The survival package

by Thomas Lumley

This is another in our series of articles describing the recommended packages that come with the R distribution. The survival package is a port of code written by T. M. Therneau, of the Mayo Clinic, code that has also been part of S-PLUS for many years. Porting the survival package in 1997 was my introduction to R programming. It exposed quite a few bugs in R and incompatibilities with S (some of which, on reflection, might have been better left incompatible).

Overview

Survival analysis, also called event history analysis in the social sciences and reliability analysis in engineering, is concerned with the time until an event occurs. The complication that makes survival analysis interesting is that individuals are often observed for only part of the relevant time period. They may be recruited well after time zero, and the study may end with only a small proportion of events having been seen.

In medical statistics the most common survival analyses are estimation of the distribution of time to the event, testing for equality of two distributions, and regression modelling of the rate at which events occur. These are all covered by the survival package.

Additional tools for regression modelling are in the Design and eha packages, and the muhaz package allows estimation of the hazard function (the analogue in survival analysis of density estimation).

Specifying survival data

In the popular 'counting process' formulation of survival analysis each record in a dataset has three variables describing the survival time. A `start` variable specifies when observation begins, a `stop` variable specifies when it ends, and an `event` variable is 1 if observation ended with an event and 0 if it ended without seeing the event. These variables are bundled into a `"Surv"` object by the `Surv()` function. If the `start` variable is zero for everyone it may be omitted.

In `data(veteran)`, `time` measures the time from the start of a lung cancer clinical trial and `status` is 1 if the patient died at `time`, 0 if follow-up ended with the patient still alive. In addition, `diagtime` gives the time from initial diagnosis until entry into the clinical trial.

```
> library(survival)
> data(veteran)
## time from randomisation to death
> with(veteran, Surv(time,status))
  [1]  72  411  228  126  118   10   82
  [8] 110  314  100+  42    8  144   25+
  ...

## time from diagnosis to death
> with(veteran, Surv(diagtime*30,
                     diagtime*30+time,
                     status))
  [1] ( 210, 282 ] ( 150, 561 ]
  [3] (  90, 318 ] ( 270, 396 ]
  ...
  [9] ( 540, 854 ] ( 180, 280+]
  ...
```

The first `"Surv"` object prints the observation time, with a + to indicate that the event is still to come. The second object prints the start and end of observation, again using a + to indicate end of follow-up before an event.

One and two sample summaries

The `survfit()` function estimates the survival distribution for a single group or multiple groups. It produces a `"survfit"` object with a `plot()` method. Figure 1 illustrates some of the options for producing more attractive graphs.

Two (or more) survival distributions can be compared with `survdiff`, which implements the logrank test (and the G^ρ family of weighted logrank tests). In this example it is clear from the graphs and the tests that the new treatment (`trt=2`) and the standard treatment (`trt=1`) were equally ineffective.

```
> data(veteran)
> plot(survfit(Surv(time,status)~trt,data=veteran),
      xlab="Years since randomisation",
      xscale=365, ylab="% surviving", yscale=100,
      col=c("forestgreen","blue"))

> survdiff(Surv(time,status)~trt, data=veteran)

Call:
survdiff(formula = Surv(time, status) ~ trt,
         data = veteran)

       N Observed Expected (O-E)^2/E
trt=1 69       64     64.5   0.00388
trt=2 68       64     63.5   0.00394
      (O-E)^2/V
trt=1   0.00823
trt=2   0.00823

 Chisq= 0  on 1 degrees of freedom, p= 0.928
```

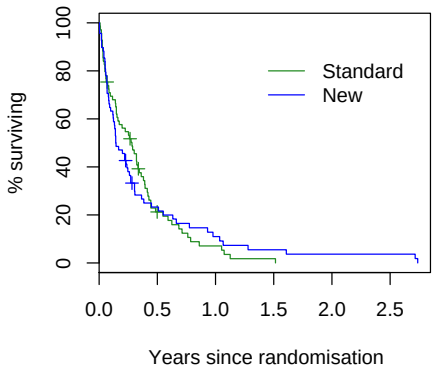

Figure 1: Survival distributions for two lung cancer treatments

Proportional hazards models

The mainstay of survival analysis in the medical world is the Cox proportional hazards model and its extensions. This expresses the hazard (or rate) of events as an unspecified baseline hazard function multiplied by a function of the predictor variables.

Writing $h(t;z)$ for the hazard at time t with predictor variables $Z = z$ the Cox model specifies

$$\log h(t,z) = \log h_0(t) e^{\beta z}.$$

Somewhat unusually for a semiparametric model, there is very little loss of efficiency by leaving $h_0(t)$ unspecified, and computation is, if anything, easier than for parametric models.

A standard example of the Cox model is one constructed at the Mayo Clinic to predict survival in patients with primary biliary cirrhosis, a rare liver disease. This disease is now treated by liver transplantation, but at the same there was no effective treatment. The model is based on data from 312 patients in a randomised trial.

```
> data(pbc)
> mayomodel<-coxph(Surv(time,status)~edtrt+
                       log(bili)+log(protime)+
                       age+platelet,
                 data=pbc, subset=trt>0)
> mayomodel
Call:
coxph(formula = Surv(time, status) ~ edtrt +
          log(bili) + log(protime) +
           age + platelet, data = pbc,
           subset = trt > 0)


                coef exp(coef)
edtrt         1.02980     2.800
log(bili)     0.95100     2.588
log(protime)  2.88544    17.911
age           0.03544     1.036
platelet     -0.00128     0.999
              se(coef)     z       p
edtrt         0.300321  3.43 0.00061
log(bili)     0.097771  9.73 0.00000
log(protime) 1.031908  2.80 0.00520
age           0.008489  4.18 0.00003
platelet      0.000927 -1.38 0.17000

Likelihood ratio test=185  on 5 df, p=0  n= 312
```

The `survexp` function can be used to compare predictions from a proportional hazards model to actual survival. Here the comparison is for 106 patients who did not participate in the randomised trial. They are divided into two groups based on whether they had edema (fluid accumulation in tissues), an important risk factor.

```
> plot(survfit(Surv(time,status)~edtrt,
       data=pbc,subset=trt==-9))
> lines(survexp(~edtrt+
              ratetable(edtrt=edtrt,bili=bili,
                    platelet=platelet,age=age,
                    protime=protime),
              data=pbc,
              subset=trt==-9,
              ratetable=mayomodel,
              cohort=TRUE),
        col="purple")
```

The `ratetable` function in the model formula wraps the variables that are used to match the new sample to the old model.

Figure 2 shows the comparison of predicted survival (purple) and observed survival (black) in these 106 patients. The fit is quite good, especially as people who do and do not participate in a clinical trial are often quite different in many ways.

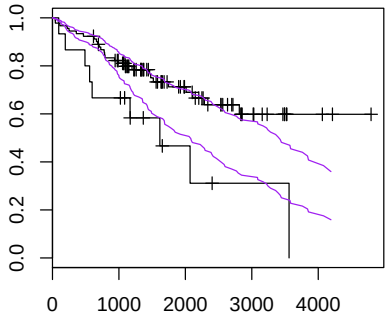

Figure 2: Observed and predicted survival

The main assumption of the proportional hazards model is that hazards for different groups are in fact proportional, *i.e.* that β is constant over time. The

cox.zph function provides formal tests and graphical diagnostics for proportional hazards

```
> cox.zph(mayomodel)
                rho chisq      p
edtrt        -0.1602 3.411 0.0648
log(bili)     0.1507 2.696 0.1006
log(protime) -0.1646 2.710 0.0997
age          -0.0708 0.542 0.4617
platelet     -0.0435 0.243 0.6221
GLOBAL           NA 9.850 0.0796

## graph for variable 1 (edtrt)
> plot(cox.zph(mayomodel)[1])
```

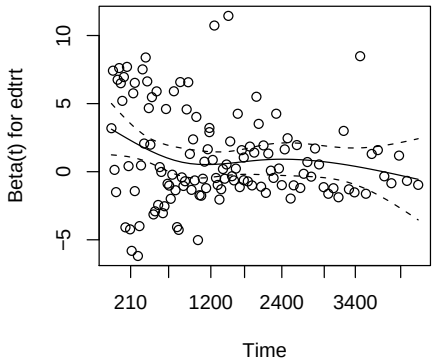

Figure 3: Testing proportional hazards

There is a suggestion that edtrt and log(protime) may have non-proportional hazards, and Figure 3 confirms this for edtrt. The curve shows an estimate of β at each point in time, obtained by smoothing the residuals on the graph. It appears that edtrt is quite strongly predictive for the first couple of years but has little predictive power after that.

Additional features

Computation for the Cox model extends straightforwardly to situations where z can vary over time, and when an individual can experience multiple events of the same or different types. Interpretation of the parameters becomes substantially trickier, of course. The coxph() function allows formulas to contain a cluster() term indicating which records belong to the same individual and a strata() term indicating subgroups that get their own unspecified baseline hazard function.

Complementing coxph is survreg, which fits linear models for the mean of survival time or its logarithm with various parametric error distributions. The parametric models allow more flexible forms of censoring than does the Cox model.

More recent additions include penalised likelihood estimation for smoothing splines and for random effects models.

The survival package also comes with standard mortality tables for the US and a few individual states, together with functions for comparing the survival of a sample to that of a population and computing person-years of followup.

Thomas Lumley
Department of Biostatistics
University of Washington, Seattle

useR! 2004

The R User Conference

by John Fox
McMaster University, Canada

More than 200 R users and developers converged on the Technische Universität Wien for the first R users' conference — userR! 2004 — which took place in Vienna between May 20 and May 22. The conference was organized by the Austrian Association for Statistical Computing (AASC) and sponsored by the R Foundation for Statistical Computing, the Austrian Science Foundation (FWF), and MedAnalytics (http://www.medanalytics.com/). Torsten Hothorn, Achim Zeileis, and David Meyer served as chairs of the conference, program committee, and local organizing committee; Bettina Grün was in charge of local arrangements.

The conference program included nearly 100 presentations, many in keynote, plenary, and semi-plenary sessions. The diversity of presentations — from bioinformatics to user interfaces, and finance to fights among lizards — reflects the wide range of applications supported by R.

A particularly interesting component of the program were keynote addresses by members of the R core team, aimed primarily at improving programming skills, and describing key features of R along with new and imminent developments. These talks reflected a major theme of the conference — the close relationship in R between use and development. The keynote addresses included:

- Paul Murrell on grid graphics and programming
- Martin Mächler on good programming practice