A Package for Survival Analysis in S

Terry M. Therneau
Mayo Foundation

January 27, 1999

Contents

1 Introduction

Over the last ten years I have been using the S package as a personal tool for my investigations of survival analysis. This work gained a large amount of momentum during my joint efforts with Patricia Grambsch and Tom Fleming on residuals for survival models, some of which is described in [53]. The set of routines based on that work has been posted to statlib for several years under the name "survival2" and is also included as a standard part of the S-plus package from Statistical Sciences. With the advent of the new object-oriented methods within S, laid out in the book "Statistical models in S" by John Chambers and Trevor Hastie [9], I became interested in adapting some of these methods to the survival package.

The execution of this task has turned out to be a substantial effort, but it afforded a chance to upgrade several of the routines with new features, and I think the collection has been much improved in both functionality and ease of use. Substantial opportunities for further improvements still remain, however.

Section 2 gives a very terse overview of the available commands, without attempting to explain the various options. Section 3 contains the formal statististical details for the methods.

Section 4 gives detailed examples. This is the place to expend most of your reading effort, at least for early users. Sections 5, 6 and 7 discuss three separate areas of the package's interaction with S in general: side effects, missing values, and conflicts with existing S functions.

There are several good texts on survival analysis; examples are Miller [43] and Kalbfleisch and Prentice [33]. A more technical treatment of the Cox model, including the important extensions of time dependent-covariates and multiple events, can be found in Fleming and Harrington [20], and in Andersen et al. [2].

My thanks to Sin-Ho Jung, Paul Novotny, and particularly Peter Thall for helpful comments on an earlier draft of this manuscript. Frank Harrell continues to provide feedback on the methods, along with new ideas for improvement. (If I ever find time to implement all of Frank's suggestions, this will become an *amazing* survival package!)

2 Overview

The summary below is purposefully very terse. If you are familiar with survival analysis *and* with other S modeling functions it will provide a good summary. Otherwise, just skim the section to get an overview of the type of computations available from this package, and move on to section 3 for a fuller description.

Surv() A *packaging* function; like I() and C() it doesn't transform its argument. This is used for the left hand side of all the formulas.

- `Surv(time, status)` – right censored data
- `Surv(time, s==3)` – right censored data, a value of 3 = death
- `Surv(t1, t2, status)` – counting process data, as in the current agreg() function
- `Surv(t1, ind, type='left')` – left censoring
- Methods

  Math.Surv
  Ops.Surv
  Summary.Surv
  [.Surv
  is.na.Surv
  print.Surv

coxph() Cox's proportional hazards model.

- `coxph(Surv(time, status) ~x, data=aml)` – standard Cox model
- `coxph(Surv(t1, t2, stat) ~ (age + surgery)* transplant)` – time dependent data.
- `y <- Surv(t1, t2, stat)`
  `coxph(y ~ strata(inst) * sex + age + treat)` – Stratified model, with a separate baseline per institution, and institution specific effects for sex.
- `coxph(y ~ offset(x1) + x2)` – force in a known term, without estimating a coefficient for it.
- Options

  method='breslow' the most common approximation for tied death times
  method='efron' a more accurate method for handling ties (default)
  method='exact' the exact partial likelihood (conditional logistic)
  subset selection
  missing value actions
  convergence options
- Methods

  print
  summary
  residuals — martingale, deviance, score, and Schoenfeld
  predict — $X'\beta$, risk, expected # events, and terms
  plot
  surv.fit — see below

survreg Parametric survival models.

- `survreg(Surv(time, stat) ~ x, dist='logistic')` - Fit a log-logistic distribution.
- Options

  distribution = extreme value, logistic, or Gaussian
  link = log or identity
  optionally fix the scale parameter
  subset selection
  missing value actions
  convergence options
- Methods: like coxph (but not all are completed yet)

survdiff One and k-sample versions of the Fleming-Harrington G^ρ family. Includes the logrank and Gehan-Wilcoxon as special cases.

- `survdiff(Surv(time, status) ~ sex + treat)` – Compare th e 4 sub-groups formed by sex and treatment combinations.
- `survdiff(Surv(time, status) ~ offset(pred))` - One-sample test
- Methods

  print

survexp Predicted survival for an age and sex matched cohort of subjects, given a baseline matrix of known hazard rates for the population. Most often these are US mortality tables, but we have also used local tables for stroke rates.

- `survexp(entry.dt, birth.dt, sex)` – Defaults to US white, average cohort survival
- `pred <- survexp(entry, birth, sex, futime, type='individual')` Data to enter into a one sample test for comparing the given group to a known population.
- Options

  matrix of known hazards (US white and Minnesota white are included free!)
  time points at which to compute a result (expected survival)
  individual or cohort output
- Methods: inherits from survfit

survfit Fit a survival curve.

- `survfit(Surv(time, status))` – Simple Kaplan-Meier
- `survfit(Surv(time, status) ~ rx + sex)` – Four groups
- `fit <- coxph(Surv(time, stat) ~ rx + sex)`
  `survfit(fit)` – Prediction at the mean x's
- `survfit(fit, list(rx=1, sex=2))` – Predict elsewhere than the mean
- Options

  Greenwood or Tsiatis variance estimate
  subset selection
  missing value actions
  case weights (not the same as risk weights)
  Kaplan-Meier or Fleming-Harrington estimate of survival
  log, log-log, plain, or no confidence intervals
- Methods

  print
  summary
  plot
  lines
  points

strata marks a variable or group of variables as strata. Similar to the base S function `interaction` except: coxph notices it as special, and there is a different labeling style.

- `strata(rx)`
- `strata(rx, sex)`
- `strata(rx, na.group=T)` – Make NA a separate group rather than omitting NA.

cox.zph Computes a test of proportional hazards for the fitted Cox model.

- `cox.zph(coxfit)`
- Options:

  transform – Optional time transform
  global=T/F – Include the global test
- Methods

  print
  plot

naresid, naprint A new idea for missing values.

- A change to the base function `model.frame` allows a global na.action to be specified, e.g. `options(na.action='na.omit')`.
- The na.action routine adds an attribute *na.action* to the data, with appropriate class, e.g., class='omit' for na.omit(). Coxph and other routines add this to their output without modification.
- The print routines will call naprint(fit$na.action). The returned character string is added to the printout. My naprint.omit returns "14 observations deleted due to missing".
- The resid and predict routines call `x<- naresid(fit$na.action, x)`, where `x` is a vector or matrix. The naresid.omit routine re-inserts missing values so that the residual matches the input data.

3 Mathematical Backround

Let $Y_i(t)$, $i = 1, \ldots, n$ be the indicator that subject i is at risk and under observation at time t. Let $N_i(t)$ be the step function for the ith subject, which counts the number of "events" for that subject up to time t. The total number of deaths that have occurred up to time t will be $\overline{N}(t) = \sum N_i(t)$, and the number of subjects at risk at time t will be $\overline{Y}(t) = \sum Y_i(t)$. It will also be useful to define $d(t)$ as the number of deaths that occur exactly at time t.

3.1 Survival

3.1.1 Estimation

The most common estimate of the survival distribution, the Kaplan-Meier (KM) estimate, is a product of survival probabilities:

$$\hat{S}_{KM}(t) = \prod_{t_i < t} \frac{\overline{Y}(t_i) - d(t_i)}{\overline{Y}(t_i)} \,.$$

Graphically, the Kaplan-Meier survival curve appears as a step function with a drop at each death. Censoring times are often marked on the plot as "+" symbols.

Another approach is to estimate Λ, the cumulative hazard, using Nelson's estimate:

$$\hat{\Lambda}(t) = \sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{\overline{Y}(s)}.$$

The Nelson estimate is also a step function, it starts at zero and has a step of size $d(t)/\overline{Y}(t)$ at each death. One problem with the Nelson estimate is that it is susceptible to ties in the data. For example, assume that 3 subjects die at 3 nearby times t_1, t_2, t_3, with 7 other subjects also at risk. Then the total increment in the Nelson estimate will be $1/10 + 1/9 + 1/8$. However, if time data were grouped such that the distinction between t_1, t_2 and t_3 were lost the increment would be the lesser step $3/10$. If there are a large number of ties this can introduce significant bias. One solution is to employ a modified Nelson estimate that always uses the larger increment, as suggested by Nelson [46] and Fleming and Harrington [19]. This is not an issue with the Kaplan-Meier, with or without ties the multiplicative step will be $7/10$.

The relationship $\Lambda(t) = -\log S(t)$, which holds for any continuous distribution leads to the Fleming-Harrington (FH) [19] estimate of survival:

$$\hat{S}_{FH}(t) = e^{-\hat{\Lambda}(t)}\,. \tag{1}$$

This estimate has natural connections to survival curves for a Cox model. For sufficiently large sample sizes the FH and KM estimates will be arbitrarily close to one another, but keep in mind the fact that unless there is heavy censoring n is always small in the right hand tail of the estimated curve.

3.1.2 Variance

Several estimates of the variance of $\hat{\Lambda}$ are possible. Since $\hat{\Lambda}$ can be treated as a sum of independent increments, the variance is a cumulative sum with terms of

$$\begin{array}{ll} \frac{d(t)}{\overline{Y}(t)[\overline{Y}(t)-d(t)]} & \text{Greenwood} \\[2ex] \frac{d(t)}{\overline{Y}^2(t)} & \text{Aalen} \\[2ex] \frac{d(t)[\overline{Y}(t)-d(t)]}{\overline{Y}^3(t)} & \text{Klein,} \end{array}$$

see Klein [34] for details. Using equation (1) and the simple Taylor series approximation $\text{var} \log f \approx \text{var} f/f^2$, the variance of the KM or FH estimators is

$$\text{var}(\hat{S}(t)) = \hat{S}^2(t)\text{var}(\hat{\Lambda}(t))\,. \tag{2}$$

Klein also considers two other forms for the variance of S, but concludes that

- For computing the variance of $\hat{\Lambda}$ the Aalen formula is preferred.
- For computing the variance of $\hat{S}$, the Greenwood formula along with (2) is preferred.

Confidence intervals for $\hat{S}(t)$ can be computed on the plain scale:

$$\hat{S} \pm 1.96\,\mathrm{se}(\hat{S})\,, \tag{3}$$

on the cumulative hazard or log survival scale:

$$\exp[\log(\hat{S}) \pm 1.96\,\mathrm{se}(\hat{\Lambda})]\,, \tag{4}$$

or on a log-hazard scale:

$$\exp(-\exp[\log(-\log(\hat{S}) \pm 1.96\,\mathrm{se}(\log\hat{\Lambda})])\,, \tag{5}$$

where se refers to the standard error.

Confidence intervals based on (3) may give survival probabilities greater than 1 or less than zero. Those based on (4) may sometimes be greater than 1, but those based on (5) are always between 0 and 1. For this reason many users prefer the log-hazard formulation. Confidence intervals based on the logit of S are another alternative. Link [39, 40], however suggests that confidence intervals based on the cumulative hazard scale have the best performance. This is not surprising given their relation to the independent-increments formulation of $\hat{\Lambda}$.

A further refinement to the confidence intervals is explored in Dorey and Korn [12]. When the tail of the survival curve contains much censoring and few deaths, there will be one or more long flat segments. In all of the procedures considered thus far, the confidence intervals are constant across these intervals. Yet, it seems reasonable that as censored subjects are removed from the sample and the effective sample size decreases, that the actual reliability of the curve would also decrease. The proposed correction retains the original upper confidence limit ($S(t)$ does not rise!), and a modified lower limit which agrees with the standard limits at each death time but is based on the *effective* n between death times.

Two methods are implemented within `summary.survfit`. Peto's method assumes that $\mathrm{var}(\hat{\Lambda}(t)) = c/\overline{Y}(t)$, where $\overline{Y}$ is the number at risk and $c \equiv 1 - \hat{S}(t)$. The Peto limit is known to be conservative; the modified Peto limit chooses c such that the variance at each death time is equal to the usual estimate. The modified estimate is equal to the usual variance multiplied by $n^*/\overline{Y}$, where $\overline{Y}$ is the number at risk and n^* is the number at risk at the last jump in the curve (last death time). This is the recommended solution. A different modification is explored in Dorey and Korn, but they found it to be anti-conservative (it is also harder to calculate).

3.1.3 Mean and median

For the Kaplan-Meier estimate, the estimated mean survival is undefined if the last observation is censored. One solution, used here, is to redefine the estimate to be zero beyond the last observation. This gives an estimated mean that is biased towards zero, but there are no compelling alternatives that do better. With this definition, the mean is estimated as

$$\hat{\mu} = \int_0^T \hat{S}(t)dt$$

where $\hat{S}$ is the Kaplan-Meier estimate and T is the maximum observed follow-up time in the study. The variance of the mean is

$$\mathrm{var}(\hat{\mu}) = \int_0^T \left(\int_t^T \hat{S}(u)du\right)^2 \frac{d\overline{N}(t)}{\overline{Y}(t)(\overline{Y}(t) - \overline{N}(t))}$$

where $\bar{N} = \sum N_i$ is the total counting process and $\bar{Y} = \sum Y_i$ is the number at risk.

The sample median is defined as the first time at which $\hat{S}(t) \leq .5$. Upper and lower confidence intervals for the median are defined in terms of the confidence intervals for S: the upper confidence interval is the first time at which the upper confidence interval for $\hat{S}$ is $\leq .5$. This corresponds to drawing a horizontal line at 0.5 on the graph of the survival curve, and using intersections of this line with the curve and its upper and lower confidence bands. In the very rare circumstance that the survival curve has a horizontal portion at exactly 0.5 (e.g., an even number of subjects and no censoring before the median) then the average time of that horizonal segment is used. This agrees with usual definition of the median for even n in uncensored data.

3.2 Expected Survival

3.2.1 Individual Expected Survival

The survival tables published by the Department of the Census contain 1 year survival probabilities by age and sex, optionally subgrouped as well by race and geographic region. The entry for age 21 in 1950 is the probability that a subject who turns 21 during 1950 will live to his or her 22nd birthday. The tables stored in S contain the daily hazard rate λ rather than the probability of survival p

$$p = \exp(-365.25 * \lambda)$$

for convenience. If a, s, y are subscripts into the age by sex by calendar year table of rates, then the cumulative hazard for a given subject is the simply the sequential sum of $\lambda_{asy} *$ number of days in state a, s, y. Looked at another way, the patient progresses through the rate table on a diagonal line whose starting point is (date of entry, age at entry, sex), see Berry [6] for a nice graphical illustration.

Let $\lambda_i(t)$ and $\Lambda_i(t)$ be the derived hazard and cumulative hazard functions, respectively, for subject i, starting at their time of entry to the study. Then $S_i(t) = \exp(-\Lambda_i(t))$ is the subject's expected survival function.

Some authors use the product form $S = 1 - \prod(1 - q_k)$ where the q are yearly probabilities of death, and yet others an equation similar to actuarial survival estimates. Numerically it makes little difference which form is chosen, and the S functions use the hazard based formulation for its convenience.

3.2.2 Cohort expected survival

The expected survial curve for a cohort of n subjects is an "average" of the n individual survival curves for the subjects. There are 3 main methods for combining these; for some data sets they can give substantially different results. Let S_e be the expected survival for the cohort as a whole, and S_i, λ_i be the individual survival and hazard functions. All three methods can be written as

$$S_e(t) = \exp\left(-\int_0^t \frac{\sum \lambda_i(s) w_i(s)}{\sum w_i(s)}\, ds\right) \tag{6}$$

and differ only in the weight function w_i.

A weight function of $w_i(t) = S_i(t)$ corresponds to the *exact* method. This is the oldest and most commonly used technique, and is described in Ederer, Axtel and Cutler [13]. An equivalent expression for the estimate is

$$S_e(t) \quad = \quad (1/n) \sum S_i(t) \tag{7}$$

The exact method corresponds to selecting a population matched control for each subject in the study, and then computing the expected survival of this cohort *assuming complete follow-up.*

The exact method is most appropriate when doing forcasting, sample size calculations or other predictions of the "future" where censoring is not an issue.

A common use of the expected survival curve is to plot it along with the Kaplan-Meier of the sample in order to assess the relative survival of the study group. When used in this way, several authors have shown that the Ederer method can be misleading if censoring is not independent of age and sex (or whatever the matching factors are for the referent population). Indeed, independence if often not the case. For example, in a long study it is not uncommon to allow older patients to enroll only after the initial phase. A severe example of this is demonstrated in Verheul et al. [56], concerning aortic valve replacement over a 20 year period. The proportion of patients over 70 years of age was 1% in the first ten years, and 27% in the second ten years. Assume that analysis of the data took place immediately at the end of the study period. Then the Kaplan-Meier curve for the latter years of follow-up time is guarranteed to be "flatter" than the earlier segment, because it is computed over a much younger population. The Ederer or exact curve will not reflect this bias, and makes the treatment look better than it is. The exact expected survival curve forms a reference line, in reality, for what the Kaplan-Meier will be when followup is complete, rather than for what the Kaplan-Meier is now.

In Hakulinen's method [25, 26], each study subject is again paired with a fictional referent from the cohort population, but this referrent is now treated as though he/she were followed in the same way as the study patients. Each referrent thus has a maximum *potential* follow-up, i.e., they will become censored at the analysis date. Let $c_i(t)$ is a censoring indicator which is 1 during the period of potential follow-up and 0 thereafter; the weight function for the Hakulinen or *cohort* method is $w_i(t) = S_i(t)c_i(t)$.

If the study subject is censored then the referrent would presumably be censored at the same time, but if the study subject dies the censoring time for his/her matched referrent will be the time at which the study subject *would have been censored.* For observational studies or clinical trials where censoring is induced by the analysis date this should be straightforward, but determination of the potential follow-up could be a problem if there are large numbers lost to follow-up. (However, as pointed out long ago by Berkeson, if a large number of subjects are lost to follow-up then any conclusion is subject to doubt. Did patients stop responding to follow-up letters at random, because they were cured, or because they were at death's door?)

In practice, the program will be invoked using the actual follow-up time for those patients who are censored, and the *maximum* potential follow-up for those who have died. By the maximum potential follow-up we mean the difference between enrollment date and the average last contact date, e.g., if patients are contacted every 3 months on average and the study was closed six months ago this date would be 7.5 months ago. It may true that the (hypothetical) matched control for a case who died 30 years ago would have little actual chance of such long follow-up, but this is not really important. Almost all of the numerical difference between the Ederer and Hakulinen estimates results from censoring those patients who were most recently entered on study. For these recent patients, presumably, enough is known about the operation of the study to give a rational estimate of potential follow-up.

The Hakulinen formula can be expressed in a product form

$$S_e(t+s) = S_e(t) * \frac{\sum p_i(t,s)S_i(t)c_i(t)}{\sum S_i(t)c_i(t)}, \tag{8}$$

where $p_i(t,s)$ is the conditional probability of surviving from time t to time $t+s$, which is $\exp(\Lambda_i(t) - \Lambda_i(t+s))$. The formula is technically correct only over time intervals $(t, t+s)$ for which c_i is constant for all i, i.e., censoring only at the ends of the interval.

The conditional estimate is advocated by Verheul [56], and was also suggested as a computation simplification of the exact method by Ederer and Heise [14]. For this estimate the weight function $w_i(t)$ is defined to be 1 while the subject is alive and at risk and 0 otherwise. It is clearly related to Hakulinen's method, since $E(w_i(t)) = S_i(t)c_i(t)$.

Most authors present the estimator in the product-limit form $\prod[1 - d(t)/n(t)]$, where d and n are the numerator and denominator terms within the integral of equation (6). One disadvantage of the product-limit form is that the value of the estimate at time t depends on the number of intervals into which the time axis has been divided, for this reason we use the integral form (6) directly.

One advantage of the conditional estimate, shared with Hakulinen's method, is that it remains consistent when the censoring pattern differs between age-sex strata. This advantage was not noted by the Ederer and Heise, and the "exact" calculation was adapted as the preferred method [13, 25]. A problem with the conditional estimator is that it has a much larger variance than either the exact or Hakulinen estimate. In fact, the variance of these latter two can usually be assumed to be zero, at least in comparison to the variance of the Kaplan-Meier of the sample. Rate tables are normally based on a very large sample size so the individual λ_i are very precise, and the censoring indicators c_i are based on the the study design rather than on patient outcomes. The conditional estimate $S_c(t)$, however, depends on the actual death times and w_i is a random variable. I am not aware of a reference where this variance has been worked out, however.

The main argument for use of the conditional estimate, however, is that we often want to make conditional statements about the survival. For instance, in studies of a surgical intervention such as hip replacement, the observed and expected survival curves often will initially diverge due to surgical mortality, and then appear to become parallel. It is tempting to say that survival beyond hospital discharge is "equivalent to expected". This is a conditional probability statement, and it should not be made unless a conditional estimate was used.

A hypothetical example may make this clearer. For simplicity assume no censoring. Suppose we have studies of two diseases, and that their age distributions at entry are identical. Disease A kills 10% of the subjects in the first month, independent of age or sex, and thereafter has no effect. Disease B also kills 10% of its subjects in the first month, but predominately affects the old. After the first month it exerts a continuing though much smaller force of mortality, still biased toward the older ages. With proper choice of the age effect, studies A and B will have almost identical survival curves; as the patients in B are always younger, on average, than those in A. Two different questions can be asked under the guise of "expected survival":

- What is the overall effect of the disease? In this sense both A and B have the same effect, in that the 5 year survival probability for a diseased group is $x\%$ below that of a matched population cohort. The Hakulinen estimate would be preferred because of its lower variance. It estimates the curve we "would have gotton" if the study had included a control group.

- What is the ongoing effect of the disease? Detection of the differential effects of A and B after the first month requires the conditional estimator. We can look at the slopes of the curves to judge if they have become parallel.

The actual curve generated by the conditional estimator remains difficult to interpret, however. One wag in our department has suggested calling it the "lab rat" estimator, since the control subject is removed from the calculation ("sacrificed") whenever his/her macthing case dies. I suggest that Hakulinen's cohort estimate is the most appropriate estimator. If there is a question about delayed effects, as in the above example (there would be an apparent flattening of the Kaplan-Meier curves after the first month), then one can plot a new curve using only those patients who survived at least one month.

Other suggestions for exploring conditional effects can be found in the literature under the heading of relative survival. Hakulinen [27] for instance, suggests dividing the patients into disjoint age groups and computing the ratio of observed/expected survival separately within each strata. However, this estimate can have an unacceptable variance due to small numbers in the subgroups.

3.2.3 Approximations

The Hakulinen cohort estimate (8) is "Kaplan-Meier like" in that it is a product of conditional probabilities and that the time axis is partitioned according to the observed death and censoring times. Both the exact and conditional estimators can be written in this way as well. They are unlike a KM calculation, however, in that the ingredients of each conditional estimate are the n distinct individual survival probabilities at that time point rather than just a count of the number at risk. For a large data set this requirement for $O(n)$ temporary variables can be a problem. An approximation is to use longer intervals, and allow subjects to contribute partial information to each interval. For instance, in (8) replace the 0/1 weight $c_i(t)$ by $\int_t^{t+s} c_i(u)du/s$, which is the proportion of time that subject i was uncensorted during the interval $(t, t+s)$. If those with fractional weights form a minority of those at risk during the interval the approximation should be reliable. (More formally, if the sum of their weights is a minority of the total sum of weights). By Jensen's inequality the approximation will always be biased upwards, but it is very small. For the Stanford heart transplant data used in the examples an exact 5 year estimate using the cohort method is 0.94728, an approximate cohort computation using only the half year intervals yields 0.94841.

The Ederer estimate is unchanged under re-partioning of the time axis.

3.2.4 Testing

All of the above discussion has been geared towards a plot of $S_e(t) = \exp(-\Lambda_e(t))$, which attempts to capture the proportion of patients who will have died by t. When comparing observed to expected survival for testing purposes, an appropriate test is the one-sample log-rank test [28] $(O-E)^2/E$, where O is the observed number of deaths and

$$\begin{aligned} E &= \sum_{i=1}^{n} e_i \\ &= \sum_{i=1}^{n} \int \lambda_i(s) Y_i(s) \end{aligned} \tag{9}$$

is the expected number of deaths, given the observation time of each subject. This follows Mantel's concept of 'exposure to death' [42], and is the expected number of deaths during this exposure. Notice how this differs from the expected number of deaths $nS_e(t)$ in the matched cohort at time t. In particular, E can be greater than n. Equation (9) is referred to as the person-years estimate of the expected number of deaths. The log-rank test is usually more powerful than one based on comparing the observed survival at time t to $S_e(t)$; the former is a comparison of the entire observed curve to the expected, and the latter is a test for difference at one point in time.

Tests at a particular time point, though less powerful, will be appropriate if some fixed time is of particular interest, such as 5 year survival. In this case the test should be based on the cohort estimate. The H_0 of the test is "is suvival different that what a control-group's survival would have been". A pointwise test based on the exact estimate may well be invalid if there is censoring. A pointwise test based on the conditional estimate has two problems. The first is that an appropriate variance is difficult to construct. The second, and more damming one, is that it is unclear exactly what alternative is being tested against.

Hartz, Giefer and Hoffman [29] argue strongly for the pointwise tests based on a expected survival estimate equivalent to (8), and claim that such a test is both more powerful and more logical than the person-years approach. Subsequent letters to the editor [30, 31] challenged these views, and it appears that the person-years method is preferred.

Berry [6] provides an excellent overview of the the person-years method. Let the e_i be the expected number of events for each subject, treating them as an $n=1$ Poisson process. We have

$$\begin{aligned} e_i &= \int_0^\infty Y_i(s) \lambda_i(s) ds \\ &= \Lambda_i(t_i)\,, \end{aligned}$$

where t_i is the observed survival or censoring time for a subject. This quantity e_i is the total amount of hazard that would have been experienced by the population-matched referrent subject, over the time interval that subject i was actually under observation. If we treat e_i as though it were the follow-up time, this corrects for the backround mortality by, in effect, mapping each subject onto a time scale where the baseline hazard is 1.

Tests can now be based on a Poisson model, using δ_i as the response variable (1=dead, 0=censored), and e_i as the time of observation (an `offset` of $\log e_i$). The intercept term of the model estimates the overall difference in hazard between the study subjects and the expected population. An intercept-only model is equivalent to the one sample log-rank test. Covariates in the model estimate the effect of a predictor on *excess* mortality, whereas an ordinary Poisson or Cox model would estimate its effect on total mortality.

Andersen and Væth [4] consider both multiplicative and additive models for excess risk. Let λ_i^* be the actual hazard function for the individual at risk and λ_i be, as before, that for his/her matched control from the population. The multiplicative hazard model is

$$\lambda_i^*(t) = \beta(t)\lambda_i(t)\,.$$

If $\beta(t)$ were constant, then

$$\hat{\beta}_0 \equiv \frac{\sum N_i}{\sum e_i}$$

is an estimate of the *standard mortality ratio* or SMR, which is identical to `exp(intercept)` in the Poisson model used by Berry (assuming a log link). Their estimate over time is based on a modified Nelson hazard estimate

$$\hat{B}'(t) = \int_0^t \frac{\sum dN_i(s)}{\sum Y_i(s)\lambda_i(s)} ds\,,$$

which estimates the integral of $\beta(t)$. If the SMR is constant then a plot of $\hat{B}'(t)$ versus t should be a straight line through the origin.

For the additive hazard model

$$\lambda_i^*(t) = \alpha(t) + \lambda_i(t)$$

the integral $A(t)$ of α is estimated as $\log[S_KM(t)/S_c(t)]$, the difference between the Kaplan-Meier and the conditional estimator, when plotted on log scale. Under the hypothesis of a constant additive risk, a plot of $\hat{A}(t)$ versus t should approximate a line through the origin.

3.3 Cox Model

Let $Z_{ij}(t)$ be the jth covariate of the ith person (possibly time dependent), where $i = 1, \ldots, n$ and $j = 1, \ldots, p$; and $Z_i(t)$ be the entire covariate set for a subject, represented as a $p \times 1$ column vector. Define $r_i(t)$ to be $\exp[\beta' Z_i(t)]$, i.e., the risk score for the ith subject. In actual practice β will be replaced by $\hat{\beta}$ and the subject weights r_i by $\hat{r}_i$.

The Cox model assumes that the risk for subject i is

$$\lambda(t; Z_i) = \lambda_0(t) r_i(t)$$

where λ_0 is an unspecified baseline hazard. Assuming no tied death times, the log partial likelihood is defined as

$$l(\beta) = \sum_{i=1}^{n} \int_0^\infty \left[Y_i(t) r_i(t) - \log\{\sum_j Y_j(t) r_j(t)\} \right] dN_i(t) \,.$$

The first derivative is the p by 1 vector

$$\begin{aligned} U(\beta) &= \sum_{i=1}^{n} \int_0^\infty [Z_i(t) - \bar{Z}(\beta,t)]\, dN_i(t) && (10) \\ &= \sum_{i=1}^{n} \int_0^\infty [Z_i(t) - \bar{Z}(\beta,t)]\, dM_i(\beta,t) && (11) \end{aligned}$$

and the p by p information matrix is

$$\mathcal{I}(\beta) = \sum_{i=1}^{n} \int_0^\infty \frac{\sum_j Y_j(t) r_j(t) [Z_i(t) - \bar{Z}(t)][Z_i(t) - \bar{Z}(\beta,t)]'}{\sum_j Y_j(t) r_j(t)} dN_i(t) \,, \qquad (12)$$

where $\bar{Z}$ is the weighted mean of those still at risk at time t

$$\bar{Z}(\beta,t) = \frac{\sum Y_i(t) r_i(t) Z_i(t)}{\sum Y_i(t) r_i(t)} \,.$$

The martingale residual M_i is defined below.

The above notation is derived from the counting process representation, as found in Fleming and Harrington [20]. It allows very naturally for several extensions to the original Cox model formulation:

- multiple events per subject,
- discontinuous intervals of risk — Y_i may change states from 1 to 0 and back again multiple times,
- left truncation — subjects need not enter the risk set at time 0.

This extension is known as the *multiplicative hazards model.*

3.3.1 Computation

The S function `coxph` accommodates these extensions by a simple programming artifice. The input data set is assumed to consist of observations or rows of data, each of which contains the covariate values Z, a status indicator 1=event 0=censored and an optional stratum indicator, along with the time interval $(start, stop]$ over which this information applies. In the notation above, this means that each row is treated as a separate subject whose Y_i variable is 1 on the interval $(start, stop]$ and zero otherwise. Within the program, it means that the risk set at time t only uses the applicable rows of the data.

The code has no specific "hooks" to accommodate time-dependent covariates, time-dependent strata, multiple events, or any of the other special features mentioned above. Rather, it is the responsibility of the user to first construct an appropriate data set. The strategy, originally motivated by sloth, led to a fitting program that is simpler, shorter, easier to debug, and more efficient than one with multiple specific options. A significantly more important benefit has become apparent over time, i.e., the flexibility inherent in building a data set has allowed analyses that were not considered by the original coder — left truncation is a case in point.

The more common way to deal with time-dependent Cox models is to have a computation occur at each death time. For example, BMDP and SAS PHREG do this. One advantage of that procedure over this one is the ability to code continuously time-dependent covariates: `coxph` only accommodates step functions. However, I have yet to find an example where this was a deficiency. In the common case of repeated measurements on each subject, the data set for `coxph` is quite easy to set up, since it and the original measurements consist of one line of data per visit. On a small set of cases, I have compared these results to a fit using full linear interpolation of the covariate, and the regression coefficients were essentially identical.

When only a few subjects have time-dependent variates, this method can be much more efficient. In a recent study here with multiple lab tests, the proportion of subjects with 1,2, etc. tests was geometrically decreasing. Only 2/100 patients had 5 values. Thus most patients had only 1 line of data in the constructed `coxph` data set. Let $r(t)$ be the number of subjects in the risk set at death time t, p be the number of covariates, and $s(t)$ the number of rows in this strata (when set up for `coxph`). The S calculation has at each death time a search over $s(t)$ terms with a sum over $r(t)$ of them, the BMDP calculation has a sum over $r(t)$ terms, each of which requires a call to the computation subroutine. So the total S time is $O(p * p * E(r)) + O(a * E(s))$, and the total BMDP time is $O(p * p * E(r)) + O(b * E(r))$, each times the number of events. If the subroutine is at all complex ($b >> a$) then S wins.

The `coxph` function will often run much faster when there are stratification variables in the model. When strata are introduced the program spends less time searching out whom is part of the current risk set since it need look only within the strata; without strata it has to scan the entire data set.

If the start time is omitted, it is assumed that $start = 0$. In this case the algorithm is equivalent to a standard Cox model. Computation is more rapid, since the risk sets can be accumulated rather than performing a separate search per death time.

3.3.2 Residuals

The Breslow (or Tsiatis or Link) estimate of the baseline hazard is

$$\hat{\Lambda}_0(\beta, t) = \int_0^t \frac{\sum_{i=1}^n dN_i(s)}{\sum_{i=1}^n Y_i(s) r_i(\beta, s)}.$$

The martingale residual at time t is

$$M_i(t) = N_i(t) - \int_0^t r_i(\beta, s) Y_i(s) d\hat{\Lambda}_0(\beta, s).$$

The program returns the residual at $t = \infty, \beta = \hat{\beta}$. If there are no time-dependent covariates, then $r_i(t) = r_i$ and can be factored out of the integral, giving $\widehat{M}_i = N_i - \hat{r}_i \hat{\Lambda}_0(\hat{\beta}, t_i)$.

The deviance residual is a normalizing transform of the martingale residual

$$d_i = \text{sign}(\widehat{M}_i) * \sqrt{-\widehat{M}_i - N_i \log((N_i - \widehat{M}_i)/N_i)}$$

In practice, it has not proven to be very useful.

The other two residuals are based on the score process $U_{ij}(t)$ for the ith subject and the jth variable:

$$U_{ij}(\beta, t) = \int_0^t (Z_{ij}(s) - \bar{Z}_j(\beta, s)) d\widehat{M}_i(\beta, s)\,.$$

The score residual is then defined, for each subject and each variable (an n by p matrix) as $U_{ij}(\hat{\beta}, \infty)$. It is the sum of the score process over time. The usual score vector $U(\beta)$, as in equation (10), is the column sum of the matrix of score residuals.

The martingale and score residuals are integrals over time of some object. One consequence of this is that they work naturally for the `coxph` formulation. Specificly, in setting up a multiplicative hazard model, a single subject is broken up into multiple lines of the input data, as though he were a set of different individuals observed over disjoint times. After the coxph function is finished, the residual for that person is just the sum of the residuals for these "pseudo" subjects. This property is not true for the deviance residual, however.

The Schoenfeld residuals [50] are defined as a matrix

$$s_{ij}(\beta) = Z_{ij}(t_i) - \bar{Z}_j(\beta, t_i) \tag{13}$$

with one row per death time and one column per covariate, where i, t_i are the subject and the time that the event occurred. Again, this works very well with the `coxph` formulation, since the residuals are completely independent of how subjects are broken up into time intervals. The Schoenfeld residuals are also related to the score process U_{ij}. Sum the score process up over individuals to get a total score process $\sum_i L_{ij}(\beta, t) = U(\beta, t)$. This is just the score vector at time t, so that at $\hat{\beta}$ we must have $U(\hat{\beta}, 0) = U(\hat{\beta}, \infty) = 0$. Because $\hat{\Lambda}$ is discrete, our estimated score process will also be discrete, having jumps at each of the unique death times. There are two simplifying identities for these residuals:

$$U(\beta, t) = \sum_i \int_0^t Z_{ij}(s) dM_i(\beta, s) = \sum_i \int_0^t (Z_{ij}(s) - \bar{Z}_j(\beta, s)) dN_i(s) \qquad (14)$$

Note that $d\widehat{M}_i(t)$ is zero when subject i is not in the risk set at time t. Since the sums are the same for all t, each increment of the processes must be the same as well. Comparing the second of these to (13), we see that the Schoenfeld residuals are the increments or jumps in the total score process. There is a small nuisance with tied death times: under the integral formulation the $O - E$ process has a single jump at each death time, leading to one residual for each unique event time, while under the Schoenfeld representation there is one residual for each event. In practice, the latter formulation has been found to work better for both plots and diagnostics, as it leads to residuals that are approximately equivariant. For the alternative of one residual per *unique* death time, both the size and variance of the residual is proportional to the number of events.

The last and most general residual is the entire score process R_{ijk} where i indexes subjects, k indexes the event times, and j indexes the covariate.

$$R_{ijk} = [Z_{ij}(t_k) - \bar{Z}_j(t_k)][dN_i(t_k) - r_i(t_k) d\hat{\Lambda}_0(t_k)].$$

The score and Schoenfeld residuals are seen to be marginal sums of this array. Lin, Wei and Ying [38] suggest a global test of the proportional hazards model based on the maximum of the array.

3.3.3 Variance of the Residuals

Plots of the martingale residuals, and the subsequent fits that search for an appropriate functional form, may benefit from an adjustment for the variance of each residual. This is particularly true in a counting process model where some subjects may generate many "observations" whilst others contribute only 1 or 2; thus the amount of information per observation, i.e. the expected count, will vary markedly.

The martingale residuals are essentially an (observed − expected) number of events for each subject, $M_i = O_i - E_i$. Not surprisingly, they are approximately independent with variance of E_i, similar to a Poisson process or the counts in a 2 way table. Since the residuals must sum to zero, they are not precisely independent.

Chen and Wang [10] derive an exact variance matrix for the residuals. Let $\eta = X\hat\beta$ be the vector of linear predictors for the n subjects. Then with straightforward but tedious algebra we obtain the derivatives of the log partial likelihood l

$$\begin{aligned}\frac{\partial l}{\partial \eta} &= M = O - E \\ \frac{\partial^2 l}{\partial \eta^2} &= V = \text{diag}(E) - A\,,\end{aligned}$$

where M is the vector of martingale residuals, E is the vector of expected events, and A is defined below. By analogy with other maximum likelihood expressions, we may consider V to be the variance matrix for M, and Chen and Wang suggest using the adjusted residual $V^{-1/2}M$ for diagnostic plots. A has elements

$$a_{ij} = \int_0^\infty Y_i(s)Y_j(s)r_i(s)r_j(s)\frac{d\overline{N}(s)}{(\sum_k Y_k(s)r_k(s))^2}\,,$$

and is closely related to the Aalen estimate of the variance of $\hat\Lambda_0$, see equation (3.1.2). Note that the ith expected value E_i can be written as

$$E_i = \int_0^\infty Y_i(s)r_i(s)\frac{d\overline{N}(s)}{\sum_k Y_k(s)r_k(s)}\,,$$

so that the a_{ij} terms are of smaller order than E.

Since $\sum M_i = 0$, V is not of full rank, and it is easy to verify that each row of V sums to zero. However, for even moderate sample sizes calculation of the symmetric square root of V can take nearly forever in S, the correction to the naive variance is slight, and the exact variance does not seem very useful.

3.3.4 Tied data

For untied data, the terms in the partial likelihood look like

$$\left(\frac{r_1}{\sum_i r_i}\right)\left(\frac{r_2}{\sum_{i>1} r_i}\right)\ldots,$$

where $r_1, r_2, \ldots, r_i$ are the per subject risk scores. Assume that the real data are continuous, but the data as recorded have tied death times. For instance, we might have several subjects die on day 1 of their hospital stay, but of course they did not all perish at the same moment. For a simple example, assume 5 subjects in time order, with the first two both dying at the same recorded time. If the time data had been more precise, then the first two terms in the likelihood would be either

$$\left(\frac{r_1}{r_1+r_2+r_3+r_4+r_5}\right)\left(\frac{r_2}{r_2+r_3+r_4+r_5}\right)$$

or

$$\left(\frac{r_2}{r_1+r_2+r_3+r_4+r_5}\right)\left(\frac{r_1}{r_1+r_3+r_4+r_5}\right),$$

but we don't know which. Notice that the product of the numerators remains constant, but that of the denominators does not. How do we approximate this?

The Breslow approximation is the most commonly used, as it is the easiest to program. It simply uses the complete sum $\sum r_i$ for both denominators. Clearly, if the proportion of ties is large this will deflate the partial likelihood.

The Efron approximation uses $.5r_1 + .5r_2 + r_3 + r_4 + r_5$ as the second denominator, based on the idea that r_1 and r_2 each have a 50% chance of appearing in the "true" second term. If there were 4 tied deaths, then the ratios for r_1 to r_4 would be 1, 3/4, 1/2, and 1/4 in each of the four denominator terms, respectively. Though it is not widely used, the Efron approximation is only slightly more difficult to program than the Breslow version. In particular, since the downweighting is independent of w and thus of β, the form of the derivatives is unchanged.

There are several ways to approach an "exact" calculation. One is to use the average of the two possible denominators as the denominator for the second term. This calculation quickly gets cumbersome if the number of tied subjects d who perish at a given time is at all large, since it is the average of d terms for the second denominator, $\binom{d}{2}$ terms for the third, etc. Note that if the risk scores for the tied subjects were all equal, then the Efron approximation agrees precisely with this exact calculation.

Another tack is to use the marginal probability that subjects 1 and 2 both expire before subjects 3, 4 and 5. The form of the likelihood changes considerably in this case, and the product of terms 1 and 2 is replaced by

$$\int_0^\infty \left[1-\exp\left(-\frac{r_1}{r_3+r_4+r_5}\right)\right]\left[1-\exp\left(-\frac{r_2}{r_3+r_4+r_5}\right)\right] e^{-t}\,dt$$

If there are d subjects r_1 to r_d tied some given time, and we let s be the sum of the remaining scores in the risk set, the above integral expands to

$$1-\sum_{i=1}^{d}\frac{s}{r_i+s}+\sum_{i\neq j}\frac{s}{r_i+r_j+s}-\sum_{i\neq j\neq k}\frac{s}{r_i+r_j+r_k+s}+\ldots,$$

which is the same amount of work as the average denominator calculation. (Though similar, the two expressions are not equivalent for $d > 2$). Some tedious algebra shows that if the risk scores for the tied subjects are all equal, this equals $d!$ times the Efron approximation, and thus leads to exactly the same solution for $\hat\beta$. This would imply that the first and second "exact" methods would be close for actual data sets.

The exact logistic likelihood, or exact partial likelihood, comes from viewing the data as genuinely discrete. The denominator in this case is $\sum_{i \neq j} r_i r_j$ if there are two subjects tied, $\sum_{i \neq j \neq k} r_i r_j r_k$ if there are three subjects tied, etc. The compute time for this case will be even larger than for the calculation above. If there are ties, the value can be considerably different than the first exact method.

The SAS phreg procedure implements the second and third exact method. A small amount of empiric checking has verified that the Efron approximation is very close to the exact marginal likelihood, and so only the exact partial likelihood has been implemented in the S package.

Because of the superiority of the Efron approximation, the coxph function has departed from all other Cox regression programs (that I know of) by making it the default option rather than the Breslow approximation. Note that when there are no ties, all the methods reduce to the same form.

The Efron approximation also induces changes in the residuals' definitions. In particular, the Cox score statistic is still

$$U = \sum_{i=1}^{n} \int_0^\infty (Z_i(s) - \bar Z(s))\, dN_i(s)\,, \tag{15}$$

but the definition of $\bar Z(s)$ has changed if there are tied deaths at time s. If there are d deaths at s, then there were d different values of $\bar Z$ used at the time point. The Schoenfeld residuals use $\bar{\bar Z}$, the average of these d values, in the computation.

The martingale and score residuals require a new definition of $\hat\Lambda$. If there are d tied deaths at time t, we again assume that in the exact (but unknown) untied data there are events and corresponding jumps in the cumulative hazard at $t \pm \epsilon_1 < \ldots < t \pm \epsilon_d$. Then each of the tied subjects will in expectation experience all of the first hazard increment, but only $(d-1)/d$ of the second, $(d-2)/d$ of the next, and etc. If we equate observed to expected hazard at each of the d deaths, then the total increment in hazard at the time point is the sum of the denominators of the weighted means. Returning to our earlier example of 5 subjects of which 1 and 2 are a tied deaths:

$$d\hat\Lambda(t) = \frac{1}{r_1 + r_2 + r_3 + r_4 + r_5} + \frac{1}{r_1/2 + r_2/2 + r_3 + r_4 + r_5}\,.$$

For the null model where $r_i = 1$ for all i, this agrees with the suggestion of Nelson (1969) to use $1/5 + 1/4$ rather than $2/5$ as the increment to the cumulative hazard.

The score residuals do not work out to as neat a formula, though the computation is no harder. For subject 1 in the example, the residual at time 1 is the sum $a + b$ of the 2 terms:

$$\begin{aligned} a &= \left(Z_1 - \frac{r_1 Z_1 + r_2 Z_2 + \ldots + r_5 Z_5}{r_1 + r_2 + \ldots + r_5} \right) \left(\frac{dN_1}{2} - \frac{r_1}{r_1 + r_2 + \ldots + r_5} \right) \text{ and} \\ b &= \left(Z_1 - \frac{r_1 Z_1/2 + r_2 Z_2/2 + \ldots + r_5 Z_5}{r_1/2 + r_2/2 + \ldots + r_5} \right) \left(\frac{dN_1}{2} - \frac{r_1/2}{r_1/2 + r_2/2 + \ldots + r_5} \right) . \end{aligned}$$

This product does not neatly collapse into $(Z_1 - \bar{\bar{Z}})\, d\widehat{M}_i$ but is nevertheless fairly easy to compute. To those who wish to check the algebra: start with the expanded (d term) definition of the increment to U, and repeat the infinitesimal jackknife calculations of Cain and Lange [8].

This argument carries through as well to predicted survival. There is a change in weights but no change in the form of the equations.

The connection between residuals and the exact partial likelihood is not as precise, e.g. the score residuals will not correspond to an infinitesimal jackknife. The exact calculation is used only rarely, the form of the computations will be quite different, and it thus appears to be not worth the bother. If residuals are requested after a Cox fit with the `exact` method, the Breslow formulae are used.

3.3.5 Tests for Proportional Hazards

The key ideas of this section are taken from Grambsch and Therneau [23]. Most of the common alternatives to proportional hazards can be cast in terms of a *time-varying coefficient* model. That is, we assume that

$$\lambda(t; Z) = \lambda_0(t) e^{\beta_1(t) Z_1 + \beta_2(t) Z_2 + \ldots} .$$

(If Z_j is a 0-1 covariate, such as treatment, this formulation is completely general in that it encompasses all alternatives to proportional hazards.) The proportional hazards assumption is then a test for $\beta(t) = \beta$, which is a test for zero slope in the appropriate plot of $\hat{\beta}(t)$ on t.

Let i index subjects, j index variables, and k index the death times. Then let s_k be the Schoenfeld residual and V_k be the contribution to the information matrix at time t_k (see equation 12). Define the rescaled Schoenfeld residual as

$$s_k^* = \hat{\beta} + s_k V_k^{-1} .$$

The main results are:

- $E(s_k^*) = \beta(t_k)$, so that a smoothed plot of s^* versus time gives a direct estimate of $\hat{\beta}(t)$.

- Many of the common tests for proportional hazards are linear tests for zero slope, applied to the plot of r^* versus $g(t)$ for some function g. In particular, the Z:PH test popularized in the SAS PHGLM procedure corresponds to $g(t)$ = rank of the death time. The test of Lin [36] corresponds to $g(t) = K(t)$, where K is the Kaplan-Meier.

- Confidence bands, tests for individual variables, and a global test are available, and all have the fairly standard "linear models" form.

- The estimates and tests are affected very little if the individual variance estimates V_k are replaced by their global average $\overline{V} = \sum V_k/d = \mathcal{I}/d$. Calculations then require only the Schoenfeld residuals and the standard Cox variance estimate $\mathcal{I}^{-1}$.

For the global test, let $g_1(t), g_2(t), \ldots$ be the desired transformations of time for variables 1, 2, etc, and G_k be the diagonal matrix $g_1(t_k), g_2(t_k), \ldots$. Then

$$T = \left(\sum G_k s_k\right)' D^{-1} \left(\sum G_k s_k\right)$$

is asymptotically χ^2 on p degrees of freedom, where

$$D = \sum G_k V_k G_k - \left(\sum G_k V_k\right) \left(\sum V_k\right)^{-1} \left(\sum G_k V_k\right)' .$$

Because the s_k sum to zero, a little algebra shows that the above expression is invariant if G_k is replaced by $G_k - cI$ for any constant c. Subtraction of a mean will, however, result in less computer round-off error.

In any rational application, we will have $g_1 = g_2 = \ldots = g$, and then we can replace each matrix G_k with a scalar $g(t_k) \equiv g_k$ in the above formulas. A further simplification occurs by using $\overline{V}$, leading to

$$T = \left[\sum (g_k - \bar{g}) s_k\right]' \left[\frac{d\mathcal{I}^{-1}}{\sum (g_k - \bar{g})^2}\right] \left[\sum ((g_k - \bar{g}) s_k\right] \tag{16}$$

For a given covariate j, the diagnostic plot will have $y_k = s^*_{kj}$ on the vertical axis and g_k on the horizontal. The variance matrix of the y vector is $\Sigma_j = (A - cJ) + cI$, where A is a dxd diagonal matrix whose kth diagonal element is $V^{-1}_{k,jj}$, $c = \mathcal{I}^{-1}_{jj}$, J is a dxd matrix of ones and I is the identity. The cI term reflects the uncertainty in s^* due the the $\hat{\beta}$ term. If only the shape of $\beta(t)$ is of interest (e.g., is it linear or sigmoid) the term could be dropped. If absolute values are important (e.g. $\beta(t) = 0$ for $t > 2$ years) it should be retained.

For smooths that are linear operators, such as splines or the lowess function, the final smooth is $\hat{y} = Hy$ for some matrix H. Then $\hat{y}$ will be asymptotically normal with mean 0 and variance $H\Sigma_j H'$. Standard errors can be computed using ordinary linear model methods. Although the covariance between residuals is roughly $-1/d$ times their variance, it cannot be neglected in computing standard errors for the smooth. For larger smoothing spans, simulations showed up to 25% inflation in the width of the interval if covariance was ignored.

If V_k is replaced with $\overline{V}$, then Σ_j simplifies to $\mathcal{I}_{jj}^{-1}((d+1)I - J)$. With the same substitution, the component-wise test for linear association is

$$t_j = \frac{\sum(g_k - \bar{g})y_k}{\sqrt{d\mathcal{I}_{jj}^{-1}\sum(g_k - \bar{g})^2}} \tag{17}$$

The `cox.zph` function uses (16) as a global test of proportional hazards, and (17) to test individual covariates. The plot method for `cox.zph` uses a natural spline smoother (`lowess` might be preferred, but the necessary H matrix is not readily obtained); confidence bands for the smooth are based on the full covariance matrix, with $\overline{V}$ replacing V_k.

Formulae aside, reasonably accurate results can be obtained by using other methods directly on the residuals. The return value of`cox.zph` contains both the $g(t)$ vector and the y matrix that are appropriate for plots. These can be used as the x and y data for a `gam` model with identity link and Gaussian errors. for example. The size of the confidence band will be conservative (too large) since, as discussed above, the correlation between the data points has been ignored. This effect will tend to decrease as the sample size increases, since a smaller fraction of the data will be in any smoothing window. Secondly, the overall estimate of variation may be larger, since it is estimated using the variation of each y value from the fitted function; the V_k estimates are based on the variation of each y from its risk set.

Though the simulations in Grambsch and Therneau (1993) did not uncover any situations where the simpler formulae based on $\overline{V}$ were less reliable, such cases could arise. The substitution trades a possible increase in bias for a substantial reduction in the variance of the individual V_k. It is likely to be unwise in those cases where the variance of the covariates, within the risk sets, differs substantially between different risk sets. Two examples come to mind. The first would be a stratified Cox model, where the strata represent different populations. In a muli-center clinical trial for instance, inner city, Veterans Administration and suburban hospitals often service quite disparate populations. In this case a separate average $\overline{V}$ should be formed for each strata. A second example is where the covariate mix changes markedly over time, perhaps because of aggressive censoring of certain patient types.

These cases have not been addressed directly in the software. However, `coxph.detail` will return all of the V_k matrices, which can then be used to construct specialized tests for such situations.

Clearly, no one scaling function $g(t)$ will be optimal for all situations. The `cox.zph` function directly supports four common choices: identity, log, rank, and 1 - Kaplan-Meier. By default, it will use the last of these, based on the following rationale. Since the test for proportional hazards is essentially a linear regression of the scaled residual on $g(t)$, we would expect this test to be adversely effected if there are outliers in x. We would also like the test to be only mildly (if at all) effected by the censoring pattern of the data. The Kaplan-Meier transform appears to satisfy both of these criteria.

3.3.6 Robust Variance

Robust variance calculations are based on the *sandwich estimate*

$$V = ABA'$$

where $A^{-1} = \mathcal{I}$ is the usual information matrix, and B is a "correction term". The genesis of this formula can be found in Huber [32], who discusses the behavior of any solution to an estimating equation

$$\sum_{i=1}^{n} \phi(x_i, \hat{\beta}) = 0 \,.$$

Of particular interest is the case of a maximum likelihood estimate based on distribution f (so that $\phi = \partial \log f / \partial \beta$), when in fact the data are observations from distribution g. Then, under appropriate conditions, $\hat{\beta}$ is asymptotically normal with mean β and covariance $V = ABA'$, where

$$A \;=\; \left(\frac{\partial E \Phi(\beta)}{\partial \beta} \right)^{-1}$$

and B is the covariance matrix for $\Phi = \sum \phi(x_i, \beta)$. Under most situations the derivative can be moved inside the expectation, and A will be the inverse of the usual information matrix. This formula was rediscovered by White [58] [59] (under less general conditions I believe, but all these papers are a bit over my head), and is also known in the econometric literature as White's method.

Under the common case of maximum likelihood estimation we have

$$\begin{aligned} \sum \phi &= \sum_{i=1}^{n} \frac{\partial \log f(x_i)}{\partial \beta} \\ &\equiv \sum_{i=1}^{n} u_i(\beta) \,. \end{aligned}$$

Then by interchanging the order of the expectation and the derivative, A^{-1} is the expected value of the information matrix, which will be estimated by the observed information $\mathcal{I}$. Since $E[u_i(\beta)] = 0$,

$$\begin{aligned} B = \mathrm{var}(\Phi) &= E(\Phi^2) \\ &= \sum_{i=1}^{n} E[u_i'(\beta)u_i(\beta)] + \sum_{i \neq j} E[u_i'(\beta)u_j(\beta)] \end{aligned} \tag{18}$$

where $u_i(\beta)$ is assumed to be a row vector. If the observations are independent, then the u_i will also be independent and the cross terms in equation (18) above will be zero. Then a natural estimator of B is

$$\begin{aligned} \widehat{B} &= \sum_{i=1}^{n} u_i'(\hat{\beta})u_i(\hat{\beta}) \\ &= U'U\,, \end{aligned}$$

where U is the matrix of *score residuals*, the ith row of U equals $u_i(\hat{\beta})$. The column sums of U are the efficient score vector Φ.

As a simple example consider generalized linear models. McCullagh and Nelder [44] maintain that overdispersion "is the norm in practice and nominal dispersion the exception." To account for overdispersion they recommend inflating the nominal covariance matrix of the regression coefficients $A = (X'WX)^{-1}$ by a factor

$$c = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{V_i} / (n-p)\,,$$

where V_i is the nominal variance. Smith and Heitjan [51] show that AB may be regarded as a multivariate version of this variance adjustment factor, and that c and AB may be interpreted as the average ratio of actual variance $(y_i - \mu_i)^2$ to nominal variance V_i. By premultiplying by AB, each element of the nominal variance-covariance matrix A is adjusted differentially for departures from nominal dispersion.

When the observations are not independent, the estimator B must be adjusted accordingly. The "natural" choice $(\sum u_i)^2$ is not available of course, since $\Phi(\hat{\beta}) = 0$ by definition. However, a reasonable estimate is available when the correlation is confined to subgroups. In particular, assume that the data comes from clustered sampling with $j = 1, 2, \ldots, k$ clusters, where there may be correlation within cluster but observations from different clusters are independent. Using equation (18), the cross-product terms between clusters can be eliminated, and the resulting equation rearranged as

$$\mathrm{var}(\Phi) = \sum_{j=1}^{k} \tilde{u}_j(\beta)'\tilde{u}_j(\beta)\,,$$

where $\tilde{u}_j$ is the sum of u_i over all subjects in the jth cluster. This leads to the *modified sandwich estimator*

$$V = A(\tilde{U}'\tilde{U})A\,,$$

where the collapsed score matrix $\tilde{U}$ is obtained by replacement of each cluster of rows in U by the sum of those rows. If the total number of clusters is small, then this estimate will be sharply biased towards zero, and some other estimate must be considered. In fact, rank$(V) < k$, where k is the number of clusters. Asymptotic results for the modified sandwich estimator require that the number of clusters tend to infinity.

Application of these results to the Cox model requires an expression for the score residuals matrix U. Equations (10) and (11) show the partial likelihood written in two forms, and (14) yet another; which should be used as a basis for our work? One way to proceed is to define a weighted Cox partial likelihood, and then let

$$u_i(\beta) \equiv \left(\frac{\partial U}{\partial w_i}\right)_{w=1}\,,$$

where w is the vector of weights; this approach isused in Cain and Lange [8] to define a leverage or influence measure for Cox regression. In particular, they derive the leverage matrix

$$L = U\mathcal{I}^{-1}\,,$$

where L_{ij} is the approximate change in $\hat{\beta}$ when observation i is removed from the data set. Their estimate can be recognized as a form of the *infinitesimal jackknife*, see for example the discussion in Efron [16] for the linear models case. The same leverage estimate is derived using a slightly different argument by Reid and Crépeau [49]. They mention, but do not persue, the use of $L'L$ as a variance estimate.

Specific applications of the sandwich and modified sandwich estimators, detailed below, have all re-derived this result as part of their development.

In fact the connection to the jackknife is quite general. For any model stated as an estimating equation, the Newton-Raphson iteration has step

$$\Delta\beta = 1'(U\mathcal{I}^{-1})\,,$$

the column sums of the matrix $L = U\mathcal{I}^{-1}$. At the solution $\hat{\beta}$ the iteration's step size is, by definition, zero. Consider the following approximation to the jackknife

1. treat the information matrix $\mathcal{I}$ as fixed
2. remove observation i
3. beginning at the full data solution $\hat{\beta}$, do one Newton-Raphson iteration.

This is equivalent to removing one row from L, and using the new column sum as the increment. Since the column sums of $L(\hat\beta)$ are zero, the increment must be $\Delta\beta = -L_{i\cdot}$. That is, the rows of L are an approximation to the jackknife, and the sandwich estimate of variance $L'L$ is an approximation to the jackknife estimate of variance.

When the data are correlated, the appropriate form of the jackknife is to leave out an entire *subject* at time, rather than one observation, i.e., the grouped jackknife. To approximate this, we leave out groups of rows from L, leading to $\tilde L'\tilde L$ as the approximation to the jackknife.

Lin and Wei [37] show the applicability of Huber's work to the partial likelihood, and derive the ordinary Huber sandwich estimate $V = \mathcal{I}^{-1}(U'U)\mathcal{I}^{-1} = L'L$. They also discuss situations in which this is estimate is preferable, including the important cases of omitted covariates and incorrect functional form for the covariate. The relationship of their estimate to the leverage matrix L is not noted by the authors.

Lee, Wei and Amato [35] consider highly stratified data sets which arise from inter observation correlation. As an example they use paired eye data on visual loss due to diabetic retinopathy, where photocoagulation was randomly assigned to one eye of each patient. There are $n/2 = 1742$ clusters (patients) with 2 observations per cluster. Treating each pair of eyes as a cluster, they derive the modified sandwich estimate $V = \tilde L'\tilde L$, where $\tilde L$ is derived from L in the following way. L will have one row, or observation, per eye. Because of possible correlation, we want to reduce this to a leverage matrix $\tilde L$ with one row per individual. The leverage (or row) for an individual is simply the sum of the rows for each of their eyes. (A subject, if any, with only one eye would retain that row of leverage data unchanged). The resulting estimator is shown to be much more efficient than analysis stratified by cluster. A second example given in Lee, Wei and Amato concerns a litter-matched experiment. In this case the number of rats/litter may vary.

Wei, Lin and Weissfeld [57] consider multivariate survival times. An example is the measurement of both time to progression of disease and time to death for a group of cancer patients. The data set again contains $2n$ observations, time and status variables, subject id, and covariates. It also contains an indicator variable `etype` to distinguish the event type, progression vs. survival. The suggested model is stratified on event type, and includes all strata$\times$covariate interaction terms. One way to do this in S is

```
> fit2 <- coxph(Surv(time, status) ~ (rx + size + number)*strata(etype), ...)
> Ltilde <- residuals(fit2, type='dfbeta', collapse=subject.id)
> newvar <- t(Ltilde) %*% Ltilde
```

The thrust of the computation is to obtain a per *subject* leverage matrix.

Actually, WLW lay out the calculation in a different way. Their approach is to fit each of the two models separately. They then concatenate the coefficient vectors, and build up a joint variance matrix using products of the individual leverage matrices. If

L_1 and L_2 are the leverage matrices from the first and second fit, respectively, then WLW suggest the variance matrix

$$\begin{pmatrix} L_1' L_1 & L_1' L_2 \\ L_2' L_1 & L_2' L_2 \end{pmatrix}$$

However, closer examination (and a little algebra) shows that their approach is equivalent to the first option. Using the grouped jackknife approach, as suggested here, rather than separate fits for each event type has some practical advantages:

- It is easier to program, particularly when the number of events per subject is large. (See the example in section 4.5.)
- Other models can be encompassed, in particular one need not include all of the strata$\times$covariate interaction terms.
- There need not be the same number of events for each subject. The method for building up a joint variance matrix requires that all of the score residual matrices be of the same dimension, which is not the case if information on one of the failure types was not collected for some subjects.

The main difference, then, between the techniques used in the Lee et al. and Wei et al. papers is whether or not to stratify the analysis based on the failure type. If the event types are distinct, such as "survival" and "progression" this seems to be a wise idea, since the baseline hazard functions for the two types of event are likely to differ. In the case of the eye data, there is no good reason to assume that left and right eyes (or should it be "dominant" versus "non-dominant"?) differ in their hazard function, and the risk set is structured to include both eye types. The case is less clear when there are multiple sequential events per subject, but the events ("failures") are all of the same type. Examples include repeated infections in patients with an immune disorder (see the example of chronic granulotomous disease (CGD) discussed in [20]), repeated fatal or non-fatal infarctions in cardiac patients, or the recurrent bladder cancer data found in [57]. If each event may influence subsequent events, e.g., each infarction damages the remaining myocardium, the consensus appears to be that stratification is preferable, using first event, second event, etc. as the strata. In other data sets, such as CGD, strata may not be desired. In this case the data set should be set up using intervals of risk (start, stop], so that a subject is not counted twice in the same risk set. The S code for the modified sandwich estimator will be identical to that for the stratified case. A situation that included both multiple failure types and multiple events/subject of one of the types could involve both strata *and* disjoint risk intervals in setting up the data set.

3.3.7 Weighted Cox models

A Cox model that includes case weights has been considered by Binder [7] in the context of survey data. If w_i are the weights, then the modified score statistic is

$$U(\beta) = \sum_{i=1}^{n} w_i u_i(\beta) \,. \tag{19}$$

The individual terms u_i are still $Z_i(t) - \bar{Z}(t)$; the weighted mean $\bar{Z}$ is changed in the obvious way to include both the risk weights r and the external weights w. The information matrix can be written as $I = \sum \delta_i w_i v_i$, where δ_i is the censoring variable and v_i is a weighted covariance matrix. Again, the definition of v_i changes in the obvious way from equation (12). If all of the weights are integers, then for the Breslow approximation this reduces to ordinary case weights, i.e., the solution is identical to what one would obtain by replicating each observation w_i times. With the Efron approximation or the exact partial likelihood approximation, of course, replication of a subject would result in a correction for ties. The `coxph` function allows general case weights. Residuals from the fit will be such that the sum of weighted residuals =0, and the returned values from the coxph.detail function will be the individual terms u_i and v_i, so that U and I are weighted sums. The sandwich estimator of variance will have `t(L) %*% diag(w^2) %*% L` as its central term. The estimate of $\hat{\beta}$ and the sandwich estimate of its variance are unchanged if each w_i is replace by cw_i for any $c > 0$.

Using weights appropriate to the survey sampling scheme, Binder suggests use of the modified sandwich estimate $\mathcal{I}^{-1}B\mathcal{I}^{-1}$ where $B = \mathrm{var}(U)$ "can be estimated using design based methods", though he gives no specifics on what these might be. His derivation of the score residual vectors u_i differs from the above, but the same result is obtained, and shown in the last equation of his section 3. In a simulation study he compares the naive, sandwich, and modified sandwich estimators, with the latter being the most reliable.

Lin [36] also develops a weighted Cox model, in the context of tests for proportional hazards. His estimates of the score and hence of $\hat{\beta}$ are based on equation (19), but without redefinition of $\bar{Z}$ to include weights. It is thus not related to case weights, but rather to weighted log-rank statistics such as the Tarone-Ware family [43]. Estimates for this model can be obtained from S in three steps; assume that w is the weight variable:

1. Use `coxph` with w as weights and $-\log(w)$ as an offset to estimate Lin's weighted $\hat{\beta}$.

2. Fit a second cox model, without weights or an offset, but with the coefficients constrained to equal the results of the first model. (Use initial values and `iter=0`.) The `coxph.detail` function can be applied to this second model to obtain the individual v_i estimates.

3. Estimate the variance of $\hat\beta$ as ABA, where $A = (\sum w_i v_i)^{-1}$ and $B = \sum w_i^2 v_i$.

Tests for proportional hazards are more easily accomplished, however, using the `cox.zph` function.

An exciting use of weights is presented in Pugh et al. [48], for inference with missing covariate data. Let π_i be the probability that none of the covariates for subject i is missing, and p_i be an indicator function which is 0 if any of the covariates actually is NA, so that $E(p_i) = \pi_i$. The usual strategy is to compute the Cox model fit over only the complete cases, i.e., those with $p_1 = 1$. If information is not missing at random, this can lead to serious bias in the estimate of $\hat\beta$. A weighted analysis with weights of p_i/π_i will correct for this imbalance. There is an obvious connection between this idea and survey sampling: both reweight cases from underrepresented groups.

In practice π_i will be unknown, and the authors suggest estimating it using a logistic regression with p_i as the dependent variable. The covariates for the logistic regression may be some subset of the Cox model covariates (those without missing information), as well as others. In an example, the authors use a logistic model with follow-up time and status as the predictors. Let T be the matrix of score residuals from the logistic model, i.e.

$$T_{ij} = \frac{\partial}{\partial \alpha_j}[p_i \log \pi_i(\alpha) + (1-p_i)\log(1-\pi_i(\alpha))],$$

where α are the coefficients of the fitted logistic regression. Then the estimated variance matrix for $\hat\beta$ is the sandwich estimator $\mathcal{I}^{-1} B \mathcal{I}^{-1}$, where

$$B = U'U - [U'T][T'T]^{-1}[T'U]\,.$$

This is equivalent to first replacing each row of U with the residuals from a regression of U on T, and then forming the product $U'U$. Note that if the logistic regression is completely uninformative ($\hat\pi_i$ = constant), this reduces to the ordinary sandwich estimate.

For either of the Breslow or the Efron approximations, the extra programming to handle weights is modest. For the Breslow method the logic behind the addition is also straightforward, and corresponds to the derivation given above. For tied data and the Efron approximation, the formula is based on extending the basic idea of the approximation, $E(f(r_1, r_2, \ldots)) \approx f(E(r_1), E(r_2), \ldots)$ to include the weights, as necessary. Returning to the simple example of section 3.3.4 above, the second term of the partial likelihood is either

$$\left(\frac{w_1 r_1}{w_1 r_1 + w_3 r_3 + w_4 r_4 + w_5 r_5}\right)$$

or

$$\left(\frac{w_2 r_2}{w_2 r_2 + w_3 r_3 + w_4 r_4 + w_5 r_5}\right)\,.$$

To compute the Efron approximation, separately replace the numerator with $.5(w_1 r_1 + w_2 r_2)$ and the denominator with $.5w_1 r_1 + .5w_2 r_2 + w_3 r_3 + w_4 r_4 + w_5 r_5$.

3.3.8 Estimated survival

The methods for expected survival after a Cox model parallel those for expected survival based on a population. One can create individual survival curves or cohort survival curves, and the latter can be in any of the Ederer, Hakulinen, or conditional frameworks.

For individual survival, recall that the estimated cumulative hazard for a subject with covariate process $x(t)$ is

$$\Lambda(t; x) = \int_0^t e^{\hat\beta' x(s)} d\hat\Lambda_0(s)',,$$

with either the Breslow or Efron estimator of $\hat\Lambda_0$. The choice of the Breslow or Efron estimate should be consistent with the option chosen to break ties when the Cox model was fit (this is the default action of the software). The estimated survival function is then $\hat S(t; x) = \exp(-\hat\Lambda(t; x))$.

If the vector of coefficients $\hat\beta$ were treated as fixed, then the variance of the cumulative hazard would be

$$V(t) = \int_0^t e^{2\hat\beta' x(s)} \frac{d\bar N(s)}{(\sum Y_i(s) e^{\hat\beta' Z_i(s)})^2}$$

for the Breslow estimator, which is a natural extension of the Aalen estimate of variance in Nelson's hazard estimator. If Efron's method were used, the variance estimate will have a slightly larger increment at each tied time.

The actual variance for the cumulative hazard must also account for the error in estimation of $\hat\beta$ Tsiatis [55] and Link [39, 40] have derived this, and with some rearragement their result can be written as

$$V_c(t) = V(t) + d(t)' \mathcal{I}^{-1} d(t) \tag{20}$$

$$d(t) = \int_0^t [x(s) - \bar Z(s)] \frac{d\bar N(s)}{\sum Y_i(s) e^{\hat\beta' Z_i(s)}}$$

where V_c is the variance under the Cox model, V is the naive variance given above, Z is the covariate set for the fitted model, x is the covariate vector for this curve, and $\mathcal{I}^{-1}$ is the variance matrix for $\hat\beta$. The increase in variance is largest for a covariate set that is far from the mean. The vector $d(t)$ is also the score residual process for a hypothetical new subject with covariates x and no events, which is a measure of the leverage of such an observation on the estimate of β. It is intuitive that if the covariate value(s) x has small leverage on $\hat\beta$, then the variance in $\hat\beta$ will have small effect on the curve.

The calculations for a weighted model have the same form, using the jumps in the weighted cumulative hazard function

$$\hat\Lambda_0(t) = \sum_{i=1}^n \int_0^t w_i \frac{dN_i(s)}{\sum Y_j(s) w_j \hat r_j(s)} \,.$$

Notice: In an earlier version of this document the above was incorrectly factored to form the equation below, which is wrong.

$$V_c(t) = \int_0^t \{1 + [x(s) - \bar Z(s)]' \mathcal{I}^{-1} (x(s) - \bar Z(s)]\} dV(s)' \,,$$

Gail and Byar [21] have extended this result to the estimation of multiple curves, and show that if $x(s)$ and $x^*(s)$ were two separate covariate vectors, then the covariance of the two estimated cumulative hazards is of exactly the same form as above, with the obvious substitutions of $x(s) + x^*(s)$ in place of $2x(s)$, and d^* for one of the two vectors d. No results are given for a stratified Cox model. (The cross product terms could presumably be integrated with respect to $(dV_k + dV_{k^*})/2$; however, this creates extra complexity in the programming and has not yet been implimented.)

Since the variance formula involves the covariate average over time $\bar Z(t)$ for the *original study*, it cannot usually be calculated from published summaries, e.g., to esimate the expected survival of the patient in front of you based on a report in the literature. However, for most Cox fits $\bar Z(t)$ will vary only slightly over time, and a reasonable approximation could be made if the report contained both the initial averages and the confindence intervals for any particular case.

This curve is applicable to a single patient, and is the appropriate object to plot when considering the predicted survival for some future patient who has some particular set of covariates. Another use is to fit a stratified model, for example

```
Surv(time, status) ~ age + sex + strata(treatment)
```

Then a plot of the pair of curves is a comparison of the treatment groups, adjusted to common age and sex values. This can be useful when two treatments are unbalanced with respect to an important covariate.

It is common practice to use these curves for group survival as well. The curve for the "average" patient, i.e., the curve corresponding to a fictional subject with mean values for each covariate, is then used as the predicted survival curve for the entire cohort. Though convenient this procedure is incorrect. What should be done follows from exactly the same discussion as found above in the section on expected survival for a population matched reference. Either the Ederer, Hakulinen, or conditional computation can be used.

One use of these cohort averages is to summarize and present the results of a study. This issue is reviewed in Thomsen, Keiding and Altman [54], albeit using very different terms. Center stage is given to the analog of the Ederer estimate, referred to as the *direct adjusted survival curve*, a term coined earlier by Makuch [41]. Using a simple example, Thomsen et. al. demonstrate that the estimate based on a single average patient will be badly biased whenever there is large variation in the individual risk scores $\beta' x$.

A second use is for historical comparison. With the increased availability of published regression analyses of survival with specific diseases it has become possible to make historical comparisons between the observed survival in a (possibly fairly small) study group and the survival that was to be expected from a published regression analysis. One area where this has been used is in transplantation, where randomized studies are logistically (and perhaps ethically) impossible, so that the best answer available to the question "what would have happened to these patients if they had not been transplanted?" will be via comparison with published pre-transplatation-therapy experience [22]. In this case λ_i will come from the older study, with the original Cox model covariates as the matching variables. Follow-up and censoring time will come from the new data set.

A variance estimate for the direct adjusted curve S_d is derived in Gail and Byar [21]. Let

$$S_d = (1/n) \sum_{i=1}^{n} S_i(t)$$

Where S_i is the individual survival curve for subject i in the *new* data set. This is calculated as above, using $x_i(t)$ for the new subject but risk sets and variance based on the original Cox model fit. Then

$$\text{var}(S_d) = (1/n)^2 [\sum \text{var}(S_i) + \sum_{i \neq j} \text{cov}(S_i, S_j)].$$

Thomsen et al. also discuss the conditional estimate

$$\exp\left[-\int_0^t \frac{\sum Y_i(s) \lambda_0(s) e_i^{\eta}(s)}{\sum Y_i(s)} \, ds \right].$$

They conclude that the curve itself is "not easy to interpret" because it mixes observed mortality, through Y_i, with expected mortality, through λ_i. However, the difference in log survival curves can be used as an estimate of excess mortality as is done in Andersen and Væth [4].

This author believes that extension of Hakulinen's cohort method is the most appropriate way to combine expected curves in the Cox model. However, I am not aware of any discussion of this in the literature. The failure of the Ederer method, remember, occurs when there is significant change in the enrollment criteria over the course of a study. This is of major concern in historical reviews that span > 10 years, as the age profile often changes dramatically. For a Cox model, this will be an issue whenever there is similar change in the values at entry for one of the variables that was included in the model.

To conclude this section I must comment on a common *abuse* of the Cox model expected curve. This is the comparison of the expected curve for a study to the Kaplan-Meier of the same study as a test for "goodness of fit". This is nonsense since:

1. Some difference between the two can arise because of different approximations, i.e., the S functions default to an estimate that is comparable to the Fleming-Harrington method. This will differ from the Kaplan-Meier in the tails, where n is small.

2. Some simple algebra shows that the conditional estimator *is* the F-H estimate of the raw data, independent of the value of $\hat{\beta}$.

3. If the censoring pattern is not independent of risk, then the Ederer estimate will differ from the K-M because of the censoring effect, even if the Cox model is completely correct.

4. For most data sets the value of $\bar{Z}(t)$ varies only slowly with time. In this case the individual survival curve for the average subject $\bar{Z}(0)$ will also be approximately equal to the F-H estimator, independent of $\hat{\beta}$.

5. If a data set includes time-dependent covariates, the individual survival curve for any fixed x can be very surprizing.

4 Examples

4.1 Simple Cox Models

The first example uses data from a study of ovarian cancer [15]. This data appears in the SAS supplemental procedure guide for the COXREG and SURVTEST procedures. The variables are

- futime: The number of days from enrollment until death or censoring, whichever came first.

- fustat: An indicator of death (1) or censoring (0).

- age: The patient age in years (actually, the age in days divided by 365.25)

- residual.dz: An indicator of the extent of residual disease.

- rx: The treatment given.
- ecog.ps: A measure of performance score or functional status, using the Eastern Cooperative Oncology Group's scale. It ranges from 0 (fully functional) to 4 (completely disabled). Level 4 subjects are usually considered too ill to enter a randomized trial such as this.

```
> ovarian <- read.table("data.ovarian", row.names=NULL,
                col.names= c("futime", "fustat", "age",
                             "residual.dz", "rx", "ecog.ps"))
```

A listing of the data is given below.

futime	fustat	age	residual.dz	rx	ecog.ps
59	1	72.3315	2	1	1
115	1	74.4932	2	1	1
156	1	66.4658	2	1	2
268	1	74.5041	2	1	2
329	1	43.1370	2	1	1
353	1	63.2192	1	2	2
365	1	64.4247	2	2	1
377	0	58.3096	1	2	1
421	0	53.3644	2	2	1
431	1	50.3397	2	1	1
448	0	56.4301	1	1	2
464	1	56.9370	2	2	2
475	1	59.8548	2	2	2
477	0	64.1753	2	1	1
563	1	55.1781	1	2	2
638	1	56.7562	1	1	2
744	0	50.1096	1	2	1
769	0	59.6301	2	2	2
770	0	57.0521	2	2	1
803	0	39.2712	1	1	1
855	0	43.1233	1	1	2
1040	0	38.8932	2	1	2
1106	0	44.6000	1	1	1
1129	0	53.9068	1	2	1
1206	0	44.2055	2	2	1
1227	0	59.5890	1	2	2

Here is a simple survival model for age and its result. Age is perhaps the only important variable in this data set.

```
> coxph(Surv(futime, fustat)~ age, ovarian)
```

```
Call:  coxph(formula = Surv(futime, fustat) ~ age,
        data=ovarian)

     coef exp(coef) se(coef)    z       p
age 0.162      1.18   0.0497 3.25 0.00116

Likelihood ratio test=14.3  on 1 df, p=0.000156    n= 26
```

For a more complicated model, the result should probably be saved in a temporary variable. It can then be printed multiple times, and residuals and/or predicted values may be extracted.

```
> fit <- coxph(Surv(futime, fustat)~ residual.dz + rx + ecog.ps, ovarian)
> print(fit)
Call:  coxph(formula = Surv(futime, fustat) ~ residual.dz + rx + ecog.ps)

              coef exp(coef) se(coef)      z      p
residual.dz  1.347     3.844    0.680  1.980 0.0478
         rx -0.749     0.473    0.595 -1.260 0.2078
    ecog.ps  0.453     1.573    0.590  0.767 0.4431

Likelihood ratio test=6.03  on 3 df, p=0.11    n= 26
```

The print function is invoked automatically, so in the example above the user could have typed "fit" instead of "print(fit)". A more complete printout is produced by the summary function. This adds confidence intervals, Wald and score tests, and an R^2 measure based on work of Nagelkirke [45]. This measure needs to be proven over time, but certainly is one of the easier ones to implement that I've seen, and appears very well founded. An option to the summary function can be used to get confidence intervals at levels other than .95.

The stratified Cox model can be obtained by using a strata directive within the fit.

```
> fit <- coxph(Surv(futime, fustat)~ age + ecog.ps + strata(rx), ovarian)
> summary(fit)
Call:
coxph(formula = Surv(futime, fustat) ~ age + ecog.ps + strata(rx),
        data=ovarian)

N= 26

           coef exp(coef) se(coef)      z       p
    age  0.1385     1.149    0.048  2.885 0.00391
ecog.ps -0.0967     0.908    0.630 -0.154 0.87799

        exp(coef) exp(-coef) lower .95 upper .95
```

```
    age      1.149       0.871      1.045       1.26
ecog.ps      0.908       1.102      0.264       3.12

Rsquare= 0.387   (max possible= 0.874 )
Likelihood ratio test= 12.7  on 2 df,    p=0.00174
Efficient score test = 12.2  on 2 df,    p=0.0022
```

After the fit is completed residuals from the fitted model can be obtained using the `resid` function. By default the martingale residuals are produced, also available are deviance, score, and Schoenfeld residuals. For any of these it is sufficient to give the shortest unique abbreviation of the residual type. Two common transformations of the score residuals can also be requested: `dbeta` and `dfbetas`. These are the approximate change in the coefficient vector if observation i is dropped, and that change scaled by the variance of β.

Martingale residuals are used most often in an assessment of functional form, score residuals play a role in assessing influential or leverage data points as well as computation of robust "sandwich" variance estimators, and the Schoenfeld residuals are useful in assessing time trends or lack of proportionality in one of the coefficients of the model. Deviance residuals, though they have an interesting theoretical justification, have not proven very useful in practice.

```
> fit <- coxph(Surv(futime, fustat) ~ age + residual.dz + rx + ecog.ps,
                    ovarian)
> mresid <- resid(fit)
> dresid <- resid(fit, "dev")
> sresid <- resid(fit, "scor")
> resid(fit, "scho")
             age residual.dz         rx    ecog.ps
 59   2.69315678  0.06761161 -0.1256239 -0.5072536
115   5.36390193  0.08039118 -0.1493686 -0.6031317
156  -0.89877404  0.10683988 -0.1985109  0.1984379
268   6.95664457  0.12857952 -0.2389036  0.2388158
329 -15.73656567  0.28889884 -0.5367805 -0.4634169
353   4.06104424 -0.70587652  0.4535120  0.5282024
365   5.50035871  0.25348266  0.4796229 -0.4413864
431  -8.06809462  0.27490178 -0.4297023 -0.5248323
464  -2.15471513  0.23158423  0.5066040  0.4814387
475   0.57065101  0.25226661  0.5518479  0.5244351
563   0.06487254 -0.47274521  0.3319974  0.2747028
638   1.64752693 -0.50593435 -0.6446946  0.2939883
```

The martingale and deviance residuals are each a vector of length n, where n is the number of subjects in the data. The score residuals form an n by p matrix, with one column per regressor variable, and are components of the first derivative of the partial likelihood. By definition, the column sums of the score residual matrix will be zero at $\hat{\beta}$. The Schoenfeld residuals have one row for each death in the data set and p columns, the time point of the death is returned as the row label of the matrix. As with other models in S, a factor variable may be expanded into multiple contrasts in the X matrix (though there are none in this example). It will then appear as multiple columns in the score or Schoenfeld residuals as well.

Tests for proportional hazards are based on rescaled Schoenfeld residuals, and can be obtained with cox.zph. They are based on Grambsch and Therneau [23], and are discussed further in section on mathematical backround.

```
> temp <- cox.zph(fit)
> print(temp)
                 rho   chisq      p
        age -0.03989 0.02618 0.8715
residual.dz -0.14168 0.24626 0.6197
         rx  0.13246 0.20012 0.6546
    ecog.ps  0.48448 1.88192 0.1701
     GLOBAL       NA 3.36086 0.4993
> plot(temp)
> plot(temp, var=2)
```

The plot shows time (or a monotone function of time) on the x axis and the rescaled residuals on the y axis, with one plot per covariate. An overlaid smooth curve is an estimate of $\beta(t)$, a time-dependent regression coefficient. Under proportional hazards we must have $\beta(t) = \beta$, i.e., the hazard ratio does not vary with time. The standard printout includes the correlation coefficients ρ for the plots of each variable along with tests for $\rho = 0$, and a global test of proportional hazards based on all the covariates. (There is no applicable value of ρ for the global test). The var option can be used to create only a single plot, in this case for the second variable in the model statement: (residual.dz). This is useful when one wishes to add a main title, a horizontal line at $y = 0$, or other annotation to the plot.

Predicted values are available based on either the linear predictor $\eta = X'\hat{\beta}$, the risk for an individual relative to the average subject within the data set $r = \exp(\eta)$, the expected number of events for an individual over the time interval that they were observed to be at risk (which is a component of the martingale residual), or for individual components of the linear predictor η.

```
> lp   <- predict(fit, type='lp', se.fit=T)
> risk <- predict(fit, type='risk')
> expt <- predict(fit, type='expected')
> term <- predict(fit, type='terms')
> round(risk,2)
```

```
    1    2    3     4    5    6    7    8    9   10   11   12   13   14   15
14.43 18.9 9.71 26.49 0.38 1.14 2.16 0.44 0.54 0.93 1.21 1.18 1.71 5.21 0.42

  16   17   18   19  20   21   22  23   24   25   26
1.27 0.16 1.66 0.86 0.1 0.23 0.31 0.2 0.25 0.17 0.72
> fit
```

An optional `data` argument to the predict function is a new data set. It allows one to obtain predicted values for subjects who are not part of the original study. Because it is positioned as the second argument to `predict`, the examples above must explicitly include "type =" (or a second comma), where this was not necessary in `residual`. Another option to the predict function is to return standard errors of the predicted values.

4.2 Fitted survival curves

The survfit function is used for fitting a survival curve, either to original data or after a Cox model or parametric model fit.

```
> sf <- survfit(Surv(futime, fustat) ~ rx + residual.dz, ovarian)
> summary(sf)
```

The above example would result in four survival curves, indexed by the two levels of treatment and the two levels of residual disease. The right hand side of the formula is interpreted differently than it would be for an ordinary linear or Cox model. Technically, the formula should have been expressed using a * operator instead of +, since the desired result is for all four levels or 'interactions'. We process the + symbol as though it were an interaction, as well as producing labels that are longer (and hopefully more readable) than the default labels generated by the * operator.

Each of the following formulas would have produced the same output curves, though the first has different labels.

```
Surv(futime, fustat) ~ interaction(rx,residual.dz)
Surv(futime, fustat) ~ strata(rx,residual.dz)
```

Another example is shown below, for a somewhat smaller study of acute myelogenous leukemia; the data can be found on page 49 of Miller [43],

```
> aml<- data.frame(time=c(9, 13, 13, 18, 23, 28, 31, 34, 45, 48, 161,
                           5,  5,  8,  8, 12, 16, 23, 27, 30, 33, 43, 45),
             status= c(1,1,0,1,1,0,1,1,0,1,0, 1,1,1,1,1,0,1,1,1,1,1,1),
             group = as.factor(c(rep("Maintained", 11),
                                 rep("Nonmaintained", 12) )))
```

```
>sf <- survfit(Surv(time, status) ~ group, aml)
>sf
Call: survfit.formula(formula = Surv(time, status) ~ group, data=aml)

                     n events mean se(mean) median 0.95CI 0.95CI
   group=Maintained 11      7 52.6    19.83     31     18     NA
group=Nonmaintained 12     11 22.7     4.18     23      8     NA
```

Similarly to other S model programs, the print function gives a very short synopsis and the summary function provides more complete information. For survfit the default print contains only n, the total number of events, the mean survival time and its standard error, median survival, and confidence intervals for the median. The mean is based on a truncated estimate, i.e., the survival curve is assumed to go to zero just past the last follow-up time. It is thus an under estimate of the mean when the last observation(s) is censored. The confidence interval for the median is based on the confidence intervals for the survival curve: the lower and upper limits are the intersection of a horizontal line at .5 with the lower and upper confidence bands for $S(t)$. The confidence interval will change if a different confidence level or confidence type is specified in the `survfit` call. If the upper confidence band for $S(t)$ never reaches 0.5, as in the example above, then the upper confidence limit for the median is unknown.

```
>summary(sf)
Call: survfit.formula(formula = Surv(time, status) ~ group, data=aml)

                group=Maintained
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    9     11       1    0.909  0.0867       0.7541        1.000
   13     10       1    0.818  0.1163       0.6192        1.000
   18      8       1    0.716  0.1397       0.4884        1.000
   23      7       1    0.614  0.1526       0.3769        0.999
   31      5       1    0.491  0.1642       0.2549        0.946
   34      4       1    0.368  0.1627       0.1549        0.875
   48      2       1    0.184  0.1535       0.0359        0.944

                group=Nonmaintained
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    5     12       2   0.8333  0.1076       0.6470        1.000
    8     10       2   0.6667  0.1361       0.4468        0.995
   12      8       1   0.5833  0.1423       0.3616        0.941
   23      6       1   0.4861  0.1481       0.2675        0.883
   27      5       1   0.3889  0.1470       0.1854        0.816
   30      4       1   0.2917  0.1387       0.1148        0.741
   33      3       1   0.1944  0.1219       0.0569        0.664
   43      2       1   0.0972  0.0919       0.0153        0.620
```

```
   45       1         1   0.0000       NA            NA            NA
```

By default, the summary includes one row for each time at which a death occurred, and for each of these times lists the number of subjects who are at risk just prior to the event, the number of events that occurred, the survival estimate, its standard error, and upper and lower 95% confidence intervals calculated on the hazard scale.

Options to the survfit routine include estimates based on the Nelson cumulative hazard estimator instead of the Kaplan-Meier (as proposed by Fleming and Harrington), confidence intervals based on the log hazard scale, and the level for the confidence interval. An option to the summary routine allows the listing to be printed for selected survival times and/or to include censored time points within the printout.

Survival estimates for the data set as a whole can be generated by using the null model, i.e. `~1` as the right-hand side of the formula.

```
> survfit(Surv(time, status) ~1, aml)
```

It is also allowable to leave off the `~1`; the same result will be obtained.

The survfit function now supports case weights via the weight argument. One obvious use of this feature is for data sets with multiplicities of the input lines, i.e., instead of having one line of data appear three times it could appear once with a weight of three. For instance, if weights of two are attached to each of the 23 cases in the AML data this effectively doubles the data set, the resultant survival is identical to the original, but the variance of the hazard is halved. Case weights were included as an option in the program less for this reason, however, than to facilitate various manipulations that require fractional case weights, see for example Turnbull [52], who uses this in an EM algorithm for left censored data. When `survfit` is used with fractional weights, the returned variance estimate is probably worthless.

The survfit function also can generate predicted survival curves for a Cox model by using the resultant fit as a first argument to the function.

```
> attach(ovarian)
> fit <- coxph(Surv(futime, fustat)~ age + ecog.ps + strata(rx))
> survfit(fit)
```

This will produce two survival curves, one for each of the two `rx` strata; each curve will be for a "pseudo cohort" whose age and ecog performance score are equal to the mean values for the data set that was fit.

An important aspect of the new modeling language is the (largely undocumented) number of *side effects*. If the last line above were executed without first attaching the `ovarian` data frame (perhaps in a later S session) it would fail. This is because estimation of the survival curve requires some data summaries that were not saved in the *fit* object. In order to obtain them, the data is reconstructed based on the call that produced the *fit* object, and this reconstruction requires essentially the same environment to exist as was originally present. However, if the original fit had been produced using the `data=ovarian` argument to `coxph`, no attachment would be necessary. More subtle errors arise if important options differ at the later call, i.e., `na.action` or `contrasts`. A summary of the side effects in survival models is found later in this document.

The original version of survfit had an option for *risk weights*. This option had to do with survival curves following a Cox model with known coefficients, and is now obtained in a more obvious way. The example below shows a fit with risk weights, i.e. $\exp(X'\beta)$, where $X'\beta = .1$ to 2.3.

```
> survfit(coxph(Surv(aml$time, aml$status) ~ offset(1:23/10)))
```

This example is contrived: normally the offset statement would contain the result of a coxph fit, perhaps on a different set of data. I have used this feature on two occasions in my own analyses. In each case the second data set had a small number of patients, and we wished to test the effect of variable "x" after adjusting for a set of baseline variates known to effect survival. Though the data set was not large enough to model and estimate all of the variables concurrently, it was sufficient when the regression coefficients for the baseline variates were fixed at the values from an earlier, larger study. For instance, if the earlier study had a variable `old` and a fitted regression coefficient of 2.3, the new model would have had `offset(2.3 *old)` as a term in the model.

Another use of offsets is to model the *relative* mortality of a population. This is discussed in example Andersen, et al. [3] using diabetes in the county of Fyn. In this model the covariates were age at onset, sex, and the population mortality or hazard "phaz" which depends on age and sex and is set up as a time dependent covariate. Then a model for absolute mortality is

```
Surv(start, stop, status) ~ age + sex
```

and the model for relative mortality is

```
Surv(start, stop, status) ~ age + sex + offset(log(phaz))
```

As a comparison of the two models, they suggest fitting a third model

```
Surv(start, stop, status) ~ age + sex + log(phaz)
```

where a values of $\hat\beta_3 = 0$ corresponds to absolute mortality and $\hat\beta_3 = 1$ to a model for the relative mortality. The fitted value of -.39 suggests that the model for absolute mortality is preferred.

Note that when a curve is requested using only the `offset` directive, using the same data set that created the coefficients for the offset, the resultant survival estimate is identical to the curve based on the model's `fit` component but the variance will be deflated. The survival curve from a Cox model includes in its variance a term that accounts for the error in $\hat{\beta}$, and an offset model implicitedly states that β is known without error.

In order to get curves for a pseudo cohort other than one centered at the mean of the covariates, use the newdata argument to survfit. The newdata argument is a list, data frame or matrix which contains in each row all variables found in the right-hand side of the equation that was fit, excluding strata terms. Multiple rows in the new data frame will result in multiple "cohorts" of subjects, for example:

```
> fit <- coxph(Surv(futime, fustat)~ age + ecog.ps + strata(rx),
              ovarian)
> survfit(fit, newdata=list(age=c(30,50), ecog.ps=c(2,2))
```

This will produce two survival curves, the first for an imagined cohort of subject who were age 30 with a performance score of 2, and the second for an age 30 group. A more complete example is given below. (Amendment: if the model includes a strata by covariate interaction, use of the `newdata` argument currently leads to failure, including a very nonintuitive error message. This is a bug, and should be fixed in the next release.)

In keeping with other S models, the newdata argument can also be the number of a data frame. (I don't like this, but it was important to be compatible). When the formula has only one variable on the right-hand side of the equation and the given newdata argument is a single integer, then the intent of the user is ambiguous. For instance

```
> fit <- coxph(y ~ x)
> survfit(fit, newdata=3)
```

Is the intention to have $x = 3$ or to use the variable x found in data frame number 3? To avoid this anomaly, be explicit by writing `survfit(fit, newdata=list(x=3))`. (And for those who are curious, the ambiguity is, if necessary, resolved by the following rule: If the number is a positive integer, this routine assumes it is a frame number, otherwise the number is assumed to be the actual value for the covariate.)

Here is another example from the Fleming data set. Note the use of `x=T`. This causes a copy of the final X matrix to be saved in the `fit` object, including any transformations, dummy variables that represent factors, etc. When subsequent calculations are planned for a fitted Cox model, such as residuals or survival curves, this will save significant computation time (at the expense of a larger `fit` object) since x does not have to be reconstructed.

```
> fit <- coxph(Surv(futime, fustat) ~ age + ecog.ps + strata(rx),
        ovarian, x=T)
> summary(fit)
Call:
```

```
coxph(formula = Surv(futime, fustat) ~ age + ecog.ps + strata(rx),
        data=ovarian, x=T)
N= 26

           coef exp(coef) se(coef)      z       p
    age  0.1385     1.149    0.048  2.885 0.00391
ecog.ps -0.0967     0.908    0.630 -0.154 0.87799

        exp(coef) exp(-coef) lower .95 upper .95
    age     1.149      0.871     1.045      1.26
ecog.ps     0.908      1.102     0.264      3.12

Rsquare= 0.387   (max possible= 0.874 )
Likelihood ratio test= 12.7  on 2 df,   p=0.00174
Efficient score test = 12.2  on 2 df,   p=0.0022

> summary(survfit(fit))
Call: survfit.coxph(object = fit)

                rx=1
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   59     13       1    0.978  0.0266       0.9275            1
  115     12       1    0.951  0.0478       0.8620            1
  156     11       1    0.910  0.0760       0.7722            1
  268     10       1    0.861  0.1055       0.6776            1
  329      9       1    0.736  0.1525       0.4909            1
  431      8       1    0.627  0.1704       0.3680            1
  638      5       1    0.333  0.2296       0.0865            1

                rx=2
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
  353     13       1    0.943   0.141        0.703            1
  365     12       1    0.880   0.144        0.638            1
  464      9       1    0.789   0.165        0.523            1
  475      8       1    0.697   0.177        0.424            1
  563      7       1    0.597   0.182        0.329            1

> summary(survfit(fit, list(age=c(30,70), ecog.ps=c(2,3))))
Call: survfit.coxph(object = fit, newdata = list(age = c(30, 70),
                        ecog.ps = c( 2, 3)))

                rx=1
 time n.risk n.event survival1 survival2
   59     13       1     0.999   0.87905
  115     12       1     0.999   0.74575
```

```
  156      11        1       0.998   0.57399
  268      10        1       0.996   0.41765
  329       9        1       0.992   0.16677
  431       8        1       0.988   0.06492
  638       5        1       0.973   0.00161

                   rx=2
 time n.risk n.event survival1 survival2
  353      13        1       0.999    0.7093
  365      12        1       0.997    0.4739
  464       9        1       0.994    0.2494
  475       8        1       0.991    0.1207
  563       7        1       0.987    0.0489
```

The first call to survfit asks only for a single curve at the mean of the covariates. (The value of those means is stored in the fit object as fit$means.) The second call asks for curves for two hypothetical cohorts, one has an age of 30 and a performance score of two, the second is age 70 with a performance score of three. The printout requires some explanation. The printouts for the two treatment strata are listed in sequence: since the event times are different in the two strata they cannot be listed side-by-side. The survivals for the two age * ps cohorts are listed side by side since they are computed at the same time points.

Age is a very significant variable in the model: survival of a subject age 70 is significantly worse than one age 30. The standard error and confidence intervals are computed in the second example as they were in the first, and are present in the returned survival structure, but since their inclusion would be too wide for the paper the printing routine leaves them off.

The functions for predicted survival, unfortunately, share a basic flaw with S prediction in other models. If the model equation involves any special functions such as `ns`, `poly` *or involves factor variables*, then naive use of `survfit` will result in incorrect answers. In particular, for a factor it is important that `newdata` has the same number of levels as the original data, or contrasts will not be coded as they were originally.

```
> fit <- coxph(Surv(time, status) ~group, data=aml)     # group is a factor
> srv <- survfit(fit, list(group=1))                    # Wrong answer
> srv <- survfit(fit, list(group=1:2))                  # Wrong answer
> srv <- survfit(fit, list(group=aml$group[1]))         # Ok
> srv <- survfit(fit, list(group=factor(2, levels=1:2)))# Ok
> srv <- survfit(fit, list(group='Nonmaintained'))      # Error message
```

The same problems can be revealed using `lm(time~group, aml)` and the `predict` function. Admittedly, this is not much of a defense. For further discussion see sections 4.2.3 and 6.3.6 of Chambers and Hastie [9].

4.3 Complex Cox Models

The more complex Cox models usually involve time-dependent data. This is handled by using the *counting process* style of notation developed by Andersen and Gill [1]; for a technical reference see Fleming and Harrington [20]. The example below reprises an analysis of the Stanford heart transplant study found in Kalbfleisch and Prentice [33], section 5.5.3. (The data itself is taken from Crowley and Hu [11], as the values listed in the appendix of Kalbfleisch and Prentice are rounded and do not reproduce the results of their section 5.5).

The covariates in the study are

(age of acceptance in days /365.25) - 48
(date of acceptance in days since 1 Oct 1967) /365.25
prior surgery (1=yes, 0=no),

along with the time-dependent transplant variable. From the time of admission to the study until the time of death a patient was eligible for a heart transplant. The time to transplant depends on the next available donor heart with an appropriate tissue-type match.

In the analysis data frame, a transplanted patient is represented by two rows of data. The first is over the time period from enrollment (time 0) until the transplant, and has transplant=0, the second is over the period from transplant to death or last follow-up and has transplant=1. All other covariates are the same on the two lines. Subjects without a transplant are represented by a single row of data. Each row of data contains two variables `start` and `stop` which mark the time interval (start, stop] for the data, as well as an indicator variable `event` which is 1 if there was a death at time `stop`. Consider a subject who was transplanted at day 10, and then followed up until day 31. His first row of data applies over the time interval (0,10] and his second over the interval (10,31]. S code to create this data frame can be found in the Examples subdirectory of the source code.

Here is the code to fit the six models found in Kalbfleisch and Prentice. Note the use of the options statement, which forces the interaction terms to be coded in terms of dummy variables; see the S documentation for `contr.sum`. (The S default `contr.helmert` tries to create orthogonal contrasts, which rarely makes sense except in balanced anova designs — but that is just my opinion.) Since the data set contains tied death times, we must use the same approximation as K&P in order to match their coefficients.

```
> options(contrasts=c("contr.treatment", "contr.poly"))
> sfit.1 _ coxph(Surv(start, stop, event)~ (age + surgery)*transplant,
                         data=jasa1, method='breslow')
> sfit.2 _ coxph(Surv(start, stop, event)~ year*transplant,
                         data=jasa1, method='breslow')
```

```
> sfit.3 _ coxph(Surv(start, stop, event)~ (age + year)*transplant,
                         data=jasa1, method='breslow')
> sfit.4 _ coxph(Surv(start, stop, event)~ (year + surgery)*transplant,
                         data=jasa1, method='breslow')
> sfit.5 _ coxph(Surv(start, stop, event)~ (age +surgery)*transplant +year,
                         data=jasa1, method='breslow')
> sfit.6 _ coxph(Surv(start, stop, event)~ age*transplant + surgery + year,
                         data=jasa1, method='breslow')
> summary(sfit.1)
Call:
coxph(formula = Surv(start, stop, event) ~ (age + surgery) * transplant)

N= 172

                     coef exp(coef) se(coef)      z     p
               age  0.0138     1.014   0.0181  0.763 0.446
           surgery -0.5457     0.579   0.6109 -0.893 0.372
        transplant  0.1181     1.125   0.3277  0.360 0.719
    age:transplant  0.0348     1.035   0.0273  1.276 0.202
surgery:transplant -0.2916     0.747   0.7582 -0.385 0.701

                   exp(coef) exp(-coef) lower .95 upper .95
               age     1.014      0.986     0.979      1.05
           surgery     0.579      1.726     0.175      1.92
        transplant     1.125      0.889     0.592      2.14
    age:transplant     1.035      0.966     0.982      1.09
surgery:transplant     0.747      1.339     0.169      3.30

Rsquare= 0.07   (max possible= 0.969 )
Likelihood ratio test= 12.4  on 5 df,   p=0.0291
Efficient score test = 12  on 5 df,   p=0.0345

> sfit.2
Call:  coxph(formula = Surv(start, stop, event) ~ year * transplant)

                  coef exp(coef) se(coef)      z      p
           year -0.265     0.767    0.105 -2.518 0.0118
     transplant -0.282     0.754    0.514 -0.549 0.5831
year:transplant  0.136     1.146    0.141  0.967 0.3337

Likelihood ratio test=8.61  on 3 df, p=0.035    n= 172
```

One line of the above printout may generate confusion: `N = 172`. This is the number of *observations* in the data set, not the number of subjects. There are actually 103 patients, of which 69 had a transplant and are thus represented using 2 rows of data.

When there are time dependent covariates, the predicted survival curve can present something of a dilemma. The usual call's result is for a pseudo cohort whose covariates do not change–

```
>fit1 <- survfit(sfit.2, c(year=2, transplant=0) )
>fit2 <- survfit(sfit.2, c(year=2, transplant=1) )
```

The second curve, fit2, represents a cohort of patients whose `transplant` variable is always 1, even on day 0, i.e., patients who had no waiting time for a transplant. There were none of these in the study, so just what does it represent? Time dependent covariates that represent repeated measurements on a patient, such as a blood enzyme level, are particularly problematic. Since the model depended on these covariates, a proper predicted survival requires specification of the "future covariate history" for the patient in question. Because of this, it is all too easy to create predicted survival curves for "patients" that never would or perhaps never could exist. See the discussion of *internal* and *external* covariates in section 5.3 of Kalbfleisch and Prentice for a more complete exposition on these issues.

It is possible to obtain the projected survival for some particular pattern of change in the time dependent covariates. This requires a data frame with multiple lines, along with the flag `individual=T` to signal the survfit function that only one curve is desired. The example below gives the curve for a cohort aged 50, with prior surgery and a transplant at 6 months. That is, over the time interval (0,.5] the covariate set is (50, 1, 0), and over the time interval (.5, 3] it is (50, 1, 1). (The example uses days instead of years, however). In order to specify the time points both y and X variables must be present in the supplied data frame, though the value for event will be ignored.

```
> data <- data.frame(start=c(0,183), stop=c(183,3*365), event=c(1,1),
                     age=c(50,50),  surgery=c(1,1), transplant=c(0,1))
> survfit(sfit.1, data, individual=T)
```

Another useful extension is time dependent strata. The following examples come from a study of liver transplantation. As in the heart transplant study, there is a variable waiting time for any patient. One question of interest is the efficacy of liver transplant, another is the utility of a particular risk score developed at the Mayo Clinic. The variables for a given patient are:

- (tstart, tstop, status]: the time interval, open on the left and closed on the right. Status is 1 if the subject died at time tstop. All times are in days since enrollment in the study.
- base.rsk: The risk score at study entry. This covariate was defined by the "Mayo model" analysis, on an independent data set. The actual definition involves 5 variables: .871*log(bilirubin) + .039*age + ...

- trisk: time-dependent risk score. The latest value we have for the risk score (most patients are evaluated about once a year).
- transplant: time-dependent transplant status.
- tx.time: For transplanted patients, the number of days from enrollment to transplant.

There are 83 patients, who generate approximately 600 observations in the constructed data frame. The number of observations for a given patient depends on the number of determinations of his/her risk score, and on whether they were transplanted.

```
> attach(timedep)
> options(contrasts=c("contr.treatment", "contr.poly"))
> yy <- Surv(tstart, tstop, status)
> tfit1 <- coxph(yy ~ base.rsk, subset=(trplant==0))
> tfit2 <- coxph(yy ~ offset(base.rsk) + trplant)
> tfit3 <- coxph(yy ~ trisk, subset=(trplant==0))
> tfit4 <- coxph(yy ~ trisk *strata(trplant))
```

The first fit is a validity test of the Mayo model, it uses the baseline risk and only the time points before transplant. The fitted coefficient was 1.01, which is almost too good an agreement! The second fit is a test of the efficacy of transplant, after adjustment for the baseline risk. Given the fit of the first model, along with the fact that this data set is much smaller than the one used to develop the risk score, it seemed reasonable to not re-fit the risk variable.

Fit 3 tests the question of whether the most recent score has the same utility as the baseline score. The fitted coefficient was 1.3, which validates the 'medically obvious' assertion that recent lab data is more predictive than older values.

Fit 4 uses both the pre and post transplant data. A separate coefficient for trisk is fit to the pre and post experience, as well as a separate baseline hazard. The question is: adjusting for the effect of transplant, is the risk score's predictive ability the same in the pre and post-tx groups?

With a time dependent strata such as that in fit 4 above, there is a question of alignment. The Cox model's inference depends on the comparison of each subject with the other subjects in his strata who were "at risk" at the time of his death. Say that a particular patient has a transplant on day 200 and then dies on day 300. Should the risk group for his death be all transplanted patients who were alive 300 days after enrollment in the study, or all those alive 100 days after their transplant? If we believe the latter, and it does seem more plausible this particular disease and intervention, then a patient's time clock should restart at 0 after a transplant. Another argument is that the "baseline hazard" for a given subject is more dependent on their recent major surgery than on the time since enrollment. A third, self serving reason for realignment at the transplant date is that it will increase the power of the model, since it tends to maximize the average number of subjects in a risk set. The re-aligned model is fit by

```
> yyy <- Surv(tstart, tstop, status,
              origin=ifelse(trplant==1, tx.time, 0))
> coxph(yyy ~ trisk*strata(trplant) )
```

In SAS or BMDP, it is possible to mimic time dependent strata by breaking a subject into two new subjects. Because each subject's time interval implicitly starts at 0 in these packages, there is effectively a realignment of the data.

4.4 Differences in survival

There is a single function survdiff to test for differences between 2 or more survival curves. It implements the G^ρ family of Fleming and Harrington [18]. A single parameter ρ controls the weights given to different survival times, $\rho = 0$ yields the log-rank test and $\rho = 1$ the Peto-Wilcoxon. Other values give a test that is intermediate to these two. The default value is $\rho = 0$.

The interpretation of the formula is the same as for survfit, i.e., variables on the right hand side of the equation jointly break the patients into groups.

```
> survdiff(Surv(time, status)~ group, aml)

               N Observed Expected (O-E)^2/E
   Maintained 11        7   10.689     1.273
Nonmaintained 12       11    7.311     1.862

 Chisq= 3.4  on 1 degrees of freedom, p= 0.06534
```

For one-sample tests see the section on expected survival.

4.5 Competing Risks

This running example is taken from the paper by Wei, Lin and Weissfeld (WLW) [57]. They use a data set on time to recurrence of bladder cancer; a copy of the data may be obtained from statlib. A portion of the data is shown below:

```
                                             Recurrence Time
Treatment   Follow-up   Initial  Initial    ----------------
  group       time      number     size      1    2    3    4

    1          0          1         1
    1          1          1         3
```

```
1        4         2       1
1        7         1       1
1       10         5       1
1       10         4       1       6
1       14         1       1
1       18         1       1
1       18         1       3       5
1       18         1       1      12   16
1       23         3       3
1       23         3       3      10   15
1       23         1       1       3   16   23
1       23         1       1       3    9   21
   .                    .
   .                    .
   .                    .
```

Code for reading in this data can be found in the Examples directory. After reading it in, we have created the following data set

```
> bladder[1:20,]
   id rx size number start stop event enum
 1  1  1    1      3     0    1     0    1
 2  1  1    1      3     1    1     0    2
 3  1  1    1      3     1    1     0    3
 4  1  1    1      3     1    1     0    4
 5  2  1    2      1     0    4     0    1
 6  2  1    2      1     4    4     0    2
 7  2  1    2      1     4    4     0    3
 8  2  1    2      1     4    4     0    4
 9  3  1    1      1     0    7     0    1
10  3  1    1      1     7    7     0    2
11  3  1    1      1     7    7     0    3
12  3  1    1      1     7    7     0    4
13  4  1    5      1     0   10     0    1
14  4  1    5      1    10   10     0    2
15  4  1    5      1    10   10     0    3
16  4  1    5      1    10   10     0    4
17  5  1    4      1     0    6     1    1
18  5  1    4      1     6   10     0    2
19  5  1    4      1    10   10     0    3
20  5  1    4      1    10   10     0    4
    .                    .
    .                    .
    .                    .
```

Notice that this data set has exactly 4 observations for each subject. A second data set, bladder2, has had all of the rows with `start==stop` removed, and also has a fifth observation for some subjects (those with follow-up after the fourth recurrence).

The model explored in WLW is easily fit by the following commands. The key addition to the model is `cluster(id)`, which asserts that subjects with the same value of the variable `id` may be correlated. In order to compare the results directly to WLW, we wish to look at a different set of contrasts than the S default. These are created "by hand"

```
> options(contrasts='contr.treatment')
> wfit <- coxph(Surv(stop, event)~ (rx + size + number)* strata(enum) +
                 cluster(id), bladder, method='breslow')

> rx <- c(1,4,5,6)                 # the coefficients for the treatment effect
> cmat <- diag(4); cmat[,1] <- 1;          # a contrast matrix

> cmat %*% wfit$coef[rx]                   # the coefs in WLW (table 5)
[1] -0.5175702 -0.6194396 -0.6998691 -0.6504161

> wvar <- cmat %*% wfit$var[rx,rx] %*% t(cmat)  # var matrix (eqn 3.2)
> sqrt(diag(wvar))
[1] 0.3075006 0.3639071 0.4151602 0.4896743
```

The same coefficients can also be obtained, as WLW do, by performing four separate fits, and then combining the results.

```
> fit1 <- coxph(Surv(stop, event) ~ rx + size + number, bladder,
                subset=(enum=1), method='breslow')
> fit2 <- coxph(Surv(stop, event) ~ rx + size + number, bladder,
                subset=(enum=2), method='breslow')
> fit3 <- coxph(Surv(stop, event) ~ rx + size + number, bladder,
                subset=(enum=3), method='breslow')
> fit4 <- coxph(Surv(stop, event) ~ rx + size + number, bladder,
                subset=(enum=4), method='breslow')

> sc1 <- resid(fit1, type='score')
> sc2 <- resid(fit2, type='score')
> sc3 <- resid(fit3, type='score')
> sc4 <- resid(fit4, type='score')

> t11 <- fit1$var %*% t(sc1) %*% sc1 %*% fit1$var
> t12 <- fit1$var %*% t(sc1) %*% sc2 %*% fit2$var
        .
        .
        .
```

```
> t44 <- fit4$var %*% t(sc4) %*% sc4 %*% fit4$var

> wvar.all <- cbind(rbind( t11,   t12,   t13,     t14),
                    rbind(t(t12), t22,   t23,     t24),
                    rbind(t(t13), t(t23), t33,     t34),
                    rbind(t(t14), t(t24), t(t34),  t44))
> wvar <- wvar.all[c(1,4,7,10), c(1,4,7,10)]
```

The first of the individual fits is based on time from the start of the study until the first event, for all patients; the second is based on time from the start of the study until the second event, again for all patients, and etc. (In order to match results with WLW I have chosen the Breslow approximation above, but with the large number of ties this approximation may be ill advised.) The two approaches give exactly the same answer.

A major advantage of the compressed form, beyond the need for less typing, is that it allows us to easily fit submodels that are not available using separate `coxph` calls for each strata. In particular, the model

```
Surv(stop, event) ~ rx + (size + number)*strata(enum) +cluster(id),
```

differs only in that there is no treatment by strata interaction, and gives an average treatment coefficient of -.60, which is near to the weighted average of the marginal fits (based on the diagonal of /Cowvar) suggested in WLW.

The authors also give the results for two suggestions proposed by Prentice et al [47]. For time to first event these are the same as above. For the second event they use only patients who experienced at least one event, and use either the time from start of study (method a) or the time since the occurrence of the last event. The code for these is quite easy:

```
> fit2pa <- coxph(Surv(stop, event) ~ rx + size + number, bladder2,
                  subset=(enum==1))
> fit2pb <- coxph(Surv(stop-start,  event) ~ rx + size + number, bladder2,
                  subset=(enum==2))
```

Lastly, the authors also make use of an Andersen-Gill model in the comparison. This model has the advantage that it uses all of the data directly, but because of correlation it may again underestimate the variance of the relevant coefficients. A method to address this is given in a paper by Lee, Wei and Amato [35]; it is essentially the same method found in the WLW paper.

```
> afit <- coxph(Surv(start, stop, event) ~ rx + size + number + cluster(id),
                     data=bladder2)
> sqrt(diag(afit$var))
[1] 0.24183067 0.05689293 0.07221747
> sqrt(diag(afit$naive.var))
[1] 0.20007253 0.04800789 0.07025758
```

The naive estimate of standard error is .20, the correct estimate of .24 is intermediate between the naive estimate and the linear combination estimate. (Since this model does not include strata by covariate interactions, I would not expect an exact match). Further discussion on these estimators can be found in section 3.3.6.

4.6 Expected Survival

Expected survival is closely related to a standard method in epidemiology called *person years*, which consists of counting the total amount of follow-up time contributed by the subjects within any of several strata. Person-years analysis is accomplished within S by the `pyears` function. The main complication of this function, as opposed to a straightforward use of `tapply`, is that a subject may contribute to several different cells of the output array during his follow-up. For instance, if the desired output table were treatment group * age in years, a subject with 4 years of observation would contribute to 5 different cells of the table (4 cells if he entered the study exactly on his birthdate). The following counts up years of observation for the Stanford heart patients by age group and surgical status.

```
age <- jasa$accept.dt - jasa$birth.dt
ptable <- pyears(futime ~ tcut(age, c(0,50,60,70,100)) + surgery, jasa)
```

The `tcut` function has the same arguments as `cut`, but indicates as well that the category is time based. If `cut` had been used in the formula above, the final table would be based only on each subject's age at entry. With `tcut`, a subject who entered at age 58.5 and had 4 years of follow up would contribute 1.5 years to the 50-60 category and 2.5 years to the 60-70 category. A consequence of this is that the age and fu.time variables must be in the same units for the calculation to proceed correctly, in this case both should be in years given the cutpoints that were chosen. The surgery variable is treated as a factor, exactly as it would be in `surv.fit`.

The output of `pyears` is a list of arrays containing the total amount of time contributed to each cell and the number of subjects who contributed some fraction of time to each cell. If the response variable is a `Surv` object, then the output will also contain an array with the observed number of events for each cell. If a rate table is supplied, the output will contain an array with the expected number of events in each cell. These can be used to compute observed and expected rates, along with confidence intervals.

A rate table argument is optional for person-years, but is required to compute expected survival. Rate tables are included with the package for the US population and for Minnesota, Florida, and Arizona (states where Mayo has a clinic). The addition of rate tables for other areas is a tedious but straightforward exercise. US and state rate tables contain the expected hazard rate for a subject, stratified by age, sex, calendar year, and optionally by race. (User created rate tables have no restrictions on the number or names of the stratifications; see the documentation for survexp.uswhite for details.) When using a rate table, it is important that all time variables be in the same units as were used for the table — for the US tables this is hazard/day, so time must be in days.

```
fit <- survexp(futime ~ surgery + ratetable(age=age, sex='Male',
                  year=accept.dt), data=jasa, ratetable=survexp.us)
```

The formula contains an observation time futime, a grouping variable surgery which will cause the output to contain 2 curves, and a special function ratetable. The purpose of this function is to match the data frame's variables to the proper dimensions of the ratetable. Argument order is unimportant, the necessary key words 'age', 'sex', and 'year' are contained in the dimid attribute of the actual ratetable survexp.us. The jasa data set does not contain a sex variable, so we have defaulted the value to "Male". Default values such as this must either be an integer subscript or match one of the dimnames. The above example produces a cohort survival curve, which is almost always plotted along with the observed (Kaplan-Meier) survival of the data for visual comparison. There are 3 different methods for calculating the cohort curve, which are discussed in more detail in section 3.2.2. They are the conditional method shown above, which uses the actual death or censoring time, the method of Hakulinen, which instead uses the potential follow-up time of each subject, and the uncensored population method of Ederer, Axtel and Cutler, which requires no response variable.

Formal tests of observed versus expected survival are usually based not on the cohort curve directly but on the individual expected survival probabilities for each subject. These probabilities are always based on the actual death/censoring time:

```
surv.prob <- survexp(futime ~ ratetable(age=age, sex='Male',
                        year=accept.dt), data=jasa, cohort=F)
newtime <- -log(surv.prob)   #the cumulative hazard for each subject
test <- glm(fustat ~ offset(log(newtime)), family=poisson)
```

When cohort=F the survexp function returns a vector of survival probabilities, one per subject. The negative log of the survival probability can be treated as an "adjusted time" for the subject for the purposes of modeling. The one-sample log-rank test for equivalence of the observed survival to that of the referent population is then the test for intercept=0 in the Poisson regression model shown. A test for treatment difference, adjusted for any age-sex differences between the two arms, would be obtained by adding a treatment variable to the model.

User created rate tables can also be used. Table 1 shows yearly death rates per 100,000 subjects based on their smoking status [60]. A stored raw data set contains this data, with the "Never smoked" data replicated where the lower table shows blanks,

Males			Former smokers (1-20 cig/day)					
			Duration of abstinence (yr)					
Age	Never Smoked	Current Smokers	< 1	1-2	3-5	6-10	11-15	≥ 16
45-49	186.0	439.2	234.4	365.8	159.6	216.9	167.4	159.5
50-54	255.6	702.7	544.7	431.0	454.8	349.7	214.0	250.4
55-59	448.9	1,132.4	945.2	728.8	729.4	590.2	447.3	436.6
60-64	733.7	1,981.1	1,177.7	1,589.2	1,316.5	1,266.9	875.6	703.0
65-60	1,119.4	3,003.0	2,244.9	3,380.3	2,374.9	1,820.2	1,669.1	1,159.2
70-74	2,070.5	4,697.5	4,255.3	5,083.0	4,485.0	3,888.7	3,184.3	2,194.9
75-79	3,675.3	7,340.6	5,882.4	6,597.2	7,707.5	4,945.1	5,618.0	4,128.9

Males			Former smokers ($\geq$ 21 cig/day)					
			Duration of abstinence (yr)					
Age	Never Smoked	Current Smokers	< 1	1-2	3-5	6-10	11-15	≥ 16
45-49		610.0	497.5	251.7	417.5	122.6	198.3	193.4
50-54		915.6	482.8	500.7	488.9	402.9	393.9	354.3
55-59		1,391.0	1,757.1	953.5	1,025.8	744.0	668.5	537.8
60-64		2,393.4	1,578.4	1,847.2	1,790.1	1,220.7	1,100.0	993.3
65-69		3,497.9	2,301.8	3,776.6	2,081.0	2,766.4	2,268.1	1,230.7
70-74		5,861.3	3,174.6	2,974.0	3,712.9	3,988.8	3,268.6	2,468.9
75-79		6,250.0	4,000.0	4,424.8	7,329.8	6,383.0	7,666.1	5,048.1

Table 1: *Table 1*

followed by the same data for females. A rate table is created using the following S code.

```
temp <- matrix(scan("data.smoke"), ncol=8, byrow=T)/100000
smoke.rate <- c(rep(temp[,1],6), rep(temp[,2],6), temp[,3:8])
attributes(smoke.rate) <- list(
    dim=c(7,2,2,6,3),
    dimnames=list(c("45-49","50-54","55-59","60-64","65-69","70-74","75-79"),
                  c("1-20", "21+"),
                  c("Male","Female"),
                  c("<1", "1-2", "3-5", "6-10", "11-15", ">=16"),
                  c("Never", "Current", "Former")),
    dimid=c("age", "amount", "sex", "duration", "status"),
    factor=c(0,1,1,0,1),
    cutpoints=list(c(45,50,55,60,65,70,75),NULL, NULL,
                                  c(0,1,3,6,11,16),NULL),
    class='ratetable'
    )
is.ratetable(smoke.rate)
```

The smoking data cross-classifies subjects by 5 characteristics: age group, sex, status (never, current or former smoker), the number of cigarettes consumed per day, and, for the prior smokers, the duration of abstinence. In our S implementation, a ratetable is an array with 4 extra attributes of which one is the class. In order to cast the above data into a single array, the rates for never and current smokers needed to be replicated across all 6 levels of the duration, we do this in first creating the `smoke.rate` vector. The vector of rates is then saddled with a list of descriptive attributes. The dim and dimnames are as they would be for an array, and give its shape and printing labels, respectively. Dimid is the list of keywords that will be recognized by the `ratetable` function, when this table is later used within the `survexp` or `pyears` function. For the US total table, for instance, the keywords are "age", "sex", and "year". These keywords must be in the same order as the array dimensions. The factor attribute identifies each dimension as fixed or mobile in time. For a subject with 15 years of follow-up, for instance, the sex category remains fixed over this 15 years, but the age and duration of abstinence continue to change; more than 1 of the age groups will be referenced to calculate his/her total hazard. For each dimension that is not a factor, the starting value for each of the rows of the array must be specified so that the routine can change rows at the appropriate time, this is specified by the cutpoints. The cutpoints are null for a factor dimension.

Because these attributes must be self-consistent, it is wise to carefully check them for any user created rate table. The `is.ratetable` function does this automatically.

As a contrived example, we can apply this table to the Stanford data, assuming that all of the subjects were current heavy smokers (after all, they have heart disease). The max follow-up for any subject in the Stanford data set was April 1 1974.

```
ptime <- mdy.date(4,1,74) - jasa$entry.dt   #max potential fu
ptime <- ifelse(jasa$fustat==1, ptime, jasa$futime) / 365.24
exp4 <- survexp(ptime ~ ratetable(age=(age=(age/365.24), status="Current",
                                  amount=2, duration=1, sex='Male'),
          data=jasa, ratetable=smoke.rate, conditional=F, scale=1)
```

This example does illustrate some points. For any factor variable, the `ratetable` function allows use of either a character name or the actual column number. Since I have chosen the current smoker category, duration is unimportant, and any value could have been specified. The most important point is to note that `age` has been rescaled. This table contains rates per year, whereas the US tables contained rates per day. It is crucial that all of the time variables (age, duration, etc) be scaled to the same units, or the results may not be even remotely correct. The US rate tables were created using days as the basic unit since year of entry will normally be a julian date; for the smoking data years seemed more natural.

An optional portion of a rate table, not illustrated in the example above, is a `summary` attribute. This is a user written function which will be passed a matrix and can return a character string. The matrix will have one column per dimension of the ratetable, in the order of the `dimid` attribute, and will have already been processed for illegal values. To see an example of a summary function, type `attr(survexp.us, 'summary')` at the S prompt. In this summary function the returned character string lists the range of ages and calendar years in the input, along with the number of males and females. This string is included in the output of `survexp`, and will be listed as part of the printed output. This printout is the only good way of catching errors in the time units; for instance, if the string contained "age ranges from .13 to .26 years", it is a reasonable guess that age was given in years when it should have been stated in days.

As an aside, many entries in the smoke.rate table are based on small samples. In particular, the data for females who are former smokers contains 2 zeros. Before serious use these data should be smoothed. As a trivial example:

```
newrate <- smoke.rate
temp <- newrate[ ,1,2, ,3]
fit <- gam(temp ~ s(row(temp)) + s(col(temp)))
newrate[,1,2,,3] <- predict(fit)
```

A realistic effort would begin and end with graphical assessment, and likely make use of the individual sample sizes as well. The raw rates data, but not the sample sizes, has been included in the Examples directory.

5 Parametric Regression

5.1 Usage

The `survreg` function implements the class of parametric accelerated failure time models. Assume that the survival time y satisfies $\log(y) = X'\beta + \sigma W$, for W from some given distribution. Then if $\Lambda_w(t)$ is the cumulative hazard function for W, the cumulative hazard function for subject i is $\Lambda_w[\exp(-\eta_i/\sigma)t]$, that is, the time scale for the subject is accelerated by a constant factor. A good description of the models is found in chapter 3 of Kalbfleisch and Prentice [33].

The following fits a Weibull model to the lung cancer data set included in S-Plus.

```
> fit <- survreg(Surv(time, status) ~ age + sex + ph.karno, data=lung,
        dist='weibull')
> fit
Call:
survreg(formula = Surv(time, status) ~ age + sex + ph.karno, data = lung, dist
         = "weibull")

Coefficients:
 (Intercept)          age       sex    ph.karno
    5.326344 -0.008910282 0.3701786 0.009263843

Scale= 0.7551354

Loglik(model)= -1138.7   Loglik(intercept only)= -1147.5
        Chisq= 17.59 on 3 degrees of freedom, p= 0.00053
n=227 (1 observations deleted due to missing)
```

The code for the routines has undergone substantial revision between releases 4 and 5 of the code. Calls to the older version are not compatable with all of the changes, users can use the `survreg.old` function if desired, which retains the old argument style (but uses the newer maximization code). Major additions included penalzed models, strata, user specified distributions, and more stable maximization code.

5.2 Strata

In a Cox model the `strata` statement is used to allow separate baseline hazards for subgroups of the data, while retaining common coefficients for the other covariates across groups. For parametric models, the statement allows for a separate scale parameter for each subgroup, but again keeping the other coefficients common across groups. For instance, assume that separate "baseline" hazards were desired for males and females in the lung cancer data set. If we think of the intercept and scale as the baseline shape, then an appropriate model is

```
> sfit <- survreg(Surv(time, status) ~ sex + age + ph.karno + strata(sex),
        data=lung)
> sfit
Coefficients:
 (Intercept)       sex          age   ph.karno
    5.059089 0.3566277 -0.006808082 0.01094966

Scale:
     sex=1     sex=2
 0.8165161 0.6222807

Loglik(model)= -1136.7   Loglik(intercept only)= -1146.2
        Chisq= 18.95 on 3 degrees of freedom, p= 0.00028
```

The intercept only model used for the likelihood ratio test has 3 degrees of freedom, corresponding to the intercept and two scales, as compared to the 6 degrees of freedom for the full model.

This is quite different from the effect of the `strata` statement in `censorReg`; there it acts as a 'by' statement and causes a totally separate model to be fit to each gender. The same fit (but not as nice a printout) can be obtained from `survreg` by adding an explicit interaction to the formula:

```
Surv(time,status) ~ sex + (age + ph.karno)*strata(sex)
```

5.3 Penalized models

Let the linear predictor for a `survreg` model be $\eta = X\beta + Z\omega$, and consider maximizing the penalized log-likelihood

$$PLL = LL(y; \beta, \omega) - p(\omega; \theta)\,,$$

where β and ω are the unconstrained effects, respectively, X and Z are the covariates, p is a function that penalizes certain choices for ω, and θ is a vector of tuning parameters.

For instance, ridge regression is based on the penalty $p = \theta \sum \omega_j^2$; it shrinks coefficients towards zero.

Penalties have been implemented in `survreg` in exactly the same way as in `coxph`, and the reader is referred to that documentation for a complete description of the capabilities. In particular the exact same penalty functions, e.g., ridge(), pspline(), and frailty() can be used in the formula of a parametric survival model. The values of the tuning parameter(s) θ may differ, however. The Cox model is closely related to exponential regression; the baseline hazard estimate causes the other coefficients to behave consistent with a scale of 1. For a frailty term added to a Weibull model with estimated scale of c, a tuning parameter of $c\theta$ appears to be similar to a value of θ in a Cox fit. See for instance Louis and xxx.

More work on understanding these models is clearly in order.

5.4 Specifying a distribution

The fitting routine is quite general, and can accept any distribution that spans the real line for W, and any monotone transformation of y. The standard set of distributions is contained in a list `survreg.distributions`. Elements of the list are of two types. Basic elements are a description of a distribution. Here is the entry for the logistic family:

```
logistic = list(
    name  = "Logistic",
    variance = function(parm) pi^2/3,
    init  = function(x, weights, ...) {
        mean <- sum(x*weights)/ sum(weights)
        var  <- sum(weights*(x-mean)^2)/ sum(weights)
        c(mean, var/3.2)
        },
    deviance= function(y, scale, parms) {
        status <- y[,ncol(y)]
        width <- ifelse(status==3,(y[,2] - y[,1])/scale, 0)
        center <- y[,1] - width/2
        temp2 <- ifelse(status==3, exp(width/2), 2) #avoid a log(0) message
        temp3 <- log((temp2-1)/(temp2+1))
        best <- ifelse(status==1, -log(4*scale),
                                  ifelse(status==3, temp3, 0))
        list(center=center, loglik=best)
        },
    density = function(x, ...) {
        w <- exp(x)
        cbind(w/(1+w), 1/(1+w), w/(1+w)^2, (1-w)/(1+w), (w*(w-4) +1)/(1+w)^2)
        },
    quantile = function(p, ...) log(p/(1-p))
    )
```

- Name is used to label the printout.
- Variance contains the variance of the distribution. For distributions with an optional parameter such as the t-distribution, the `parm` argument will contain those parameters.
- Deviance gives a function to compute the deviance residuals. More on this is explained below in the mathematical details.

- The density function gives the necessary quantities to fit the distribution. It should return a matrix with columns $F(x)$, $1 - F(x)$, $f(x)$, $f'(x)/f(x)$ and $f''(x)/f(x)$, where f' and f'' are the first and second derivatives of the density function, respectively.

- The quantiles function returns quantiles, and is used for residuals.

The reason for returning both F and $1 - F$ in the density function is to avoid round off error when $F(x)$ is very close to 1. This is quite simple for symmetric distributions, in the Gaussian case for instance we use `qnorm(x)` and `qnorm(-x)` respectively. (In the intermediate steps of iteration very large deviates may be generated, and a probabilty value of zero will cause further problems.)

Here is an example of the second type of entry:

```
exponential = list(
    name  = "Exponential",
    dist  = "extreme",
    scale =1 ,
    trans = function(y) log(y),
    dtrans= function(y) 1/y ,
    itrans= function(x) exp(x)
    )
```

This states that an exponential fit is computed by fitting an extreme value distribution to the log transformation of y. (The distribution pointed to must not itself be a pointer to another). The extreme value distribution is restricted to have a scale of 1. The first derivative of the transformation, `dtrans`, is used to adjust the final log-likelihood of the model back to the exponential's scale. The inverse transformation `itrans` is used to create predicted values on the original scale.

The formal rules for an entry are that it must include a name, either the "dist" component or the set "variance","init", "deviance", "density" and "quantile", an optional scale, and either all or none of "trans", "dtrans" and "itrans".

The `dist="weibull"` argument to the `survreg` function chooses the appropriate list from the survreg.distributions object. User defined distributions of either type can be specified by supplying the appropriate list object rather than a character string. Distributions should, in general, be defined on the entire real line. If not the minimizer used is likely to fail, since it has no provision for range restrictions.

Currently supported distributions are

- basic
  - (least) Extreme value
  - Gaussian

  - Logistic
  - t-distribution

- transformations
  - Exponential
  - Weibull
  - Log-normal ('lognormal' or 'loggaussian')
  - Log-logistic ('loglogistic')

5.5 Residuals

5.5.1 Response

The target return value is $y - \hat{y}$, but what should we use for y when the observation is not exact? We will let $\hat{y}_0$ be the MLE for the location parameter μ over a data set with only the observation of interest, with σ fixed at the solution to the problem as a whole, subject to the constraint that μ be consistent with the data. That is, for an observation right censored at $t = 20$, we constain $\mu \geq 20$, similarly for left censoring, and constrain μ to lie within the two endpoints of an interval censored observation. To be consistent as the width of an interval censored observation goes to zero, this definition does require that the mode of the density lies at zero.

For exact, left, and right censored observations $\hat{y}_0 = y$, so that this appears to be an ordinary response residual. For interval censored observations from a symmetric distribution, $\hat{y}_0 =$ the center of the censoring interval. The only unusual case, then, is for a non-symmetric distribution such as the extreme value. As shown later in the detailed information on distributions, for the extreme value distribution this occurs for $\hat{y}_0 = y^l - \log(b/[exp(b) - 1])$, where $b = y^u - y^l$ is the length of the interval.

5.5.2 Deviance

Deviance residuals are response residuals, but transformed to the log-likelihood scale.

$$d_i = sign(r_i)\sqrt{LL(y_i, \hat{y}_0; \sigma) - LL(y_i, \eta_i; \sigma)}$$

The definition for $\hat{y}_0$ used for response residuals, however, could lead to the square root of a negative number for left or right censored observations, e.g., if the predicted value for a right censored observation is less than the censoring time for that observation. For these observations we let $\hat{y}_0$ be the *unconstrained* maximum, which leads to $yhat_0 = -\infty$ and $+\infty$ for right and left censored observations, respectively, and a log-likelihood term of 0.

The advantages of these residuals for plotting and outlier detection are nicely detailed in McCullagh and Nelder [44]. However, unlike GLM models, deviance residuals for interval censored data are not free of the scale parameter. This means that if there are interval censored data values and one fits two models A and B, say, that the sum of the squared deviance residuals for model A minus the sum for model B is *not* equal to the difference in log-likelihoods. This is one reason that the current `survreg` function does not inherit from class `glm`: `glm` models use the deviance as the main summary statistic in the printout.

5.5.3 Dfbeta

The `dfbeta` residuals are a matrix with one row per subject and one column per parameter. The ith row gives the approximate change in the parameter vector due to observation i, i.e., the change in $\hat{\beta}$ when observation i is added to a fit based on all observations but the ith. The `dfbetas` residuals scale each column of this matrix by the standard error of the respective parameter.

5.5.4 Working

As shown in section 5.7 below, the Newton-Raphson iteration used to solve the model can be viewed as an iteratively reweighted least squares problem with a dependent variable of "current prediction - correction". The working residual is the correction term.

5.5.5 Likelihood displacement residuals

Escobar and Meeker [17] define a matrix of likelihood displacement residuals for the accelerated failure time model. The full residual information is a square matrix $\ddot{A}$, with dimension the number of pertubations considered. Three examples are developed in detail, all with dimension n, the number of observations.

Case weight pertubations measure the overall effect on the parameter vector of dropping a case. Let V be the variance matrix of the model, and L the n by p matrix with elements $\partial L_i / \partial \beta_j$, where L_i is the likelihood contribution of the ith observation. Then $\ddot{A} = LVL'$. The residuals function returns the diagonal values of the matrix. Note that LV equals the `dfbeta` residuals.

Response pertubations correspond to a change of 1 σ unit in one of the response values. For a Gaussian linear model, the equivalent computation yields the diagonal elements of the hat matrix.

Shape pertubations measure the effect of a change in the log of the scale parameter by 1 unit.

The matrix residual type returns the raw values that can be used to compute these and other LD influence measures. The result is an n by 6 matrix, containing columns for

$$L_i \quad \frac{\partial L_i}{\partial \eta_i} \quad \frac{\partial^2 L_i}{\partial \eta_i^2} \quad \frac{\partial L_i}{\partial \log(\sigma)} \quad \frac{\partial L_i}{\partial \log(\sigma)^2} \quad \frac{\partial^2 L_i}{\partial \eta \partial \log(\sigma)}$$

5.6 Predicted values

5.6.1 Linear predictor and response

The linear predictor is $\eta_i = x_i'\hat{\beta}$, where x_i is the covariate vecor for subject i and $\hat{\beta}$ is the final parameter estimate. The standard error of the linear predictor is $x_i'Vx_i$, where V is the variance matrix for $\hat{\beta}$.

The predicted response is identical to the linear predictor for fits to the untransformed distributions, i.e., the extreme-value, logistic and Gaussian. For transformed distributions such as the Weibull, for which $\log(y)$ is from an extreme value distribution, the linear predictor is on the transformed scale and the response is the inverse transform, e.g. $\exp(\eta_i)$ for the Weibull. The standard error of the transformed response is the standard error of η_i, times the first derivative of the inverse transformation.

5.6.2 Terms

Predictions of type terms are useful for examination of terms in the model that expand into multiple dummy variables, such as factors and p-splines. The result is a matrix with one column for each of the terms in the model, along with an optional matrix of standard errors. Here is an example using psplines on the 1980 Stanford data

```
> fit <- survreg(Surv(time, status) ~ pspline(age, df=3) + t5, stanford2,
        dist='lognormal')
> tt <- predict(fit, type='terms', se.fit=T)
> yy <- cbind(tt$fit[,1], tt$fit[,1] -1.96*tt$se.fit[,1],
                          tt$fit[,1] +1.96*tt$se.fit[,1])
> matplot(stanford2$age, yy, type='l', lty=c(1,2,2))

> plot(stanford2$age, stanford2$time, log='y',
        xlab='Age', ylab='Days', ylim=c(.1, 10000))
> matlines(stanford2$age, exp(yy+ attr(tt$fit, 'constant')), lty=c(1,2,2))
```

The second plot puts the fit onto the scale of the data, and thus is similar in scope to figure 1 in Escobar and Meeker [17]. Their plot is for a quadratic fit to age, and without T5 mismatch score in the model.

5.6.3 Quantiles

If predicted quantiles are desired, then the set of probability values p must also be given to the `predict` function. A matrix of n rows and p columns is returned, whose ij element is the p_jth quantile of the predicted survival distribution, based on the covariates of subject i. This can be written as $X\beta + z_q\sigma$ where z_q is the qth quantile of the parent distribution. The variance of the quantile estimate is then cVc' where V is the variance matrix of (β, σ) and $c = (X, z_q)$.

In computing confidence bands for the quantiles, it may be preferable to add standard errors on the untransformed scale. For instance, consider the motor reliability data of Kalbfleisch and Prentice [?].

```
> fit <- survreg(Surv(time, status) ~ temp, data=motors)
> q1 <- predict(fit, data.frame(temp=130), type='quantile',
                p=c(.1, .5, .9), se.fit=T)
> ci1 <- cbind(q1$fit, q1$fit - 1.96*q1$se.fit, q1$fit + 1.96*q1$se.fit)
> dimnames(ci1) <- list(c(.1, .5, .9), c("Estimate", "Lower ci", "Upper ci"))
> round(ci1)
    Estimate Lower ci Upper ci
0.1    15935     9057    22812
0.5    29914    17395    42433
0.9    44687    22731    66643

> q2 <- predict(fit, data.frame(temp=130), type='uquantile',
                p=c(.1, .5, .9), se.fit=T)
> ci2 <- cbind(q2$fit, q2$fit - 1.96*q2$se.fit, q2$fit + 1.96*q2$se.fit)
> ci2 <- exp(ci2)  #convert from log scale to original y
> dimnames(ci2) <- list(c(.1, .5, .9), c("Estimate", "Lower ci", "Upper ci"))
> round(ci2)
    Estimate Lower ci Upper ci
0.1    15935    10349    24535
0.5    29914    19684    45459
0.9    44687    27340    73041
```

Using the (default) Weibull model, the data is fit on the $\log(y)$ scale. The confidence bands obtained by the second method are asymmetric and may be more reasonable. They are also guarranteed to be > 0.

This example reproduces figure 1 of Escobar and Meeker [17].

```
> plot(stanford2$age, stanford2$time, log='y',
        xlab='Age', ylab='Days', ylim=c(.01, 10^6), xlim=c(1,65))
> fit <- survreg(Surv(time, status) ~ age + age^2, stanford2,
        dist='lognormal')
> qq <- predict(fit, newdata=list(age=1:65), type='quantile',
         p=c(.1, .5, .9))
```

```
> matlines(1:65, qq, lty=c(1,2,2))
```

Note that the percentile bands on this figure are really quite a different object than the confidence bands on the spline fit. The latter reflect the uncertainty of the fitted estimate and are related to the standard error. The quantile bands reflect the predicted distribution of a subject at each given age (assuming no error in the quadratic estimate of the mean), and are related to the standard deviation of the population.

5.7 Fitting the model

With some care, parametric survival can be written so as to fit into the iteratively reweighted least squares formulation used in Generalized Linear Models of McCullagh and Nelder [44]. A detailed description of this setup for general maximum likelihood computation is found in Green [24].

Let y be the data vector (possibly transformed), and x_i be the vector of covariates for the ith observation. Assume that

$$z_i \equiv \frac{y_i - x_i'\beta}{\sigma} \sim f$$

for some distribution f, where y may be censored.

Then the likelihood for y is

$$l = \left(\prod_{exact} f(z_i)/\sigma \right) \left(\prod_{right} \int_{z_i}^{\infty} f(u)du \right) \left(\prod_{left} \int_{-\infty}^{z_i} f(u)du \right) \left(\prod_{interval} \int_{z_i^l}^{z_i^u} f(u)du \right),$$

where "exact", "right", "left", and "interval" refer to uncensored, right censored, left censored, and interval censored observations, respectively, and z_i^l, z_i^u are the lower and upper endpoints, respectively, for an interval censored observation. Then the log likelihood is defined as

$$LL = \sum_{exact} g_1(z_i) - log(\sigma) + \sum_{right} g_2(z_i) + \sum_{left} g_3(z_i) + \sum_{interval} g_4(z_i, z_i^*), \qquad (21)$$

where $g_1 = \log(f)$, $g_2 = \log(1 - F)$, etc.

Derivatives of the LL with respect to the regression parameters are:

$$\begin{aligned}
\frac{\partial LL}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial g}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} \\
&= \sum_{i=1}^{n} x_{ij} \frac{\partial g}{\partial \eta_i} && (22) \\
\frac{\partial^2 LL}{\partial \beta_j \beta_k} &= \sum x_{ij} x_{ik} \frac{\partial^2 g}{\partial \eta_i^2}, && (23)
\end{aligned}$$

where $\eta = X'\beta$ is the vector of linear predictors.

Ignore for a moment the derivatives with respect to σ (or treat it as fixed). The Newton-Raphson step defines an update δ

$$(X^T DX)\delta = X^T U,$$

where D is the diagonal matrix formed from $-g''$, and U is the vector g'. The current estimate β satisfies $X\beta = \eta$, so that the new estimate $\beta + \delta$ will have

$$\begin{aligned}(X^T DX)(\beta + \delta) &= X^T D\eta + X^T U \\ &= (X^T D)(\eta + D^{-1}U)\end{aligned}$$

Thus if we treat σ as fixed, iteration is equivalent to IRLS with weights of $-g''$ and adjusted dependent variable of $\eta - g'/g''$. At the solution to the iteration we might expect that $\hat{\eta} \approx y$; and a weighted regression with y replacing η gives, in general, good starting estimates for the iteration. (For an interval censored observation we use the center of the interval as 'y'). Note that if all of the observations are uncensored, then this reduces to using the linear regression of y on X as a starting estimate: $y = \eta$ so $z = 0$, thus $g' = 0$ and g'' = a constant (all of the supported densities have a mode at zero).

This clever starting estimate is introduced in Generalized Linear Models (McCullagh and Nelder [44]), and works extremely well in that context: convergence often occurs in 3-4 iterations. It does not work quite so well here, since a "good" fit to a right censored observation might have $\eta >> y$. Secondly, the other coefficients are not independent of σ, and σ often appears to be the most touchy variable in the iteration.

Most often, the routines will be used with $\log(y)$, which corresponds to the set of accelerated failure time models. The transform can be applied implicitly or explicitly; the following two fits give identical coefficients:

```
> fit1 <- survreg(Surv(futime, fustat)~ age + rx, fleming, dist='weibull')
> fit2 <- survreg(Surv(log(futime), fustat) ~ age + rx, data=fleming,
                  dist='extreme')
```

The log-likelihoods for the two fits differ by a constant, i.e., the sum of $d\log(y)/dy$ for the uncensored observations, and certain predicted values and residuals will be on the y versus $\log(y)$ scale.

5.8 Derivatives

This section is very similar to the appendix of Escobar and Meeker [17], differing only in our use of $\log(\sigma)$ rather than σ as the natural parameter. Let f and F denote the density and distribution functions, respectively, of the distributions. Using (21) as the definition of $g1, \ldots, g4$ we have

$$
\begin{aligned}
\frac{\partial g_1}{\partial \eta} &= -\frac{1}{\sigma}\left[\frac{f'(z)}{f(z)}\right] \\
\frac{\partial g_4}{\partial \eta} &= -\frac{1}{\sigma}\left[\frac{f(z^u) - f(z^l)}{F(z^u) - F(z^l)}\right] \\
\frac{\partial^2 g_1}{\partial \eta^2} &= \frac{1}{\sigma^2}\left[\frac{f''(z)}{f(z)}\right] - (\partial g_1/\partial \eta)^2 \\
\frac{\partial^2 g_4}{\partial \eta^2} &= \frac{1}{\sigma^2}\left[\frac{f'(z^u) - f'(z^l)}{F(z^u) - F(z^l)}\right] - (\partial g_4/\partial \eta)^2 \\
\frac{\partial g_1}{\partial \log \sigma} &= -\left[\frac{zf'(z)}{f(z)}\right] \\
\frac{\partial g_4}{\partial \log \sigma} &= -\left[\frac{z^u f(z^u) - z^l f(z^l)}{F(z^u) - F(z^l)}\right] \\
\frac{\partial^2 g_1}{\partial (\log \sigma)^2} &= \left[\frac{z^2 f''(z) + zf'(z)}{f(z)}\right] - (\partial g_1/\partial \log \sigma)^2 \\
\frac{\partial^2 g_4}{\partial (\log \sigma)^2} &= \left[\frac{(z^u)^2 f'(z^u) - (z^l)^2 f'(z_l)}{F(z^u) - F(z^l)}\right] - \partial g_1/\partial \log \sigma(1 + \partial g_1/\partial \log \sigma) \\
\frac{\partial^2 g_1}{\partial \eta \partial \log \sigma} &= \frac{zf''(z)}{\sigma f(z)} - \partial g_1/\partial \eta(1 + \partial g_1/\partial \log \sigma) \\
\frac{\partial^2 g_4}{\partial \eta \partial \log \sigma} &= \frac{z^u f'(z^u) - z^l f'(z^l)}{\sigma[F(z^u) - F(z^l)]} - \partial g_4/\partial \eta(1 + \partial g_4/\partial \log \sigma)
\end{aligned}
$$

To obtain the derivatives for g_2, set the upper endpoint z_u to ∞ in the equations for g_4. To obtain the equations for g_3, left censored data, set the lower endpoint to $-\infty$.

After much experimentation, a further decision was made to do the internal iteration in terms of $\log(\sigma)$. This avoids the boundary condition at zero, and also helped the iteration speed considerably for some test cases. The changes to the code were not too great. By the chain rule

$$
\begin{aligned}
\frac{\partial LL}{\partial \log \sigma} &= \sigma \frac{\partial LL}{\partial \sigma} \\
\frac{\partial^2 LL}{\partial (\log \sigma)^2} &= \sigma^2 \frac{\partial^2 LL}{\partial \sigma^2} + \sigma \frac{\partial LL}{\partial \sigma} \\
\frac{\partial^2 LL}{\partial \eta \partial \log \sigma} &= \sigma \frac{\partial^2}{\partial \eta \partial \sigma}
\end{aligned}
$$

At the solution $\partial LL/\partial\sigma = 0$, so the variance matrix for σ is a simple scale change of the returned matrix for $\log(\sigma)$.

5.9 Distributions

5.9.1 Gaussian

Everyone's favorite distribution. The continual calls to Φ may make it slow on censored data, however. Because there is no closed form for Φ, only the equations for g_1 simplify from the general form given in section 2 above.

$$\begin{aligned}
\mu = 0 \quad &, \quad \sigma^2 = 1 \\
F(z) &= \Phi(z) \\
f(z) &= \exp(-z^2/2)/\sqrt{2\pi} \\
f'(z) &= -zf(z) \\
f''(z) &= (z^2-1)f(z)
\end{aligned}$$

For uncensored data, the standard glm results are clear by substituting $g_1 = -z/\sigma$ into equations 1-5. The first derivative vector is equal to $X'r$ where $r = -z/\sigma$ is a scaled residual, the update step $I^{-1}U$ is independent of the estimate of σ, and the maximum likelihood estimate of $n\sigma^2$ is the sum of squared residuals. None of these hold so neatly for right censored data.

5.9.2 Extreme value

If y is Weibull then $\log(y)$ is distributed according to the (least) extreme value distribution. As stated above, fits on the latter scale are numerically preferable because it removes the range restriction on y. A Weibull distribution with the scale restricted to 1 gives an exponential model.

$$\mu = -\gamma = .5722\ldots, \ \sigma^2 = \pi^2/6$$

$$\begin{aligned}
F(z) &= 1 - \exp(-w) \\
f(z) &= we^{-w} \\
f'(z) &= (1-w)f(z) \\
f''(z) &= (w^2 - 3w + 1)f(z)
\end{aligned}$$

where $w \equiv exp(z)$.

The mode of the distribution is at $f(0) = 1/e$, so for an exact observation the deviance term has $\hat{y} = y$. For interval censored data where the interval is of length

$b = z^u - z^l$, it turns out that we cover the most mass if the interval has a lower endpoint of $a = \log(b/(\exp(b) - 1)))$, and the resulting log-likelihood is

$$\log(e^{-e^a} - e^{-e^{a+b}}).$$

Proving this is left as an exercise for the reader.

5.9.3 Logistic

This distribution is very close to the Gaussian except in the extreme tails, but it is easier to compute. However, some data sets may contain survival times close to zero, leading to differences in fit between the lognormal and log-logistic choices. (In such cases the rationality of a Gaussian fit may also be in question). Again let $w = \exp(z)$.

$$\mu = 0, \ \sigma^2 = \pi^2/3$$

$$\begin{aligned} F(z) &= w/(1+w) \\ f(z) &= w/(1+w)^2 \\ f'(z) &= f(z)\,(1-w)/(1+w) \\ f''(z) &= f(z)\,(w^2 - 4w + 1)/(1+w)^2 \end{aligned}$$

The distribution is symmetric about 0, so for an exact observation the contribution to the deviance term is $-\log(4)$. For an interval censored observation with span $2b$ the contribution is

$$\log\left(F(b) - F(-b)\right) = \log\left(\frac{e^b - 1}{e^b + 1}\right).$$

5.9.4 Other distributions

Some other population hazards can be fit into this location-scale framework, others do not.

Distribution	Hazard
Weibull	$p\lambda(\lambda t)^{p-1}$
Extreme value	$(1/\sigma)e^{(t-\eta)/\sigma}$
Rayleigh	$a + bt$
Gompertz	bc^t
Makeham	$a + bc^t$

We can see that an extreme value distribution on $t' = \log(t)$ is equivalent to a Weibull hazard on t, with $p - 1 = 1/\sigma$.

The Makeham hazard seems to fit human mortality experience beyond infancy quite well, where a is a constant mortality which is independent of the health of the subject (accidents, homicide, etc) and the second term models the Gompertz assumption that "the average exhaustion of a man's power to avoid death is such that at the end of equal infinitely small itervals of time he has lost equal portions of his remaining power to oppose destruction which he had at the commencement of these intervals". For older ages a is a neglible portion of the death rate and the Gompertz model holds.

Clearly

- The Wiebull distribution with $p = 2$ ($\sigma = .5$) is the same as a Rayleigh distribution with $a = 0$. It is not, however, the most general form of a Rayleigh.
- The extreme value and Gompertz distributions have the same hazard function, with $\sigma = 1/\log(c)$, and $\exp(-\eta/\sigma) = b$.

It would appear that the Gompertz can be fit with an identity link function combined with the extreme value distribution. However, this ignores a boundary restriction. If $f(x; \eta, \sigma)$ is the extreme value distribution with paramters η and σ, then the definition of the Gompertz densitiy is

$$\begin{array}{ll} g(x; \eta, \sigma) = 0 & x < 0 \\ g(x; \eta, \sigma) = cf(x; \eta, \sigma) & x >= 0 \end{array}$$

where $c = \exp(\exp(-\eta/\sigma))$ is the necessary constant so that g integrates to 1. If η/σ is far from 1, then the correction term will be minimal and `survreg` should give a reasonable fit to Gompertz data. If not, the distribution can not be made to easily conform to the general fitting scheme of the program.

The Makeham distribution falls into the gamma family (equation 2.3 of Kalbfleisch and Prentice, Survival Analysis), but with the same range restriction problem.

6 Side Effects

The basic problem is that it is inefficient to return *everything* one might ever need from a model fit. Depending on the routine, the returned result might be larger than the original data. In a Cox model, for instance, the standard errors for a predicted survival curve depend on the difference between the chosen values of the covariates and the weighted means of the covariates at each time point, i.e., the mean at time t over all of the subjects still at risk at t weighted by their fitted risk score. This is essentially the same size as the X matrix.

The routines for a Kaplan-Meier survival, differences in survival, and for expected survival return all necessary information, and have no side effects.

6.1 Cox model

Several of the downstream results require the y or survival data of the Cox model (most often just the status variable). Others require the right hand side variables and/or the strata information as well. By default, the coxph function saves y in its result, but not the X matrix. Optionally, one can specify that coxph save the X matrix (including the strata variables), or even the entire model frame. The predict, residuals, and survfit methods only reconstruct what they have to, so the side effects depend on what is saved.

If either X or y must be recreated, the program is forced to recreate both. This is due to missing values — if only y were recreated, it might include extra observations that were omitted in the original fit due to missing values in X. BEWARE, if X is recreated and the current settings for either the `na.action` or `contrasts` options are different than they were when the model first was run, then the new matrix may be different than the old. Resulting answers will be nonsense, *and there will be no warning.* The same is true if the original fit did not include a `data=` argument and the list of attached data frames is not identical to the original environment, though in this case there is some possibility of an error message due to unmatched variable names.

The predict function has four types: linear predictor, risk, expected number of events, and terms. By default it does not compute the standard errors of the predictions. A data frame may be included to get predictions for a new subject. The following objects may need to be computed:

- Without a new data frame
  - type='terms': X
  - type='lp' with standard errors: X
  - type='risk' with standard errors: X
  - type='expected': y
- With a new data frame
  - type='expected': y

The residuals function has four types. Martingale residuals have no side effects, deviance residuals require y, and score and Schoenfeld residuals require both y and X.

Fitted survival curves for a Cox model always require y. If standard errors are desired (the default) or if there were strata in the fit then X is required as well.

6.2 Parametric survival models

The matrix type of residual requires y; if it is not present the routine fails with an error message. At present this is the only side effect, though if and when a survfit method is created for the parametric models there may be more.

7 Missing values

The handling of missing values has always been a flaw in S, a particularly odious one to those who, like me, deal with medical data. (The original survival functions broke from the S mold by automatically deleting missing values from the input data set.) The latest version of S has begun to address this issue via the `na.action` function. Though this is an admirable start, I feel that more needs to be done to get a "complete" solution.

For the survival routines, missing values impact in 4 areas –

- Global choice of a default missing value strategy.
- Possible data reduction in passing from the data frame to the model matrix.
- A report of the effects (if any) of the action, included in the standard printout for the model.
- Possible data expansion or other modification in computing the residuals or predicted values.

By the last of these I mean that the following S code should work independent of the current na.action.

```
> fit <- coxph( Surv(tt, ss) ~ x1 + x2)
> plot(x1, resid(fit))
```

That is, the residual from a model is the same shape as the *input* data, independent of reductions that may have occurred in the intervening X matrix. Other actions in this last step are possible. For instance, if the na.action had imputed numbers for the missing values, the predicted value for such observations should perhaps be set to NA or otherwise marked as unreliable.

Of the four effects, the second is dealt with by the `na.action` extension to base S. The first, the ability to choose a global action, is accomplished with a change to `model.frame.default`. If neither the particular function call nor the data frame contained an explicit na.action, the routine now checks for an na.action option. To make na.omit the global default, type

```
> options(na.action='na.omit')
```

Because `model.frame.default` is called by all of the S modeling functions, this action will apply to all of them, not just survival.

In order to implement the third and fourth effects, it was necessary to pass na.action information as part of the returned model fit. This in turn requires the na.action routine to store something in the model frame that is passed to the fitting function. Because the definition of what "should be done" to label the printout or to modify the residuals depends on the na.action chosen, the fitting, print, residual, and predict routines do not try to manipulate the passed information directly, but use it as an argument to `naprint` and `naresid` methods. This allows the actions taken to be extremely general, and more importantly, they are easily modified by the user.

Specifically

1. The na.action routine optionally adds an attribute named *na.action* to the data frame. The class of the attribute object should be the name of the na action, but the contents or form of the object is unrestricted and depends on the particular na.action.

2. The fitting routine adds this attribute to its output, as a component named *na.action*.

3. The print routines for coxph, survfit, and etc. each contain a call to the naprint method, with the passed na.action object as its argument. This method is assumed to return a character string appropriate for inclusion in the printout.

4. The residual and predicted value routines each pass their final result through a call to the naresid method, whose arguments are the na.action object and the vector or matrix of results.

The package includes a modified version of `na.omit` along with the methods `naprint.omit` and `naresid.omit`.

The default naprint method returns a null string, and the default naresid method returns its second argument, this mimics the unmodified S. This allows the survival routines to work with "standard" S na.action functions. The standard modeling functions such as `lm` ignore any attributes on the data frame, and work "as usual" with the modified na.omit function.

References

[1] Andersen, P.K. and Gill, R.D. (1982). Cox's regression model for counting processes: A large sample study. *Ann. Stat.* **10**, 1100-20.

[2] Andersen, P.K., Borgan, O., Gill, R.D., and Keiding, N. (1991) *Statistical Models Based on Counting Processes.* Springer-Verlag, New York.

[3] Andersen, P.K., Borch-Johnsen, K., Deckert, T., Green, A., Hougaard, P., Keiding, N., and Kreiner, S. (1985). A Cox regression model for the relative mortality and its application to diabetes mellitus survival data. *Biometrics* **41**, 921-32.

[4] Andersen, P.K. and Væth, M. (1989). Simple parametric and non-parametric models for excess and relative mortality. *Biometrics* **45**, 523-35.

[5] Barlow, W.E. and Prentice, R.L. (1988). Residuals for relative risk regression. *Biometrika* **75**, 65-74.

[6] Berry, G. (1983) The analysis of mortality by the subject years method. *Biometrics* **39**, 173-84.

[7] Binder, D.A. (1992). Fitting Cox's proportional hazards models from survey data. *Biometrika* **79**, 139-47.

[8] Cain, K.C. and Lange, N.T. (1984). Approximate case influence for the proportional hazards regression model with censored data. *Biometrics* **40**, 493-9.

[9] Chambers, J.M. and Hastie, T.J. (1992). *Statistical Models in S.* Wadsworth and Brooks/Cole, California.

[10] Chen, C. and Wang, P.C. (1991). Diagnostic plots in Cox's regression model. *Biometrics* **47**, 841-50.

[11] Crowley, J. and Hu, M. (1977), Covariance analysis of heart transplant data. *J. Am. Stat. Assoc.* **72**, 27-36.

[12] Dorey, F.J. and Korn, E.L. (1987). Effective sample sizes for confidence intervals for survival probabilities. *Statistics in Medicine* **6**, 679-87.

[13] Ederer, F., Axtell, L.M. and Cutler, S.J. (1961). The relative survival rate: A statistical methodology. National Cancer Institute Mongraphs **6**, 101-21.

[14] Ederer, F. and Heise, H. (1977). Instructions to IBM 650 programmers in processing survival computations, *Methodological Note No. 10, End Results Evaluation Section*, National Cancer Institute.

[15] Edmunson, J.H., Fleming, T.R., Decker, D.G., Malkasian, G.D., Jefferies, J.A., Webb, M.J., and Kvols, L.K. (1979), Different Chemotherapeutic Sensitivities and Host Factors Affecting Prognosis in Advanced Ovarian Carcinoma vs. Minimal Residual Disease. *Cancer Treatment Reports* **63**, 241-47.

[16] Efron, B. (19xx) *The Jackknife, the bootstrap, and other resampling plans.* SIAM CMBS-NSF Monograph **38**.

[17] Escobar, L.A. and Meeker W.Q., Jr. (1992). Assessing influence in regression analysis with censored data. *Biometrics* **48**, 507-528.

[18] Fleming, T.R. and Harrington, D.P. (1981), A class of hypothesis tests for one and two sample censored survival data. *Communications in Statistics* **A10(8)**, 763-94.

[19] Fleming, T.R. and Harrington, D.P. (1984), Nonparametric estimation of the survival distribution in censored data. *Comm. in Statistics A* **13**, 2469-86.

[20] Fleming, T.R. and Harrington, D.P. (1991), *Counting Processes and Survival Analysis*, Wiley, New York.

[21] Gail, M.H. and Byar, D.P. (1986). Variance calculations for direct adjusted survival curves, with applications to testing for no treatment effect. *Biom J* **28**, 587-99.

[22] Grambsch, P.M., Dickson, E.R., Wiesner, R.H. and Langworthy, A. (1989). Application of the Mayo PBC survival model to liver transplant patients. *Mayo Clinic Proc* **64**, 699-704.

[23] Grambsch, P. and Therneau, T.M. (1994). Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, **81**, 515-26.

[24] Green P.J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives (with discussion). *JRSSB* **46**, 149-92

[25] Hakulinen, T. (1982). Cancer survival corrected for heterogeneity in patient withdrawal. *Biometrics* **38**, 933.

[26] Hakulinen, T. and Abeywickrama, K.H. (1985). A computer program package for relative survival analysis. *Computer Programs in Biomedicine* **19**, 197-207.

[27] Hakulinen, T. (1977). On long term relative survival rates. *J. Chronic Diseases* **30**, 431-43.

[28] Harrington, D.P. and Fleming, T.R. (1982). A class of rank test proceedures for censored survival data. *Biometrika*, **69**, 553-66.

[29] Hartz, A.J., Giefer, E.E. and Hoffmann, G.G. (1983). A comparison of two method for calculating expected mortality. *Statistics in Medicine* **2**, 381-6.

[30] Hartz, A.J., Giefer, E.E. and Hoffmann, G.G. (1984). Letter and rejoinder on "A comparison of two method for calculating expected mortality." *Statistics in Medicine* **3**, 301-2.

[31] Hartz, A.J., Giefer, E.E. and Hoffmann, G.G. (1985). Letters and rejoinder on "A comparison of two method for calculating expected mortality." *Statistics in Medicine* **4**, 105-9.

[32] Huber, P.J. (1967). The behavior of maximum likelihood estimates under non-standard conditions. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **1**, 221-33.

[33] Kalbfleisch, J.D. and Prentice R.L. (1980). *The Statistical Analysis of Failure Time Data*, Wiley, New York.

[34] Klein, J.P. (1991). Small sample moments of some estimators of the variance of the Kaplan-Meier and Nelson-Aalen estimators. *Scand. J. Statistics* **18**, 333-40.

[35] Lee, E.W., Wei, L.J. and Amato D. (1992). *Cox-type regression analysis for large number of small groups of correlated failure time observations.* In Klein, J.P and Goel, P.K. (eds), Survival Analysis, State of the Art, 237-247, Kluwer Academic Publishers, Netherlands.

[36] Lin, D.Y. (1991). Goodness-of-fit analysis for the Cox regression model based on a class of parameter estimators. *JASA* **86**, 725-728.

[37] Lin, D.Y. and Wei, L.J. (1989). The robust inference for the Cox Proportional Hazards Model. *J. Am. Stat. Assoc.* **84**, 1074-79.

[38] Lin, D.Y., Wei, L.J. and Ying, Z. (1992). Checking the Cox model with cumulative sums of martingale-based residuals. *Technical Report #111*, Dept. of Biostatistics, U. of Washington.

[39] Link, C.L. (1984). Confidence intervals for the survival function using Cox's proportional hazard model with covariates. *Biometrics* **40**, 601-10.

[40] Link, C.L. (1986). Confidence intervals for the survival function in the presence of covariates. *Biometrics* **42**, 219-20.

[41] Makuch, R.W. (1982). Adjusted survival curve estimation using covariates. *J Chronic Diseases* **3**, 437-43.

[42] Mantel, N. (1966). Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports* **50**, 163-6.

[43] Miller, R.G. (1981), *Survival Analysis*, Wiley, New York.

[44] McCullagh, P. and Nelder, J.A. (1983). *Generalized Linear Models.* Chapman and Hall.

[45] Nagelkirke, N. (1991). A note on a general definition of the coefficient of determination. *Biometrika* **78**, 691-2.

[46] Nelson, W.B. (1969). Hazard plotting for incomplete failure data. *J. Quality Technology* **1**, 27-52.

[47] Prentice, R.L., Williams, B.J. and Peterson, A.V. (1981). On the regression analysis of multivariate failure time data. *Biometrika* **68**, 373-89.

[48] Pugh, M., Robins, J., Lipsitz, S., and Harrington, D. (1992). Inference in the Cox proportional hazards model with missing covariates. *Technical Report 758Z*, Department of Biostatistics, Harvard School of Public Health.

[49] Reid, N. and Crépeau, H. (1985). Influence functions for proportional hazards regression. *Biometrika* **72**: 1-9.

[50] Schoenfeld, D. (1980). Chi-squared goodness-of-fit tests for the proportional hazards regression model. *Biometrika* **67**, 145-53.

[51] Smith, P.J. and Hietjan, D.F. (to appear). Testing and adjusting for overdispersion in generalized linear models. *J Royal Statistical Soc, Series C.*

[52] Turnbull, B.W. (1974) Nonparametric estimation of a survivorship function with doubly censored data. *JASA* **69**, 169-73.

[53] Therneau, T.M., Grambsch P.M. and Fleming, T.R. (1990). Martingale based residuals for survival models. *Biometrika* **77**, 147-60.

[54] Thomsen, T.L., Keiding, N. and Altman, D.G. (1991), A note on the calculation of expected survival, illustrated by the survival of liver transplant patients. *Statistics in Medicine* **10**, 733-8.

[55] Tsiatis, A.A. (1981). A large sample study of Cox's regression model. *Ann. Statistics* **9**, 93-108.

[56] Verheul, H.A., Dekker, E., Bossuyt, P., Moulijn, A.C. and Dunning A.J. (1993). Backround mortality in clinical survival studies, *Lancet* **341**, 872-5.

[57] Wei, L.J., Lin, D.Y. and Weissfeld, L. (1989). Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. *J. Am. Stat. Assoc.* **84**, 1065-73.

[58] White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica* **48**, 817-30.

[59] White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrika* **50** 1-25.

[60] *The Health Benefits of Smoking Cessation* (1990). US Department of Health and Human Services. Public Health Service, Centers for Disease Control, Center for Chronic Disease Prevention and Health Promotion, Office on Smoking and Health. DHHS Publication No (CDC)90-8416.