Pseudovalues for survival data

Terry Therneau

August 26, 2021

1 Introduction

Let Y be survival time and f some function of interest, for which we desire to estimate $E(f(Y))$ over the population. With complete data on each observation's survival time, we can compute the expectation in the straightforward way as the simple mean $\sum f(y_i)/n$. Due t censoring, survival data is unfortunately incomplete.

Pseudo-values are based on a simple idea. Suppose the data are incomplete but we have an estimator $\hat{\theta} = E(f(Y)$ of the quantity of interest, e.g., f = survival probability at time 45, and $\hat{\theta}$ the Kaplan-Meier estimate at time 45. The pseudo-observation for y_i is then defined as

$$\hat{\theta}_i = \hat{\theta} + (n-1)(\hat{\theta} - \hat{\theta}^{-i}) \tag{1}$$

where $\hat{\theta}^{-i}$ is defined as the value of the estimate when subject i is omitted from the sample. An an illustration, evaluate this formula when θ is an ordinary mean:

$$\begin{aligned} \hat{\theta} + (n-1)(\hat{\theta} - \hat{\theta}^{-i}) &= \sum_j y_j/n + (n-1)\left(\sum_j y_j/n - \sum_{j \neq i} y_j/(n-1)\right) \\ &= \sum_j y_j - \sum_{j \neq i} y_j \\ &= y_i \end{aligned}$$

The idea is to use the pseudo-observations $\hat{\theta}_i$ as a replacement for the incompletely observed data y_i. These pseudo values are not censored, allowing for ordinary statistical methods to be applied. An good overview of pseudo-value based methods is provided by Andersen and Perme [1].

The `pseudo` function in the survival package uses values based on the infinitesimal jackknife (IJ), i.e.,

$$\hat{\theta}_i = \hat{\theta} + n\frac{\partial \hat{\theta}}{\partial w_i} \tag{2}$$

Why use IJ based pseudo-values instead of jackknife based values?

1. The largest advantage is compuatational; IJ values can be assembled more quickly.

2. The survival package already makes extensive use of IJ values to compute robust variance estimates, so these naturally fit into that framework. We take advantage of existing test suites to ensure accurate computations.

Both estimates are linear approximations to a functional surface, something nicely pointed out by Efron [2]: the IJ is a tangent plane to the surface and the jackknife a secant plane. As such, they share the property that the average of the pseudo values will recover the starting estimate, mean($\hat{\theta}_i$) = $\hat{\theta}$. The choice of n for the IJ based pseudo value makes the compuation for an ordinary mean exact, as was shown above for $n-1$ and the jackknife pseudovalue, and in fact Efron argues out that the use of $n-1$ is somewhat arbitrary, i.e., other statistics than the mean might be more exact using n, $n+1$ or some other multiplier. The largest potential deficit of the IJ based approach is that the literature is small — most of the direct examinations have focused on jackknife based values. Examples show that the numerical difference between the two is miniscule, with the possible exception of large outliers, but a critic might call the IJ approach "pseudo" pseudovalues and claim that they are as yet unproven.

2 Residual mean survival time

We will start with one of the more compelling uses, which provides modeling tools for the mean time spent in a state. For any postive probability distribution, a well known identity is that mean is equal to the area under the survival curve.

$$\mu = \int S(t)dt$$

This extends to the multi-state case, where the expected time in state will be equal to the area under the P(state) curve for that state.

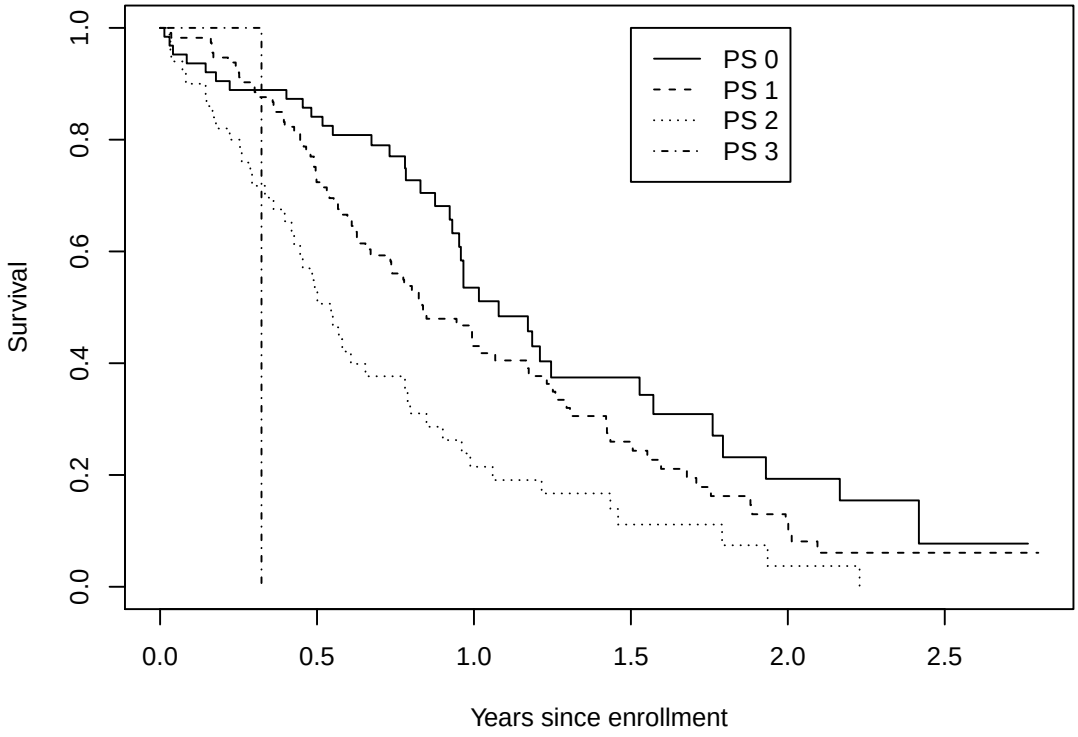
Since the Kaplan-Meier gives an unbiased estimate of Pr(in the alive state), we can use the area under the KM to estimate the mean time to death. However, since the entire $S(t)$ curve is usually not available, i.e., the KM terminates before reaching 0, we instead estimate the restricted mean survival time (RMST), using the area under the KM up to some specified point τ. This is interpreted as the expected number of life years, out of the first τ years since intitation. If $\tau = 10$ and the area under the curve were 8.1, the relevant phrase would be an "expected liftime of 8.1 out of the next 10 years". Other common labels the sojourn time, or for a multi-state model, the restricted mean time in state (RMTS). For a multi-state model, the sojourn time for all states must sum to τ, i.e. everone has to be somewhere.

Common choices for τ are either a fixed time of interest such as 2, 5 or 10 years, the last observed event time, or the point at which any one of the curves has no subjects.

Here is a simple example using the lung cancer data set. For this cohort of subjects with advanced disease, the strongest predictor is performance score.

```
> lfit0 <- survfit(Surv(time/365.25, status) ~ 1, data=lung)
> lfit1 <- survfit(Surv(time/365.25, status) ~ ph.ecog, data=lung)
> print(lfit1, rmean=2.5)
Call: survfit(formula = Surv(time/365.25, status) ~ ph.ecog, data = lung)
```

```
   1 observation deleted due to missingness
             n events rmean* se(rmean) median 0.95LCL
ph.ecog=0   63     37  1.251    0.1085  1.079   0.953
ph.ecog=1  113     82  1.035    0.0692  0.838   0.734
ph.ecog=2   50     44  0.708    0.0855  0.545   0.427
ph.ecog=3    1      1  0.323    0.0000  0.323      NA
           0.95UCL
ph.ecog=0    1.572
ph.ecog=1    1.175
ph.ecog=2    0.789
ph.ecog=3       NA
    * restricted mean with upper limit =  2.5
> plot(lfit1, lty=1:4, xlab="Years since enrollment", ylab="Survival")
> legend(1.5, 1, c("PS 0", "PS 1", "PS 2", "PS 3"), lty=1:4)
```



The strongest predictor in the data set is ECOG performance score, which has levels of 0–3. For a single categorical predictor such as this, it is easy to obtain the RMST and its standard error directly from the print routine for `survfit`. We can also do this with pseudo-values, which allow for a multivariate model. The result shows a loss of about .26 years for each 1 point increase in the performance score, and about .32 years longer RMST for females.

```
> pmean <- pseudo(lfit0, times=2.5, type="RMST")
> ldata <- data.frame(lung, pmean= c(pmean),
```

```
                      id=1:nrow(lung))
> afit1 <- lm(pmean ~ ph.ecog + sex + age, data=ldata)
> round(summary(afit1)$coefficients, 2)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.19       0.38    3.14     0.00
ph.ecog        -0.26       0.07   -3.74     0.00
sex             0.32       0.10    3.26     0.00
age            -0.01       0.01   -1.12     0.26
```

The IJ residuals for observations that are censored early in the study are of necessity small, as they have less opportunity to affect the results, which in turn means that the pseudo values are not equivariant. (An observation censored befor the first event has residual 0, so its pseudo value has no variance at all.) In such a case theory argues for using White's variance estimate for the linear model fit, which accounts for possible heterscadisity. This is the same as the "working indepence" estimate of a GEE model, or the variance estimate from a survey sampling approach. We compute both of these below.

```
> afit2 <- geese(pmean ~ ph.ecog + sex + age, data=ldata,
                 subset= (!is.na(ph.ecog)))
> round(summary(afit2)$mean, 3)
            estimate san.se   wald     p
(Intercept)    1.192  0.389  9.399 0.002
ph.ecog       -0.255  0.067 14.526 0.000
sex            0.322  0.099 10.519 0.001
age           -0.006  0.006  1.217 0.270
> #
>
> ldesign <- svydesign(data=ldata, id= ~id, weights=NULL,
                 variables= ~ . -id)
> afit3 <- svyglm(pmean ~ ph.ecog + sex + age, design=ldesign)
> round(summary(afit3)$coefficients, 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.192      0.390   3.059    0.002
ph.ecog       -0.255      0.067  -3.803    0.000
sex            0.322      0.100   3.236    0.001
age           -0.006      0.006  -1.101    0.272
```

The estimated coefficients will be identical for all three approaches, as evidenced above. In this particular case the robust and normal standard errors hardly differ, which we have found to be the usual case when there is a single psuedo value per subject. When there are psuedo values at multiple reporting times, and thus mulitple observations per subject, then the correction becomes essential. Deficiencies in R's `geese` function lead us to use the survey sampling approach from here forward, in particular that any missing values cause it to fail, the input data is required to be sorted by id, and that standard extraction functions such as `coef` and `vcov` have not been implemented. The `survey` package requires that the design be specified in advance; the result object can then be used to fit multiple models.

Stratified pseudo values can also be computed, i.e., where each is based on the observation's leverate within a subset. For example, the `lung` data set comes from a multi-center study, and contains an identifier for the instutition from which the subject was recruited. We might want to adjust for the possibility that different instutions recruit from different patient populations. One way to do so is to add `factor(inst)` to the model, but another is to base the result off per-institution curves.

```
> lfit4 <- survfit(Surv(time/365.25, status) ~ inst, data=lung)
> pmean4 <- pseudo(lfit4, time=2.5, type="auc")
> ldata$pmean4 <- c(pmean4)
> afit4a <- lm(pmean4 ~ ph.ecog + sex + age, ldata)
> afit4b <- lm(pmean4 ~ ph.ecog + sex + age + factor(inst), ldata)
> # survey package does not allow a missing strata
> temp <- subset(ldata, !is.na(inst))
> ldesign4 <- svydesign(data=temp,  id= ~id, weights=NULL, strata= ~inst,
                        variables= ~ .-id -inst)
> afit4c <- svyglm(pmean4 ~ ph.ecog + sex + age, design=ldesign4)
> afit4d <- svyglm(pmean4 ~ ph.ecog + sex + age + factor(inst), design=ldesign4)
> round(summary(afit4c)$coefficients, 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.110      0.377   2.942    0.004
ph.ecog       -0.266      0.069  -3.858    0.000
sex            0.427      0.099   4.313    0.000
age           -0.006      0.006  -1.131    0.259
> round(summary(afit4d)$coefficients[1:4,], 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.224      0.384   3.184    0.002
ph.ecog       -0.316      0.063  -5.038    0.000
sex            0.366      0.101   3.621    0.000
age           -0.008      0.005  -1.424    0.156
```

3 Survival and probability in state

3.1 AML data

As a first case for this endpoint look at a simple survival data set, the well known AML study with 23 subjects, which records the time to relapse for patients with acute myelogenous leukemia. Relapse times range from 5 to 48 months, 5 of the 23 patient times are censored. Get the pseudo values for 12 and 24 month survival.

```
> with(aml, Surv(time, status))
 [1]   9   13   13+  18   23   28+  31   34   45+  48  161+
[12]   5    5    8    8   12   16+  23   27   30   33   43
[23]  45
```

```
> fit1 <- survfit(Surv(time, status) ~1, aml)
> rr1 <- resid(fit1, times=c(12,24))
> pv1 <- pseudo(fit1, times= c(12, 24))
> round(pv1[1:8,], 4)
     [,1]    [,2]
[1,]    0  0.0000
[2,]    1  0.0000
[3,]    1  0.7857
[4,]    1 -0.1122
[5,]    1 -0.1122
[6,]    1  1.0306
[7,]    1  1.0306
[8,]    1  1.0306
```

Both the matrix of IJ values from `resid` and the matrix of pseudo-values from `pseudo` have one row per subject and one column per reporting time. The first censoring is at 13 months, and so at the first reporting time of 12 months the pseudo values behave exactly like a mean, and have recaptured the (uncensored) 0/1 response of "relapse within 12 months". At 24 months, after censoring enters, the pseudo values are no longer constrained to lie in $(0, 1)$.

What happens if we use these values in an ordinary regression? The `data.frame` argument causes the values to be returned in long form as a data.frame.

```
> pdata <- pseudo(fit1, times= c(12, 24), data.frame=TRUE)
> lfit1 <- lm(pseudo ~1, pdata, subset= (time==12))
> lfit2 <- lm(pseudo ~1, pdata, subset= (time==24))
> summary(lfit1)$coefficients
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 0.7391304 0.09361833 7.895146 7.34356e-08
> summary(lfit2)$coefficients
             Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 0.5465839  0.1096611 4.984302 5.472569e-05
> summary(fit1, time=c(12,24))
Call: survfit(formula = Surv(time, status) ~ 1, data = aml)

 time n.risk n.event survival std.err lower 95% CI
   12     18       6    0.739  0.0916        0.580
   24     11       4    0.547  0.1073        0.372
 upper 95% CI
        0.942
        0.803
```

The regressions have exactly reproduced the KM values at 12 and 24 months, as expected, and the estimated standard errors almost match those from the `survfit` function. The robust (IJ) variance for the Kaplan-Meier, i.e. the sum of squared IJ values, can in fact be shown to exactly equal the Greenwood estimate of variance for a KM (Anne Eaton, personal communication).

The lm difference is due to the fact that the linear model uses $n-1$ rather than n in computing a variance.

A natural follow-on is to look at covariates. In the AML data set the covariate `x` denotes the two treatment arms.

```
> cfit3 <- coxph(Surv(time, status) ~x, aml)
> cfit3
Call:
coxph(formula = Surv(time, status) ~ x, data = aml)

                 coef exp(coef) se(coef)     z      p
xNonmaintained 0.9155    2.4981   0.5119 1.788 0.0737

Likelihood ratio test=3.38  on 1 df, p=0.06581
n= 23, number of events= 18
> pdata <- cbind(pdata, x= aml$x)   # original data + pseudovalues
> lfit3 <- lm(pseudo ~ x + factor(time), pdata)
> summary(lfit3)$coefficients
                Estimate Std. Error   t value     Pr(>|t|)
(Intercept)     0.8574050  0.1245665  6.883111 1.900447e-08
xNonmaintained -0.2266929  0.1418401 -1.598229 1.173154e-01
factor(time)24 -0.1925466  0.1417060 -1.358775 1.813030e-01
> sfit3 <- survfit(cfit3, newdata=list(x= c("Maintained", "Nonmaintained")))
> summary(sfit3, times=c(12,24))
Call: survfit(formula = cfit3, newdata = list(x = c("Maintained", "Nonmaintained")))

 time n.risk n.event survival1 survival2
   12     18       6     0.841     0.648
   24     11       4     0.706     0.419
```

The above code contains a lot of ideas. First, the Cox model estimates that the subjects on the no maintainance arm have a hazard rate of about 2.5 fold higher than the maintainance arm, $p = .09$. Second, a linear model using pseudovalues has estimated the absolute probability of no recurrence, at 12 and 24 months, to be about .23 lower for the no maintainance group, $p = .12$; conversely the probability of recurrence is .23 higher. Last is a plot comparing the curves. The overall probabilities of recurrence are gathered into a table below.

	Cox, 12	Pseudo, 12	Cox, 24	Pseudo, 24
Maintainance	0.16	0.14	0.29	0.34
Nonmaintainance	0.35	0.37	0.58	0.56
Difference	0.19	0.23	0.29	0.23

One advantage of the linear model is that the pseudovalues are direct estimates of a probability, and so may be easier to communicate to study participants than a hazards ratio. The linear model contains the strong assumption that the difference in survival is the same at times 12 and 24, the Cox model the equally strong one that hazards are proportional across all time

points. Of potentially more consequepsnce, the pseudo-value has two observations for each subject, leading to correlated data. We can correct for this using a robust sandwich estimor either via survey sampling or a GEE argument. These are shown below. The `geese` function has the unfortunate feature that correct standard errors require that all of the observations for a subject are in contiguous rows of the input data. No error message arises if this does not hold, only an incorrect result.

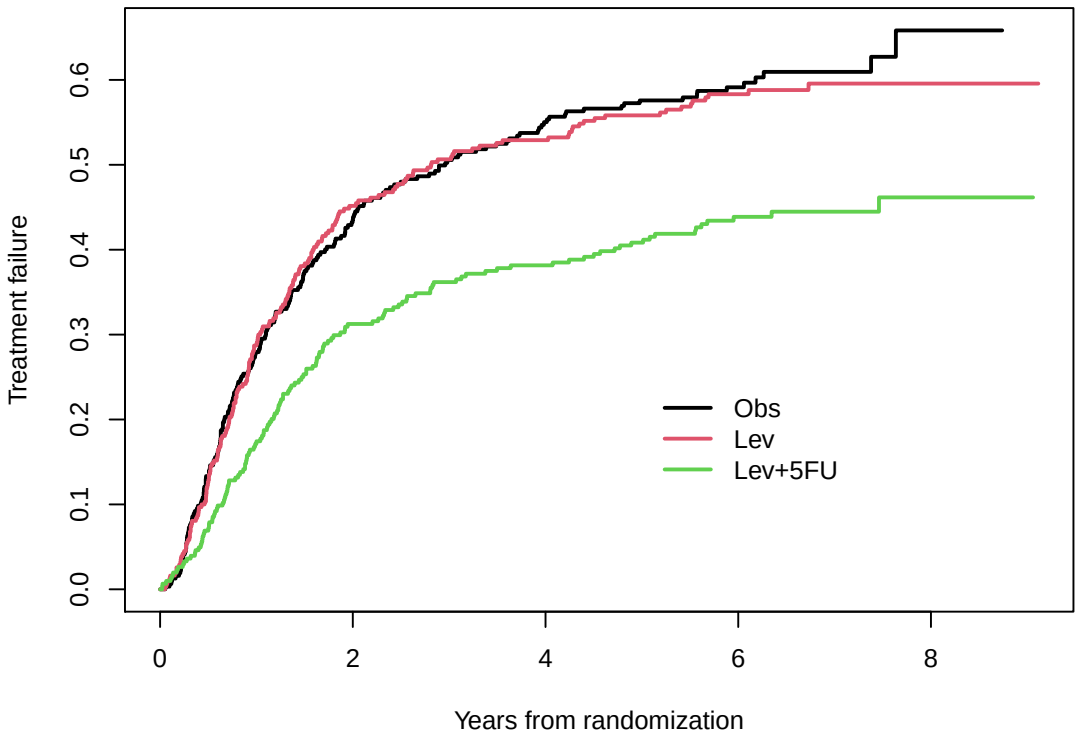
```
> # Naive lm
> summary(lfit3)$coefficients
                 Estimate Std. Error   t value     Pr(>|t|)
(Intercept)     0.8574050  0.1245665  6.883111 1.900447e-08
xNonmaintained -0.2266929  0.1418401 -1.598229 1.173154e-01
factor(time)24 -0.1925466  0.1417060 -1.358775 1.813030e-01
> #
> #survey
> library(survey)
> # When an id is constructed rather than supplied, pseudo() purposely gives
> #  it a non-standard name to avoid confusion with any user's variable names.
> #  But that name is a PITA to use in formulas
> pdata$id <- pdata[['(Id)']]
> pdesign <- svydesign(id= ~ id, varibles= ~ pseudo + x + time, weights=NULL,
                       data=pdata)
> sfit <- svyglm(pseudo ~ x + factor(time), design=pdesign)
> summary(sfit)$coefficients
                 Estimate Std. Error   t value     Pr(>|t|)
(Intercept)     0.8574050 0.09337136  9.182741 1.300627e-08
xNonmaintained -0.2266929 0.17489231 -1.296186 2.096699e-01
factor(time)24 -0.1925466 0.08847305 -2.176330 4.168743e-02
> #
> # GEE 1
> library(geepack)
> gfit1 <- geese(pseudo ~ x + factor(time), data=pdata, id= id) # wrong answer
> summary(gfit1)$mean
                 estimate     san.se      wald          p
(Intercept)     0.8574050 0.09352952 84.037788 0.00000000
xNonmaintained -0.2266929 0.13604793  2.776465 0.09565912
factor(time)24 -0.1925466 0.13700721  1.975080 0.15990964
> #
> # GEE 2
> pdata <- pdata[order(pdata$id),]  # group id values to be together
> gfit2 <- geese(pseudo ~ x + factor(time), data=pdata, id= id)
> summary(gfit2)$mean
                 estimate     san.se      wald          p
(Intercept)     0.8574050 0.09131899 88.155586 0.00000000
xNonmaintained -0.2266929 0.17104806  1.756466 0.18506593
factor(time)24 -0.1925466 0.08652835  4.951706 0.02606494
```

All of the approaches give the same coefficient estimates, differing only in the estimated standard error.

3.2 Colon cancer

Now work with a larger data set, using time to failure (recurrence or death) in the colon cancer study. The data set has 2*929 observations, the first set for time to recurrence and the second for time to death. For anyone without recurrence, their follow-up time for recurrence is equal to their follow-up time for death.

```
> cdata <- subset(colon, etype==1, -etype)  # time to recurrence
> temp  <- subset(colon, etype==2)
> cdata$status <- pmax(cdata$status, temp$status)
> trtsurv <- survfit(Surv(time, status) ~ rx, cdata, id=id)
> plot(trtsurv, fun="event", xscale= 365.25, col = 1:3, lwd=2,
       xlab= "Years from randomization", ylab="Treatment failure")
> legend(5*365, .25, levels(cdata$rx), col=1:3, lwd=2, bty='n')
> ccox <- coxph(Surv(time, status) ~ rx + sex + age + extent + node4+
                    obstruct + perfor + adhere,  data=cdata)
```



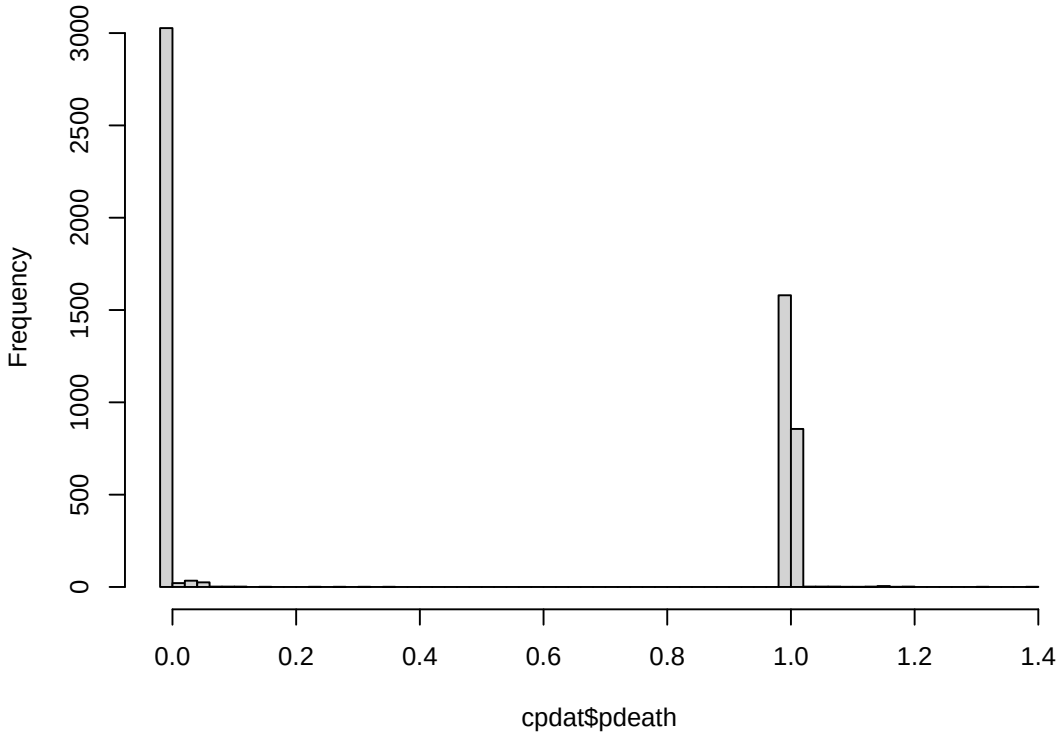
The graph shows that the Levamisole + 5-FU treatment are is markedly superior to either observation or 5-FU alone (the standard treatment at the time). There are few further events

after 4–5 years and the curves flatten out. Having 4 or more positive lymph nodes and greater local spread of the disease are also potent predictors. Now look at pseudo values for the data.

```
> csurv <- survfit(Surv(time, status)  ~1, data=cdata, id=id)
> cptemp <- pseudo(csurv, time= 1:6 * 365.25, data.frame=TRUE)
> cptemp$pdeath <- 1- cptemp$pseudo
> cptemp$year <- round(cptemp$time/365.25)
> cpdat <- merge(cptemp,
                 subset(cdata,,c(id, rx, extent, node4)), by="id")
```

Most users will prefer a model for the risk of death, i.e., for pseudo values based on the probability of death $1 - S(t)$. It is easy to show that these are simply $1 - p(S)$ where $p(S)$ are the pseudovalues for S. This was saved as the variable `pdeath`. We then merge the pseudo data, which has multiple rows per subject, with the original data, by the subject identifier `id` found in the colon cancer data set. Because id was specified in the survfit call it carries through and properly labels the pseudo values.

```
> hist(cpdat$pdeath, nclass=50, main=NULL)
```



This shows that for a large data set with moderate censoring, the pseudo-values are clustered around 0 and 1, mimicing binomial data. To assess survival, which values should we use: year 4 alone, 2 or 3 of the 6 years, or all of them? One important consideration for all of this is the choice of a transformation function f, where we assume that $E(y) \approx f(\eta) + \epsilon$. (Generalized linear model literature normally focuses on the *link* function $g = f^{-1}$.) The ideal function f will

- Transform treatment effects to a common scale over time. That is, a separate intercept for each time point will suffice. No interactions between covariates and time are needed.
- Cause multivariate effects to be additive. That example, the effect of treatment is the same for those with and without 4+ positive lymph nodes. No between covariate interactions are needed.
- Normalize the variance so that ϵ is constant across time and across covariates. Alternatively, choose a distribution that properly maps between the predicted mean and variance.
- Bound predicted values to the range of 0–1.

Satisfying all four of these at once is likely to be impossible. Logistic regression, most user's immediate response to estimation of a yes/no question, fails directly. First, it does not accomodate response values outside the range of 0–1, and secondly the variance for a predicted value near 0 or 1 is assumed to drop to zero. The maximum value of 1.4 for `cpdat$pdeath` fails both of these.

As a first pass, look at the between curve difference for the other arms vs. 5-FU across time, based on the Kaplan-Meier estimates. For this particular data set and these time points, absolute difference between the curves is more stable across time than logit differences.

```
> temp1 <- summary(trtsurv, times= 1:6*365.25)
> temp2 <- matrix(temp1$surv, nrow=6)
> temp3 <- rbind("Obs vs 5FU"= temp2[,1]- temp2[,2],
              "Lev vs 5FU"= temp2[,3]- temp2[,2])
> cat("absolute difference\n")
absolute difference
> round(temp3*100, 1)
           [,1] [,2] [,3] [,4] [,5] [,6]
Obs vs 5FU  0.8  1.6  0.1 -2.1 -1.8 -0.8
Lev vs 5FU 11.3 13.9 14.5 14.7 15.0 14.4
> #
> cat("logit difference\n")
logit difference
> logit <- function(x) log(1/(1-x))
> temp4 <- rbind("Obs vs 5FU"= logit(temp2[,1])- logit(temp2[,2]),
              "Lev vs 5FU"= logit(temp2[,3])- logit(temp2[,2]))
> round(temp4*100, 1)
           [,1] [,2] [,3] [,4] [,5] [,6]
Obs vs 5FU  2.7  3.6  0.2 -3.9 -3.1 -1.4
Lev vs 5FU 49.9 36.8 33.6 32.7 31.3 28.5
```

Based on this, do a first model with linear effects. The next question is which time points to use, and how many. First, look at 6 models with a single time point, each containing treatment, extent and nodes, but summarize only the levamisole coefficient.

```
> onetime <- matrix(0, 6, 4,
                    dimnames=list(paste("Time", 1:6),
                                  c("coefficient", "se.glm", "coef", "se.survey")))
> cpdesign <- svydesign(id= ~id, weights=NULL, data=cpdat,
                        variables = ~ . - id)
> for (i in 1:6) {
      fit1 <- lm(pdeath ~ rx + extent + node4, data=cpdat, subset= (year==i))
      onetime[i,1:2] <- summary(fit1)$coefficients[3,1:2]
      sfit1 <- svyglm(pdeath ~ rx+ extent + node4, design=cpdesign,
                      subset= (year==i))
      onetime[i,3:4] <- summary(sfit1)$coefficients[3, 1:2]
  }
> onetime <- cbind(onetime, "coef/se"= onetime[,3]/onetime[,4])
> round(onetime[, c(1,2,4,5)],3)
       coefficient se.glm se.survey coef/se
Time 1      -0.099  0.033     0.032  -3.091
Time 2      -0.114  0.037     0.037  -3.100
Time 3      -0.135  0.038     0.038  -3.585
Time 4      -0.160  0.038     0.038  -4.225
Time 5      -0.159  0.038     0.038  -4.196
Time 6      -0.144  0.038     0.038  -3.763
```

If you are only going to use one time point, then a robust variance is in this case not necesary, and the best time point in terms of z statistic or power, by an admittedly small margin, is 4–5 years when the results have largely matured. This happens to be the largest estimated gain for levamisole, reducing the absolute failure rate by 16%. A robust variance is called for not only when there are multiple observations per subject, but in the case of heteroscedaticity in the variance. We should therefore not be too quick to declare it unnecessary based on a single example.

Now consider combination of times 2,4,6, or all 6 time points.

```
> pfun <- function(x,d=2) printCoefmat(x, digits=d, P.values=TRUE,
                                       has.Pvalue=TRUE, signif.stars= FALSE)
> sfit1 <- svyglm(pdeath ~ rx + extent + node4,
                          design= cpdesign, subset=(year==4))
> pfun(summary(sfit1)$coefficients[2:5,])
          Estimate Std. Error t value Pr(>|t|)
rxLev       -0.026      0.038    -0.7      0.5
rxLev+5FU   -0.160      0.038    -4.2    3e-05
extent       0.150      0.031     4.9    1e-06
node4        0.274      0.034     8.0    4e-15
> #
> sfit3 <- svyglm(pdeath ~ rx + extent + node4 + factor(year),
                          design= cpdesign,
                  subset= (year==2 | year==4 | year==6))
> pfun(summary(sfit3)$coefficients[2:5,],)
```

```
          Estimate Std. Error t value Pr(>|t|)
rxLev      -0.0091     0.0353    -0.3      0.8
rxLev+5FU  -0.1395     0.0351    -4.0    8e-05
extent      0.1501     0.0283     5.3    1e-07
node4       0.2774     0.0322     8.6   <2e-16
> sfit6 <- svyglm(pdeath ~ rx + extent + node4 + factor(year),
                  design= cpdesign)
> pfun(summary(sfit6)$coefficients)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.1787     0.0800    -2.2     0.03
rxLev          -0.0081     0.0335    -0.2     0.81
rxLev+5FU      -0.1353     0.0330    -4.1    4e-05
extent          0.1382     0.0271     5.1    4e-07
node4           0.2711     0.0308     8.8   <2e-16
factor(year)2   0.1530     0.0118    12.9   <2e-16
factor(year)3   0.2112     0.0134    15.8   <2e-16
factor(year)4   0.2404     0.0140    17.1   <2e-16
factor(year)5   0.2675     0.0145    18.4   <2e-16
factor(year)6   0.2911     0.0151    19.3   <2e-16
```

For the coefficient of interest, the treatment effect, the use of 1, 3, or 6 time points hardly changes the t-statistic or the estimate. All the remaining coefficients are also fairly stable. The year coefficients show the baseline rate creeping up.

Logit link Although a linear link was indicated for the colon data, the logit link is more common. Directly using it will fail, however: consider the following

```
> tryCatch( svyglm(pdeath ~rx + extent + node4 + factor(year), design=cpdesign,
                   family= gaussian(link = "logit")),
            error= function(e) e)
<simpleError in family$linkfun(mustart): Value -0.000235585 out of range (0, 1)>
```

The same out of range error occurs with the simple `glm` function. The issue is that the glm function uses the logit link of $f(x) = \log(x/(1-x))$ to create starting estimates for the iteration, and values outside of $(0, 1)$ lead to a missing value. (That is actually the only place the link is used.) Two choices are to give explicit initial values, or to define our own link. Here is an example of the first, it give starting estimates which are "good enough".

```
> sfit6b <- svyglm(pdeath ~rx + extent + node4 + factor(year), design=cpdesign,
                   family= gaussian(link = "logit"),
                   etastart= pmax(.05, pmin(.95, cpdat$pdeath)))
```

A problem with this is that if we add a subset argument, then the starting estimate also needs to be explicitly trimmed down. An alternate is to define our own link function, based on the code found in the documentation for `family`, or on examination of the glm `make.link` function. We call the function `blogit` for "bounded logit".

```
> blogit <- function(edge=.05) {
      new <- make.link("logit")
      new$linkfun <- function(mu) {
          x <- (pmax(edge, pmin(mu, 1-edge)))
          log(x/(1-x))
      }
      new
  }
> sfit3c <- svyglm(pdeath ~rx + extent + node4 + factor(year), design=cpdesign,
                family= gaussian(link = blogit()), subset=(year %in% c(2,4,6)))
> pfun(summary(sfit3c)$coefficients)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.759      0.477    -5.8    1e-08
rxLev          -0.066      0.159    -0.4      0.7
rxLev+5FU      -0.645      0.163    -4.0    8e-05
extent          0.765      0.158     4.8    2e-06
node4           1.228      0.153     8.0    4e-15
factor(year)4   0.399      0.043     9.2   <2e-16
factor(year)6   0.633      0.054    11.8   <2e-16
```

Though common, results from the logit link are harder to interpret than the linear link. The latter estimates that Levamisole+5FU will reduce the probability of death by 14%, on average, for time points from years 2–6, while the logit link predicts an exp(-.645)= .52 odds of relapse.

log-log link If a Cox model holds, then

$$S(t) = e^{-\Lambda_0(t)e^{(X\beta)}}$$
$$\log(-\log(S(t)) = \log[\Lambda_0(t)] + X\beta$$

which motivates using the following link function.

```
> bloglog <- function(edge=.05) {
      new <- list(linkfun = function(mu) {
                                x <- (pmax(edge, pmin(mu, 1-edge)))
                                log(-log(x))} ,
                  linkinv = function(eta) exp(-exp(eta)),
                  mu.eta  = function(eta) -exp(eta)* exp(-exp(eta)),
                  valideta= function(eta) TRUE,
                  name="loglog")
      class(new) <- "link-glm"
      new
  }
> sfit3d <- svyglm(pdeath ~rx + extent + node4 + factor(year), design=cpdesign,
                family= gaussian(link = bloglog()), subset=(year %in% c(2,4,6)))
> pfun(summary(sfit3d)$coefficients)
```

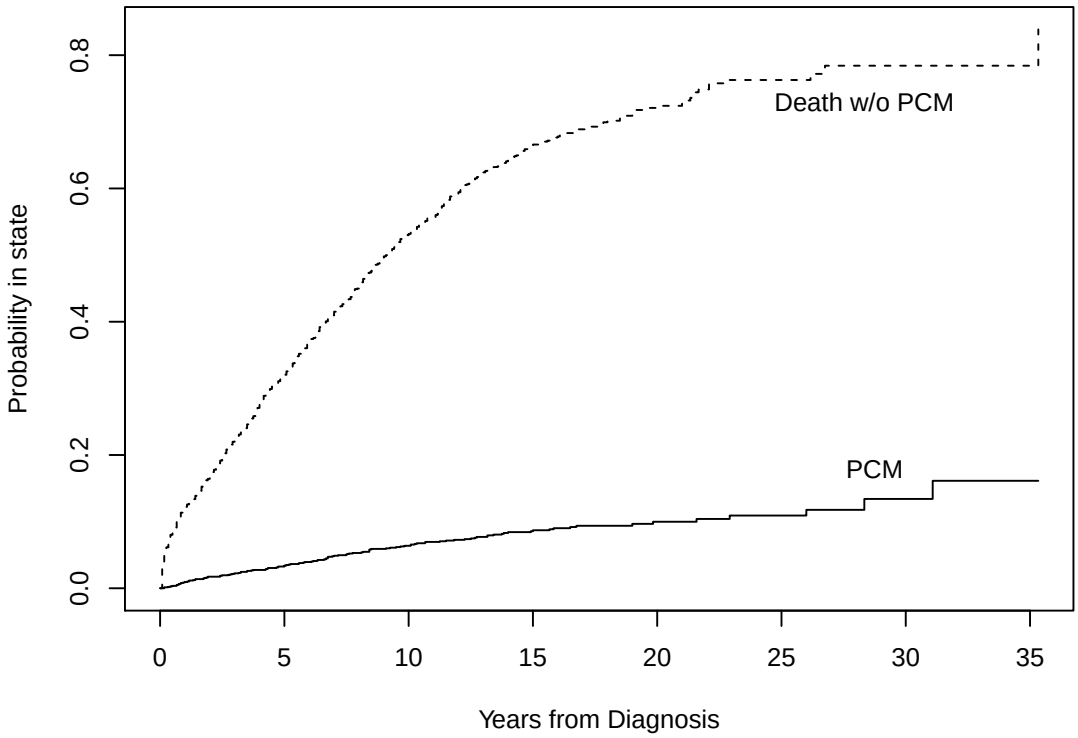
```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.438      0.292     4.9    1e-06
rxLev            0.038      0.112     0.3      0.7
rxLev+5FU        0.426      0.109     3.9    1e-04
extent          -0.498      0.099    -5.0    6e-07
node4           -0.901      0.118    -7.7    5e-14
factor(year)4   -0.269      0.029    -9.3   <2e-16
factor(year)6   -0.431      0.037   -11.7   <2e-16
> cfit <- coxph(Surv(time, status) ~ rx + extent + node4, cdata)
> round(summary(cfit)$coefficients, 3)
            coef exp(coef) se(coef)      z Pr(>|z|)
rxLev     -0.049     0.953    0.104 -0.468     0.64
rxLev+5FU -0.482     0.618    0.113 -4.262     0.00
extent     0.527     1.694    0.108  4.864     0.00
node4      0.819     2.268    0.093  8.849     0.00
```

We should in theory get similar coefficients to the `coxph` fit. Why are they larger?

4 Competing risks

Pseudo values based on the multi-state Aalen-Johansen estimate allow a look at the absolute risk of an outcome, in the presence of a competing risk. We will use the `mgus2` data set, which has been used to illustrate competing risks elsewhere in the vignettes.

```
> mdata <- mgus2
> mdata$etime <- with(mdata, ifelse(pstat==1, ptime, futime))
> mdata$event <- factor(with(mdata, ifelse(pstat==1, 1, 2*death)), 0:2,
                        c("censor", "PCM", "Death"))
> mdata$age10 <- mdata$age/10  # age in decades
> msurv <- survfit(Surv(etime, event) ~1, data=mdata, id=id)
> plot(msurv, lty=1:2, xscale=12,
       xlab="Years from Diagnosis", ylab="Probability in state")
> text(c(345, 340), c(.18, .73), c("PCM", "Death w/o PCM"))
```



At 30 years post diagnosis approximately 13% of the subjects have experienced a plasma cell malignancy (PCM), while 78% have died without PCM. The pseudo-value matrix has 3 columns correponding to each of the 3 states. We are interested in factors that promote malignancy, so model the PCM outcome at 30 years.

```
> mps <- pseudo(msurv, time=30*12, type="pstate")
> dim(mps)
[1] 1384    3
> mdata$ps.pcm <- mps[,2]
> mfit1 <- lm(ps.pcm ~ age10 + sex + mspike, data= mdata)
> pfun(summary(mfit1)$coefficients)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.397      0.129     3.1    0.002
age10         -0.053      0.017    -3.2    0.002
sexM          -0.054      0.041    -1.3    0.186
mspike         0.119      0.036     3.3    9e-04
```

We see that males have a slightly lower 30 risk, though not significant on this scale, which age and diagnosis and the sized of the serum monoclonal spike are very important. How do these results change when using more time points and/or a different link function?

```
> mps6 <- pseudo(msurv, time= 12*seq(5, 30, by=5),
                 data.frame=TRUE) # every 5 years
```

```
> mdata6 <- merge(mdata, subset(mps6, state== "PCM"), by="id")
> mdata6$year <- mdata6$time/12
> mdesign <- svydesign(data=mdata6, id=~id, weights=NULL, variables= ~. -id)
> mpfit1 <- svyglm(pseudo ~ age10 + sex + mspike,  design=mdesign,
                   subset= (year==10))
> mpfit2 <- svyglm(pseudo ~ age10 + sex + mspike,  design=mdesign,
                   subset= (year==20))
> mpfit3 <- svyglm(pseudo ~ age10 + sex + mspike,  design=mdesign,
                   subset= (year==30))
> mpfit6 <- svyglm(pseudo ~ age10 + sex + mspike + factor(year),  design=mdesign)
> temp <- rbind(yr10 = coef(mpfit1)[1:4], yr20= coef(mpfit2)[1:4],
                yr30 = coef(mpfit3)[1:4], all= coef(mpfit6)[1:4])
> round(temp,4)
     (Intercept)   age10    sexM mspike
yr10      0.0072  0.0002 -0.0167 0.0555
yr20      0.1316 -0.0188 -0.0090 0.0911
yr30      0.3970 -0.0527 -0.0537 0.1186
all       0.0873 -0.0187 -0.0192 0.0765
```

With multiple time points, the use of a simple linear link is not sensible: the spread between groups grows with time, and so the coefficients are not constant over time. The overall incidence rate of PCM in the study is nearly constant at 1% per year, so we allow for linear growth by adding time by covariate interactions.

```
> mpfit7 <- svyglm(pseudo ~ (age10 + sex + mspike) * year, design=mdesign)
> pfun(summary(mpfit7)$coefficients)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.14728    0.06120    -2.4    0.016
age10        0.02026    0.00827     2.4    0.014
sexM         0.00164    0.01798     0.1    0.927
mspike       0.02322    0.01781     1.3    0.192
year         0.01649    0.00530     3.1    0.002
age10:year  -0.00222    0.00073    -3.0    0.002
sexM:year   -0.00119    0.00141    -0.8    0.398
mspike:year  0.00305    0.00139     2.2    0.028
```

References

[1] P. K. Andersen and M. P. Perme. Pseudo-observations in survival analysis. *Stat Methods Medical Research*, 19:71–99, 2010.

[2] B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans.* SIAM, Philadelphia, 1982.