$, however, this simple interval can sometimes be inconsistent, giving CI endpoints that lie outside of the legal range of $[0, 1]$. In the case of $\hat{p}$ there is a long history of methods to address this issue, going back at least as far as the 1956 paper by Anscombe [2], but there is less literature for the concordance or AUC. Newcombe [5] provides corrected methods, but with the caveat that they "have the drawback that on account of the large number of outcomes they are computationally practicable only for very small sample

sizes." The tree based computations used in the `concordance` function might well address the speed issue, but have not been implemented.

Here we pursue another avenue, which is to consider a transformation based confidence interval, in much the same way as is done for confidence intervals of a survival curve. That is, we use

$$g^{-1}\left[g(C) \pm z\sigma(g(C))\right]$$

for some transformation function g. For survival curves, the g functions $\log(p)$, $\log(p/(1-p)$, $\log(-\log(1-p))$ and $\arcsin(p)$ have all been found to be superior to the simple interval.

For the concordance, consider the Fisher z-transform, widely used for the correlation coefficient r

$$z = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right) \tag{10}$$

Since Somers' D and r are targeted at similar concepts, we might hazard that a similar transformation of Somers' D, which also ranges from -1 to 1, would also be close to equivariant. Since $D = 2C - 1$ we have

$$\begin{aligned} z_c &= \frac{1}{2}\log\left(\frac{1+(2C-1)}{1-(2C-1)}\right) \\ &= \frac{1}{2}\log\left(\frac{C}{1-C}\right) \end{aligned}$$

which we recognize as the inverse of the logistic link used `glm` models.

We can get the standard error of z_c by retrieving the individual dfbeta values and performing a transformation. The dfbeta value is defined as $d_i = C - C_{-i}$ where the latter is the C statistic omitting the ith observation.

```
> zci <- function(fit, p=.95) {
      ilogist   <- function(p) log(p/(1-p))         # inverse logistic
      logistic  <- function(x) exp(x)/(1 + exp(x))
      temp <- concordance(fit, influence =1)
      cminus <- temp$concordance - temp$dfbeta # values of concordance, without i

      newd    <- ilogist(temp$concordance) - ilogist(cminus) # dfbeta on new scale
      new.sd  <- sqrt(sum(newd^2))
      old.sd  <- sqrt(sum(temp$dfbeta^2))  # same as sqrt(temp$var)

      z <-  qnorm((1-p)/2)
      old.ci  <- temp$concordance + c(z, -z)*old.sd
      new.ci  <- logistic(ilogist(temp$concordance) + c(z, -z)* new.sd)
      rbind(old = old.ci, new= new.ci)
  }
> round(zci(colonfit), 4)
      [,1]   [,2]
old 0.6302 0.6810
new 0.6298 0.6805
```

The two intervals hardly differ, which is what we would expect for a value far from 1. As a second example, create a small dataset with a concordance that is close to 1. As shown below, the z-transform shifts the CI towards zero, as it should, but also avoids the out-of-bounds endpoint.

```
> set.seed(1953)
> ytest <- matrix(rexp(20), ncol=2) %*% chol(matrix(c(1, .98, .98, 1), 2))
> cor(ytest)
          [,1]      [,2]
[1,] 1.0000000 0.9422072
[2,] 0.9422072 1.0000000
> lfit <- lm(ytest[,1] ~ ytest[,2])
> zci(lfit)
         [,1]      [,2]
old 0.8419721 1.0246946
new 0.7253801 0.9867027
```

4 Details

This section documents a few details - most readers can skip it.

The usual convention for survival data is to assume that censored values come after deaths, even if they are recorded on the same day. This corresponds to the common case that a subject who is censored on day 200, say, was actually seen on that day. That is, their survival is strictly greater than 200. As a consequence, censoring weights G actually use $G(t-)$ in the code: if 10 subjects are censored at day 100, and these are the first censorings in the study, then an event on day 100 should not be given a larger weight. (Both the Uno and Schemper papers ignore this detail.)

When using weights of $S(t)$ the program actually uses a weight of $nS(t-)$ where n is the number of observations in the dataset. The reason is that for a stratified model the weighted number of concordant, discordant and tied pairs is calculated separately for each stratum, and then added together. If one stratum were much smaller or larger than the others we want to preserved this fact in the sum.

References

[1] D. G. Altman and P. Royston. What do we mean by validating a prognostic model? *Stat. in Medicine*, 19:453–73, 2000.

[2] F. J. Anscombe. One estimating binomial response relations. *Biometrika*, 43:461–464, 1956.

[3] M. Göen and G. Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92:965–970, 2005.

[4] E. L. Korn and R. Simon. Measures of explained variation for survival data. *Stat. in Medicine*, 9:487–503, 1990.

[5] R. G. Newcombe. Confidence intervals for an effect size measure based on the mann–whitney statistic. part 2: asymptotic methods and evaluation. *Stat. in Medicine*, pages 559–73, 2006.

[6] R Newson. Confidence intervals for rank statistics: Somers' D and extensions. *Stata Journal*, 6(3):309–334, 2006.

[7] R. Peto and J. Peto. Asymptotically efficient rank invariant test procedures (with discussion). *J. Royal Stat. Soc. A*, 135(2):185–206, 1972.

[8] Ross L Prentice and P Marek. A qualitative discrepancy between censored data rank tests. *Biometrics*, 35(4):861–867, 1979.

[9] M. Schemper, S. Wakounig, and G. Heinze. The estimation of average hazard ratios by weighted Cox regression. *Stat. in Medicine*, 28(19):2473–2489, 2009.

[10] T. M. Therneau and D. A. Watson. The concordance statistic and the Cox model. Technical Report 85, Department of Health Science Research, Mayo Clinic, 2015.

[11] H. Uno, T. Cai, M. J. Pencina, R. B D'Agnostino, and L. J. Wei. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Stat. in Medicine*, 30(10):1105–1117, 2011.