External Validation of Survival Models

Terry Therneau

Nov 2024

1 Introduction

Another vignette dicusses the important issue of validating software, i.e., the task of verifying that the survival package produces numerically correct answers as the result of its computations. This vignette is concerned with the *scientific* validation of a fitted model, which has two facets: how accurately does the fitted model describe or summarize the data at hand, on which it was built, but more importantly, how useful is the model for prediction or summary of future patients. We will mostly focus on the latter, known as external validation, and largely on single and multi-state hazard models.

The hazard model assumes that

$$\lambda_i(t) = e^{\beta_0(t) + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots}$$

If $beta_0(t)$ is a constant this reduces to Poisson regression, if it is a spline this the equation for parametric proportional hazards, and if it is a general non-parametric function we have the Cox proportional hazards model. We will assume from the outset that the key assumptions of the model have been explored and are satisfactory.

1. Proportional hazards: each of the $\beta_i, i > 0$ is constant over time, or equivalently, no interaction terms with β_0 are required.

2. Functional form: each x_i has a linear effect

3. Additivity: No interactions between any x_i terms are required.

2 What does it mean to validate a model?

> "If you don't know where you are going, you might end up someplace else." – Yogi Berra

The key validation question is whether a model is useful, a question that in our option, needs to be immediately clarified by a more careful definition of "success", which will, in our opinion, almost always be application specific. As a hypothetical example, assume that a cancer clinical trial has resulted in a fitted model, along with predicted survival curves specific to each treatment and covaiate pattern. Assume that the disease in question has a high early death rate, and one proposed use of the results is to identify subjects with expected survival of < 6 months for referral to supportive care. For this purpose, the performance of the model wrt separating 1, 2, and 3+ year survivors is immaterial; validation should focus on the metric of interest. Another use might be to identify a subset with expected survival of more than 1 year for enrollment in a trial of long term adjuvant therapy, e.g., a therapy that is not expected to have an influence on the earliest recurrences.

More general validation of an existing model, using a separate external dataset, should not be a yes/no type of exercise, but rather an assessment of which aspects of the prediction are most reliable and which are may be less so. This will particularly be true in multi-state models, where it may well be true that one transition is poorly predicted while the others are successful.

Much of the thinking and machinery for model validation has been developed in the context of binomial models. Survival data is more complex, in three ways.

- There are at least 3 possible assessments
    1. The probability in state j, $p_j(t)$
    2. The expected number of visits to state j, $E(N_j(t))$
    3. The expected total sojourn time in state j, $s_j(t)$
- Each of these can be assessed at one or more chosen times τ.
- The validation data set is subject to censoring.

For a simple alive/dead survival 1 and 2 above are the same: the expected number of visits to the death state by time $t =$ the probability of death by time t. The same is true for any absorbing state in a mulit-state model. Which of the measures is most appropriate, and even more so which time points τ might be a rational focus, will depend on the application target, i.e., on the situation specfic defintion of "successful" model prediction.

Censoring of the validation data is of course a central technical issue, as it always is for time to event models. There are essentially four ways to deal with it

1. Create uncensored data specific to a chosen assessment time τ, then use standard methods for uncensored data. Two approaches are

- Use the redistribut-to-the-right (RTTR) algorithm to reassign the case weights of those censored prior to τ to other subjects in the data.
- Replace the response with pseudovalues at time τ

2. Apply standard censored data methods to the validation data, and compare the results to the target model's predictions. I will sometimes call this the "$\hat{y}$ vs $\hat{y}$" approach.

3. For assessment type 2, the total number of events, we can compare the observed events to the expected number given the per-subject followup. (Adjust the prediction to the data, rather than the data to the prediction.) This arises naturally out of the counting process approach, and is closely related to standardized mortality ratios (SMR), a measure with a very long history.

4. Ignore censoring. A disastrous approach, but one that sometime appears in practice.

One of the problems in the field, in our opinion, is the attempt to force survival into the Procrustean bed of binomial methods. This leads to a focus on a single timepoint τ, $p(\tau)$ as the target, and use of the RTTR to create a binomial response. This has the (small) positive benefit of making the results look familiar, at the cost of potentially large inefficiency. There is, after all, a reason that the Cox model is used for the analysis of censored data than RTTR(τ) followed by logistic regression. I have often heard the argument that "we cannot do anything else, because the readers will not understand it", but it is not one I accept. Assuming a stupid audience is not a justification for sub-par methods.

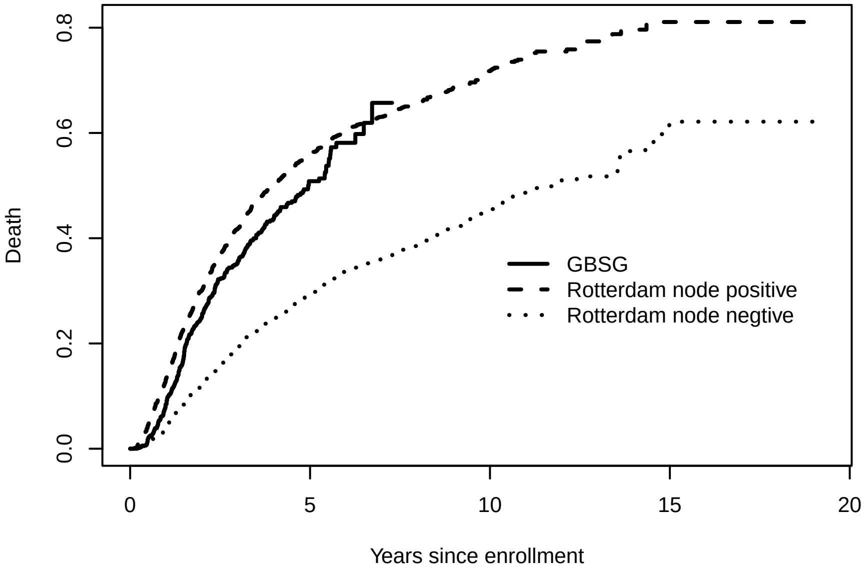
3 Data

One challenge with any presentation on external validation is the need for data set pairs, one to fit the model and another to assess it. There is a dearth of such available, but we have tried to assemble some useful cases.

3.1 Breast cancer

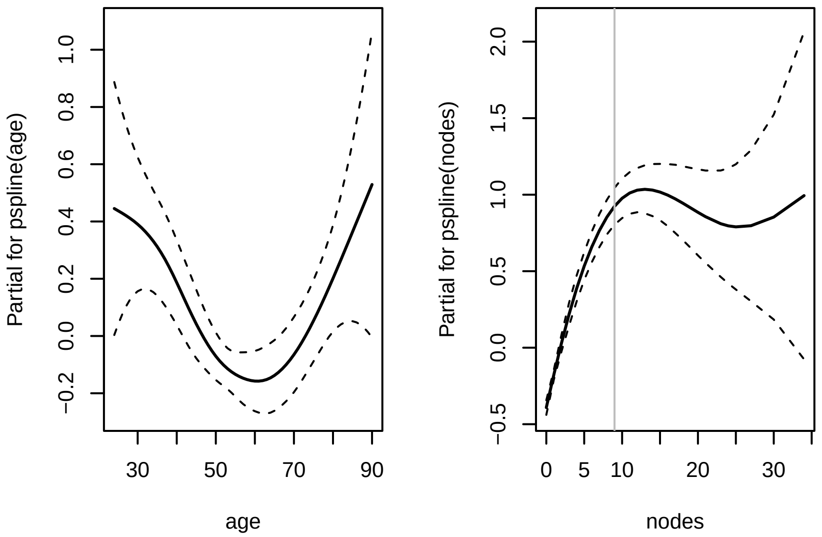
The `rotterdam` data contains information on 2892 primary breast cancer patients from the Rotterdam tumor bank. The `gbsg` data set contains the patient records of 628 patients with complete data for prognostic variables, out of 720 enrolled in 1984–1989 trial conducted by the German Breast Study Group for patients with node positive breast cancer. These data sets were introduced by Royson and Sauerbrei [?] and have been widely used. The figure below shows overall survival for the two groups. The Rotterdam study has longer follow-up.

```
> gsurv <- survfit(Surv(rfstime/365.25, status) ~1, gbsg)
> rott2 <- rotterdam
> rott2$rfs <- with(rott2, ifelse(rtime<dtime, recur, death))
> rott2$rfstime <- rott2$rtime/365.25  # plots are easier in years
> rsurv <- survfit(Surv(rfstime, rfs) ~ I(nodes==0), rott2)
>
> plot(rsurv, lty=2:3, lwd=2, conf.int=FALSE, fun="event",
     xlab="Years since enrollment", ylab="Death")
> lines(gsurv,  lwd=2, lty=1, fun="event", conf.int=FALSE)
> legend(10, .4, c("GBSG", "Rotterdam node positive", "Rotterdam node negtive"),
       lty=1:3, lwd=2, bty='n')
```



Above, we have followed the more conservative view found in the Rotterdam help file with respect to deaths after the end of progression follow-up, i.e., that using them can create a small immortal time bias with respect to the endpoint of recurrence free survival. This is the `rfs, rfstime` variable pair in data `rott2`. Now fit a model to the rotterdam data, but omit the chemo and year variables since they do not appear in the gbsg data set.

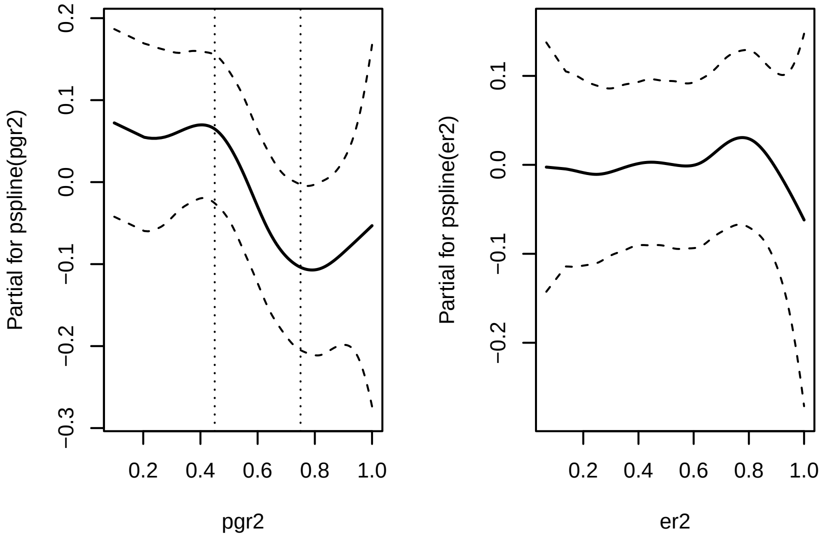
```
> rfit <- coxph(Surv(rfstime, rfs) ~ pspline(age) + meno + size + grade +
     pspline(nodes) + hormon, rott2)
> print(rfit, digits=1)
Call:
coxph(formula = Surv(rfstime, rfs) ~ pspline(age) + meno + size +
    grade + pspline(nodes) + hormon, data = rott2)

                         coef se(coef)    se2  Chisq DF      p
pspline(age), linear   -4e-04    3e-03  3e-03  1e-02  1  0.907
pspline(age), nonlin                           3e+01  3  8e-07
meno                    3e-01    1e-01  1e-01  7e+00  1  0.007
size20-50               3e-01    6e-02  6e-02  2e+01  1  2e-06
size>50                 4e-01    9e-02  9e-02  3e+01  1  2e-07
grade                   3e-01    6e-02  6e-02  3e+01  1  3e-08
pspline(nodes), linear  8e-02    6e-03  5e-03  2e+02  1 <2e-16
pspline(nodes), nonlin                         2e+02  3 <2e-16
hormon                 -3e-01    8e-02  8e-02  2e+01  1  5e-05

Iterations: 7 outer, 19 Newton-Raphson
     Theta= 1
     Theta= 1
Degrees of freedom for terms= 4.0 0.9 2.0 1.0 4.0 1.0
Likelihood ratio test=633  on 13 df, p=<2e-16
n= 2982, number of events= 1659
>
> # I dislike the color choices of termplot
> opar <- par(mfrow=c(1,2), mar=c(5,5,1,1))
> termplot2 <- function(fit, ...) termplot(fit, col.term=1, col.se=1, ...)
> termplot2(rfit, term=1, se=TRUE)
> termplot2(rfit, term=5, se=TRUE)
> abline(v=9, col="gray")
```



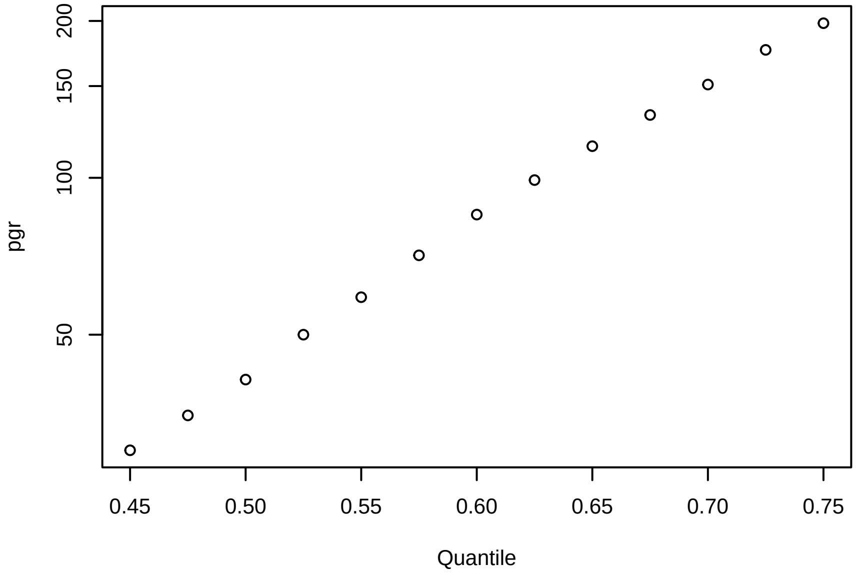
```
> par(opar)
```

Age has a very non-linear effect, with greater risk for both the youngest and oldest subjects. Risk increases with the number of nodes, up to about 8-9 and then stabilizes. Recode nodes in this way, then examing the addition of estrogen and/or progesterone receptors. Using ranks is a useful tool for a "first look" at a very skewed variable, as ER and PGR are.

```
> rfit2 <- coxph(Surv(rfstime, rfs) ~ pspline(age) + meno + size + grade+
                     pmin(nodes,8) + hormon, rott2)
> anova(rfit2, rfit)
Analysis of Deviance Table
 Cox model: response is  Surv(rfstime, rfs)
 Model 1: ~ pspline(age) + meno + size + grade + pmin(nodes, 8) + hormon
 Model 2: ~ pspline(age) + meno + size + grade + pspline(nodes) + hormon
  loglik  Chisq     Df Pr(>|Chi|)
1 -12133
2 -12130 7.5892 3.0271    0.05646
>
> er2 <- rank(rott2$er)/nrow(rott2)
> pgr2<- rank(rott2$pgr)/nrow(rott2)
> rfit3 <- update(rfit2, . ~ . + pspline(pgr2))
> rfit4 <- update(rfit3, . ~ . + pspline(er2))
> anova(rfit2, rfit3, rfit4)
Analysis of Deviance Table
 Cox model: response is  Surv(rfstime, rfs)
 Model 1: ~ pspline(age) + meno + size + grade + pmin(nodes, 8) + hormon
 Model 2: ~ pspline(age) + meno + size + grade + pmin(nodes, 8) + hormon + pspline(pgr2
 Model 3: ~ pspline(age) + meno + size + grade + pmin(nodes, 8) + hormon + pspline(pgr2
  loglik  Chisq     Df Pr(>|Chi|)
1 -12133
2 -12129 8.6451 4.0375     0.0723
3 -12128 1.1868 4.0651     0.8860
>
> opar <- par(mfrow=c(1,2), mar=c(5,5,1,1))
> termplot2(rfit4, term=7, se=TRUE)
> abline(v= c(.45, .75), lty=3)
> termplot2(rfit4, term=8, se=TRUE)
```



```
> par(opar)
>
> # the progesterone receptor effect looks fairly linear from the 45th to 75th
> #  percentile.  What values are those?
> qq <- seq(.45, .75, .025)
> quantile(rott2$pgr, qq)
  45% 47.5%   50% 52.5%   55% 57.5%   60% 62.5%   65% 67.5%   70% 
   30    35    41    50    59    71    85    99   115   132   151 
72.5%   75% 
  176   198 
> 
> plot(qq, quantile(rott2$pgr, qq), log='y',
      xlab="Quantile", ylab="pgr")
```



```
> 
> # The log values are a moderately linear transform of the ranks, over this range
> #  so create a truncated variable
> rott2$pgr3 <- pmax(30, pmin(200, rott2$pgr))
> rfit5 <- update(rfit2, .~. + log(pgr3))
> print(rfit5, digits=1)
Call:
coxph(formula = Surv(rfstime, rfs) ~ pspline(age) + meno + size + 
    grade + pmin(nodes, 8) + hormon + log(pgr3), data = rott2)
```

```
                        coef se(coef)    se2  Chisq DF      p
pspline(age), linear  1e-03    3e-03  3e-03  1e-01  1  0.739
pspline(age), nonlin                         3e+01  3  1e-06
meno                  2e-01    1e-01  1e-01  5e+00  1  0.021
size20-50             3e-01    6e-02  6e-02  2e+01  1  2e-06
size>50               4e-01    9e-02  9e-02  3e+01  1  2e-07
grade                 3e-01    6e-02  6e-02  3e+01  1  5e-07
pmin(nodes, 8)        2e-01    9e-03  9e-03  4e+02  1 <2e-16
hormon               -3e-01    8e-02  8e-02  2e+01  1  7e-05
log(pgr3)            -9e-02    3e-02  3e-02  8e+00  1  0.005

Iterations: 7 outer, 18 Newton-Raphson
     Theta= 1
Degrees of freedom for terms= 4.0 0.9 2.0 1.0 1.0 1.0 1.0
Likelihood ratio test=633  on 11 df, p=<2e-16
n= 2982, number of events= 1659
```

We have been aggressive in the fit, particularly in using the observed shape to create an "optimal" pgr3 variable. But, since such overfitting is likely common in creating a risk score we will retain this final version, and expect some imperfection to be revealed in the validations. The are some differences between the Rotterdam and GBSG covariates: GBSG has the size in mm instead of grouped, 12% of the GBSG subjects are grade 1 and none of the Rotterdam, GBSG has no patients with 0 nodes versus 48% in Rotterdam, and the Rotterdam follow-up is much longer. We have purposely ignored these, mimicing a risk score that has been built on one data set, before the potential further uses of it were known.

As a summary look at the relative importance of the components of the risk score by looking at their additive contribution. On this metric the number of nodes has by far the largest effect, followed by size, age and grade. The pgr term, on which we spent so much effort, has the least impact.

```
> rmat <- predict(rfit5, type='terms')
> round(apply(rmat, 2, sd), 2)
  pspline(age)           meno           size          grade
          0.15           0.12           0.16           0.14
pmin(nodes, 8)         hormon      log(pgr3)
          0.48           0.10           0.07
```

The survival package makes validation computations easy if the original and validation data sets have *exactly* the same variables. For the GBSG data we will need to first change size to a categorical, then add the pgr3 and rfs variables.

```
> gbsg2 <- gbsg
> gbsg2$sizec <- gbsg2$size  # sizec= continuous size in mm
> gbsg2$size <- cut(gbsg2$sizec, c(0,20, 50, 500), c("<=20", "20-50", ">50"))
> gbsg2$pgr3 <- with(gbsg2, pmax(30, pmin(200, pgr)))
> gbsg2$rfs  <- gbsg2$status
> gbsg2$rfstime <- gbsg2$rfstime/365.25
```

3.2 Amyloidosis

Still need to add these datasets.

3.3 Monoclonal gammopathy

3.4 URSO treatment

3.5 Dementia and Death

4 Discrimination

4.1 Concordance

One of the simpler measures is the relative ordering between two distributions

$$P(X_i > X_j | Y_i > Y_j) = P(Y_i > Y_j | X_i > X_j)$$

(The two forms above are numerically equivalent, just with a different viewpoint.) In the present case replace X with the model predictions $\hat{y}_i$, and Y with t_i, the observed survival times. A pair of observations is concordant if the ordering of the results agrees with the ordering of the predictions. This is a partial validation, since a prediction might have perfect concordance but poor absolute prediction. (An oracle who could predict the winner of each football match, but not the score, could do very well, but not for all tasks.) This ability to properly order the outcomes is known as discrimination.

One numerical 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 response 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) effectively score as 1/2.

- 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. 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 (gives them a tie) then a random choice is made. This leads to $c + t_x/2$ correct selections out of $c + d + t_x$ choices.

As a measure of association the concordance has two rather interesting properties. The first is that that the prediction can be shifted by an arbitrary constant without changing C, or equivalently that we do not need the intercept term of the predictor equation. A second is that, for single state survival, all 3 of our assessments lead to the same value: if the predicted probability of death is lower for subject A than B, then the expected number of death events will lower for A, as will the expected number of years in the death state, and further, this order does not depend on what target time τ might be chosen. There is thus a single concordance, which requires only the linear predictor $X\beta$ from a `coxph` fit.

4.2 Censoring

Two observations with the same response (y, survival time) value are not counted in the comparisons for C, as stated above they are considered *incomparable*. For survival data this is extended to censored times: an observation censored at 5 and a death at time 8 are also incomparable since we do not know if the first subject will or will not outlive the second. Pairs of (5+, 8) and (5+, 7+) are both incomparable.

Wlog order the data from smallest to largest time. The numerator of the concordance can then be written as

$$\sum_{i=1}^{n} \delta_i w(t_i)(r_i(t) - \overline{r}(t))$$

where δ_i is 0 for censored and 1 for uncensored observations, $r_i(t)$ is the rank of $\hat{y}_i$ among all those still at risk at time t_i, and $w(t)$ is a time dependent weight. This looks very much like the score statistic from a Cox model, and indeed for a model with a single 0/1 predictor it turns out to be exactly the Cox score statistic. Rewriting the concordance in this form also properly allows for time dependent covariates.

When there is a single binary predictor the case weights of $w(t) = 1$, $n(t)$, $S(t-)$ and $S(t-)/G(t-)$ correspond to the log-rank, Gehan-Wilcoxon, Peto-Wilcoxon, and Schemper tests for a treatment effect, where $n(t)$ is the number at risk at time t, S is the survival and G the censoring distribution. Many other weights have also been suggested. For the concordance, weights of $n(t)$ and $S(t-)/G(t-)$ correspond to the suggestions of Harrell and of Umo, respectively. At one time arguments about the relative strength and weakness of the various survival tests had a prominent role in the literature, but experience has shown that, for nearly all data sets, the choice does not matter all that much; and this once lively controversy has died away. In our limited experience, the same is true for weighted versions of the concordance, and we predict that this discussion too will fade with time. (Note that for uncensored data, $G = 1$ and all the above weights are identical.)

A variation that is important is the dichotomized concordance: replace t_i with $I(t_i <= \tau)$ for some chosen cutpoint τ, i.e., an indicator of death at or before time τ. Then

- The RTTR algorithm can be used to reassign the case weights of all those censored before τ, for whom the value of the indicator variable is uncertain, to other cases. This results in a weighted data set where the indictor is known, for all those with non-zero weight.
- Compute a weighted concordance on the remaining observations.

Since it is based on a 0/1 outcome the resulting value of C is equal to the area under a reciever operating curve (AUROC). This approach has been labeled as the "time-dependent AUROC". We consider this label a poor choice, however, because it invites confusion with time-dependent covariates, and will henceforth refer to this approach as the "dichotomized time concordance" or DTC.

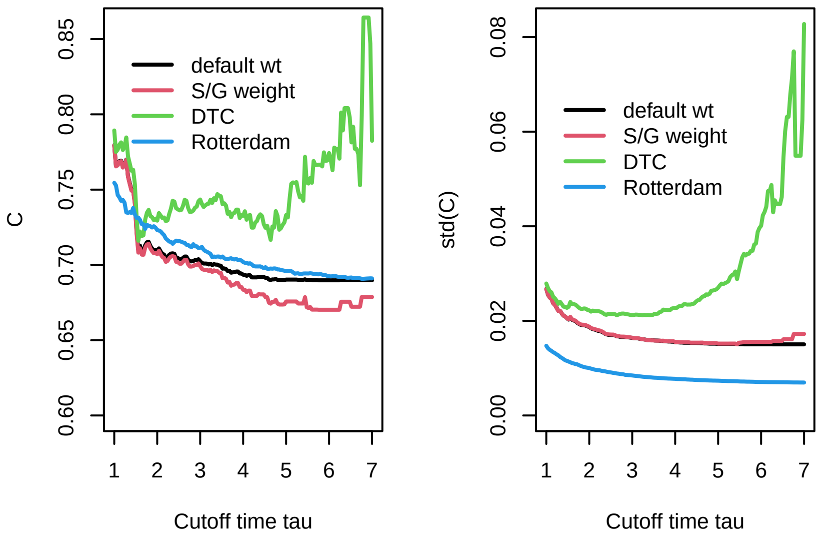
The DTC and C measure different things. To see this, divide the counts of concordant and discordant pairs in C into three groups: ($t_i \le \tau$ and $t_j \le \tau$), ($t_i > \tau$ and $t_j > \tau$), and ($t_i \le \tau$ and $t_j > \tau$); the DTC is based only on the third group. The DTC tends to be a bit larger than C, but with a larger variance. With respect to the first of these, consider all pairs with $|t_i - t_j| < c$ where c is a small constant, say 1/20 the range of t. These are normally the hardest pairs to score correctly, and C includes many more of them, DTC only has the subset which are close to τ.

4.3 Breast cancer

The next figure shows the C statistic with $n(t)$ (Harrell, default) and S/G weights (Schemper or Uno) at cutpoints τ from 1 to 7 years, along with the DTC at those same points. The C statistic at $\tau = 4$, say, is a measure of how accurate predictions are up through 4 years, and is computed by treating all pairs with both t_i and t_j greater than 4 as tied.

```
> tau <- seq(1, 7, length=151)
> reweight <- rttright(Surv(rfstime, rfs) ~ 1, gbsg2, times=tau, renorm=FALSE)
>
> Cstat <- array(0, dim=c(151,2,4)) # 3 values, 4 std
> yhat <- predict(rfit5, newdata=gbsg2)
> for (i in 1:151) {
     c1 <- concordance(rfit5, newdata=gbsg2, ymax=tau[i])
```

```
    c2 <- concordance(rfit5, newdata=gbsg2, ymax=tau[i], timewt="S/G")
    temp <- with(gbsg2,ifelse(rfs==1 & rfstime <= tau[i], 1, 0))
    c3 <- concordance(temp~ yhat, data=gbsg2, weight=reweight[,i],
                      subset=(reweight[,i] > 0))
    c4 <- concordance(rfit5, ymax=tau[i]) # concordance of the training data
    Cstat[i,,1] <- c(coef(c1), sqrt(vcov(c1)))
    Cstat[i,,2] <- c(coef(c2), sqrt(vcov(c2)))
    Cstat[i,,3] <- c(coef(c3), sqrt(vcov(c3)))
    Cstat[i,,4] <- c(coef(c4), sqrt(vcov(c4)))
}
> opar <- par(mfrow=c(1,2), mar=c(5,5,1,1))
> matplot(tau, Cstat[,1,], lwd=2, lty=1, col=1:4, type='l', ylim=c(.6, .86),
         xlab="Cutoff time tau", ylab="C")
> legend(1, .85, c("default wt", "S/G weight", "DTC", "Rotterdam"),
        lwd=2, lty=1, col=1:4, bty='n')
> matplot(tau, Cstat[,2,], lwd=2, lty=1, col=1:4, type='l',
         ylim=c(0, max(Cstat[,2,])),
         xlab="Cutoff time tau", ylab="std(C)")
> legend(1, .07, c("default wt", "S/G weight", "DTC", "Rotterdam"),
        lwd=2, lty=1, col=1:4, bty='n')
```



```
> par(opar)
```

The plot is very interesting.

- For this particular data set pair, the discrimination of the model in the GBSG data set is not too far from its performance in the Rotterdam data used to build the model. At 4 years the values are .70 and .69, less than 1 std apart. (Tgw better performance before year 1.5 is an unexplained oddity.)
- Concordance gets worse over time, which is not a surprise. Outcomes farther in the future are harder to predict in essentially all areas of life, survival time is no exception.
- The S/G weighting gives larger weights than $n(t)$ to points later in time, and consequently leads to a lower estimate. But the two remain close with a difference of approximately .01 at 4 years and .02 at 6 years. The S/G weight can become quite large at later times, leading to more bounce in the estimate over time.
- The DTC is larger, particularly at later times, but with substantially larger variance at those later times.

The IJ variance used by the `concordance` function allows computation of a variance for the difference between any two of the methods.

```
> # To be filled in
```

The plot does not tell us *which* τ cutoff to use, that has to be decided based on the goals of the validation study.

5 Expected number of events