

22S:166 Computing in Statistics

R packages

Lecture 7
September 17, 2007

Kate Cowles
374 SH, 335-0727
kcowles@stat.uiowa.edu

R packages

- *packages* contain all R functions and datasets
- contents of package become available in R when package is *loaded*
- `search()` lists displays packages currently loaded

```
> search()
[1] ".GlobalEnv"      "package:methods"  "package:stats"
[4] "package:graphics" "package:grDevices" "package:utils"
[7] "package:datasets" "Autoloads"        "package:base"
```

- `library()` displays names of packages installed (but not necessarily loaded) at site

```
> library()
Packages in library '/group/statsoft/Rlibs64':

abind          Combine multi-dimensional arrays
AnalyzefMRI    Functions for analysis of fMRI datasets stored
               in the ANALYZE format.
boa            Bayesian Output Analysis Program (BOA) for
               MCMC
car            Companion to Applied Regression
cat            Analysis of categorical-variable datasets with
               missing values
coda           Output analysis and diagnostics for MCMC
```

```
dopt          ** No title available (pre-2.0.0 install?) **
e1071         Misc Functions of the Department of Statistics
.
.
.
SuppDists    Supplementary distributions
survrec      Survival analysis for recurrent event data
tkrplot      TK Rplot
tree         Classification and regression trees
tripack      Triangulation of irregularly spaced data
tseries      Time series analysis and computational finance
xtable       Export tables to LaTeX or HTML
```

Packages in library '/usr/lib64/R/library':

```
base          The R Base Package
boot          Bootstrap R (S-Plus) Functions (Canty)
class         Functions for Classification
cluster       Cluster Analysis Extended Rousseeuw et al.
datasets      The R Datasets Package
foreign       Read Data Stored by Minitab, S, SAS, SPSS,
.
.
.
splines       Regression Spline Functions and Classes
stats         The R Stats Package
stats4        Statistical Functions using S4 Classes
survival      Survival analysis, including penalised
               likelihood.
tcltk         Tcl/Tk Interface
tools         Tools for Package Development
utils         The R Utils Package
Stata, Systat, dBase, ...
```

- `library(<package name>)` loads package

```
> library(survival)
Loading required package: splines
> search()
[1] ".GlobalEnv"      "package:survival"  "package:splines"
[4] "package:methods" "package:stats"    "package:graphics"
[7] "package:grDevices" "package:utils"    "package:datasets"
[10] "Autoloads"        "package:base"
```

- standard or base packages are considered part of R; available automatically

Contributed R packages

- hundreds of contributed packages written by authors all over the world
 - e.g. Luke Tierney and Jun Yan from UI stats dept!
- *recommended* packages come in binary distribution of R
- most available from Comprehensive R Archive Network (CRAN)

www.cran.r-project.org

Using R packages

```
> help(package="survival")

Information on package 'survival'

Description:

Title: Survival analysis, including penalised likelihood.
Maintainer: Thomas Lumley <tlumley@u.washington.edu>
Priority: recommended
Package: survival
Version: 2.26
Depends: stats, utils, graphics, splines, R (>= 2.0.0)
LazyData: Yes
LazyLoad: Yes
Author: S original by Terry Therneau, ported by Thomas Lumley
Description: survival analysis: descriptive statistics, two-sample
tests, parametric accelerated failure models, Cox model.
Delayed entry (truncation) allowed for all models;
interval censoring for parametric models. Case-cohort
designs.
License: GPL2
Packaged: Fri May 19 11:27:43 2006; tlumley
Built: R 2.3.1; x86_64-redhat-linux-gnu; 2006-06-23 12:53:09;
unix

Index:

Surv Package a survival variable
clogit Conditional logistic regression
cluster Identify clusters.
cox.zph Test the proportional hazards assumption of a Cox regressio
```

```
coxph Proportional Hazards Regression
coxph.detail Details of a cox model fit
coxph.object Proportional Hazards Regression Object
coxph.rvar Robust variance for a Cox model
frailty Frailty models by penalised likelihood
is.ratetable Verify that an object is of class ratetable.
lines.survfit Add lines to a survival plot
plot.cox.zph Graphical test of proportional hazards (requires splines)
plot.survfit Plot method for survfit.
print.survfit Short summary of a survival curve
pspline Penalised (smoothing) splines
pyears Person Years
residuals.coxph Calculate residuals for a coxph fit.
residuals.survreg Compute Residuals for survreg Objects
ridge ridge regression
strata Identify strata variables.
summary.survfit Print a Survival Curve
survdiff Test Survival Curve Differences
survexp Compute Expected Survival
survexp.fit Compute expected survival
survexp.usr Mortality tables for US and some states (requires date)
survfit Compute a survival Curve for Censored Data
survfit.object Survival Curve Object
survobrien O'Brien's test for association of a variable with survival
survreg Regression for a parametric survival model
survreg.object Parametric Survival Model Object
survSplit Split data for creating time-dependent covariates.
anova.survreg ANOVA tables for survreg objects
untangle.specials Help process the 'specials' argument of the terms functio
```

```
----- date functions -----
as.date Coerce Data to Dates
date.ddmmyy Format a Julian date
date.mdy Convert from Julian Dates to Month, Day, and
```

```
Year
date.mmdyy Format a Julian date
date.mmdyyy Format a Julian date
date.object Date Objects
mdy.date Convert to Julian Dates
```

(END)

Enter

q()

to exit the "help"

Using functions in R packages

```
> help(survreg, package="survival")
```

```
survreg          package:survival          R Documentation
```

```
Regression for a Parametric Survival Model
```

Description:

Regression for a parametric survival model. These are all time-transformed location models, with the most useful case being the accelerated failure models that use a log transformation.

Usage:

```
survreg(formula=formula(data), data=parent.frame(), weights,
subset,na.action,dist="weibull", init=NULL, scale=0,
control=survreg.control(),parms=NULL,model=FALSE, x=FALSE,
y=TRUE, robust=FALSE, ...)
```

Arguments:

formula: a formula expression as for other regression models. See the documentation for 'lm' and 'formula' for details.

data: optional data frame in which to interpret the variables

weights: Optional observation weights

subset: subset of the observations to be used in the fit.

na.action: function to be used to handle any NAs in the data.

11

an object of class 'survreg' is returned.

Compatibility note:

This routine underwent significant changes from survival4 to survival5. The `survreg.old` function gives a backwards-compatible interface. In S-PLUS the new function is called 'survReg' and the old one 'survreg'.

See Also:

'survreg.object', 'survreg.distributions', 'pspline', 'frailty', 'ridge', 'survreg.old'

Examples:

```
## These are all the same
survreg(Surv(futime, fustat) ~ ecog.ps + rx, ovarian, dist='weibull',s
survreg(Surv(futime, fustat) ~ ecog.ps + rx, ovarian,
dist="exponential")
survreg.old(Surv(futime, fustat) ~ ecog.ps + rx, ovarian, dist='extreme
```

(END)

10

dist: assumed distribution for y variable. If the argument is a character string, then it is assumed to name an element from 'survreg.distributions'. These include 'weibull', 'exponential', 'gaussian', 'logistic', 'lognormal' and 'loglogistic'. Otherwise, it is assumed to be a user defined list conforming to the format described in 'survreg.distributions'.

parms: a list of fixed parameters. For the t-distribution for instance this is the degrees of freedom; most of the distributions have no parameters.

init: optional vector of initial values for the parameters.

scale: optional fixed value for the scale. If set to <=0 then the scale is estimated.

control: a list of control values, in the format produced by 'survreg.control'.

model: if TRUE, the model frame is returned.

x: if TRUE, then the X matrix is returned.

y: if TRUE, then the y vector (or survival times) is returned.

robust: if TRUE, sandwich standard errors are computed. Defaults to TRUE when 'formula' contains a 'cluster' term.

...: other arguments which will be passed to 'survreg.control'.

Value:

12

Using functions in R packages, continued

```
> leuk <- read.table("leuk.dat")
> colnames(leuk) <- c("survtime", "cenflag", "trt")
> leuk
```

	survtime	cenflag	trt
1	1	1	0.5
2	1	1	0.5
3	2	1	0.5
4	2	1	0.5
5	3	1	0.5
6	4	1	0.5
7	4	1	0.5
8	5	1	0.5
9	5	1	0.5
10	8	1	0.5
11	8	1	0.5
12	8	1	0.5
13	8	1	0.5
14	11	1	0.5
15	11	1	0.5
16	12	1	0.5
17	12	1	0.5
18	15	1	0.5
19	17	1	0.5
20	22	1	0.5
21	23	1	0.5
22	6	1	-0.5
23	6	1	-0.5
24	6	1	-0.5

```

25     6     0 -0.5
26     7     1 -0.5
27     9     0 -0.5
28    10     1 -0.5
29    10     0 -0.5
30    11     0 -0.5
31    13     1 -0.5
32    16     1 -0.5
33    17     0 -0.5
34    19     0 -0.5
35    20     0 -0.5
36    22     1 -0.5
37    23     1 -0.5
38    25     0 -0.5
39    32     0 -0.5
40    32     0 -0.5
41    34     0 -0.5
42    35     0 -0.5
> exponanalysis <- survreg( Surv( survtime, cenflag) ~ trt,
+ data = leuk, dist="exponential")
> summary(exponanalysis)

```

```

Call:
survreg(formula = Surv(survtime, cenflag) ~ trt, data = leuk,
        dist = "exponential")
              Value Std. Error      z      p
(Intercept)  2.92      0.199 14.67 9.70e-49
trt          -1.53      0.398 -3.83 1.27e-04

```

Scale fixed at 1

```

Exponential distribution
Loglik(model)= -108.5  Loglik(intercept only)= -116.8
Chisq= 16.49 on 1 degrees of freedom, p= 4.9e-05

```

```

Number of Newton-Raphson Iterations: 4
n= 42

> weibullanalysis <- survreg( Surv( survtime, cenflag) ~ trt,
+ data = leuk, dist="weibull")

> summary(weibullanalysis)

Call:
survreg(formula = Surv(survtime, cenflag) ~ trt, data = leuk,
        dist = "weibull")
              Value Std. Error      z      p
(Intercept)  2.882      0.146 19.73 1.29e-86
trt          -1.267      0.311 -4.08 4.51e-05
Log(scale)  -0.312      0.147 -2.12 3.43e-02

Scale= 0.732

Weibull distribution
Loglik(model)= -106.6  Loglik(intercept only)= -116.4
Chisq= 19.65 on 1 degrees of freedom, p= 9.3e-06
Number of Newton-Raphson Iterations: 5
n= 42

```