External Validation of Survival Models

Terry Therneau

Nov 2024

This is an early draft.

1 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

Validation is a word that is often used but rarely defined, so much so that it has become essentially meaningless without further clarification. We will separate 3 meanings

- Software validation. This is discussed in the validation vignette.
- Internal validation: checks that further examine the fit of a model, using the original data set. These form an extension of the familiar trio of functional form, proportional hazards, and undue leverage which have been discussed elsewhere.
- External validation. Application of a model to new data set, one which was not used in the model's construction.

External validation tries to answer the question of whether a particular model is applicable outside of the data set it where it was developed. The most critical step, before computing metrics, is to carefully think through what we mean by "applicable". In what sense does a given model provide, or not provide, a useful prediction? Any careful answer to this will always be problem specific. One consequence is that this document will contain ideas on how to *think* about the problem and methods, but no final checklist with a "best" solution.

This key consideration is surprisingly absent from most papers on the topic of survival model validation; two notable exceptions which greatly influenced our thinking are Korn

and Simon [5] and Alman and Royston [1]. As an example from the first, suppose that we were using $t - \hat{t}$ as the metric for goodness of prediction, the difference between the observed and predicted time to death. Should an (observed, predicted) pair of (.5y, 1y) be treated as the same size error as (10.5y, 11y)? In a study of acute disease we might consider the first pair to be a much more consequential error than the latter. For example, in a cancer in which 5 year survival is considered a "cure", one might consider any pair (a, b) with both a and b greater than 6 to be no error at all. Or 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 same 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 yet another metric for success.

In general, the chosen validation metric needs to be *fit for purpose*. One of the most common outcomes of such a consideration, in our experience, will be to restrict the upper range of the assessment to a threshold τ; this could be 3 months or 10 years depending on the case at hand, and will be a consideration in each of our examples. A companion issue is that the validation data set might have very little information beyond some point; if only a handful of the validation subjects have more than 4 years of follow-up, for instance, then an assessment of predictive ability at 10 years will be a fool's errand, whether or not that were a target of interest.

In general validation 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.

A last note is recognized that validation is asymmetrical. Once a prediction model has been set, those predictions are, for our purpose, fixed. One would hope that the original creator of the prediction model checked its statistical validity, i.e., that proportional hazards holds, functional forms are correct, etc.; as that would give said creator's model a better chance of external success. Whether they did so or not is, however, immaterial to our assessment of how well a given prediction model works for for the external data set(s) at hand.

2 Survival Data

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 key 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.

A futher choice that needs to be made is whether to use a summary based on relative prediction or absolute prediction, commonly denoted as *discrimnation* and *calibration*. Discrimination measures whether the outcomes for two subjects are in the same relative order as the predicted values for that pair of subjects. In many cases this is sufficient, for instance if the prediction will be used to stratify treatment assignment of subjects in a new trial, then only the relative ordering of their risk is required. Individual patient counseling, on the other hand, will require absolute predictions for that patient. Likewise, predicted sample size for a trial with a survival endpoint depends on the eventual *number* of events that will occur, a task that also requires absolute predictions.

The validation data 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. 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 "yhat vs yhat" approach.

2. 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, IPC) 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.

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. A common example would be assessment of 5 year survival by comparing those dead before 5 to alive at 5 and simply ignoring all those censored before 5.

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 two of these will be closely tied to application at hand, the third is more technical.

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 [7] and have been widely used, both are now included in the survival package. Figure 1 shows overall survival for the two groups. The Rotterdam study has longer follow-up.

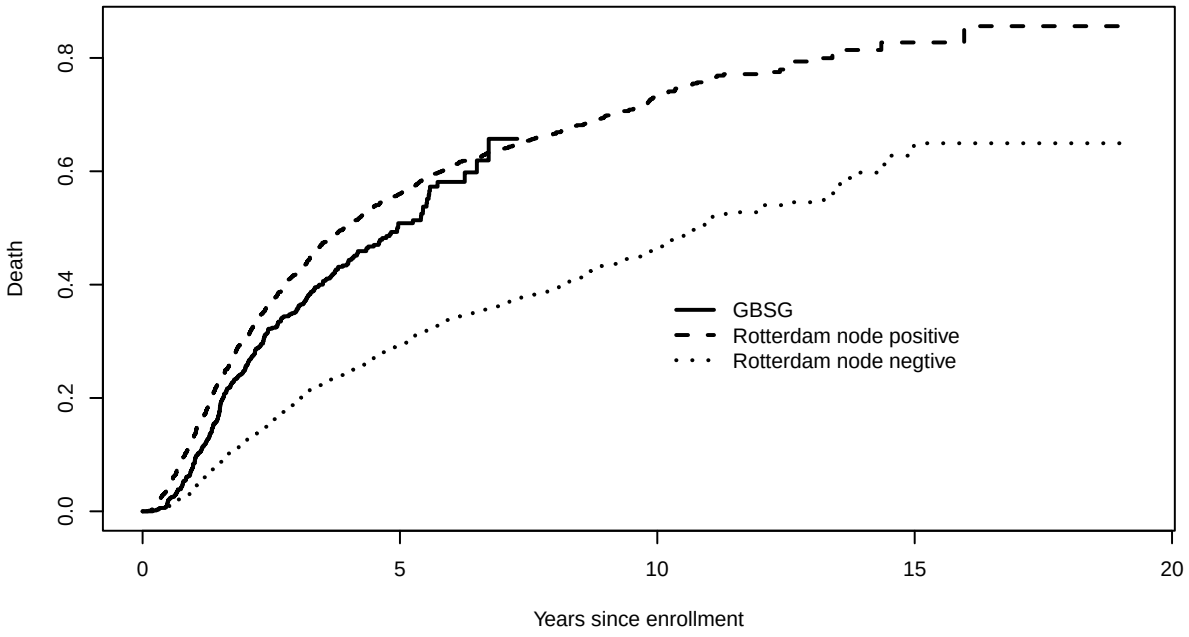


Figure 1: Rotterdam and GBSG data

```
> gsurv <- survfit(Surv(rfstime/365.25, status) ~1, gbsg)
> rott2 <- rotterdam
> rott2$rfs <- with(rott2, pmax(recur, death))
> rott2$ryear <- with(rott2, ifelse(recur==1, rtime, dtime))/365.25  # years
> rsurv <- survfit(Surv(ryear, 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')
```

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.

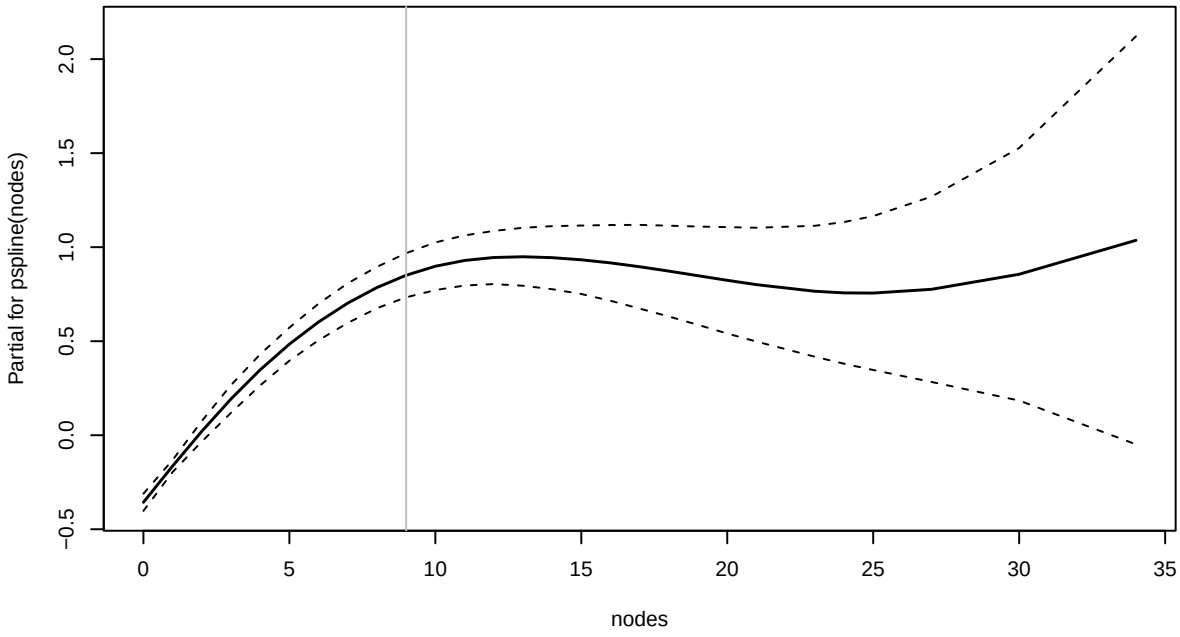
I will use 3 models: the first using only menopausal status, size, nodes, and grade, a

second adds nonlinear age, and a third using age as the time scale.

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

                        coef se(coef)   se2 Chisq DF      p
meno                   1e-01    5e-02 5e-02 4e+00  1   0.03
size20-50              3e-01    6e-02 6e-02 3e+01  1  3e-07
size>50                5e-01    8e-02 8e-02 4e+01  1  8e-10
grade                  3e-01    6e-02 6e-02 3e+01  1  6e-08
pspline(nodes), linear 7e-02    6e-03 5e-03 2e+02  1 <2e-16
pspline(nodes), nonlin                      1e+02  3 <2e-16

Iterations: 6 outer, 17 Newton-Raphson
     Theta= 1
Degrees of freedom for terms= 1 2 1 4
Likelihood ratio test=584  on 8 df, p=<2e-16
n= 2982, number of events= 1713
>
> # I dislike the color choices of termplot
> termplot2 <- function(fit, ...) termplot(fit, col.term=1, col.se=1, ...)
> termplot2(rfit, term=4, se=TRUE)
> abline(v=9, col="gray")
```



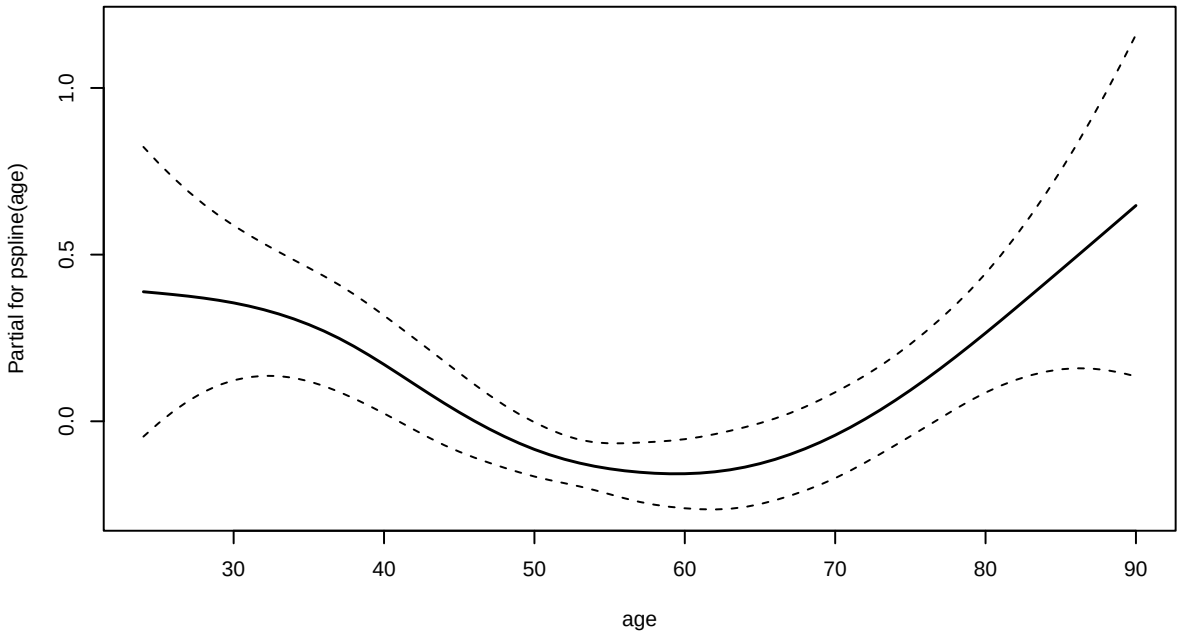
The printout shows that the nodes effect has an important non-linear component. A plot of the estimated shape is shown in figure **??**. Risk increases with the number of nodes, up to about 8-9 and then stabilizes. Recode nodes in this way.

```
> rott2$node8 <- pmin(rott2$nodes, 8)
> rfit1 <- coxph(Surv(ryear, rfs) ~  meno + size + grade+ node8, rott2)
> anova(rfit1, rfit)
Analysis of Deviance Table
 Cox model: response is  Surv(ryear, rfs)
 Model 1: ~ meno + size + grade + node8
 Model 2: ~ meno + size + grade + pspline(nodes)
  loglik  Chisq     Df Pr(>|Chi|)
1 -12522
2 -12519 5.6285 3.0835     0.1384
>
> # Now a more complex model, adding age and hormonal treatment.
> rfit2 <- update(rfit1, . ~. + pspline(age) + hormon)
```

```
> print(rfit2, digits=1)
Call:
coxph(formula = Surv(ryear, rfs) ~ meno + size + grade + node8 +
    pspline(age) + hormon, data = rott2)

                       coef se(coef)    se2  Chisq DF      p
meno                  3e-01    1e-01  1e-01  8e+00  1  0.006
size20-50             3e-01    6e-02  6e-02  2e+01  1  8e-07
size>50               4e-01    8e-02  8e-02  3e+01  1  1e-07
grade                 3e-01    6e-02  6e-02  3e+01  1  6e-08
node8                 2e-01    9e-03  9e-03  4e+02  1 <2e-16
pspline(age), linear  2e-03    3e-03  3e-03  6e-01  1  0.441
pspline(age), nonlin                         4e+01  3  6e-08
hormon               -3e-01    8e-02  8e-02  1e+01  1  1e-04

Iterations: 7 outer, 19 Newton-Raphson
     Theta= 1
Degrees of freedom for terms= 0.9 2.0 1.0 1.0 4.0 1.0
Likelihood ratio test=635  on 10 df, p=<2e-16
n= 2982, number of events= 1713
>
> termplot2(rfit2, term=5, se=T)
```



The risk appears lowest for subjects diagnosed around age 60, rising for both older and younger ages. We could go further to add progesterone receptor and estrogen receptor levels, but will stop at this point.

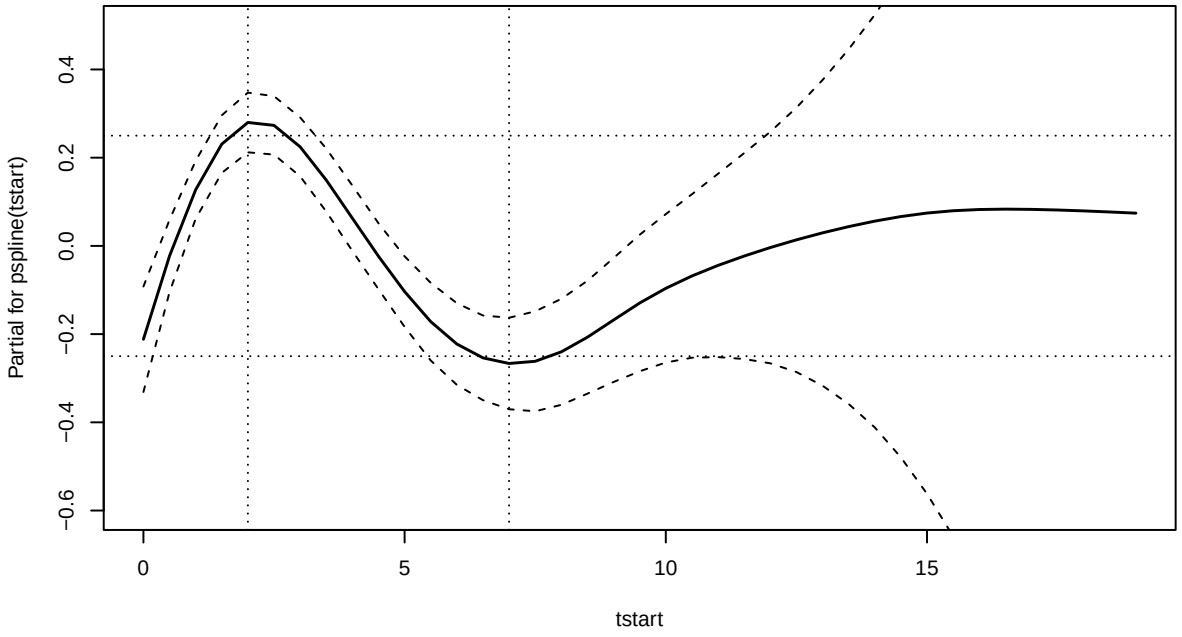
A rough index of how important each covariate is to the risk score is to compute the sd of its contribution. We see beow that nodes have by far the largest effect, followed by size, age, and grade.

```
> rmat <- predict(rfit2, type='terms')
> round(apply(rmat, 2, sd), 2)
        meno         size        grade        node8 pspline(age)
        0.14         0.16         0.14         0.47         0.15
      hormon
        0.10
```

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

since enrollment, which is by definition a time-dependent covariate: we will start with a spline, after forcing an update every .5 years. The longest followup in the Rotterdam cohort is just ovef 19 years.

```
> rott3 <- survSplit(Surv(ryear, rfs) ~ ., rott2, cut= seq(.5,19, .5))
> rott3$age1 <- rott3$age + rott3$tstart
> rott3$age2 <- rott3$age + rott3$ryear
>
> rfit3 <- coxph(Surv(age1, age2, rfs) ~ pspline(tstart) + meno + size + grade +
                   node8 + hormon, rott3)
> 2*c("age scale"= diff(rfit3$loglik), "enrollment scale"= diff(rfit2$loglik))
       age scale enrollment scale
        702.0918         634.5610
> termplot2(rfit3, term=1, se=TRUE, ylim=c(-6,5)/10)
> abline(v=c(2,7), lty=3)
> abline(h=c(-.25, .25), lty=3)
```

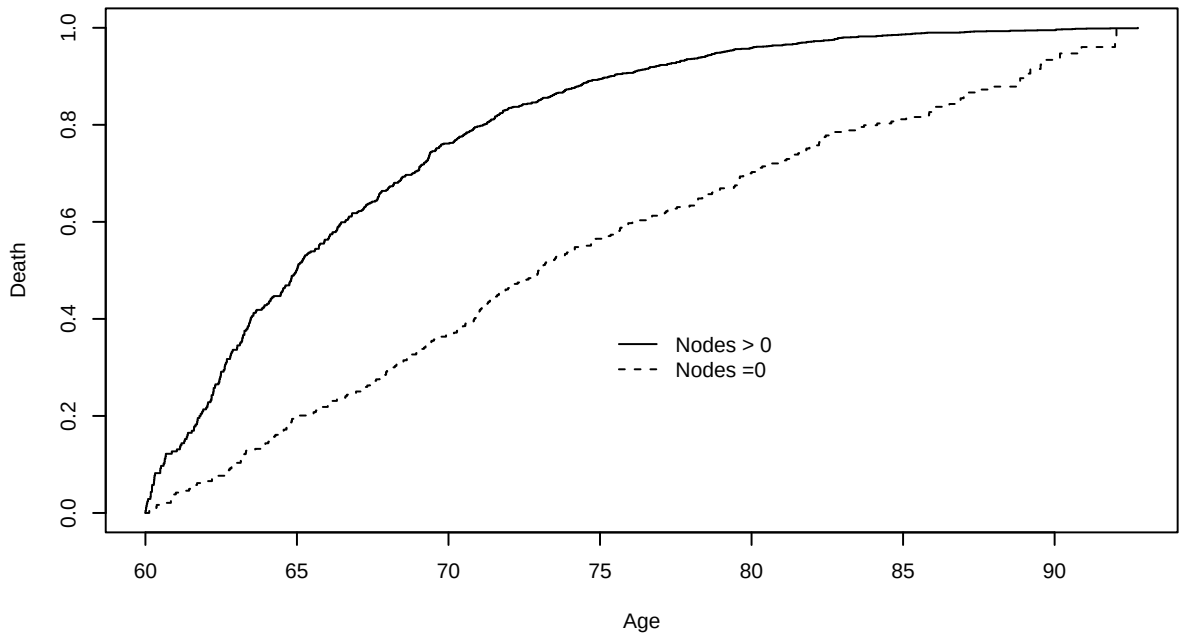


The partial likelihood for this model is actually larger than when we modeled age. The RFS rate rises by about 1.6 fold over the first 2 years after enrollment, then falls again, stabilizing after around 7 years. (The confidence intervals get crazy after 12 years so we restricted the range of the plot.)

(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.)

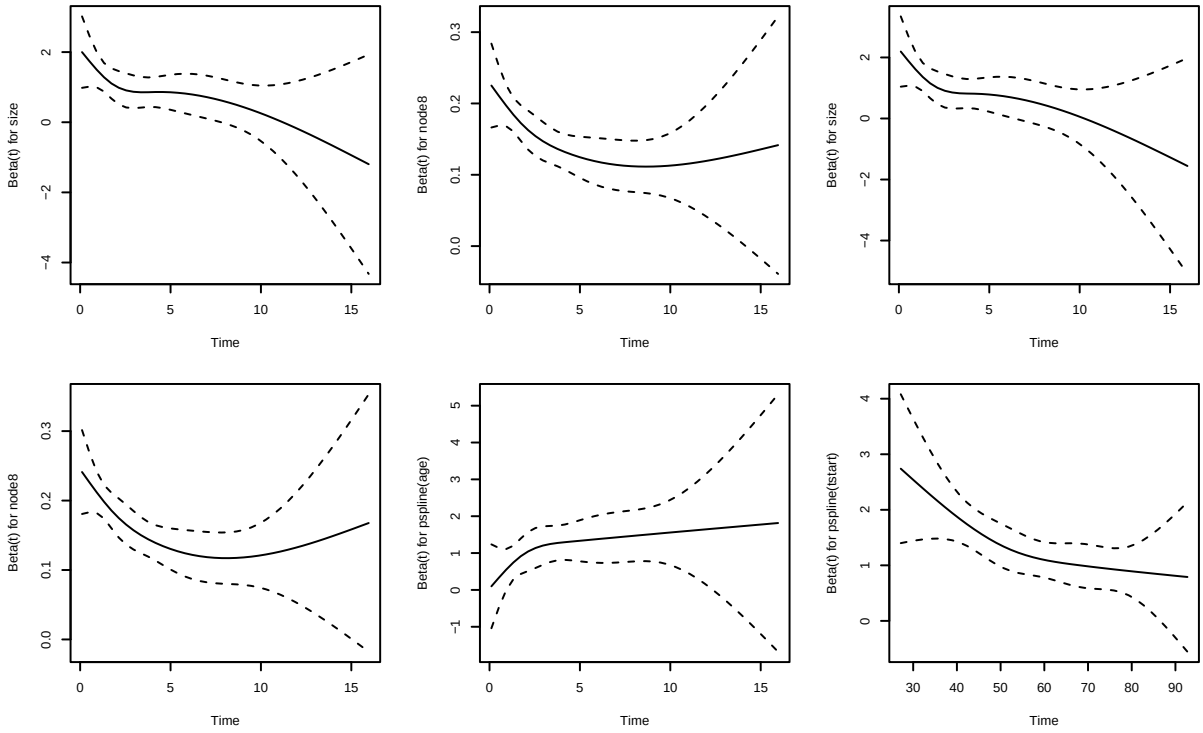
When drawing a survival curve on age scale, we also need to specify a starting point, e.g., expected survival for someone enrolled at age 60 would be

```
> surv60 <- survfit(Surv(age1, age2, rfs) ~ I(nodes==0), rott3, start.time=60)
> plot(surv60, fun='event', lty=1:2,  xlab="Age", ylab="Death")
> legend(75, .4, c("Nodes > 0", "Nodes =0"), lty=1:2, bty='n')
```



Proportional hazards All the models suffer from some failure in PH.

```
> zp1 <- cox.zph(rfit1, transform='identity') # size and nodes fail
> zp1
       chisq df       p
meno    9.75  1 0.00179
size   17.98  2 0.00012
grade   4.41  1 0.03572
node8  22.19  1 2.5e-06
GLOBAL 43.87  5 2.5e-08
> zp2 <- cox.zph(rfit2, transform='identity') # size, nodes and age
> zp2
              chisq   df       p
meno          8.946 0.92  0.0024
size         18.877 2.00 7.9e-05
grade         5.030 1.00  0.0249
node8        23.433 1.00 1.3e-06
pspline(age) 39.212 4.04 6.7e-08
hormon        0.304 1.00  0.5806
GLOBAL       80.850 9.96 3.3e-13
> zp3 <- cox.zph(rfit3, transform='identity') # time on study
> zp3
                 chisq    df       p
pspline(tstart) 39.334  4.07 6.5e-08
meno             0.845  1.00   0.358
size             2.040  2.00   0.360
grade            3.157  1.00   0.076
node8            1.610  1.00   0.204
hormon           0.163  1.00   0.686
GLOBAL          45.604 10.07 1.8e-06
>
> opar <- par(mfrow=c(2,3), mar=c(5,5,1,1))
> plot(zp1[2], resid=FALSE)
> plot(zp1[4], resid=FALSE)
> plot(zp2[2], resid=FALSE)
> plot(zp2[4], resid=FALSE)
> plot(zp2[5], resid=FALSE)
> plot(zp3[1], resid=FALSE)
```




```
> par(opar)
```

Large size or node count are particularly bad in the first years after enrollment, the age effect is less prominent in the first 2 years, and time-since-enrollment effect more prominent for younger ages.

Summary We have been aggressive in the fit, finding an "optimal" transformation for number of nodes, and allowing for a general age effect. But, since such overfitting is 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.

GBSG data 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 node8, ryear 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$status
> gbsg2$ryear <- gbsg2$rfstime/365.25
> gbsg2$node8 <- pmin(gbsg2$nodes, 8)
> gbsg2$eta1  <- predict(rfit1, newdata=gbsg2) # risk score under first model
> gbsg2$eta2  <- predict(rfit2, newdata=gbsg2) # under model 2
>
> gbsg3 <- survSplit(Surv(ryear, rfs) ~ ., gbsg2, cut= seq(.5, 9, by=.5))
> gbsg3$age1 <- with(gbsg3, age + tstart)
> gbsg3$age2 <- with(gbsg3, age + ryear)
> gbsg3$eta3 <- predict(rfit3, newdata=gbsg3) # time scale model
```

3.2 Amyloidosis

Muchtar et al [6] used data on 1007 patients with newly diagnoses light chain amyloidosis to assess the performance of 4 different predicition models that had appeared in the literature. Each of the models categorizes patients into 3-5 discrete stages of disease, and survival curves are based on those strata. The `amyloidmodel` data set contains digitized curves from the papers, and `amyloid` the survival times and per-model risk groups for each of the validation subjects.

```
> opar <- par(mfrow=c(2,2), mar=c(5,5,1,1))
> plot(survival~ month, ylim=c(0,1), lwd=2, data=amyloidmodel,
      type='l', subset= (study==2004 & stage==0))

Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> for (i in 1:2)
     lines(survival ~ month, data=amyloidmodel, subset=(study==2004 & stage==i),
           col= i+1, lwd=2)
```

```
Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> text(40, .9, "2002", cex=1.5)

Error in text.default(40, 0.9, "2002", cex = 1.5):  plot.new has not been called
yet

>
> plot(survival~ month, ylim=c(0,1), lwd=2, data=amyloidmodel,
      type='l', subset= (study==2012 & stage==0))

Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> for (i in 1:3)
     lines(survival ~ month, data=amyloidmodel, subset=(study==2012 & stage==i),
           col= i+1, lwd=2)

Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> text(40, .9, "2012", cex=1.5)

Error in text.default(40, 0.9, "2012", cex = 1.5):  plot.new has not been called
yet

>
> plot(survival~ month, ylim=c(0,1), lwd=2, data=amyloidmodel, col=4,
      type='l', subset= (study==2015 & stage==3))

Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> for (i in 0:2)
     lines(survival ~ month, data=amyloidmodel, subset=(study==2015 & stage==i),
           col= i+1, lwd=2)

Error in eval(m$data, eframe):  object 'amyloidmodel' not found

> text(40, .9, "2015", cex=1.5)

Error in text.default(40, 0.9, "2015", cex = 1.5):  plot.new has not been called
yet

>
> afit <- survfit(Surv(month, status) ~ number.organs, amyloid)
```

```
Error in eval(expr, p):  object 'amyloid' not found

> plot(afit, lty=1:4, col=1:4, lwd=2, xlab="month", ylab="survival")

Error in eval(expr, envir, enclos):  object 'afit' not found

> text(90, .9, "Validation\n2003 - 15", cex=1.5)

Error in text.default(90, 0.9, "Validation\n2003 - 15", cex = 1.5):  plot.new
has not been called yet

> par(opar)
```

This data is different in that a survival model does not directly appear, only the predicted category for each subject. Cox models were used in some of the work to help motivate the categorical staging variables and their cutoffs, but is not used in the final score. (The hematologists' firmly believe that physicians "will not understand" a continuous score.)

The 2002, 2012 and 2015 predictions differ substantially, particularly for the lowest risk goup (there are over 50 subjects and 0 deaths in the top 2015 curve). This represents an era where several new and effective treatments for amyloidosis became available. Any validation exercise will need to be cognisant of that.

3.3 Treatment trial in Primary Biliary Cirrhosis

The model and data underlying a risk score and predicted survival representing the untreated progression of primary biliary cirrhosis is well known; the `pbc` data has been part of the survival package for over 2 decades.

```
> pbc2 <- subset(pbc, id <= 312)
> pbc2$death <- 1*(pbc2$status ==2)
> pbcfit1 <- coxph(Surv(time, death) ~ age + edema + log(bili) + log(albumin) +
                     log(protime), pbc2, ties="breslow")
> pbc2$riskscore <- model.matrix(pbcfit1) %*% coef(pbcfit1)
> # or pbcfit1$linear.predictor + sum(pbcfit1$means * pbcfit1$coef)
> # For expected, matching the calculation used in the ucda data is important
> pbcfit2 <- coxph(Surv(time, death) ~ riskscore, pbc2, ties="breslow")
```

The `udca` data set contains early results of a trial of usrodoxycholic acid (UDCA) versus placebo for treatment of the disease; "early" in the sense that 3 years' enrollment plus 5

years of follow-up had produced only 16 deaths. Estimating the coefficients for the 5 risk score variables, de novo within this study, while also getting an estimate of treament effect is simply not feasable. However, we can use the PBC fit as a direct estimate of the expected death rate. The `udca` data include the Mayo risk score as computed on the 312 subset, as above.

```
> udca2 <- subset(udca, , c(id, trt, bili, riskscore))
> udca2$time <- with(udca, as.numeric(last.dt- entry.dt))
> udca2$death <- 1*(!is.na(udca$death.dt))
> pdate <- with(udca, pmin(death.dt, tx.dt, hprogress.dt, varices.dt,
                           enceph.dt, ascites.dt, last.dt, na.rm=TRUE))
> temp <- with(udca, !is.na(death.dt) | !is.na(tx.dt) | !is.na(hprogress.dt) |
                   !is.na(varices.dt) | !is.na(enceph.dt) | !is.na(ascites.dt) )
>
> udca2$ptime <- with(udca, as.numeric(pdate - entry.dt))
> udca2$pstat <- ifelse (temp, 1, 0)
> udca2$eta   <- predict(pbcfit2, newdata=udca2)
> udca2$expect<- predict(pbcfit2, newdata=udca2, type="expected")
> udca2 <- subset(udca2, !is.na(riskscore))
> pcount <- with(udca, 1*is.na(death.dt) + 1*is.na(tx.dt) + 1*is.na(hprogress.dt)
               + 1*is.na(varices.dt) + 1*is.na(enceph.dt) + 1*is.na(ascites.dt))
```

3.4 Monoclonal gammopathy

Early enrollment predicting late enrollment. Semi-competing risks. Yet to be done.

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) Yet to be done

4 Initial analysis

A few checks of the validation data should be done early in the process. These include a Kaplan-Meier plot of the outcome, overall or by subgroups, numerical and/or graphical summaries of each of the predictor variables found in the reference model, and a refit of

the reference model using the new data. The goal of this is not validation per se, but to uncover simple data issues such as a lab test in different units, a different range of subjects, or a change in time scale.

5 Concordance

The most common measure of discrimination for survival data is the concordance C. For X and Y from a continuous bivariate distribution it is defined as

$$\begin{aligned} C &= P(X_i > X_j | Y_i > Y_j) \\ &= P(Y_i > Y_j | X_i > X_j) \end{aligned}$$

The two forms are numerically equivalent, just with a different viewpoint. Image that a local newpaper has predicted the outcome of the mayor's race for 20 years. We could take a forward looking view and ask how often the voter's followed what the paper predicted (y given x), or a backwards in time view and ask how often the paper was correct (x given y). In either case the estimate is (total correct)/(total contests). 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.

One numerical wrinkle has to do with ties. There have been various suggestions, which the concordance vignette lays out in detail. Here we only summarize that discussion.

- For C count ties in the response time t as uninformative, such pairs count as neither concordant or discordant. Ties in the predictor $\hat{y}$ count as 1/2.
- For survival data, pairs such as (5+, 7) are also treated as uninformative. We do not know if subject 1 (censored at 5) will or will not have a survival time greater than subject 2 (event at 7).
- If the response is a 0/1 variable, then C = AUROC, the area under the receiver operating curve, a metric which is well established for binary outcomes. (Proving this simple theorem is harder than it looks, but the result is well known.)

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 a score of 1/2 for ties.

As a measure of association the concordance has two 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. (Multistate models, as we will see later, are more complex.)

As shown in the concordance vignette, the concordance can also be written as

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

where δ_i is 0 for censored and 1 for uncensored observations, $r_i(t)$ is the percentile rank of $\hat{y}_i$ among all those still at risk at time t_i, and $w(t)$ is a weight function. 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 and alternate time scales. Like a Cox model, at each event time all that matters is who is at risk at that time, and their covariate values at that time.

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 Uno, 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 may be true for weighted versions of the concordance, and we suspect that the debate about Harrell vs. Uno weights will also fade with time. (Note that for uncensored data, $G = 1$ and all the above weights are identical.)

5.1 Dichomized concordance

A variation that is important is the dichotomized concordance: replace t_i with $I(t_i <= \tau)$ for a chosen cutpoint τ, i.e., a 0/1 indicator of death at or before time τ. Then

- The redistribute-to-the-right (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" by [**?**]. We consider this label a poor choice, however, because it invites confusion with time-dependent covariates, and will henceforth refer to this approach as a "dichotomized concordance" or $DC(\tau)$.

The DC and C statistics measure different things. DC 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, DC only has the subset which are close to τ.

Another measure based on time dichotomy is the ordinary R^2 using the weighted RTTR data

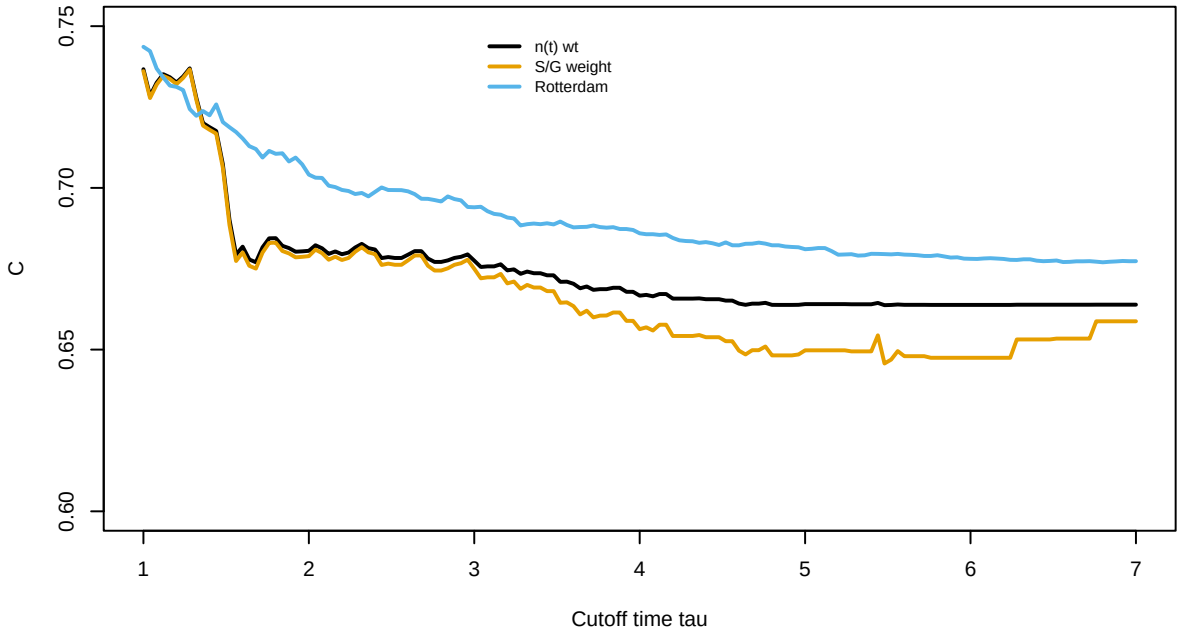
$$\begin{aligned} R^2 &= 1 - \frac{\sum_{i=1}^n w_i(\tau)\left[\delta_i - \hat{p}_i(\tau)\right]^2}{\sum_{i=1}^n w_i(\tau)\left[\delta_i - p_0(\tau)\right]^2} \\ &= 1 - N/D \end{aligned}$$

where δ is the 0/1 status indicator for observed death before τ, w are the IPC weights, $\hat{p}$ is the predicted probability of death based on the model, and p_0 is the result of fitting a null model with only an intercept. The null model predicted probability of an event at time τ is simply 1- Kaplan-Meier(τ): one of the original arguments for the RTTR was that a weighted mean of the 0/1 status variable reprises the KM [3]. That is, a weighted logistic regression of δ using only the intercept as a predictor will reprise the KM. The numerator N is known as the Brier score. van Houwelingen and Putter [8] argue that N is difficult to interpret without rescaling, since the error of the null model is very different for p near .5 or near .9, and suggest xxx. The ratio N/D is called the index of prediction accuracy in [4].

5.2 Breast cancer

Figure **??** shows the C statistic with $n(t)$ (Gehan-Wilcoxon or Harrell) 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.

```
> tau2 <- seq(1, 7, length=151)
> reweight <- rttright(Surv(ryear, rfs) ~ 1, gbsg2, times=tau2)
>
> Cstat <- array(0, dim=c(151,2,5)) # value and std, 4 measures
> for (i in 1:151) {
     c1 <- concordance(rfit1, newdata=gbsg2, ymax=tau2[i])
     c2 <- concordance(rfit1, newdata=gbsg2, ymax=tau2[i], timewt="S/G")
     ytau <- with(gbsg2,ifelse(rfs==1 & ryear <= tau2[i], 1, 0))
     c3 <- concordance(ytau ~ eta1, data=gbsg2, weight=reweight[,i],
                       subset=(reweight[,i] > 0))
     c4 <- concordance(rfit1, ymax=tau2[i])
     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)))
 }
> bfit <- brier(rfit1, newdata=gbsg2, times=tau2)
> Cstat[,1,5] <- (1+ sqrt(bfit$rsquare))/2
>
> matplot(tau2, Cstat[,1, c(1,2,4)], lwd=2, lty=1, col=1:5, type='l',
         ylim=c(.6, .75),
         xlab="Cutoff time tau", ylab="C")
> legend(3, .75, c("n(t) wt", "S/G weight", "Rotterdam"),
        lwd=2, lty=1, col=1:5, bty='n', cex=.8)
```



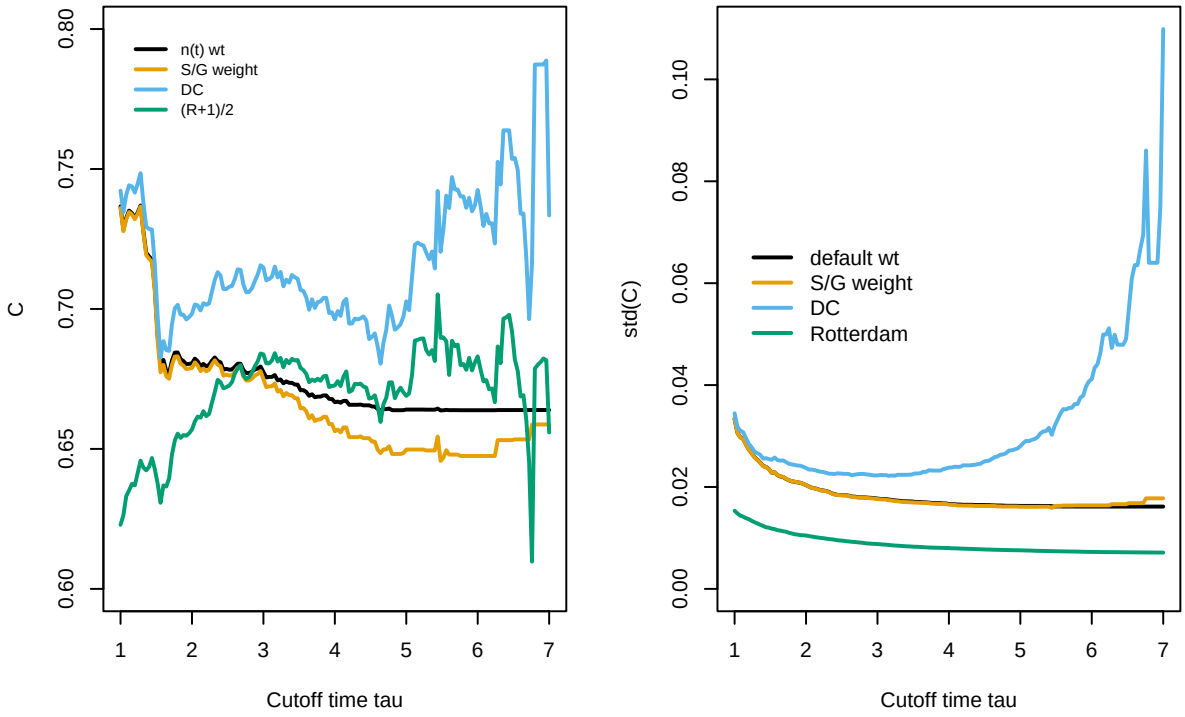
The plot is very interesting.

- 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, from year 3 forward they differ by .01 to .02. At 4 years the values are .70 and .69, less than 1 std apart. (The blip of better performance around 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.

Now add in the dichotmized measures. For R-squared, we note that $|R| = 0$ for no association but $C = 1/2$, plotting $(|R| + 1)/2$ aligns the two.



As expected the dichotomized concordance is larger. The amount variation associated with small changes in τ is a consequece of the sharp transitions in weight from > 1 to 0 when a censored observation is reassigned. Why is the DC variance grow so rapicly? As τ increases the RTTR algorithm has set more and more observations' weight to zero: by time 6 only 33/387 remain. 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.

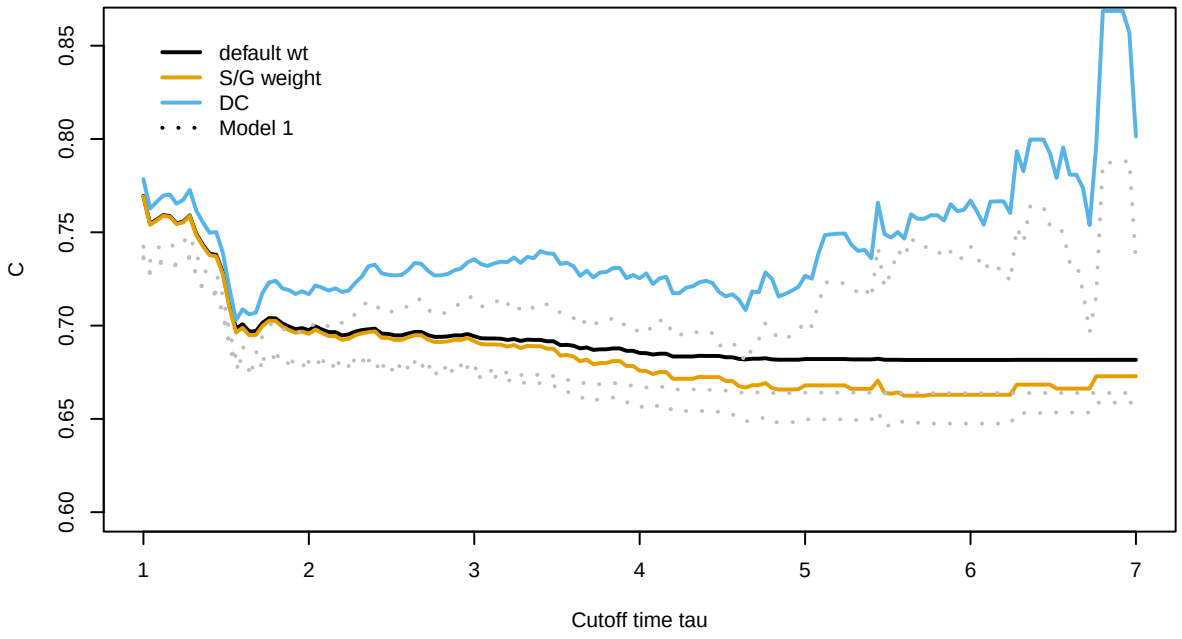
The IJ variance used by the `concordance` function also allows computation of a variance for the difference between any two of the methods.

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

Now do the same for model 2.

```
> C2 <- Cstat
> for (i in 1:151) {
     c1 <- concordance(rfit2, newdata=gbsg2, ymax=tau2[i])
     c2 <- concordance(rfit2, newdata=gbsg2, ymax=tau2[i], timewt="S/G")
```

```
    temp <- with(gbsg2,ifelse(rfs==1 & ryear <= tau2[i], 1, 0))
    c3 <- concordance(temp~ eta2, data=gbsg2, weight=reweight[,i],
                      subset=(reweight[,i] > 0))
    c4 <- concordance(rfit2, ymax=tau2[i]) # concordance of the training data
    C2[i,,1] <- c(coef(c1), sqrt(vcov(c1)))
    C2[i,,2] <- c(coef(c2), sqrt(vcov(c2)))
    C2[i,,3] <- c(coef(c3), sqrt(vcov(c3)))
    C2[i,,4] <- c(coef(c4), sqrt(vcov(c4)))
 }
> matplot(tau2, C2[,1,1:3], lwd=2, lty=1, col=1:4, type='l', ylim=c(.6, .86),
         xlab="Cutoff time tau", ylab="C")
> legend(1, .86, c("default wt", "S/G weight", "DC", "Model 1"),
        lwd=2, lty=c(1,1,1,3), col=1:3, bty='n')
> matlines(tau2, Cstat[,1,1:3], lwd=2, lty=3, col='gray')
```



The concordance lines for model 1 have been added in gray. We see that model 2 has essentially the same patterns, but higher discrimination: about .02 units at the 5 year

mark. The plots doe 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 an increase in the standard error.

```
> temp <- list(
             c1a = concordance(rfit1),
             c1b = concordance(rfit1, newdata=gbsg2),
             c2a = concordance(rfit2),
             c2b = concordance(rfit2, newdata=gbsg2),
             c3a = concordance(rfit3),
             c3b = concordance(rfit3, newdata=gbsg3))
> temp2 <- unlist(sapply(temp,  function(x) c(coef(x), sqrt(vcov(x)))))
> temp3 <- matrix(temp2, ncol=3,
        dimnames=list(c("Original C", "Original std", "GBSG C", "GBSG std"),
                      c("Model 1", "Model 2", "Model 3")))
> round(temp3, 4)
             Model 1 Model 2 Model 3
Original C    0.6728  0.6833  0.6786
Original std  0.0068  0.0067  0.0072
GBSG C        0.6639  0.6816  0.6768
GBSG std      0.0162  0.0154  0.0173
```

What is not simple for the age scale is a time cutoff; it is simply not clear what such a cutoff should be. Translation a time dependent fit to the DTC to age scale is even more 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.

5.3 Amyloid

Compare 5 different risk scores, of which a simple count of the number of affected organs is the most crude. We see, that as expected, it has the lowest concordance with outcome,

and the 2012, 2013 and 2015 modifications of the 2004 score do seem to give a slight improvemet.

```
> cfit <- concordance(Surv(month, status) ~ number.organs + r2004 + r2012 +
                          r2013 + r2015, amyloid, reverse=TRUE)
> cfit
Call:
concordance.formula(object = Surv(month, status) ~ number.organs +
    r2004 + r2012 + r2013 + r2015, data = amyloid, reverse = TRUE)

n= 1005
              concordance     se
number.organs      0.5907 0.0118
r2004              0.6941 0.0096
r2012              0.7070 0.0102
r2013              0.7206 0.0099
r2015              0.7164 0.0099

              concordant discordant tied.x tied.y tied.xy
number.organs     166888      97527 118072    126      65
r2004             196421      47936 138130     72     119
r2012             223039      64708  94740    105      86
r2013             233200      64411  84876    150      41
r2015             226046      60539  95902    131      60
```

The `reverse = TRUE` argument is needed because a high score is expected to lead to shorter survival. A second question is whether any of these are a significant improvement. We can use the variance-covariance matrix of the values to address this.

```
> contrast <- function(cmat, fit) {
     beta <- coef(fit)
     vmat <- vcov(fit)
     test <- cmat %*% beta
     vtest <- diag(cmat %*% vmat %*% t(cmat))
     cbind(contrast= c(test), sd= sqrt(vtest), z= c(test/sqrt(vtest)))
 }
>
> contrast(rbind(c1.2= c(-1,1,0,0,0), c1.3= c(-1,0,1,0,0), c1.4= c(-1,0,0,1,0),
```

```
               c1.5= c(-1,0,0,0,1), c2.3= c(0,1,-1,0,0), c2.4= c(0,-1,0,1,0),
               c2.5= c(0,-1,0,0,1), c4.5=c(0,0,0,-1,1)), cfit)
         contrast          sd         z
c1.2  0.103433581 0.013576773  7.618422
c1.3  0.116304606 0.014601834  7.965068
c1.4  0.129975659 0.014096293  9.220556
c1.5  0.125685317 0.013951311  9.008854
c2.3 -0.012871026 0.007294606 -1.764458
c2.4  0.026542079 0.003549353  7.478004
c2.5  0.022251737 0.003342559  6.657095
c4.5 -0.004290342 0.002395480 -1.791016
```

It appears that all are significantly different from the simple count, and that the final two improve on the 2004 score.

5.4 UDCA treatment

For this data the concordance is not of primary interest.

```
> concordance(Surv(time, death) ~ riskscore, udca2, reverse=TRUE)
Call:
concordance.formula(object = Surv(time, death) ~ riskscore, data = udca2,
    reverse = TRUE)

n= 169
Concordance= 0.8454 se= 0.04749
concordant discordant     tied.x     tied.y    tied.xy
      1649        288         33          1          0
> concordance(Surv(ptime, pstat) ~ riskscore, udca2, reverse=TRUE)
Call:
concordance.formula(object = Surv(ptime, pstat) ~ riskscore,
    data = udca2, reverse = TRUE)

n= 169
Concordance= 0.6611 se= 0.03347
concordant discordant     tied.x     tied.y    tied.xy
      4776       2405        176          4          0
```

6 Expected number of events

One of the more flexible, but unfortunately less known validation 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 [2] 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, which provide both estimates and their standard errors. The compensator $\Lambda_i(t_i)$ is the predicted number of events for each subject, which is obtained from a coxph model using the predict function. Below we show simple counts, then formal confidence intervals.

```
> gbsg2$expect1 <- predict(rfit1, type='expected', newdata=gbsg2)
> gbsg2$expect2 <- predict(rfit2, type='expected', newdata=gbsg2)
> gbsg3$expect3 <- predict(rfit3, type='expected', newdata=gbsg3)
>
> temp <- c(with(gbsg2, c(sum(rfs), sum(expect1), sum(expect2))),
           sum(gbsg3$expect3))
> temp2 <- rbind(rep(temp[1], 3), temp[2:4])
> temp2 <- rbind(temp2, temp2[1,]/temp2[2,])
> dimnames(temp2) <- list(c("Observed", "Expected", "O/E"), paste("Model", 1:3))
> round(temp2, 2)
         Model 1 Model 2 Model 3
Observed  299.00  299.00  299.00
Expected  269.46  244.70  246.22
O/E         1.11    1.22    1.21
>
>
> gfit1 <- glm(rfs ~ offset(log(expect1)), poisson, gbsg2, subset=(expect1 > 0))
> gfit2 <- glm(rfs ~ offset(log(expect2)), poisson, gbsg2, subset=(expect2 > 0))
> gfit3 <- glm(rfs ~ offset(log(expect3)), poisson, gbsg3, subset=(expect3 > 0))
>
> tfun <- function(fit) coef(fit) + c(0, -1.96, 1.96)*c(sqrt(vcov(fit)))
> temp3 <- cbind(tfun(gfit1), tfun(gfit2), tfun(gfit3))
> dimnames(temp3) <- list(c("SMR", "lower CI", "upper CI"), paste("Model", 1:3))
> round(exp(temp3), 2)
```

```
          Model 1 Model 2 Model 3
SMR          1.11    1.22    1.09
lower CI     0.99    1.09    0.97
upper CI     1.24    1.37    1.23
```

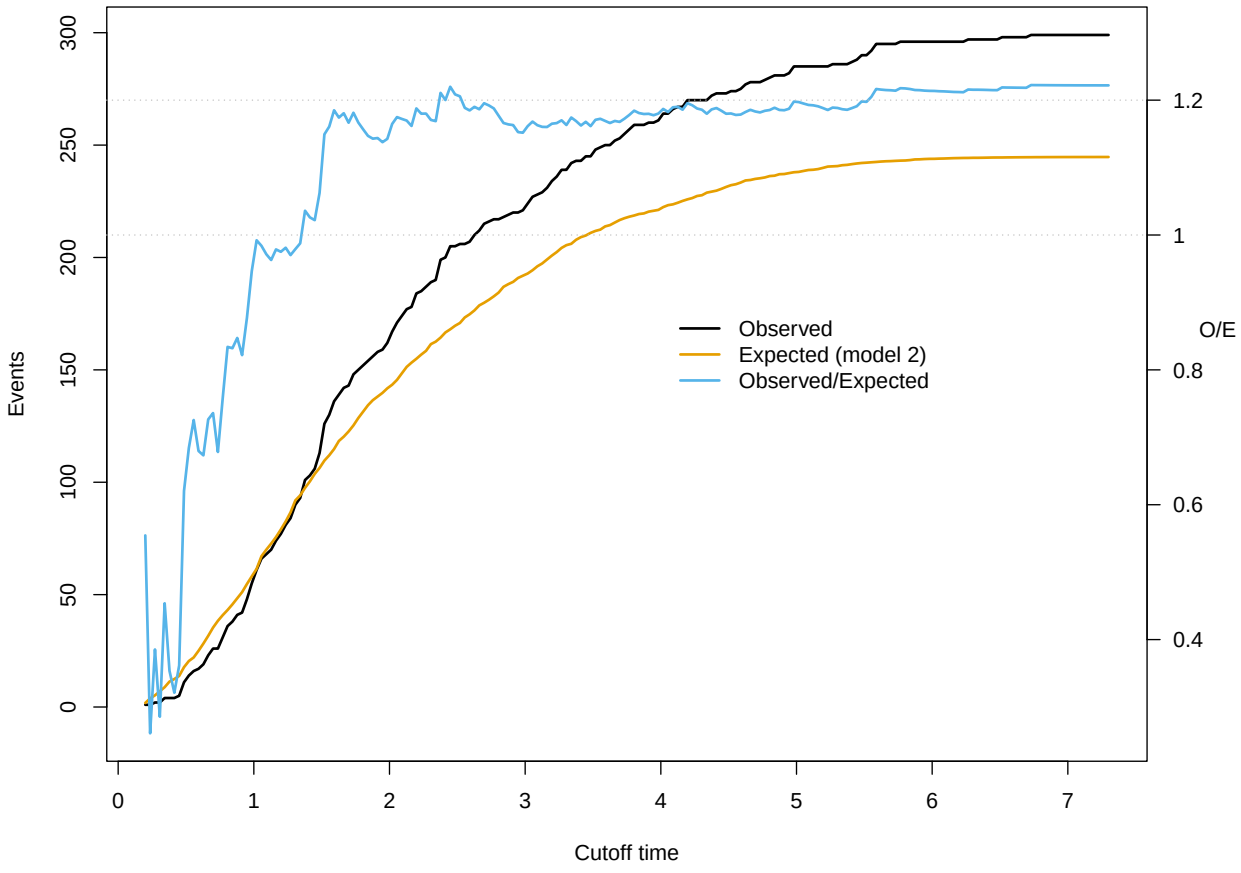
Over the full follow-up time, the gbsg data has 11%, 22%, and 9% more deaths than predicted by models 1–3, given the covariates. This is known as mean calibration. This is also a standardized mortality ratio (SMR), though that term is more common when the predicted death rate is based on national life 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$ryear <- pmin(gbsg2$ryear, tau[i])
     tdata$rfs <- ifelse(gbsg2$ryear>tau[i], 0, gbsg2$rfs)
     pp <- predict(rfit2, newdata=tdata, type='expect')
     oe[i,] <- c(sum(tdata$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="Events")
> 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 (model 2)", "Observed/Expected"),
```

```
        lty=1, lwd=2, col=1:3, bty='n')
> abline(h= tfun(c(1,1.2)), col='gray', lty=3)
```



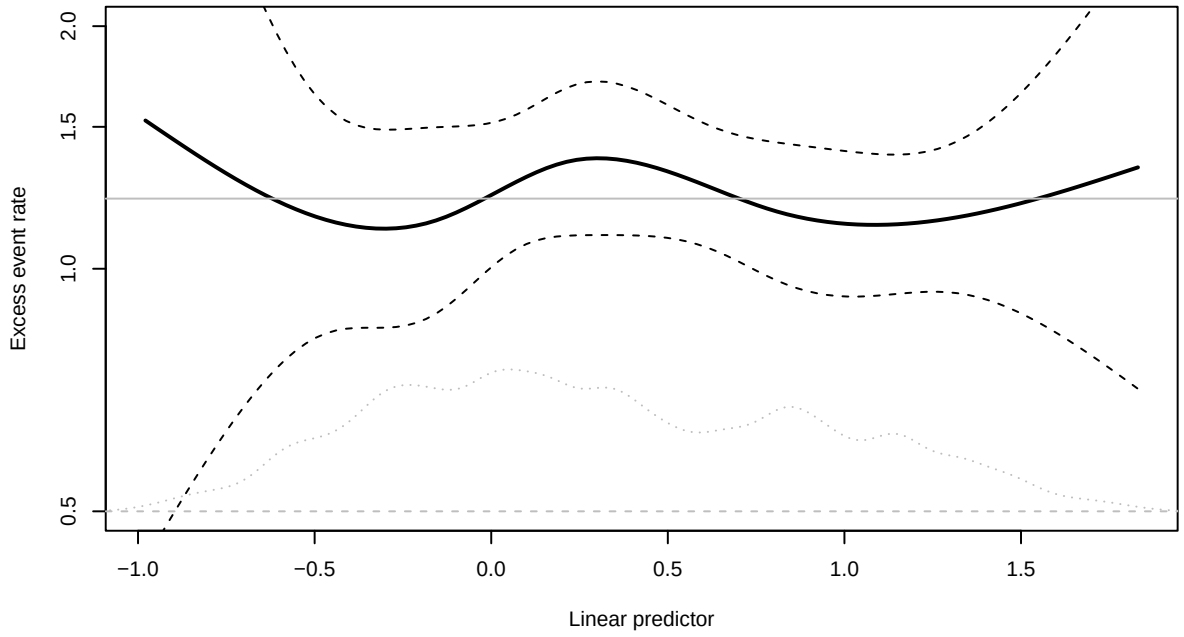

```
> par(opar)
```

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, first 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. In order to facilitate doing validation over the range of 0–5 years, add 3 more variables to the gbsg2 dataset: ryear5, rfs5, and expect5.

```
> tdata <- gbsg2
> tdata$rfs <- ifelse(tdata$ryear >5, 0, tdata$rfs)
> tdata$ryear <- pmin(tdata$ryear, 5)
> gbsg2$ryear5 <- tdata$ryear
> gbsg2$rfs5 <- tdata$rfs
> gbsg2$expect5 <- predict(rfit2, 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 1.20
(Intercept)
   1.197491
```

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 an ideal of 1.0, which would represent a perfect match to the predicted events. We have added ahorizontal lines 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 $\eta < -.5$ but only 12 events.

```
> gfit2b <- update(gfit2, . ~ . + nsp(eta2, 4))
> xx <- seq(min(gbsg2$eta2), max(gbsg2$eta2), length=100)
> yhat <- predict(gfit2b, newdata= data.frame(eta2=xx, expect2=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=exp(coef(gfit2)), col='gray', lty=1)
> #rug(gbsg2$eta)
> temp <- density(gbsg2$eta2, 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='gray')
> abline(h=.5, lty=2, col='gray')
```

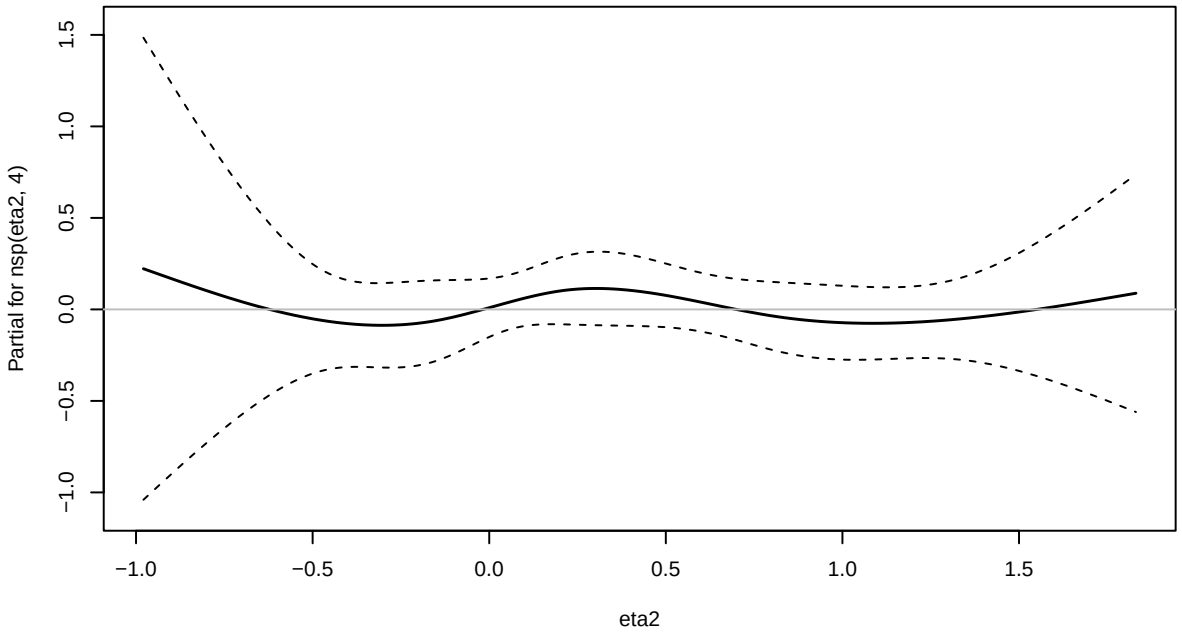


We can also divide the predicted risk score `eta` into bins to create a tableau reminiscent of the Hosmer-Lemishow approach in binomial data. In this case we lack any formal justfication for treating the last row as components of a chi-square statistic.

```
> bins <- with(gbsg2, cut(eta2, quantile(eta2, 0:5/5), include.lowest=TRUE,
                          labels= paste0("Q",1:5)))
> counts <- rbind(observed= tapply(gbsg2$rfs, bins, sum),
                  expected= tapply(gbsg2$expect2, 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  31.00 48.00 59.00 67.00 94.00
expected  27.76 36.28 44.10 61.85 74.71
O/E        1.12  1.32  1.34  1.08  1.26
(O-E)^2/E  0.38  3.79  5.03  0.43  4.98
```

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

```
> termplot2(gfit2b, se=TRUE)
> abline(0,0, col='gray')
```



A next step is 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. (Model rfit2 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.

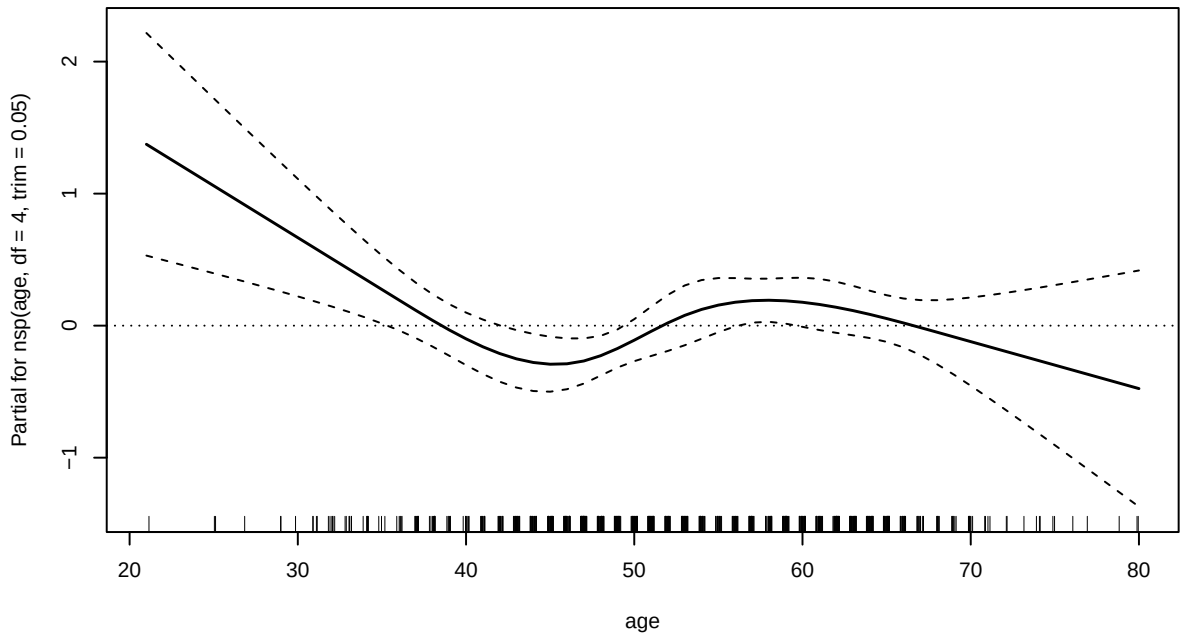
```
> test2a <- update(gfit2, . ~. +  meno + grade + hormon)
> pcoef <- function(x) printCoefmat(summary(x)$coefficients, digits=2)
> pcoef(test2a)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.22544    0.24416     0.9      0.4
meno        -0.00084    0.12080     0.0      1.0
grade       -0.00204    0.10070     0.0      1.0
hormon      -0.06237    0.12777    -0.5      0.6
>
> test2b <- update(gfit2, . ~. + factor(grade) -1)
> pcoef(test2b)
               Estimate Std. Error z value Pr(>|z|)
factor(grade)1    -0.14       0.24    -0.6      0.5
factor(grade)2     0.28       0.07     3.9    9e-05
factor(grade)3     0.11       0.11     1.0      0.3
> with(gbsg2, table(grade))    # nicer label than table(gbsg2$grade)
grade
  1   2   3
 81 444 161
> anova(gfit2, test2b, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect2))
Model 2: rfs ~ factor(grade) + offset(log(expect2)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     810.09
2       676     806.02  2   4.0712   0.1306
>
```

```
> test2c <- update(gfit2, . ~ . + size -1)
> printCoefmat(summary(test2c)$coef, digits=2)
          Estimate Std. Error z value Pr(>|z|)
size<=20     0.266      0.124     2.1    0.032
size20-50    0.201      0.070     2.9    0.004
size>50      0.068      0.183     0.4    0.711
> with(gbsg2, table(size))
size
 <=20 20-50   >50
  180   453    53
> anova(gfit2, test2c, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect2))
Model 2: rfs ~ size + offset(log(expect2)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     810.09
2       676     809.26  2  0.82629   0.6616
>
> 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
> test2d <- update(gfit2, . ~ . + ngrp -1)
> pcoef(test2d)
        Estimate Std. Error z value Pr(>|z|)
ngrp1      0.341      0.130     2.6    0.009
ngrp2-3    0.029      0.129     0.2    0.824
ngrp4-7    0.236      0.108     2.2    0.029
ngrp8+     0.206      0.103     2.0    0.045
> round(exp(coef(test2d)), 2)
  ngrp1 ngrp2-3 ngrp4-7  ngrp8+
   1.41    1.03    1.27    1.23
> anova(gfit2, test2d, test="Chisq")
Analysis of Deviance Table
```

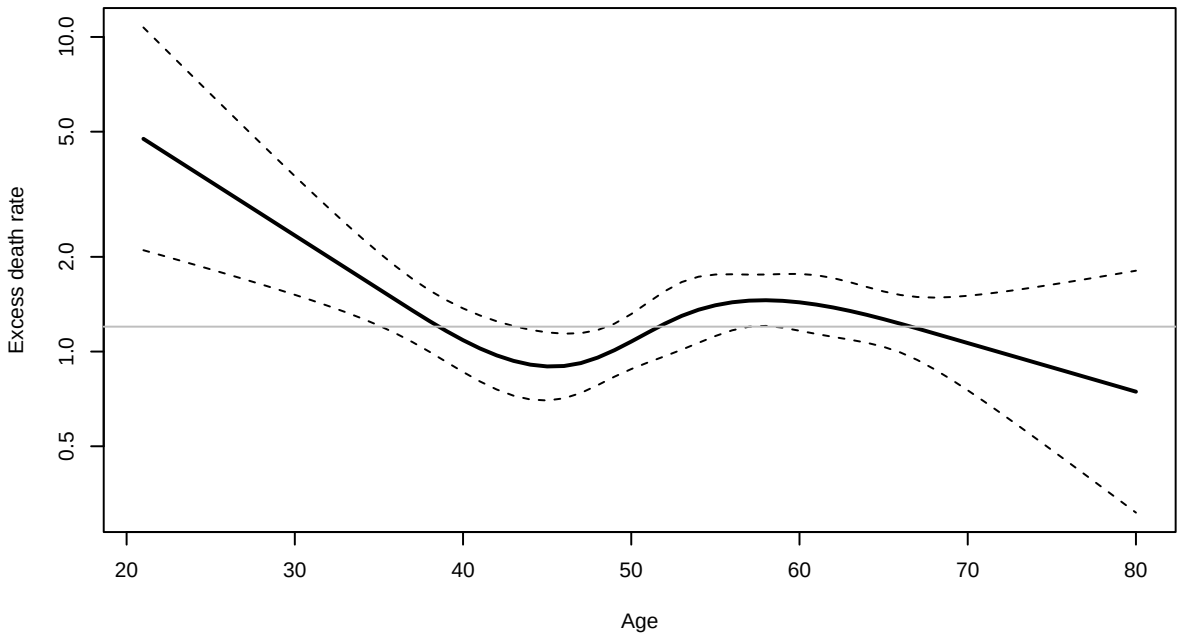
```
Model 1: rfs ~ offset(log(expect2))
Model 2: rfs ~ ngrp + offset(log(expect2)) - 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     810.09
2       675     806.98  3   3.1052   0.3757
>
> # move boundary knots in from the edge a bit, which I prefer
> test2e <- update(gfit2, . ~ . + nsp(age, df=4, trim=.05))
> termplot2(test2e, se=TRUE)
> abline(0,0, lty=3)
> rug(jitter(gbsg2$age))
```




```
> anova(gfit2, test2e, test="Chisq")
Analysis of Deviance Table

Model 1: rfs ~ offset(log(expect2))
Model 2: rfs ~ nsp(age, df = 4, trim = 0.05) + offset(log(expect2))
```

```
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       678     810.09
2       674     796.41  4   13.675 0.008408
>
> # redraw this directly on the excess risk scale
> dummy <- data.frame(age=21:80, expect2=1)
> yhat <- predict(test2e, 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

6.1 Amyloidosis

To compute expected events under the models we need the cumulative hazard, for which -log(survival) is an estimate, *unless* the estimated survival is zero.

```
> efun <- function(vtime, ttime, tsurv) {
     # vtime: fu time of the validation subject
     # ttime, tsurv, survival curves from the fitted model
     chaz <- -log(pmax(.01, tsurv)) # stop infinity
     expect <- double(length(vtime))
     z <-approx(ttime, chaz, vtime, rule=2)$y
 }
>
> # Create a matrix of expected, one row per subject, one col per model
> # 2013 only has some of the curves
> amexpect <- matrix(0, nrow(amyloid), 3)
> years <- c(2004, 2012, 2015)
> for (i in 1:3) {
     tdata <- subset(amyloid.surv, study == years[i])
     vstage <- get(paste0('r', years[i]), amyloid)
     for (stage in unique(vstage, na.rm=T)) {
         indx1 <- which(vstage== stage)
         indx2 <- which(tdata$stage == stage)
 cat(i,stage, "  ")
         amexpect[indx1, i] <- efun(amyloid$month[indx1], tdata$month[indx2],
                                    tdata$survival[indx2])
     }
 }
1 1   1 2   1 0   2 0   2 2   2 3   2 1   3 1   3 2   3 3   3 0
> colnames(amexpect)<-years
> round(colSums(amexpect), 2)
   2004    2012    2015
1669.84  892.99  760.60
```

```
> sum(amyloid$status)
[1] 583
```

The total deaths in the amloidosis validation data set is 583, while expected counts are 1670, 893 and 599 for the 2004, 2012 and 2105 predictions. The underlying reason is major gains in the treatment of AL amyloidosis over that span of years. This leaves a set of models with very similar discrimination but wildly different calibration. A natural next step would be to look within stage to see where the gains have been greatest or least.

A formal test using glm runs into an issue, however, in that the 2015 reference has 0 deaths in stage 0, while the amyloid data has 51/199 deaths in that stage. This leads to 1*log(0) in the Poisson loglikelihood for such subjects: with a poisson rate of 0 deaths have likelihood 0. In fact, the observed count above should be modified to only count deaths within the time span of the prediction model. It is actually unusual to have validation data with a longer time span than the prediction model.

6.2 UDCA

First, look at observed and expected events.

```
> ucount <- with(udca2, c("deaths"= sum(death), "progression"= sum(pstat),
                          "expected"= sum(expect)))
> ucount
     deaths progression    expected
   16.00000    63.00000    23.85245
>
> ufit1 <- glm(death ~ offset(log(expect)), poisson, data=udca2,
              subset = (expect >0))
> summary(ufit1)

Call:
glm(formula = death ~ offset(log(expect)), family = poisson,
    data = udca2, subset = (expect > 0))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.3993     0.2500  -1.597     0.11
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 79.941  on 167  degrees of freedom
Residual deviance: 79.941  on 167  degrees of freedom
AIC: 113.94

Number of Fisher Scoring iterations: 6
>
> ufit2 <- glm(death ~ trt + offset(log(expect)), poisson, data=udca2,
              subset = (expect >0))
> summary(ufit2)

Call:
glm(formula = death ~ trt + offset(log(expect)), family = poisson,
    data = udca2, subset = (expect > 0))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.09179    0.31622   0.290   0.7716
trt         -0.98988    0.51639  -1.917   0.0552

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 79.941  on 167  degrees of freedom
Residual deviance: 76.105  on 166  degrees of freedom
AIC: 112.1

Number of Fisher Scoring iterations: 6
>
> ufit3 <- coxph(Surv(time, death) ~trt + offset(riskscore), udca2)
```

This is taken up more fully in the next section.

7 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.

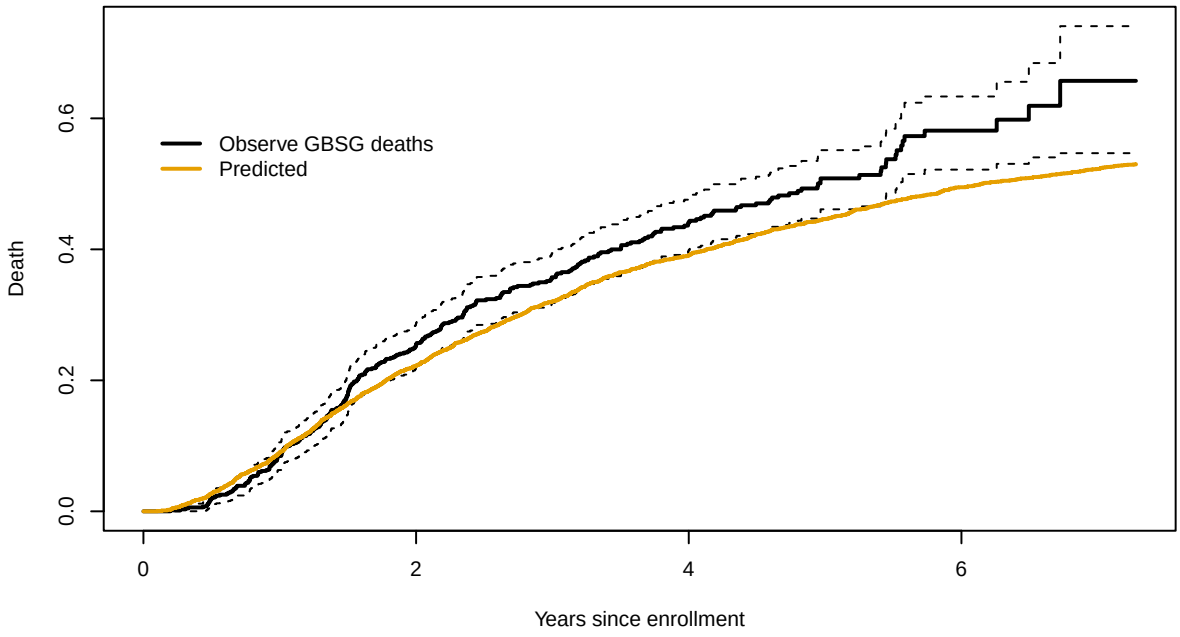
One simple approach is to compare the predicted survival curve for the validation cohort, based on the model, with the KM of the validation cohort. For the predicted curve we need to use the proper marginal estimate which is

$$\overline{S}(t) = (1/n) \sum_{i=1}^{n} \hat{S}(t; x_i) \tag{1}$$

$$\neq \hat{S}(t; \overline{x}) \tag{2}$$

where x is the set of covariate vectors for the n validation samples and $\hat{S}$ is prediction using the target model. Equation (1) is called the direct adjusted survival, or in more recent years as a g-estimate. The curve at the mean covariate, equation (2) is not the same.

```
> directall <- survfit(rfit2, newdata=gbsg2)
> dim(directall) # separate curve for all 686 subjects
data
 686
>
> # collapse for plotting
> direct <- directall
> direct$surv <- rowMeans(direct$surv)
> # std does not collapse neatly, keep plot and summary from trying to use it
> direct$std.err <- direct$upper <- direct$lower <- direct$std.err <- NULL
>
> plot(gsurv, fun='event', lwd=c(2,1,1),
      xlab="Years since enrollment", ylab= "Death")
> lines(direct, fun='event', conf.int=F, lwd=2, col=2)
> legend(0, .6, c("Observe GBSG deaths", "Predicted"), col=1:2, lwd=2, bty='n')
```




```
>
> temp1 <- 1- summary(gsurv, time=4:7)$surv
> temp2 <- 1- summary(direct, time=4:7)$surv # predicted death at years 4:7
> round(temp1/temp2,2)
[1] 1.13 1.14 1.17 1.26
```

Object `directall` 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 not identical to the SMR, but similar 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(ryear, rfs) ~ eta2, gbsg2)
> cfit2 <- coxph(Surv(ryear, rfs) ~ eta2 + offset(eta2), gbsg2)
> cfit1
```

```
Call:
coxph(formula = Surv(ryear, rfs) ~ eta2, data = gbsg2)

        coef exp(coef) se(coef)     z      p
eta2 1.01981   2.77268  0.09966 10.23 <2e-16

Likelihood ratio test=102.8  on 1 df, p=< 2.2e-16
n= 686, number of events= 299
> cfit2
Call:
coxph(formula = Surv(ryear, rfs) ~ eta2 + offset(eta2), data = gbsg2)

        coef exp(coef) se(coef)     z     p
eta2 0.01981   1.02001  0.09966 0.199 0.842

Likelihood ratio test=0.04  on 1 df, p=0.8424
n= 686, number of events= 299
> cfit3 <- coxph(Surv(ryear, rfs) ~ nsp(eta2, 4) + offset(eta2), 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. If there was overfitting during creation of the prediction model then we might expect the actual risk to rise and fall less sharply than the predictions. The label is a bit of a misnomer since the result is on a relative scale rather than absolute, so is closer to discrimination. One could have have a coefficient of 1.0 and all predicted survivals be twice what we see in the data (but consistently half!).

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 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 when assessing the overall *shape* of the curve.

The more common plot is to compute predicted survival for both the prediction model

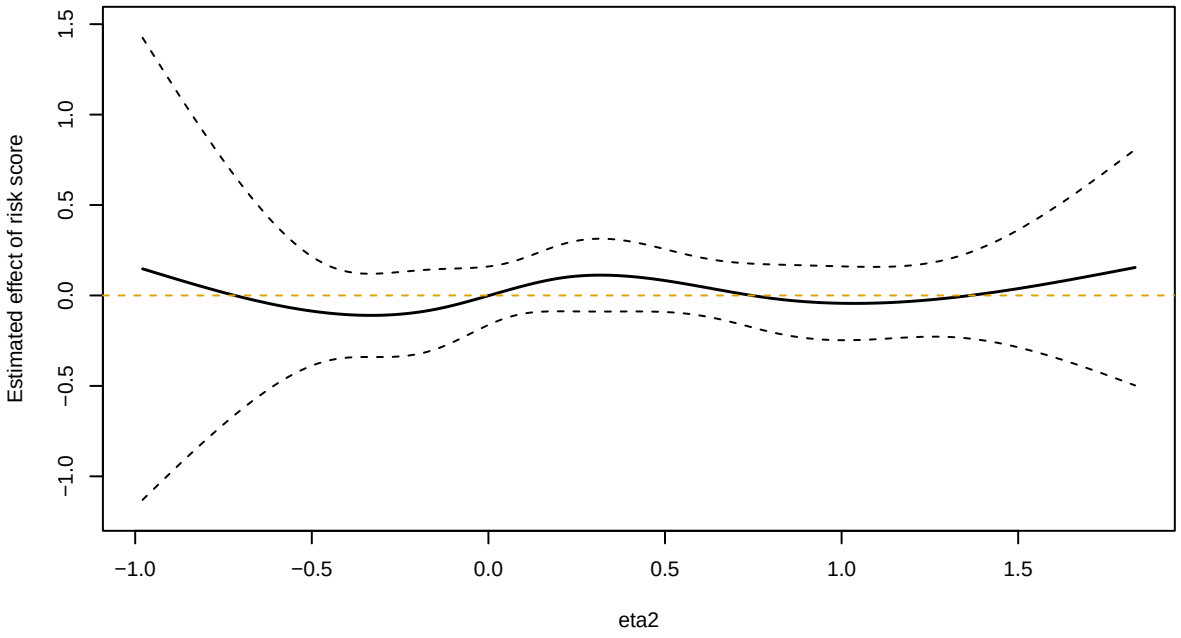
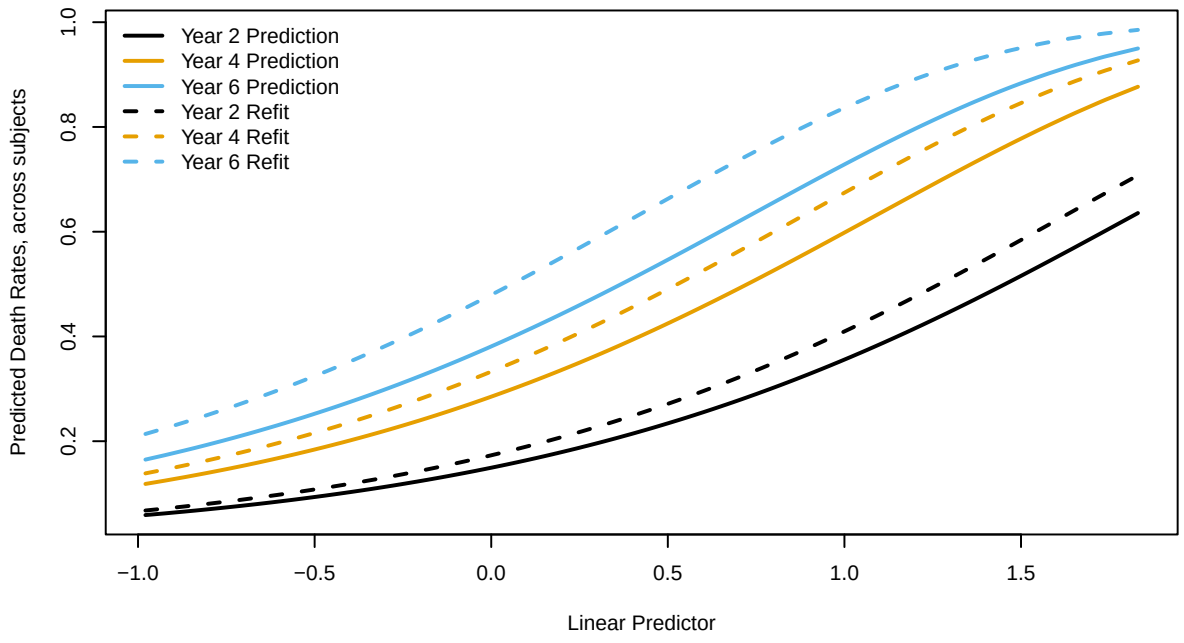


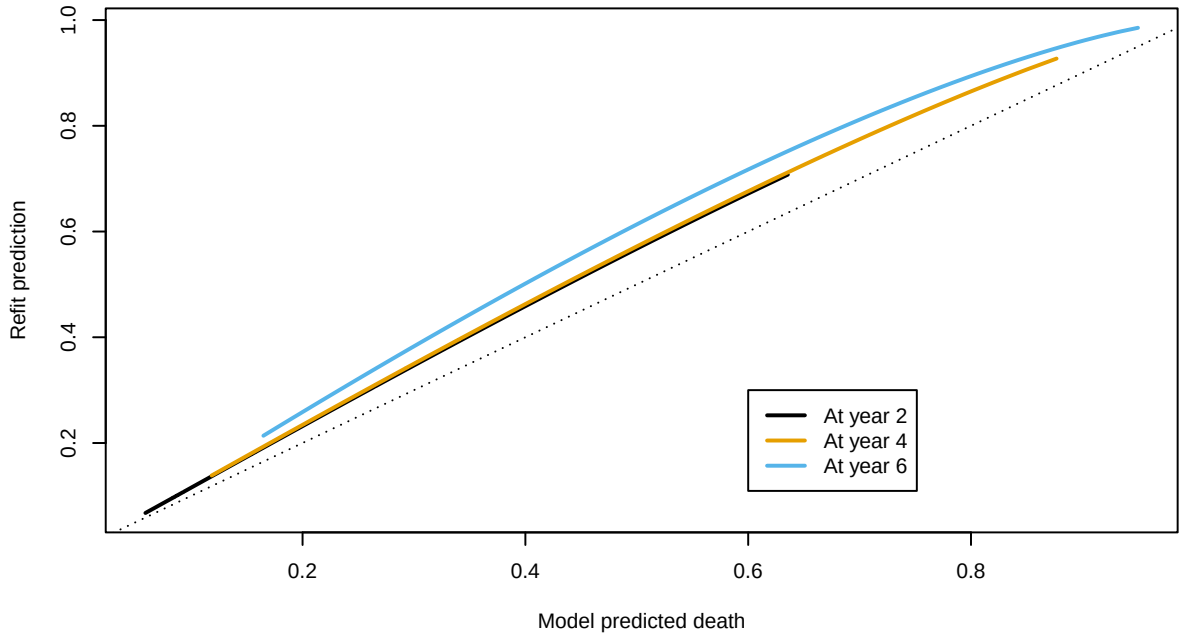
Figure 2: A check for systematic effect versus the risk score.

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. This is a calibration measure since it is on absolute scale.

```
> d2 <- survfit(cfit1, newdata=gbsg2)
> indx <- order(gbsg2$eta2) # need to be in eta order to draw lines
> yrs1 <-  findInterval(c(2,4,6), d2$time)
> yrs2 <-  findInterval(c(2,4,6), direct$time)
> temp1 <- t(d2$surv[yrs1,indx])   # Refit survival
> temp2 <- t(directall$surv[yrs2, indx]) # Predicted survival
> matplot(gbsg2$eta2[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("Prediction", "Refit"), paste),
        lty=c(1,1,1,2,2,2), col= 1:3, lwd=2, bty='n')
```



```
> matplot(1-temp2, 1-temp1, type='l', lwd=2, col=1:3, lty=1,
         xlab="Model predicted death", ylab="Refit prediction")
> abline(0,1, lty=3)
> 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 Cox 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.

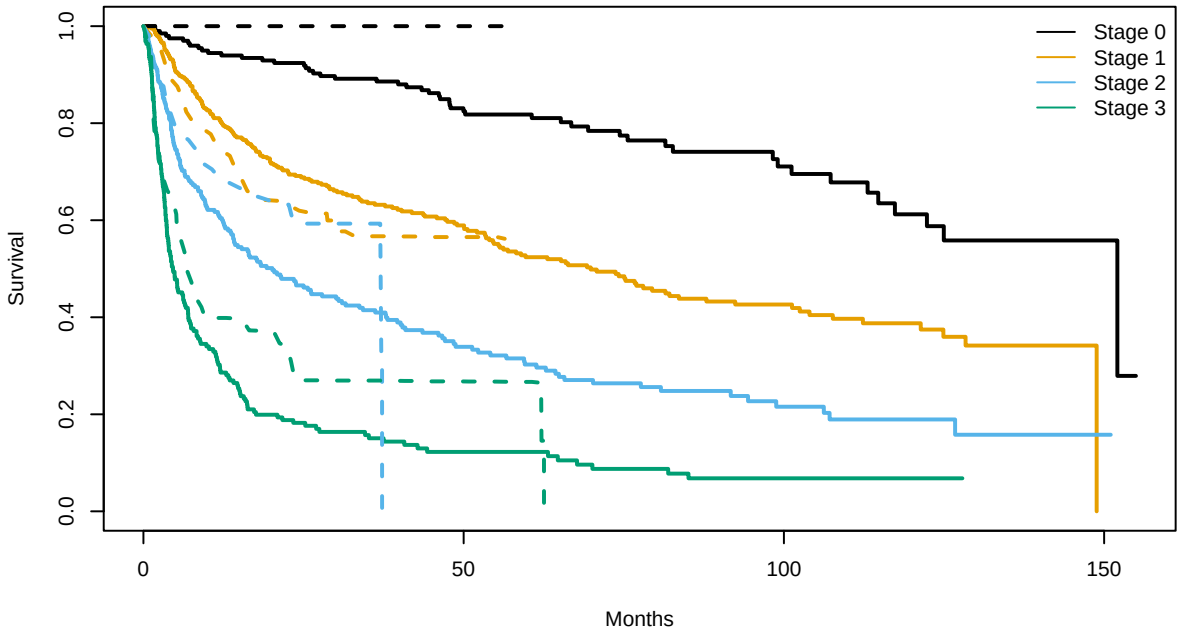
To do: Add plot of $S(t; \eta = \eta_0)$ with $\hat{S}$ from the validation model and from the refit model, 2 curves on one graph.

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 η for a selected time τ, or over potential cutpoints τ for selected values of η. To do: show it.

7.1 Amyloidosis

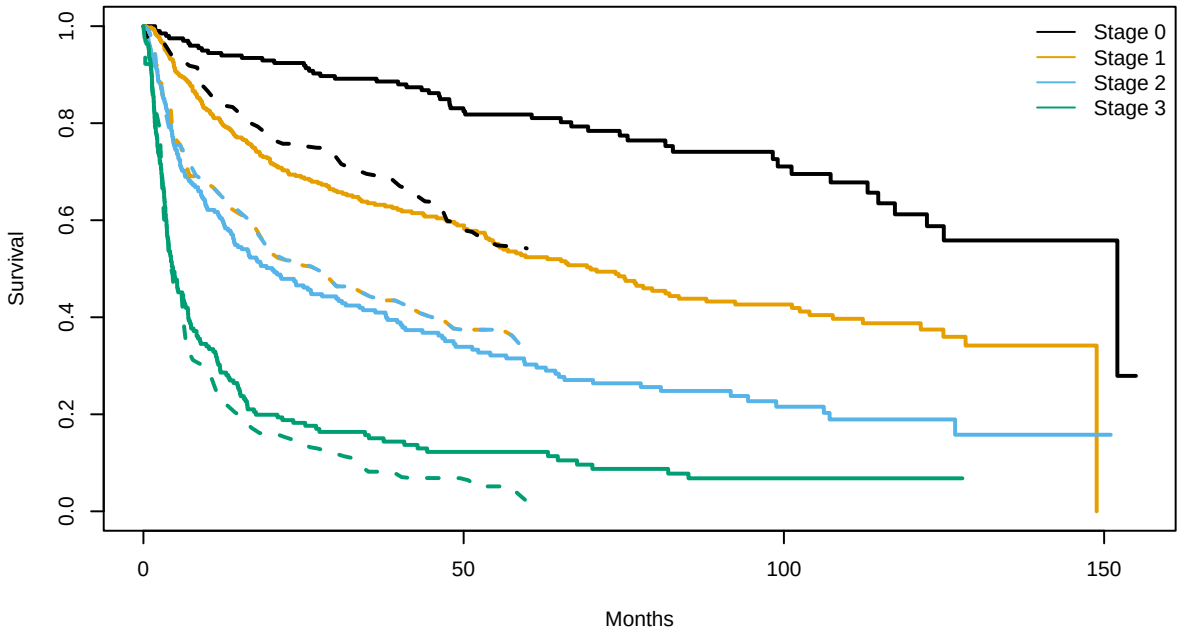
The obvious first approach with amyloidosis is to plot the survival curve of the validation data along with a predicted curve from the prediction models. It is easiest to do by stage.

```
> asurv15 <- survfit(Surv(month, status) ~ r2015, amyloid)
> plot(asurv15, col=1:4, lwd=2, xlab="Months", ylab="Survival")
> for (i in 0:3) {
     temp <- subset(amyloid.surv, stage==i & study== 2015)
     lines(temp$month, temp$survival, lwd=2, col=i+1, lty=2)
 }
> legend("topright", paste("Stage", 0:3), lty=1, col=1:4, bty='n')
```



The above shows clear flaws with our prior calculation of observed versus expected, which implictly extended all of the model curves to the right. For stages 0 and 1 this expected will be seriously biased small, for stage 3 it will be biased large. The plot for 2012 shows that our counts seriously biases there as well, perhaps more so. When using predictions with a limited time span care needs to be spent to esure that like is compared to like. Nevertheless, we can see that the greatest gains in survival were for stages 0 and 1.

```
> asurv12 <- survfit(Surv(month, status) ~ r2012, amyloid)
> plot(asurv15, col=1:4, lwd=2, xlab="Months", ylab="Survival")
> for (i in 0:3) {
     temp <- subset(amyloid.surv, stage==i & study== 2012)
     lines(temp$month, temp$survival, lwd=2, col=i+1, lty=2)
 }
> legend("topright", paste("Stage", 0:3), lty=1, col=1:4, bty='n')
```



Since the prediction did not come from a fitted model, we do not have a linear predictor η as a concise summary of the difference between curves, and the survival regression methods do not apply.

7.2 UDCA

Two analysis strategies for the UDCA study were to first increase the number of events, by using other markers of liver progression, and second to make use of the PBC risk model to reduce the total number of covariates. We have seen an absolute risk vesion of this in the SMR section.

With only 10 events, survival curves that break subjects up by risk score would be fruitless. Look at a set of regression models.

```
> ucox1 <- coxph(Surv(time, death) ~ trt, udca2)
> ucox2 <- coxph(Surv(time, death) ~ trt + log(bili), udca2)
> ucox3 <- coxph(Surv(time, death) ~ trt + offset(riskscore), udca2)
> tfun <- function(fit) c(coef(fit)[1], sqrt(diag(vcov(fit)))[1])
```

```
> temp<- matrix(c(tfun(ucox1), tfun(ucox2), tfun(ucox3)), nrow=2,
               dimnames= list(c("coef", "std"), paste("Fit", 1:3)))
> round(temp,3)
      Fit 1  Fit 2  Fit 3
coef -0.675 -0.932 -0.952
std   0.517  0.540  0.520
```

Since the study was randomized, one could argue that no adjustment for covariates is required. However, the first fit clearly shows a different treatment effect. The second fit adjusts for bilirubin, the most impactfull portion of the risk score, and the third directly uses the coefficients from the PBC risk score. We can see that option 3 does, as we expected, somewhat decrease the std of the treatment coefficient, though still not quite significant. (The glm approach avoids estimating a baseline hazard, and was perhaps a tiny bit smaller at .516.)

Use of liver progression was more successful.

```
> ucox4 <- coxph(Surv(ptime, pstat) ~ trt, udca2)
> ucox5 <- coxph(Surv(ptime, pstat) ~ trt + log(bili), udca2)
> ucox6 <- coxph(Surv(ptime, pstat) ~ trt + offset(riskscore), udca2)
> temp2 <- matrix(c(tfun(ucox4), tfun(ucox5), tfun(ucox6)), nrow=2,
               dimnames= list(c("coef", "std"), paste("Fit", 1:3)))
> round(temp2,3)
      Fit 1  Fit 2  Fit 3
coef -0.766 -0.939 -1.077
std   0.260  0.268  0.262
```

Again, we see an increase the estimated effect when adjusting for confounders, with a larger increase when more than bilirubin is included, and a decrease in std for the treatment effect. The total number of events is still too small at 63 to seriously consider regression coefficients for all 5 elements of the risk score.

8 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.

- Pseudovalues are an alternative to the RTTR which will assign a response to censored and non-censored subjects.

- Apply well known measures for binomial data to the new data, e.g., sensitivity, specificity, logistic regression, etc.

A simple weighted mean of the status values at τ, after redistributing the weights, turns out to be exactly equal to the Kaplan-Meier estimate at time τ, based on the unmodified validation data. Thus mean calibration at τ, on the probability scale, is quite easy to obtain; simply compare the predicted probability for the validation cohort as a whole to the KM at that point. Since the prediction model is considered fixed the standard error of the difference is the std of the KM at that point. One should of course use one of the better std scales for the KM rather than "plain" to form a confidence interval. The plot of KM vs $\hat{S}$ appears yet again ....

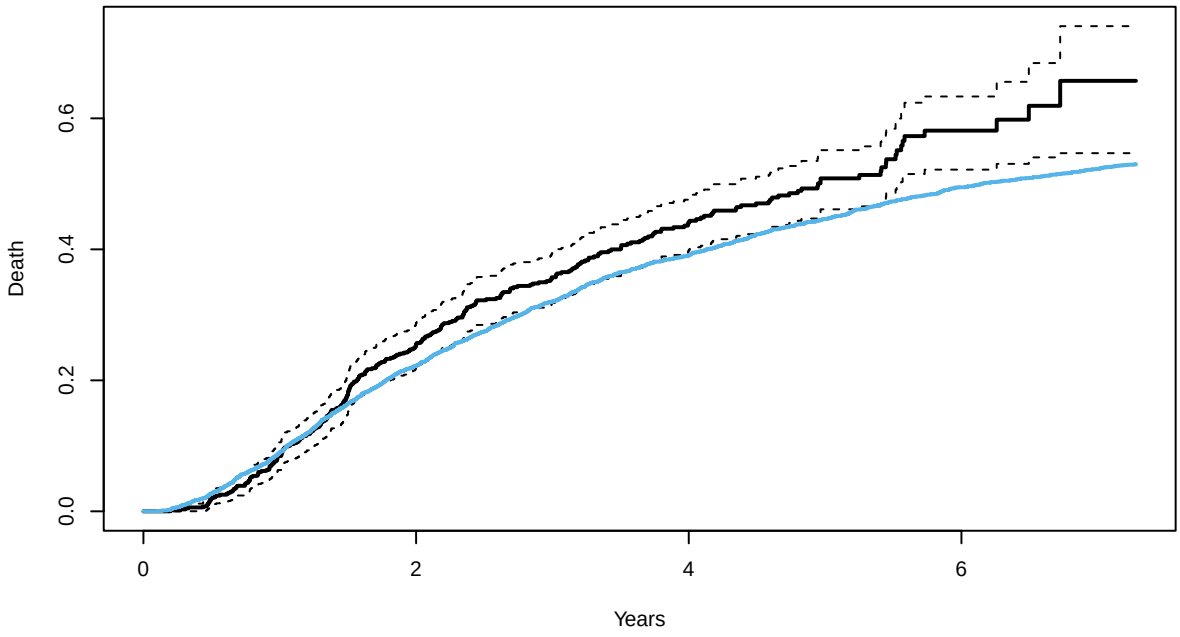
If one were to fit a weighted logistic regression to the weighted data with status as the outcome and an intercept as a predictor, the predicted probability will again equal the KM, and the transformed confidence interval for β_0 is yet another possible choice. (We point this not to argue for or against the mean only model, but to show how things fit in.)

One constraint is that the reweighting approach does not extend to age scale analyses, as neither the RTTR nor pseudovalues extend to data with delayed entry.

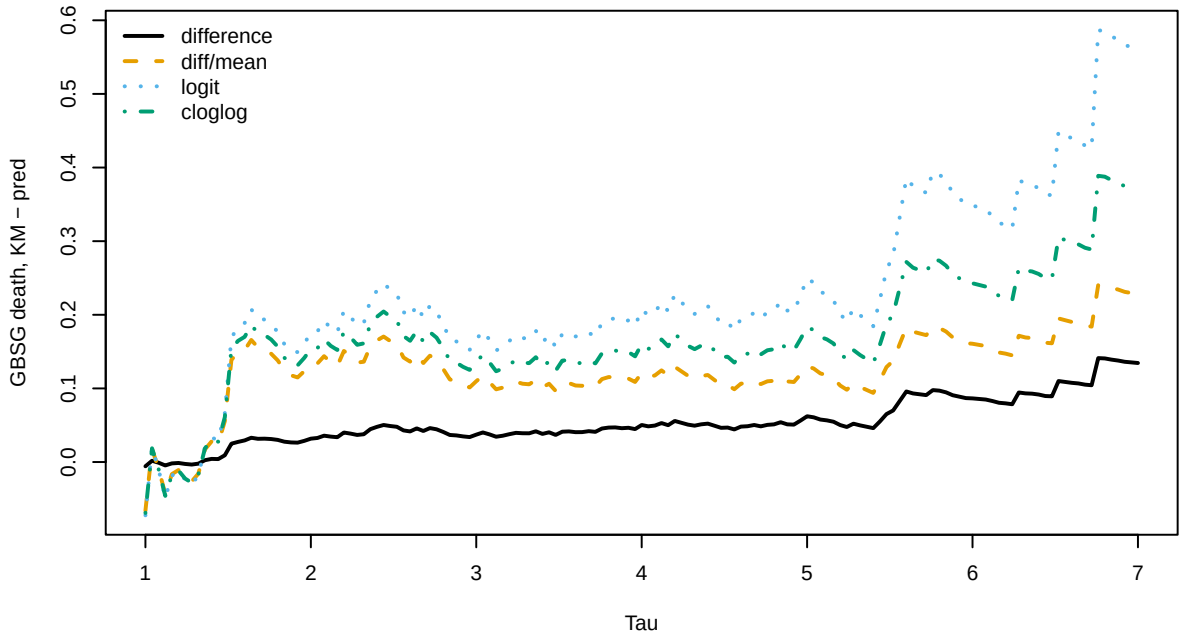
8.1 Breast cancer

As we have before, start with the overall observed and predicted survival, and look at the difference between them. The overall predicted curve `direct` was computed earlier.

```
> #opar <- par(mfrow=c(1,2), mar=c(5,5,1,1))
> plot(gsurv, lwd=c(2,1,1),  fun='event', xlab="Years", ylab="Death")
> lines(direct, lwd=2, col=3, fun='event')
```




```
> j <- seq(1,7, length=151)
> p1 <- 1- summary(gsurv, time=j)$surv
> p2 <- 1- summary(direct, time=j)$surv
> logit <- function(x) log(x/(1-x))
> clog  <- function(x) log(-log(1-x))
> ytemp <- cbind(p1-p2, 2*(p1-p2)/(p1+p2), logit(p1)- logit(p2),
                 clog(p1)- clog(p2))
>
> matplot(j, ytemp, type='l', lwd=2, xlab="Tau", ylab="GBSG death, KM - pred")
> legend("topleft", c("difference", "diff/mean","logit", "cloglog"),
        lty=1:4, col=1:4, lwd=2, bty='n')
```



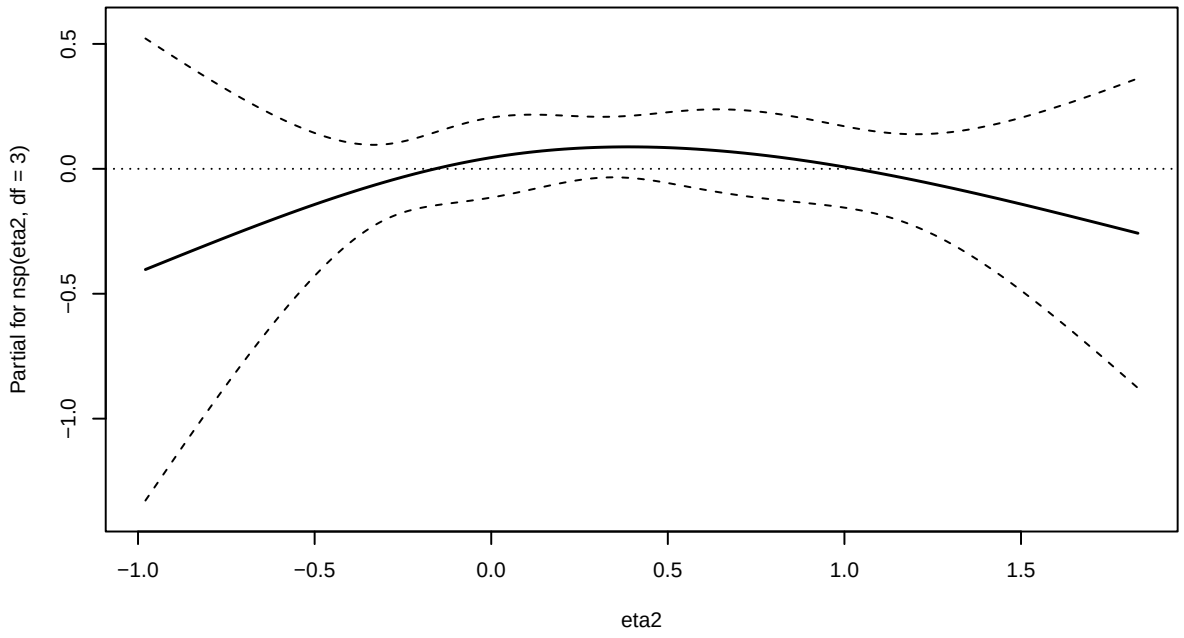

```
> #par(opar)
```

The raw difference between observed and predicted survival increases with time, which is what we would expect if there is a systematic shortcoming in the prediction model. Statisticians often pick a scale where effects will additive and approximately constant over time (though if we are honest, the transform is normally chosen purely for convenience). Both the complimentary log-log and logit have some success wrt this metric over the inteval from 2.5 to 5 years, though the discontinuous weighting process adds a lot of noise: as τ moves from left to right a censored observation's weight slowly increases and then abruptly drops to zero.

Do binomial analysis with a cutoff of $\tau = 5$ years. Normally, users default to a logistic link, but the η value from a Cox model is consistent with the cloglog link so we use the latter.

```
> tau <- 5
> d5 <- gbsg2 # temporary data set
> d5$rwt <- rttright(Surv(ryear, rfs) ~ 1, gbsg2, times=tau, renorm=FALSE)
```

```
> d5$ytau <- with(gbsg2,ifelse(rfs==1 & ryear <= tau, 1, 0))
>
> lfam <- binomial(link = "cloglog")
> lfit1 <- glm(cbind(ytau, 1-ytau) ~ eta2, lfam, weights=rwt, data= d5)
> lfit2 <- glm(cbind(ytau, 1-ytau) ~ eta2+ offset(eta2), family=lfam,
              weights= rwt, data= d5)
> lfit3 <- glm(cbind(ytau, 1-ytau) ~ nsp(eta2, df=3) + offset(eta2), lfam,
              weights= rwt, data= d5)
> termplot2(lfit3, term=1, se=TRUE)
> abline(0,0, lty=3)
```



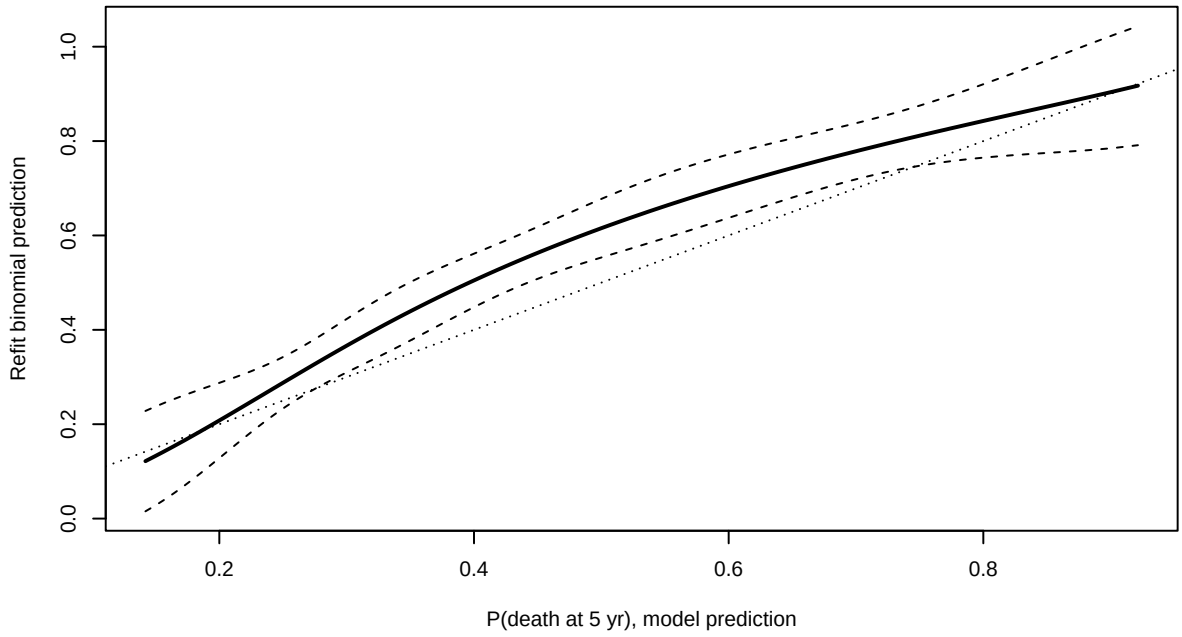
More formal (and more correct) is to use the survey package.

```
> # more formal (and correct) accounting for weights
> sdat <- svydesign(data= d5, id= ~ pid, weights= ~ rwt, variables= ~ .- pid)
> svyfit2 <- svyglm(cbind(ytau, 1-ytau) ~ eta2 + offset(eta2),
                    family=lfam, design=sdat)
```

```
>
> rbind(glm = c(coef(lfit2)[2], sd=sqrt(diag(vcov(lfit2))[2])),
       svyglm= c(coef(svyfit2)[2], sd=sqrt(diag(vcov(svyfit2))[2])))
               eta2    sd.eta2
glm    -0.005296065 0.09805892
svyglm -0.005296065 0.15630359
```

The above are not formally calibration since they do not use the absolute predictions. We can reprise the $\hat{y}$ vs $\hat{y}$ plot of the prior section, based on the clog fit.

```
> phat1 <- 1- c(summary(directall, time=5)$surv)  # predicted death at 5 years
> phat2 <- predict(lfit3, type='response', se.fit =TRUE)
> yy <- phat2$fit+ outer(phat2$se.fit, c(0, -1.96, 1.96), '*')
>
> indx <- order(d5$eta2) # so that we can plot lines
> matplot(phat1[indx], yy[indx,], type='l', lwd=c(2,1,1), lty=c(1,2,2),
         col=1, xlab="P(death at 5 yr), model prediction",
         ylab="Refit binomial prediction")
> abline(0,1, lty=3)
```



Using the set of η values found in the coxph fit to draw our curve is simply a convenience, we could as easily have chosen 50 ordered values across the range; the curve would be the same. However, the predict functions for both coxph and glm are set up to give predictions for a new set of covariates, not their linear combination. This is the reason we did not worry about the RTTR weights in the above.

We can also plot the ROC curve at 5 years. To do.

Another measure applicable to dichotomized time is the Brier score. Let $y_i = I(t_i <= \tau)$ be the dichotomized response (death by τ), $\hat{p}_i(\tau)$ the predicted probability of survival at τ from the model, and $\hat{p}_0(\tau)$ the Null model probability based on a Kaplan-Meier of the validation data, i.e., a "fit" with no covariates. Then

$$\begin{aligned} R^2 &= 1 - \frac{\sum_i (y_i - \hat{p}_i)^2}{\sum_i (y_i - p_0)^2} \\ &= 1 - N/D \end{aligned}$$

is the usual r-squared statistic, based on the 0/1 variable. The numerator N is known as the Brier score, and ratio N/D as Kattan's index of prediction.

As for the DTC, for censored data we first reassign weights for all observations censored before τ, then compute weighted sums.

9 Synthetic estimates

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

10 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.

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 potential 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.

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] G. Berry. The analysis of mortality by the subject years method. *Biometrics*, 39:173–184, 1983.

[3] B. Efron. The two sample problem with censored data. In *Proceedings of the Fifthh Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 831–853, 1967.

[4] M. W. Kattan and T. A. Gerds. The index of prediction accuracy: an intuitive measure useful for evaluating risk prediction models. *Diag Prog Res*, 2, 2018.

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

[6] E. Muchtar, T. Therneau, D. Larson, M. Gertz, M. Lacy, F. Buadi, D. Dignli, S. Hayman, P. Kapoor, W. Gonsalves, T. Kourelis, R. Warsame, A. Fonder, M. Hobbs, Y. Hwq, N. Leung, S. Russell, J. Lust, Y. Lin, R. Go, S. Zeldenrust, R. Kyle, S. Rajkumar, S. Kumar, and A. Dispensieri. Comparative analysis of staging systems in al amyloidosis. *Leukemia*, 33:811–3, 2019.

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

[8] Hans C. van Houwelingen and Hein Putter. *Dynamic prediction in clincal survival analysis*. CRC Press, 2012.