External Validation of Survival Models

Terry Therneau

Nov 2024

This is an early draft.

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 three key assumptions of the model have been explored and are satisfactory.

1. Proportional hazards: each of the β_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. More specifically "useful for the application in hand". Any definition of success must, 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 of the model 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. We might further want to stratify treatment assignment by the expected disease free survival. This leads to a quite different metric for success. For instance, mis-labeling a subject who will liver more than one year as a death before 1 year is a less severe error than the converse, the first only reduces enrollment rate while the second can bias results.

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. In this case there will be a collection of validation results, which can help guide users with respect to their particular needs.

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

- There are at least 3 possible assessments
  1. The expected number of visits to state j, $E(N_j(t))$
  2. The probability in state j, $p_j(t)$
  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 formally the same: the expected number of visits to the death state by time $t = n$ times the probability of death by time t, and the same is true for any absorbing state in a mulit-state model. Our computational approach for the two targets is however quite different. Measure 2 has been the most used, and 3 the least. 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, i.e., on the situation specfic defintion of successful prediction.

The validation will almost certainly be censored, and how to deal with this is a central technical issue, as it always is for time to event models. There are essentially four approaches.

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 observations censored prior to τ to others, resulting in a weighted subset who status at τ is known.
   - 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 1, the total number of events, we can compare the observed events to the expected number given per-subject followup. (Adjust the prediction to the data, rather than the data to the prediction.) This method arises naturally out of the counting process approach to survival data, and is closely related to standardized mortality ratios (SMR), a measure with a long history.

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

In summary, the validation of time-to-event data has three primary dimensions: choice of the appropriate summary statistic, the choice of a range or set of prediction times, and choice of a method for handling censoring. The first and two of these will be closely tied to application at hand, the third is more technical.

One of the problems in the field, in our opinion, has been the determined attempt to force survival into the Procrustean bed of binomial methods. This most often 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 positive benefit of making the results look familiar, at the cost of potentially statistical inefficiency, and far more importantly, correctly answering the wrong question. 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 this is not one which 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 Altman [1] and have been widely used. Figure 1 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.

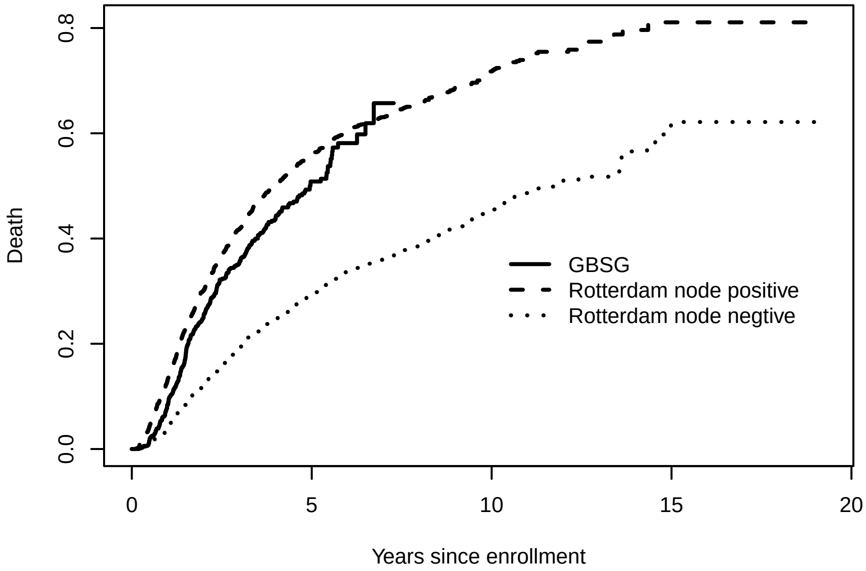
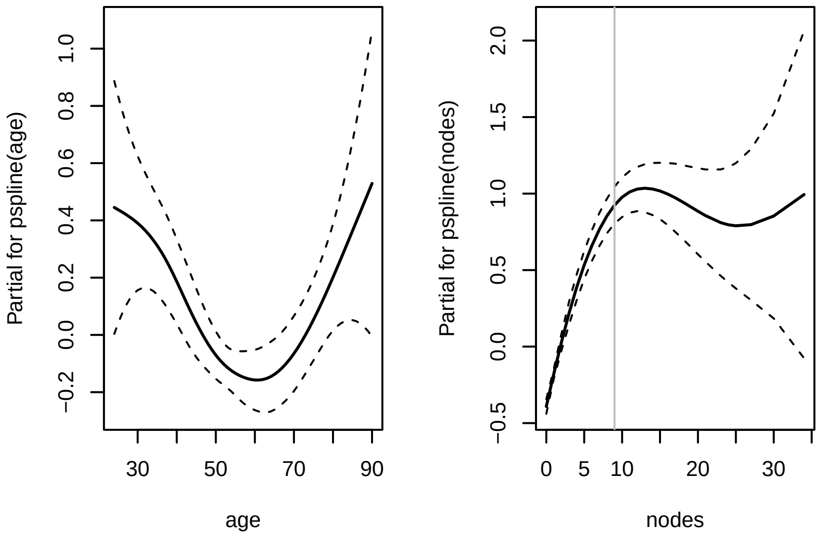


Figure 1: Rotterdam and GBSG data

```
> 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
```

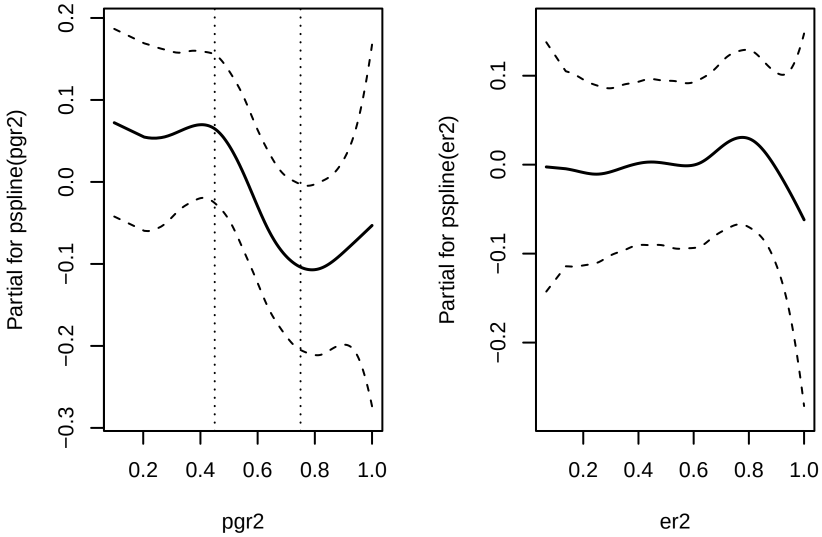
```
     Theta= 1
     Theta= 1
Degrees of freedom for terms= 4.0 0.9 2.0 1.0 4.0 1.0
Likelihood ratio test=633  on 13 df, p=<2e-16
n= 2982, number of events= 1659
>
> # I dislike the color choices of termplot
> opar <- par(mfrow=c(1,2), mar=c(5,5,1,1))
> termplot2 <- function(fit, ...) termplot(fit,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



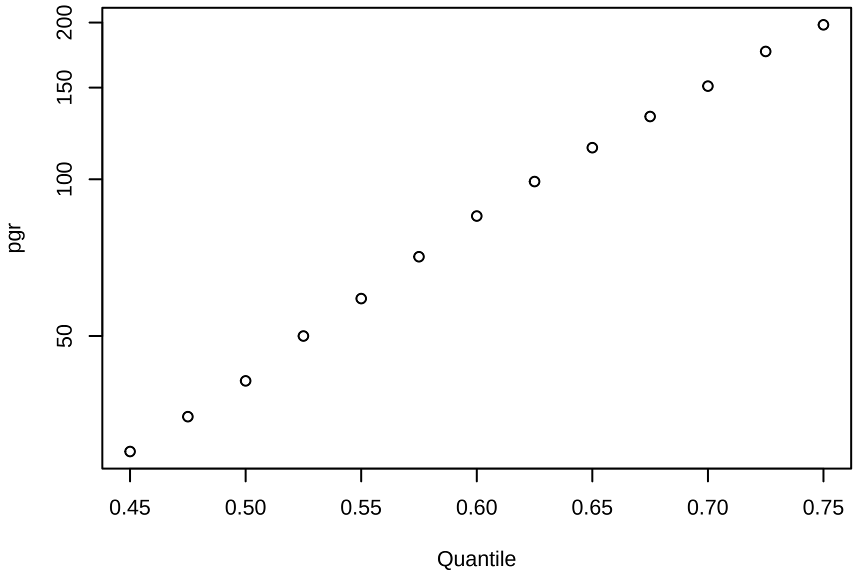
```
> par(opar)
```

Results are in figure **??**. 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 examine 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.

```
> rott2$node8 <- pmin(rott2$nodes, 8)
> rfit2 <- coxph(Surv(rfstime, rfs) ~ pspline(age) + meno + size + grade+
                     node8 + hormon, rott2)
> anova(rfit2, rfit)
Analysis of Deviance Table
 Cox model: response is  Surv(rfstime, rfs)
 Model 1: ~ pspline(age) + meno + size + grade + node8 + 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 + node8 + hormon
 Model 2: ~ pspline(age) + meno + size + grade + node8 + hormon + pspline(pgr2)
 Model 3: ~ pspline(age) + meno + size + grade + node8 + hormon + pspline(pgr2) + psplin
  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 + node8 + 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
node8                 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.

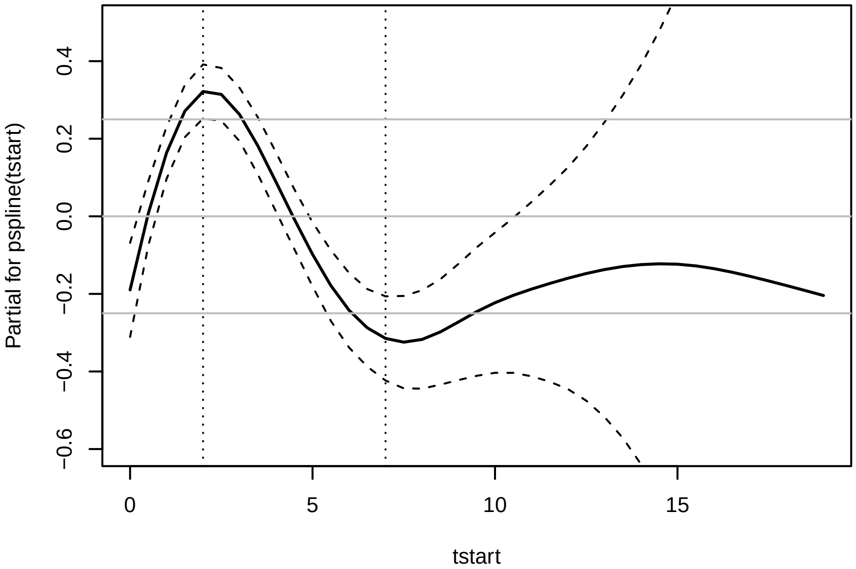
```
> rmat <- predict(rfit5, type='terms')
> round(apply(rmat, 2, sd), 2)
  pspline(age)            meno            size           grade
          0.15            0.12            0.16            0.14
pmin(nodes, 8)          hormon       log(pgr3)
          0.48            0.10            0.07
```

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$rfstime/365.25
> gbsg2$node8 <- pmin(gbsg2$nodes, 8)
```

Age scale As an alternative analysis consider doing the model with current age as the time scale and time since enrollment as a covariate. A first question is how to model time-dependent age: we will start with a spline, forcing current age to be updated every .5 years. The longest followup in the Rotterdam cohort is just ovef 19 years.

```
> rott3 <- survSplit(Surv(rfstime, rfs) ~ ., rott2, cut= seq(.5,19, .5))
> rott3$age1 <- rott3$age + rott3$tstart
> rott3$age2 <- rott3$age + rott3$rfstime
>
> rfit6 <- coxph(Surv(age1, age2, rfs) ~ pspline(tstart) + meno + size + grade +
                     node8 + hormon + log(pgr3), rott3)
> 2*c(diff(rfit5$loglik), diff(rfit6$loglik))
[1] 633.1712 739.1222
> termplot2(rfit6, term=1, se=TRUE, ylim=c(-6,5)/10)
> abline(v=c(2,7), lty=3)
> abline(h=c(-.25, 0, .25), col='gray')
```



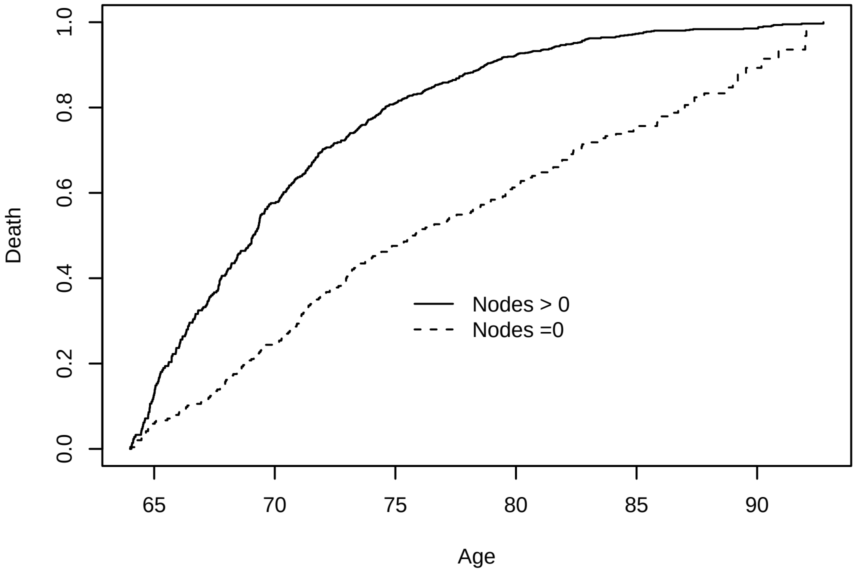
The partial likelihood for this model is actually larger than when we modeled age. The RFS rate rises by about 1.6 over the first 2 years after enrollment, then falls again, stabilizing after around 7 years. (The confidence intervals get crazy after about 12 years so we restricted the range of the plot.)

(Aside: the proper complement to a model with time-since-enrollment as the time scale and age as a covariate is one with age as a time scale and enrollment time as a covariate.

Each model controls for one of the variables by matching and the other by modeling it.)

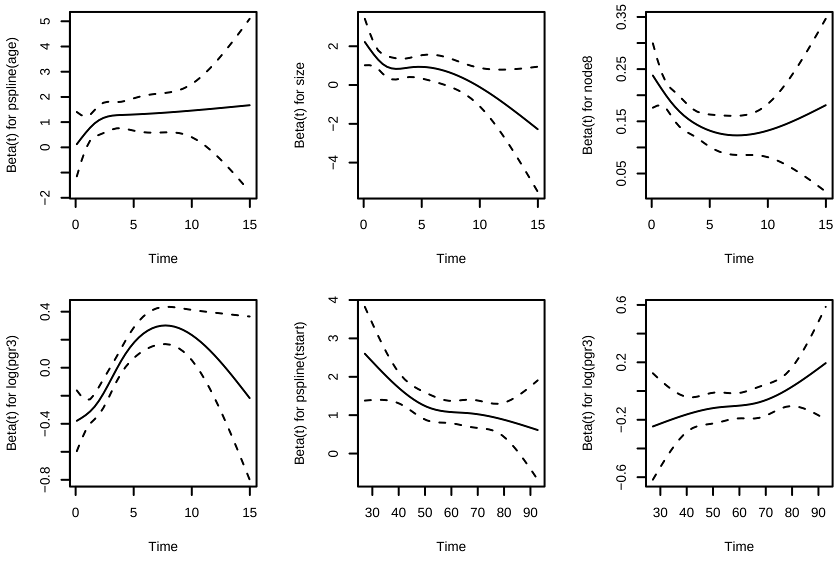
When drawing a survival curve on age scale, we also need to specify a starting point, e.g., expected survival for someone enrolled at age 64 would be

```
> surv64 <- survfit(Surv(age1, age2, rfs) ~ I(nodes==0), rott3, start.time=64)
> plot(surv64, fun='event', lty=1:2,  xlab="Age", ylab="Death")
> legend(75, .4, c("Nodes > 0", "Nodes =0"), lty=1:2, bty='n')
```



Footnote: the PH assumption is more tenable on age scale. No idea yet whether or how to discuss this.

```
> zp5 <- cox.zph(rfit5, transform='identity') # 4 hard fails
> zp5
              chisq    df       p
pspline(age) 23.447  4.04 0.00011
meno          2.626  0.92 0.09422
size         19.664  2.00 5.4e-05
grade         3.525  1.00 0.06038
node8        20.373  1.00 6.4e-06
hormon        0.379  1.00 0.53756
log(pgr3)    47.200  1.00 6.4e-12
GLOBAL       94.477 10.96 2.1e-15
> zp6 <- cox.zph(rfit6, transform='identity') # 2
> zp6
                  chisq    df       p
pspline(tstart) 31.340  4.06 2.8e-06
meno             0.524  1.00   0.469
size             1.566  2.00   0.457
grade            1.174  1.00   0.279
node8            0.527  1.00   0.468
hormon           0.111  1.00   0.739
log(pgr3)        4.520  1.00   0.034
GLOBAL          39.161 11.06 5.2e-05
>
> opar <- par(mfrow=c(2,3), mar=c(5,5,1,1))
> plot(zp5[1], resid=FALSE)
> plot(zp5[3], resid=FALSE)
> plot(zp5[5], resid=FALSE)
> plot(zp5[7], resid=FALSE)
> plot(zp6[1], resid=FALSE)
> plot(zp6[7], resid=FALSE)
```



```
> par(opar)
```

3.2 Amyloidosis

Still need to add this data. True external validation of 3 separate risk scores.

3.3 Monoclonal gammopathy

Early enrollment predicting late enrollment. Semi-competing risks.

3.4 URSO treatment

Application of the PBC risk score to a trial of UCDA.

3.5 Dementia and Death

Fits from a fixed population (Mayo Clinic Study of Aging) applied to data from the Alzheimer's Disease Neuroimaging Initiative (ADNI)

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., a 0/1 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 0/1 indictor is known, for all those with non-zero weight.
- Compute a weighted concordance on the remaining observations.

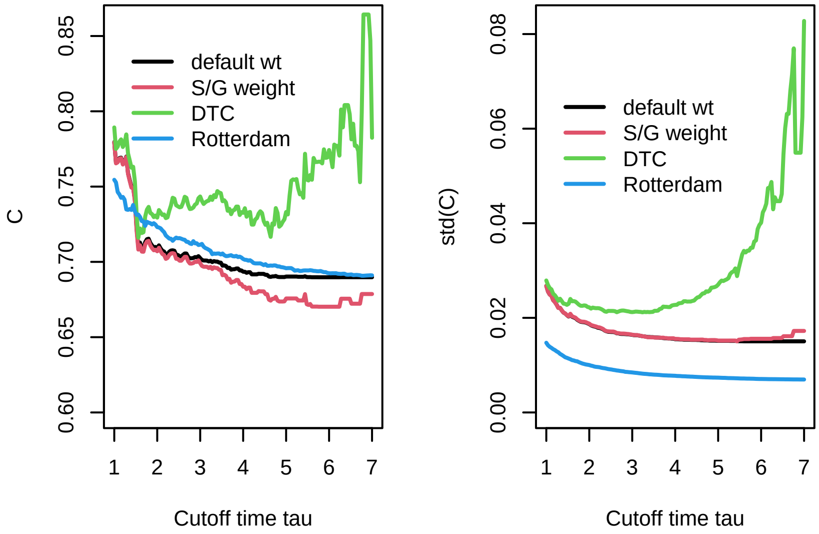
Since it is based on a 0/1 outcome the resulting value of C is equal to the area under a reciever operating curve (AUROC). This approach has been labeled as the "time-dependent AUROC". We consider this label a poor choice, however, because it invites confusion with time-dependent covariates, and will henceforth refer to this 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: both times $\leq \tau$, both times $> \tau$, and ($t_i \leq \tau$, $t_j > \tau$); the DTC is based only on the third group and so will be numerically different. The DTC tends to be a bit larger than C, but with a larger variance. With respect to why it is larger, 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. C includes many more such pairs, the DTC only has the subset which are close to τ.

4.3 Breast cancer

Figure **??** shows the C statistic with $n(t)$ (Harrell, default) and S/G weights (Schemper or Uno) at 151 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 can be 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)
>
> 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; essentially identical at 7 years. At 4 years the values are .70 and .69, less than 1 std apart. (The 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.

Why is the DTC variance so much larger? As τ increases the RTTR algorithm has set more and more observations' weight to zero: by time 6 354/686 = 52% of them. The induced 0/1 variable has counts of 296 and 36; and we know that for binomial data the precision of an estimate is proportional to the smaller group.

The IJ variance used by the `concordance` function also 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.

Age scale

Using age as the time scale does not change the underlying computation for the concordance: for the death of observation i at age a_i, the rank of η_i is compared to that for all others which are alive and at risk at that age. The total number of comparable subjects at risk at any given moment is lower, however, leading to far fewer pairs and a small increase in the standard error.

```
> c5 <- concordance(rfit5)
> c6 <- concordance(rfit6)
> c5
Call:
concordance.coxph(object = rfit5)

n= 2982
Concordance= 0.6865 se= 0.006716
concordant discordant     tied.x     tied.y    tied.xy
   2286148    1043853       1108        557          0
> c6
Call:
concordance.coxph(object = rfit6)

n= 35747
Concordance= 0.6836 se= 0.007368
concordant discordant     tied.x     tied.y    tied.xy
    395071     182551       1108         21          0
>
> sum(c6$count[1:2])/ sum(c5$count[1:2])
[1] 0.17346
>
> gbsg3 <- survSplit(Surv(rfstime, rfs) ˜ ., gbsg2, cut= seq(.5, 8, .5))
> gbsg3$age1 <- with(gbsg3, age + tstart)
> gbsg3$age2 <- with(gbsg3, age + rfstime)
> concordance(rfit6, newdata=gbsg3)
Call:
concordance.coxph(object = rfit6, newdata = gbsg3)

n= 4583
Concordance= 0.6845 se= 0.01728
concordant discordant     tied.x     tied.y    tied.xy
     11430       5260         27          3          0
```

Translation of the DTC to age scale is difficult. On a purely technical level, the RTTR algorithm is no longer available so there is not a simple way to create the 0/1 variable. More challenging perhaps than this is the question of what 0/1 variable one would define, even if there were no censoring.

5 Expected number of events

One of the more flexible, but unfortunately less known methods is to assess the observed and expected number of events. It is based on the simple fact that if the model is correct then the counting process $N_i(t) - \Lambda(t; x_i)$ is a martingale. More importantly, there can be a separate stopping point for each subject, i.e., $N_i(t_i) - \Lambda_i(t_i; x_i)$ is a martingale, where t_i is the observed follow-up time of subject i. This is a variant of the well known gambler's ruin: no stopping strategy can change the long run outcome of a game of chance.

Berry shows that N_i has the same likelihood function, up to a constant, as a Poisson distribution with rate parameter Λ_i. We can thus assess the goodness of fit using simple glm fits. The compensator $\Lambda_i(t_i)$ is the predicted number of events for each subject, which is obtained from the predict function.

```
> gbsg2$expect <- predict(rfit5, type='expected', newdata=gbsg2)
> gbsg2$eta    <- predict(rfit5, type="lp", newdata=gbsg2)
>
> temp <- with(gbsg2, c(Observed=sum(rfs), Expected= sum(expect),
              "O/E"= sum(rfs)/sum(expect)))
> round(temp,3)
Observed Expected      O/E
 299.000  248.718    1.202
>
> gfit1 <- glm(rfs ~ offset(log(expect)), poisson, gbsg2, subset=(expect > 0))
> summary(gfit1)

Call:
glm(formula = rfs ~ offset(log(expect)), family = poisson, data = gbsg2,
    subset = (expect > 0))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.18412    0.05783   3.184  0.00145

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 800.45  on 678  degrees of freedom
Residual deviance: 800.45  on 678  degrees of freedom
AIC: 1400.5
```

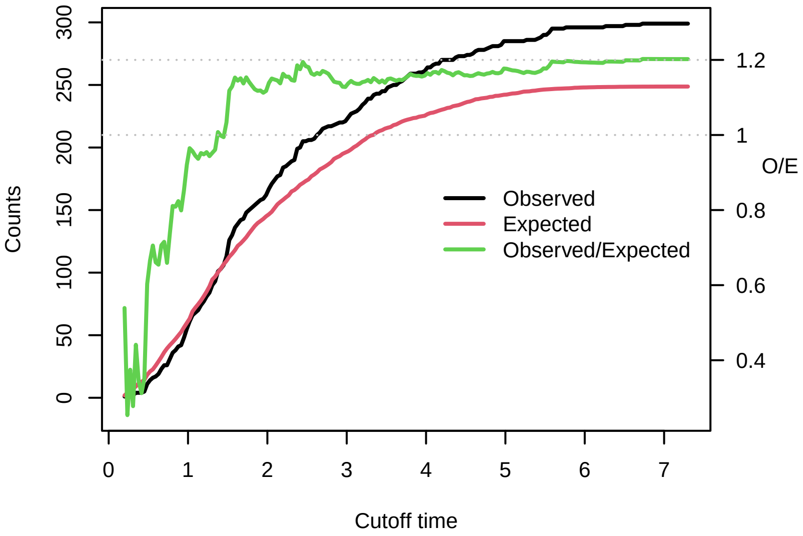
```
Number of Fisher Scoring iterations: 6
> temp2 <- coef(gfit1) + sqrt(vcov(gfit1))[1,1] *c(0, -1.96, 1.96)
> names(temp2) <- c("O/E", "lower CI", "upper CI")
> round(exp(temp2), 3)
     O/E lower CI upper CI
   1.202    1.073    1.346
```

Over the full follow-up time, the gbsg data has 20% more deaths than predicted by the model, given the covariates. This total excess is known as mean calibration. This is also as a standardized mortality ratio (SMR), though that term is more common when the predicted death rate is based on national life tables.

The use of `subset= (expect >0)` deserves comment. When there are observations in the validation data set that are censored before the first event in the training data (there are 7 such in the GBSG data), then the Poisson log-likelihood has a term of $0 \log(0)$. Using L'Hôpital's rule we know that this is 0; thus these observations do not contribute to the loglik and can be omitted. The `glm` function does not know calculus, however, and will generate NA if these observations are retained.

We can also look at the accumulation of observed and expected over time. This is currently a bit clumsy in the software, a `tmax` argument in the `predict` function, similar to ymax in concordance, would make this simpler. Nevertheless:

```
> tau <- seq(.2, 7.3, length=200)
> oe <- matrix(0, length(tau), 2)
> tdata <- gbsg2
> for (i in 1:nrow(oe)) {
     tdata$rfstime <- pmin(gbsg2$rfstime, tau[i])
     tdata$rfs <- ifelse(gbsg2$rfstime>tau[i], 0, gbsg2$rfs)
     pp <- predict(rfit5, newdata=tdata, type='expect')
     oe[i,] <- c(sum(tdata$rfs), sum(pp))
 }
> opar <- par(mar=c(5,5,1,5))
> tfun <- function(x) (x-.3)* 300
> matplot(tau, cbind(oe, tfun(oe[,1]/oe[,2])), type='l', lty=1,
                    lwd=2, col=1:3,
         xlab= "Cutoff time", ylab="Counts")
> z <- seq(.4, 1.2, by=.2)
> axis(4, tfun(z), z, las=1)
> mtext("O/E", side=4, line=2, las=1, padj= -3) # move it away from 0.8 axis label
> legend(4, 180, c("Observed", "Expected", "Observed/Expected"),
         lty=1, lwd=2, col=1:3, bty='n')
> abline(h= tfun(c(1,1.2)), col='gray', lty=3)
```



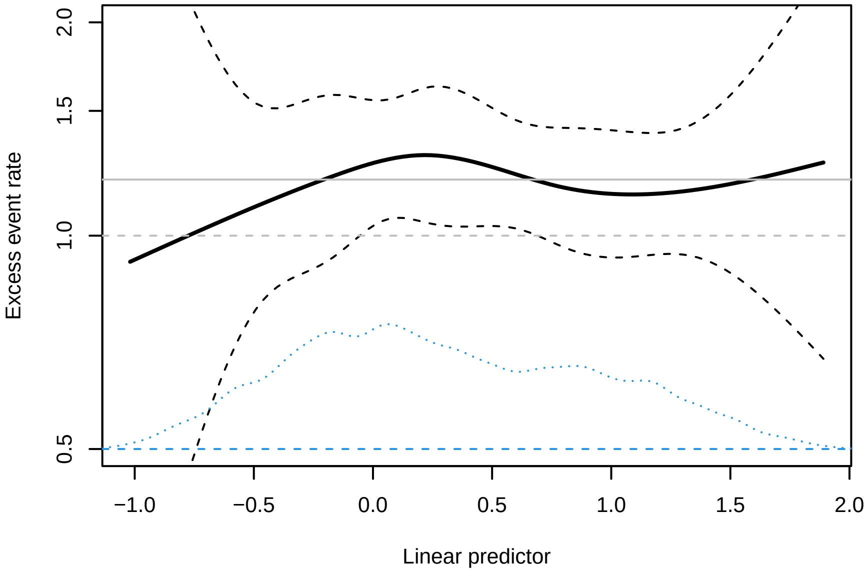
```
> par(opar)
```

The observed RFS events start out as less than expected for the first half year, then catch up by year 1, and slowly rise from 1.1 to 1.2 times expected from year 2–7. We have not yet made the crucial decision about the time period of interest. Do we want to know about predictive behavior over the first year, first 2 years, 5? The correct answer depends of course on the application at hand. This data set pair has been widely used for dicussion of validation, and most of those papers have chosen 5 years without any explicit argument for that time point. Looking ahead, however, we might see the popular RTTR based methods begin to have issues after 5 years, in the same way as the DTC statistics in Figure **??**. In order to facilitate doing validation over the range of 0–5 years, add 3 more variables to the gbsg2 dataset: rfstime5, rfs5, and expect5.

```
> tdata <- gbsg2
> tdata$rfs <- ifelse(tdata$rfstime >5, 0, tdata$rfs)
> tdata$rfstime <- pmin(tdata$rfstime, 5)
> gbsg2$rfstime5 <- tdata$rfstime
> gbsg2$rfs5 <- tdata$rfs
> gbsg2$expect5 <- predict(rfit5, tdata, type='expected')
>
> gfit5 <- glm(rfs5 ~ offset(log(expect5)), poisson, data=gbsg2,
              subset= (expect5 > 0))
> exp(coef(gfit5))   # over the shorter interval, the SMR is approx 1.18
(Intercept)
   1.176636
```

The next level of assessment is to look at the exess as a function of the prediction, known as a calibration plot. The predict call below uses expect=1 to get the death rate relative to the ideal of 1.0, which would be perfect match to the predicted events. We have added horizontal lines at 1.0 and at the overall excess. In this case, the excess does not appear to be related to the per-subject risk, e.g., it is not limited to the highest or lowest risk subjects. The hint of a drop near the left is only a hint, at no point does the 95% pointwise confidence band exclude the overall value of 1.2. A scaled density of the eta values is has been added to the bottom. There are a 60 subjects with $\eta < -.5$, risk .6 of predicted, 12 events and eventsbut the small number of events

```
> gfit2 <- update(gfit1, . ~ . + nsp(eta, 4))
> xx <- seq(min(gbsg2$eta), max(gbsg2$eta), length=100)
> yhat <- predict(gfit2, newdata= data.frame(eta=xx, expect=1), se=TRUE)
> yy <- yhat$fit + outer(yhat$se.fit, c(0, -1.96, 1.96), '*')
>
> matplot(xx, exp(yy), type='l', lty=c(1,2,2), lwd=c(2,1,1), log='y', col=1,
         ylim=c(.5, 2),
         xlab="Linear predictor", ylab="Excess event rate")
> abline(h=c(1, 1.2), col='gray', lty=2:1)
> #rug(gbsg2$eta)
> temp <- density(gbsg2$eta, adjust=.5)
> tfun <- function(y) .5 + (y-min(y))/(4*diff(range(y))) # 4 "looks good"
> lines(temp$x, tfun(temp$y), lty=3, col=4)
> abline(h=.5, lty=2, col=4)
```

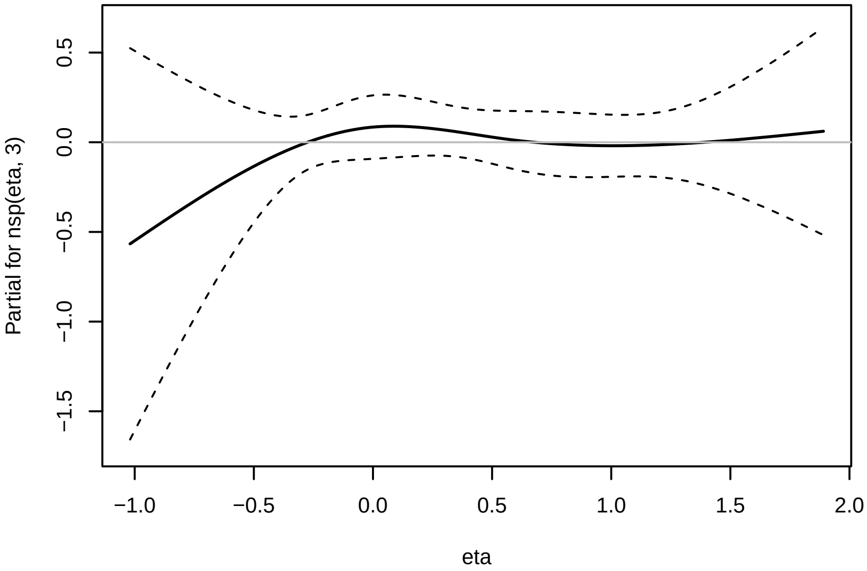


We can also divide the predicted risk score `eta` into bins to create a tableau reminiscent of the Hosmer-Lemishow approach in binomial data. In this case we lack any formal justfication for treating the last row as components of a chi-square statistic.

```
> bins <- with(gbsg2, cut(eta, quantile(eta, 0:5/5), include.lowest=TRUE,
                          labels= paste0("Q",1:5)))
> counts <- rbind(observed= tapply(gbsg2$rfs, bins, sum),
                  expected= tapply(gbsg2$expect, bins, sum))
> counts <- rbind(counts, "O/E"= counts[1,]/counts[2,],
                  "(O-E)^2/E"  = (counts[1,]-counts[2,])^2/counts[2,])
> round(counts,2)
             Q1    Q2    Q3    Q4    Q5
observed  28.00 49.00 62.00 61.00 99.00
expected  27.36 36.79 45.17 61.92 77.48
O/E        1.02  1.33  1.37  0.99  1.28
(O-E)^2/E  0.01  4.05  6.27  0.01  5.98
```

If we repeat the plot for 5 year data there is no substantive change. I use the termplot function below, which centers curves at the grand mean, the horizontal line at 0 is equivalent to one at 1.17 on the uncentered plot.

```
> gfit5b <- update(gfit5, . ~ .+ nsp(eta,3))
> termplot2(gfit5b, se=TRUE)
> abline(0,0, col='gray')
```

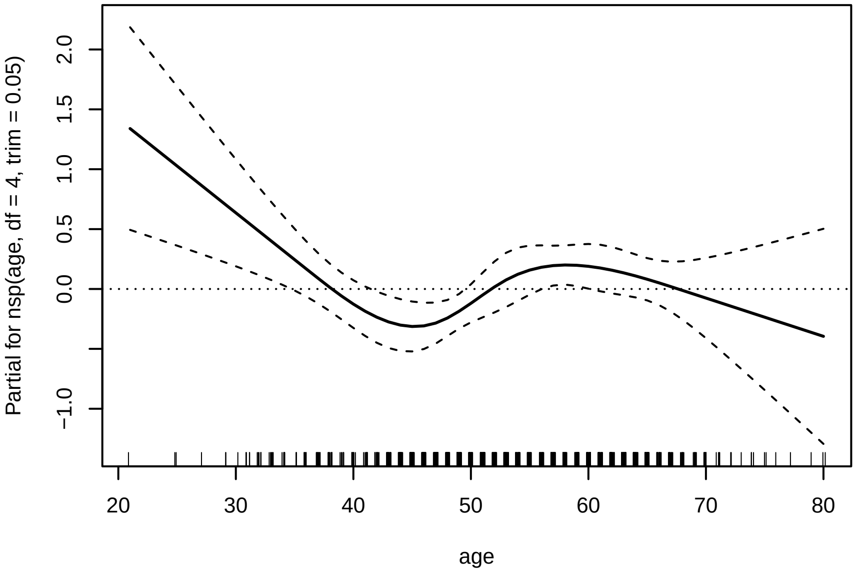


More useful may be to look at components of the risk score.

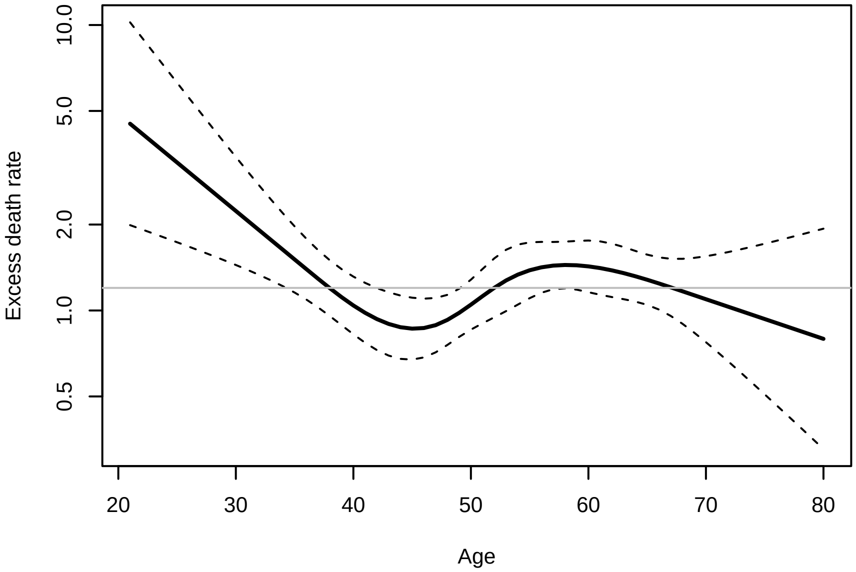
- The first fit `gfit3a` reveals nothing of consequence.
- Because the GBSG data set has subjects who are grade 1–3 while the Rotterdam data had only 2–3, a second fit looks at the three grades separately. Grade was modeled as an integer, and the pattern of excess risk coefficients suggests that perhaps the 1-2 increment in risk is in actuality larger than the 2-3 increment. (The model has overestimated risk for grade 1 and underestimated for grade 2). The overall ANOVA suggests that this could be random variation, however.
- A similar check is made for the three size categories. The medium group, which has 66% of the cases, is almost identical to the overall: exp(.186)= 1.204. There is a hint of a trend, but neither the ANOVA or a trend test are significant.
- For nodes, we have broken the distrbution into four approximatley even groups and look at the excess per group. There is not a consistent pattern of excess: largest for 1 node, smallest for 2–3, and intermediate for 4–7 and 8+.
- Last we look at age, fitting a spline to the excess.

```
> gfit3a <- update(gfit1, . ~. +  meno + grade + hormon)
> pcoef <- function(x) printCoefmat(summary(x)$coefficients, digits=2)
> pcoef(gfit3a)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.233      0.245     1.0      0.3
meno           0.039      0.121     0.3      0.7
grade         -0.027      0.101    -0.3      0.8
hormon        -0.041      0.128    -0.3      0.7
>
> gfit3b <- update(gfit1, . ~. + factor(grade) -1)
> pcoef(gfit3b)
               Estimate Std. Error z value Pr(>|z|)
factor(grade)1   -0.122      0.236    -0.5      0.6
factor(grade)2    0.263      0.070     3.7    2e-04
factor(grade)3    0.078      0.113     0.7      0.5
> with(gbsg2, table(grade))    # nicer label than table(gbsg2$grade)
grade
  1   2   3
 81 444 161
> anova(gfit1, gfit3b, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect))
Model 2: rfs ~ factor(grade) + offset(log(expect)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     800.45
2       676     796.43  2    4.024   0.1337
>
> gfit3c <- update(gfit1, . ~ . + size -1)
> printCoefmat(summary(gfit3c)$coef, digits=2)
          Estimate Std. Error z value Pr(>|z|)
size<=20     0.257      0.124     2.1    0.038
size20-50    0.186      0.070     2.7    0.008
size>50      0.034      0.183     0.2    0.854
> with(gbsg2, table(size))
size
 <=20 20-50   >50
```

```
  180   453    53
> anova(gfit1, gfit3c, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect))
Model 2: rfs ~ size + offset(log(expect)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     800.45
2       676     799.40  2   1.0495   0.5917
>
> ngrp <- cut(gbsg2$nodes, c(0,1,3,7, 60), c('1', '2-3', '4-7', '8+'))
> table(ngrp)
ngrp
  1 2-3 4-7  8+
187 189 167 143
> gfit3d <- update(gfit1, . ~ . + ngrp -1)
> pcoef(gfit3d)
        Estimate Std. Error z value Pr(>|z|)
ngrp1      0.330      0.130     2.5     0.01
ngrp2-3    0.021      0.129     0.2     0.87
ngrp4-7    0.223      0.108     2.1     0.04
ngrp8+     0.178      0.103     1.7     0.08
> round(exp(coef(gfit3d)), 2)
  ngrp1 ngrp2-3 ngrp4-7  ngrp8+
   1.39    1.02    1.25    1.19
> anova(gfit1, gfit3d, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect))
Model 2: rfs ~ ngrp + offset(log(expect)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     800.45
2       675     797.43  3   3.0208   0.3884
>
> # move boundary knots in from the edge a bit, which I prefer
> gfit3e <- update(gfit1, . ~ . + nsp(age, df=4, trim=.05))
> termplot2(gfit3e, se=TRUE)
> abline(0,0, lty=3)
> rug(jitter(gbsg2$age))
```



```
> anova(gfit1, gfit3e, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect))
Model 2: rfs ~ nsp(age, df = 4, trim = 0.05) + offset(log(expect))
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     800.45
2       674     786.59  4   13.862 0.007749
>
> # redraw this directly on the excess risk scale
> dummy <- data.frame(age=21:80, expect=1)
> yhat <- predict(gfit3e, newdata=dummy, se.fit=TRUE)
> yy <- yhat$fit + outer(yhat$se, c(0, -1.96, 1.96), '*')
> matplot(21:80, exp(yy), log='y', type='l', lty=c(1,2,2),lwd=c(2,1,1), col=1,
         xlab="Age", ylab="Excess death rate")
> abline(h=1.2, col='gray')
```



The age check shows a systematic bias in the predicted versus observed risk, and it is statistically significant. How do we parse this out? The quartiles of age in the GBSG study are 46 and 61, so approximately the middle half of the subjects lie in the upward trending portion of the plot. Referring back to the coxph fit, the predicted risk score drops over this interval (lower risk at 61). The coxph model predicts that the hazard drops by around 15% over this region, the gbsg data says "perhaps not so much". One way to look at this is to directy fit age in the GBSG data, holding all other coefficients at the rotterdam fit values.

Note– some further work makes be believe that there is something wrong with the last plot above. Come back to this later

6 Survival models

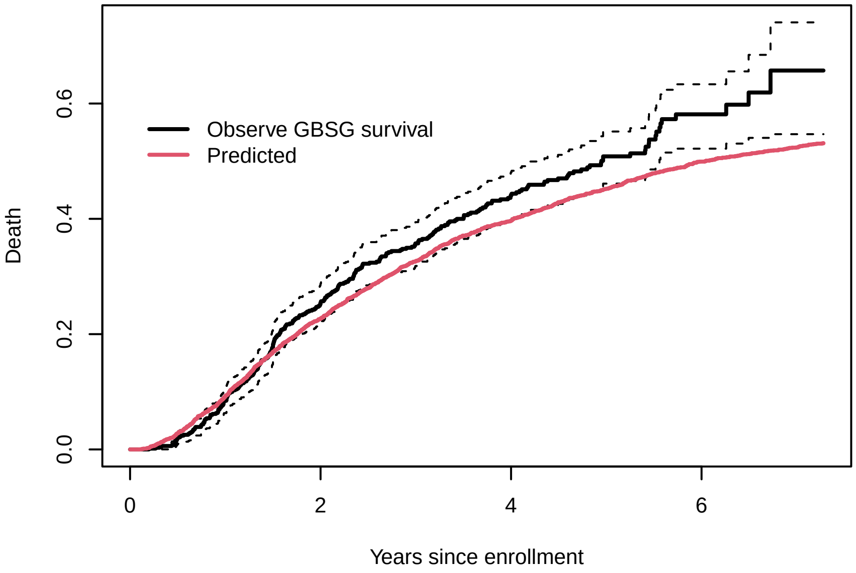
A perhaps obvious approach to the censored data in the validation data set is to use standard censored methods of the Kaplan-Meier, Cox model, etc.

One simple approach is to compare the predicted survival curve for the validation cohort, based on the model, with the KM of the validation cohort. For the predicted curve we need to use the proper marginal estimate which is

$$\begin{aligned} \overline{S}(t) &= (1/n)\sum_{i=1}^{n} \hat{S}(t; x_i) \qquad (6) \\ &\neq \hat{S}(t; \overline{x}) \qquad (7) \end{aligned}$$

where x is the set of covariate vectors for the n validation samples and $\hat{S}$ is prediction using the target model. Equation (6) is called the direct adjusted survival, or in more recent years as a g-estimate. The curve at the mean covariate, equation (7) is not the same.

```
> direct <- survfit(rfit5, newdata=gbsg2)
> dim(direct)
data
 686
>
> plot(gsurv, fun='event', lwd=c(2,1,1),
      xlab="Years since enrollment", ylab= "Death")
> dtemp <- direct
> dtemp$surv <- rowMeans(direct$surv)
> dtemp$std.err <- NULL
> lines(dtemp, fun='event', conf.int=F, lwd=2, col=2)
> legend(0, .6, c("Observe GBSG survival", "Predicted"), col=1:2, lwd=2, bty='n')
```



```
>
> temp1 <- 1- summary(gsurv, time=4:7)$surv
> temp2 <- 1- summary(dtemp, time=4:7)$surv # predicted death at years 4:7
> round(temp1/temp2,3)
[1] 1.111 1.125 1.164 1.255
```

Object `direct` contains 686 separate predicted survivals, one per subject. Their average is the cohort survival. Exacly as in the SMR plot, we see that the observed deaths start out a little smaller than expected over the first few months, are about equal at 1 year, and greater than expected after 2 years. The ratio of probability of death at 4–7 years is identical to the SMR, but not dissimilar in size.

The more common approach is to fit a Cox model with eta as the only covariate, as a way to assess the linearity of the relationship.

```
> cfit1 <- coxph(Surv(rfstime, rfs) ~ eta, gbsg2)
> cfit2 <- coxph(Surv(rfstime, rfs) ~ eta + offset(eta), gbsg2)
> cfit1
Call:
coxph(formula = Surv(rfstime, rfs) ~ eta, data = gbsg2)

       coef exp(coef) se(coef)     z      p
eta 1.03744   2.82198  0.09705 10.69 <2e-16

Likelihood ratio test=112.5  on 1 df, p=< 2.2e-16
n= 686, number of events= 299 
> cfit2
Call:
coxph(formula = Surv(rfstime, rfs) ~ eta + offset(eta), data = gbsg2)

       coef exp(coef) se(coef)     z   p
eta 0.03744   1.03815  0.09705 0.386 0.7

Likelihood ratio test=0.15  on 1 df, p=0.6996
n= 686, number of events= 299 
> cfit3 <- coxph(Surv(rfstime, rfs) ~ nsp(eta, 4) + offset(eta), gbsg2)
> 
> termplot2(cfit3, term=1, se=TRUE, ylab="Estimated effect of risk score")
> abline(0,0, lty=2, col=2)
```

Fit cfit1 is often called regression calibration, a good model should have a coefficient of 1. A test for $\beta = 0$ in cfit2 is an easy way to test $\beta = 1$ in cfit1. Likewise checking for a horizontal line in figure 2 is a visual check for systematic differences in the calibration as a function of η. Notice that this is very similar to the calibration plot in figure **??**, with two differences. First, figure 2 is centered at zero, which is a consequence of refitting the model

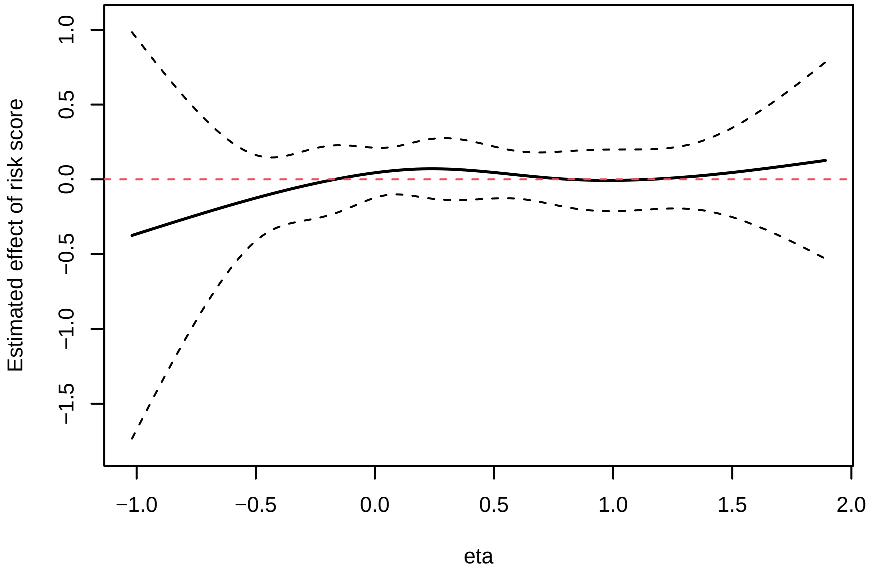


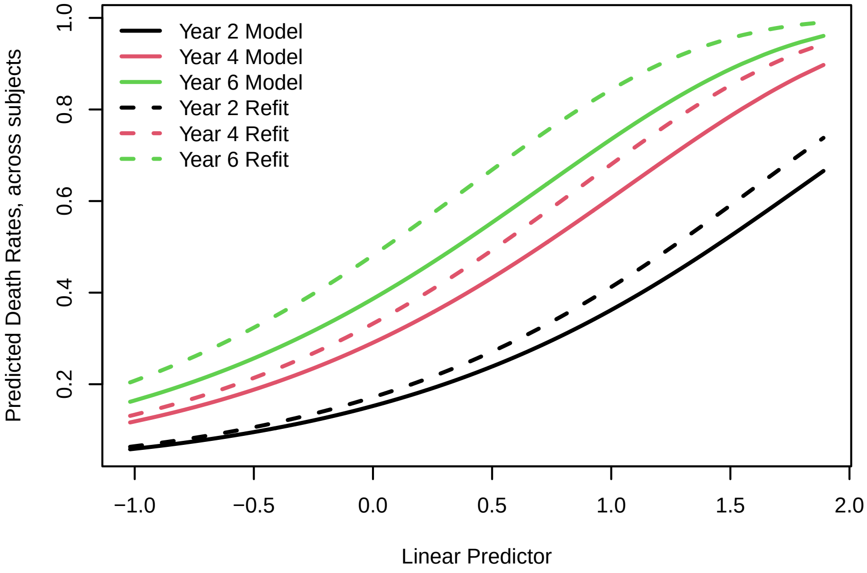
Figure 2: A check for systematic effect versus the risk score.

using coxph: O-E residuals are now guarranteed to sum to zero within the validation (these are the martingale residuals for the fit.) Because of the new baseline hazard, the figure can only assess relative change, not absolute. Second, the confidence intervals in figure **??** are a bit wider, since they also account for estimation of the intercept. (One can argue over which of these is the more useful choice.)

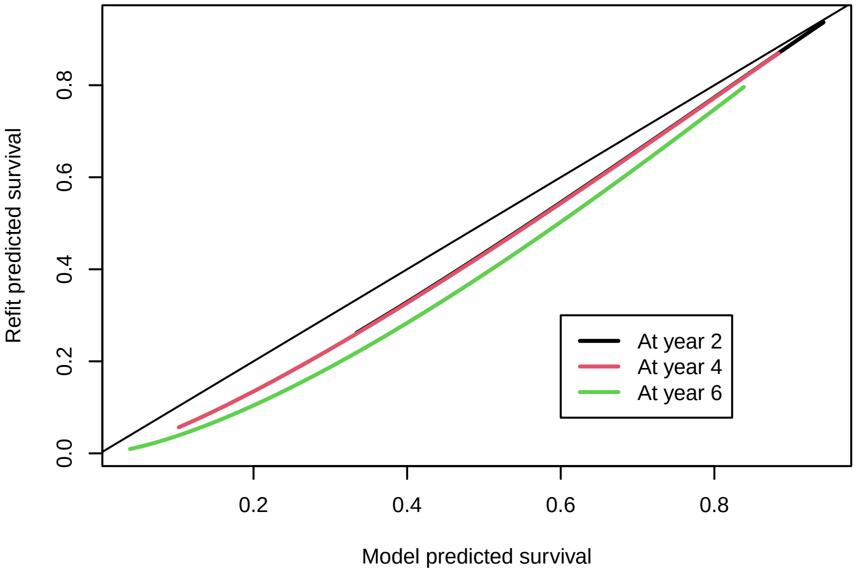
The more common plot is to compute predicted survival for both the prediction model and the refit model for a chosen time τ and range of risk scores eta, and then compare them on the same graph, what I call a $\hat{y}$ vs. $\hat{y}$ plot. An alternative but less common figure is to plot all times for a fixed risk score eta.

```
> d2 <- survfit(cfit1, newdata=gbsg2)
> indx <- order(gbsg2$eta) # need to be in eta order to draw lines
> yrs1 <-  findInterval(c(2,4,6), d2$time)
> yrs2 <-  findInterval(c(2,4,6), direct$time)
> temp1 <- t(d2$surv[yrs1,indx])   # Refit survival
> temp2 <- t(direct$surv[yrs2, indx]) # Predicted survival
> matplot(gbsg2$eta[indx], 1- cbind(temp2, temp1), type='l', lwd=2,
         lty=c(1,1,1,2,2,2), col=1:3, xlab="Linear Predictor",
         ylab="Predicted Death Rates, across subjects")
```

```
> legend("topleft", outer(paste("Year", c(2,4,6)), c("Model", "Refit"), paste),
        lty=c(1,1,1,2,2,2), col= 1:3, lwd=2, bty='n')
```



```
> matplot(temp2, temp1, type='l', lwd=2, col=1:3, lty=1,
         xlab="Model predicted survival", ylab="Refit predicted survival")
> abline(0,1)
> legend(.6, .3, c("At year 2", "At year 4", "At year 6"), lwd=2, lty=1, col=1:3)
```



The final plot above, for a single follow-up year τ, is introduced in **??**, with a couple of changes. The first is to realize that the accuracy of the "refit" survival estimate requires that the refit model be correct, in particular the proportional hazards assumption. They recommend fitting a more flexible model to address this, and we agree. A check of PH for cfit1 shows issues, for instance; we might add a time by eta interaction. (To do, show this.)

The KM comparison above is not subject to the modeling issues and gives a clear visual read of the lack of calibration. However, the model fits allow for more nuanced exploration, in particular how the calibration accuracy may change over values of the linear predictor or other covariates. A second difference is their use of the label "observed survival" on the y-axis of the above plot, which we do not agree with. The plot is a comparison of the the validation model's prediction to another estimate of survival. Given the presence of censoring this new $\hat{y}$ may be one of the best estimates possible, using as it does our best censored data tools, but it is still an estimate.

To do: Add plot of $S(t; \eta = \eta_0)$ with $\hat{S}$ from the validation model and from the refit model, 2 curves on one graph.

Age scale models Some portions of the above are unchanged by the use of age as the time scale, namely the model fits and displays of cfit1, cfit2, and cfit3 above. What does change are survival curves: we can now create predicted curves for any combination of starting age and linear predictor η, and display their results either over eta for a selected time τ, or over time for selected η. To do: show it.

7 Binomial methods

Binomial methods are the most popular, but we have left them for third. The overall approach is

- Choose a target time τ
- Use the RTTR (or IPW) approach to reassign the case weights of all those observations t_i in the validation data set which are censored before τ to other observations j with $t_j > t_i$. In the resulting data all observations with non-zero weight will have a known 0/1 status at time τ.
- Apply well known measures for binomial data to the new data, e.g., sensitivity, specificity, logistic regression, etc.

We have already encounterd this in the discrete time concordance.

Todo: fill in examples. Todo: Show that Brier score is an RTTR estimate as well. If $R^2 = 1 - numerator/denominator$, Brier= numerator and Kattan = N/D (for a binomial response).

8 Synthetic estimates

Royston and Saurbrei's D, xxx C, ... They are inapproapriate because they assume things which we want to test.

9 Summary

Binomial estimates are familiar but have higher variance (sometimes a lot).

A goal should not be to "test" for validity, but to explore in what facets the estimate does or does not perform well: covariate patterns, risk score, time ranges, metric, etc. Then focus on the specific use case.

References

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