

Package ‘RcmdrPlugin.survival’

May 19, 2012

Type Package

Title R Commander Plug-in for the survival Package

Version 1.0-1

Date 2011-08-03

Author John Fox

Maintainer John Fox <jfox@mcmaster.ca>

Depends R (>= 2.10), Rcmdr (>= 1.7.0), tcltk, survival, date, stats

Description This package provides an R Commander plug-in for the survival package, with dialogs for Cox models, parametric survival regression models, estimation of survival curves, and testing for differences in survival curves, along with data-management facilities and a variety of tests, diagnostics and graphs.

License GPL (>= 2)

LazyLoad yes

LazyData yes

Models coxph, survreg, coxph.penal

Repository CRAN

Repository/R-Forge/Project rcmdrsurvplugin

Repository/R-Forge/Revision 58

Date/Publication 2012-05-19 14:49:35

R topics documented:

RcmdrPlugin.survival-package	2
anova.coxph	3
Dialysis	3
mfrow	4
plot.coxph	5
quantile.survfit	6
Rossi	7
SurvivalData	10
unfold	10
Unfold-dialog	12
Index	13

RcmdrPlugin.survival-package
Rcmdr Plug-In Package for the survival Package

Description

This package provides an R Commander plug-in for the survival package, with dialogs for managing survival data (this to a limited extent), Cox models, parametric survival regression models, estimation of survival curves, testing for differences in survival curves, and a variety of diagnostics, tests, and displays.

Details

Package:	RcmdrPlugin.survival
Type:	Package
Version:	1.0-1
Date:	2011-08-03
License:	GPL (>= 2)
LazyLoad:	yes

The plug-in is tightly integrated with the *R Commander* interface; see the following menus: **Data -> Survival data**", **Statistics -> Survival analysis**, **Statistics -> Fit Models**, **Models -> Hypothesis tests**, **Models -> Numerical diagnostics**, **Models -> Graphs**.

Acknowledgments

I am grateful to Marilia Sa Carvalho, FIOCRUZ, Rio de Janeiro, Brazil, for many comments and suggestions, and to the following individuals for translations of messages into other languages: Philippe Grojean (French), Anton Korobeinikov (Russian), and Marilia Sa Carvalho (Portuguese).

Author(s)

John Fox

Maintainer: John Fox <jfox@mcmaster.ca>

`anova.coxph`*anova method for coxph objects*

Description

This function masks `anova.coxph` in the **survival** package; its sole purpose is to supply the argument `test = "Chisq"` as the default.

Usage

```
## S3 method for class 'coxph'  
anova(object, ..., test = "Chisq")
```

Arguments

<code>object</code>	see <code>anova.coxph</code> .
<code>...</code>	<code>anova.coxph</code> .
<code>test</code>	<code>anova.coxph</code> .

Author(s)

John Fox <jfox@mcmaster.ca>

See Also[anova.coxph](#)

`Dialysis`*Hemodialysis Data from Brazil*

Description

This data set is analyzed in Sa Carvalho et al. (2003), and consists of data on 6805 hemodialysis patients in all federally funded clinics in Rio de Janeiro State, Brazil.

Usage`data(Dialysis)`

Format

A data frame with 6805 observations on the following 7 variables.

center a numeric code indicating in which of 67 centers the patient was treated.

age of the patient.

begin The month in which treatment began, with 1 representing January 1998.

end The month in which observation terminated, either because of death or censoring. The study ended in month 44 (August, 2000).

event 1, death, or 0, censoring.

time the difference between end and begin.

disease a factor with levels congen, (congenital); diabetes; hypert (hypertension); other; and renal.

Source

M. Sa Carvalho, R. Henderson, S. Shimakura, and I. P. S. C. Sousa (2003). Survival of hemodialysis patients: Modeling differences in risk of dialysis centers. *International Journal for Quality in Health Care*, 15: 189–196.

Examples

```
summary(Dialysis)
table(Dialysis$center)
```

mfrow

Function to Compute Layout for Plot Array

Description

Given a number of plots n , find an arrangement for showing the plots in an array, set by `par(mfrow=mfrow(n))`.

Usage

```
mfrow(n, max.plots = 0)
```

Arguments

`n` number of plots
`max.plots` maximum number of plots; 0, the default, means no maximum.

Author(s)

John Fox <jfox@mcmaster.ca>

See Also

par

Examples

```
mfrow(4)
mfrow(5)
mfrow(6)
```

plot.coxph

Plot Method for coxph Objects

Description

Plots the predicted survival function from a coxph object, setting covariates to particular values.

Usage

```
## S3 method for class 'coxph'
plot(x, newdata, typical = mean, byfactors=FALSE,
     col = palette(), lty, conf.level = 0.95, ...)
```

Arguments

x	a coxph object.
newdata	a data frame containing (combinations of) values to which predictors are set; optional.
typical	function to use to compute "typical" values of numeric predictors.
byfactors	if TRUE, different lines are drawn for each unique combination of factor values, including strata; if FALSE (the default) distinct lines are drawn only for different strata, with all columns of the model matrix (including for factors) set to their means.
col	colors for lines.
lty	line-types for lines; if missing, defaults to 1 to number required.
conf.level	level for confidence intervals; note: whether or not confidence intervals are plotted is determined by <code>plot.survfit</code> , which <code>plot.coxph</code> calls; if a <code>conf.int</code> argument is supplied it is passed through.
...	arguments passed to <code>plot</code> .

Details

If `newdata` is missing then all combinations of levels of factor-predictors (or strata), if present, are combined with "typical" values of numeric predictors.

Value

Invisibly returns the summary resulting from applying `survfit.coxph` to the `coxph` object.

Author(s)

John Fox <jfox@mcmaster.ca>.

See Also

[coxph](#), [survfit.coxph](#), [plot.survfit](#).

Examples

```
require(survival)
cancer$sex <- factor(iffelse(cancer$sex == 1, "male", "female"))

mod.1 <- coxph(Surv(time, status) ~ age + wt.loss, data=cancer)
plot(mod.1)
plot(mod.1, typical=function(x) quantile(x, c(.25, .75)))

mod.2 <- coxph(Surv(time, status) ~ age + wt.loss + sex, data=cancer)
plot(mod.2)

mod.3 <- coxph(Surv(time, status) ~ (age + wt.loss)*sex, data=cancer)
plot(mod.3)

mod.4 <- coxph(Surv(time, status) ~ age + wt.loss + strata(sex), data=cancer)
plot(mod.4)

mods.1 <- survreg(Surv(time, status) ~ age + wt.loss, data=cancer)
```

quantile.survfit

Compute Quantiles for a Survival Function

Description

This methods computes quantiles for a survival function (or survival functions by strata) as returned by `survfit`. Quantiles are computed conventionally as the last observed event time before each requested quantile. Confidence intervals for the estimated quantiles are obtained from the confidence limits of the etimated survival function.

Usage

```
## S3 method for class 'survfit'
quantile(x, quantiles = c(0.25, 0.5, 0.75), ...)
```

Arguments

x a survfit object.
quantiles quantiles to compute, default is 0.25, 0.5, and 0.75.
... not used.

Value

A vector of quantiles (or a matrix, if there are strata).

Author(s)

John Fox <jfox@mcmaster.ca>

See Also

[survfit](#).

Examples

```
require(survival)
quantile(survfit(Surv(time, status) ~ 1, data=cancer))
quantile(survfit(Surv(time, status) ~ sex, data=cancer))
```

Rossi

Rossi et al.'s Criminal Recidivism Data

Description

This data set is originally from Rossi et al. (1980), and is used as an example in Allison (1995). The data pertain to 432 convicts who were released from Maryland state prisons in the 1970s and who were followed up for one year after release. Half the released convicts were assigned at random to an experimental treatment in which they were given financial aid; half did not receive aid.

Usage

Rossi

Format

A data frame with 432 observations on the following 62 variables.

week week of first arrest after release or censoring; all censored observations are censored at 52 weeks.

arrest 1 if arrested, 0 if not arrested.

fin financial aid: no yes.

age in years at time of release.

race black or other.
wexp full-time work experience before incarceration: no or yes.
mar marital status at time of release: married or not married.
paro released on parole? no or yes.
prio number of convictions prior to current incarceration.
educ level of education: 2 = 6th grade or less; 3 = 7th to 9th grade; 4 = 10th to 11th grade; 5 = 12th grade; 6 = some college.
emp1 employment status in the first week after release: no or yes.
emp2 as above.
emp3 as above.
emp4 as above.
emp5 as above.
emp6 as above.
emp7 as above.
emp8 as above.
emp9 as above.
emp10 as above.
emp11 as above.
emp12 as above.
emp13 as above.
emp14 as above.
emp15 as above.
emp16 as above.
emp17 as above.
emp18 as above.
emp19 as above.
emp20 as above.
emp21 as above.
emp22 as above.
emp23 as above.
emp24 as above.
emp25 as above.
emp26 as above.
emp27 as above.
emp28 as above.
emp29 as above.
emp30 as above.

emp31 as above.
emp32 as above.
emp33 as above.
emp34 as above.
emp35 as above.
emp36 as above.
emp37 as above.
emp38 as above.
emp39 as above.
emp40 as above.
emp41 as above.
emp42 as above.
emp43 as above.
emp44 as above.
emp45 as above.
emp46 as above.
emp47 as above.
emp48 as above.
emp49 as above.
emp50 as above.
emp51 as above.
emp52 as above.

Source

Allison, P.D. (1995). *Survival Analysis Using the SAS System: A Practical Guide*. Cary, NC: SAS Institute.

References

Rossi, P.H., R.A. Berk, and K.J. Lenihan (1980). *Money, Work, and Crime: Some Experimental Results*. New York: Academic Press.

Examples

summary(Rossi)

`SurvivalData`*Define Survival Data Dialog Box*

Description

This dialog box permits you to define a time variable (or start and stop variables), an event indicator, a strata variable or variables, and a cluster variable to be associated with the current data set. If these characteristics are defined, then they will become default choices where appropriate in other dialog boxes.

Usage

```
SurvivalData() # normally not called directly
```

Value

Used only for its side effect.

Author(s)

John Fox <jfox@mcmaster.ca>

`unfold`*Convert a Survival Data Set from "Wide" to "Long" Format*

Description

Converts a survival-analysis data frame from "wide" format, in which time-varying covariates are separate variables, one per occasion, to "long" or counting-process format in which each occasion is a separate row in the data frame.

Usage

```
unfold(data, ...)
```

```
## S3 method for class 'data.frame'
```

```
unfold(data, time, event, cov,  
       cov.names = paste("covariate", ".", 1:ncovs, sep = ""),  
       suffix = ".time", cov.times = 0:ncov, common.times = TRUE, lag = 0,  
       show.progress=TRUE, ...)
```

Arguments

<code>data</code>	a data frame to be "unfolded" from wide to long.
<code>time</code>	the column number or quoted name of the event/censoring-time variable in data.
<code>event</code>	the column number or quoted name of the event/censoring-indicator variable in data.
<code>cov</code>	a vector giving the column numbers of the time-dependent covariate in data, or a list of vectors if there is more than one time-varying covariate.
<code>cov.names</code>	a character string or character vector giving the name or names to be assigned to the time-dependent covariate(s) in the output data set.
<code>suffix</code>	the suffix to be attached to the name of the time-to-event variable in the output data set; defaults to <code>`.time`</code> .
<code>cov.times</code>	the observation times for the covariate values, including the start time. This argument can take several forms: (1) The default is integers from 0 to the number of covariate values (i.e., one more than the length of each vector in <code>cov</code>). (2) An arbitrary numerical vector with one more entry than the length of each vector in <code>cov</code> . (3) The columns in the input data set that give the observations times for each individual. There should be one more column than the length of each vector in <code>cov</code> .
<code>common.times</code>	a logical value indicating whether the times of observation are the same for all individuals; defaults to <code>TRUE</code> .
<code>lag</code>	number of observation periods to lag each value of the time-varying covariate(s); defaults to 0.
<code>show.progress</code>	if <code>TRUE</code> , the default, show a progress bar as the observations are processed.
<code>...</code>	arguments to be passed down.

Value

A data frame containing the "long" version of the data set.

Author(s)

John Fox <jfox@mcmaster.ca>

Examples

```
if (interactive()){
  head(Rossi, 2)
  Rossi.long <- unfold(Rossi, time="week", event="arrest", cov=11:62,
    cov.names="emp")
  head(Rossi.long, 50)
}
```

Description

Converts a survival-analysis data frame from "wide" format, in which time-varying covariates are separate variables, one per occasion, to "long" or counting-process format in which each occasion is a separate row in the data frame.

Usage

```
Unfold() # called via the R Commander menus
```

Details

Most of the dialog box is self-explanatory. A time-varying covariate is identified by selecting the variables constituting the covariate in the "wide" version of the data set using the variable-list box at the lower-left; specifying a name to be used for the covariate in the "long" version of the data set; and pressing the *Select* button. This process is repeated for each time-varying covariate. All time-varying covariates have to be measured on the same occasions, which are assigned times 0, 1, ... in the output data set. If the covariates are to be lagged, this is indicated via the *Lag covariates* slider near the lower right. The default lag is 0 — i.e., no lag. The output data set will include variables named *start* and *stop*, which give the counting-process start and stop times for each row, and an event indicator composed of the name of the event indicator in the "wide" form of the data set and the suffix *.time*.

The *Unfold* dialog calls the `unfold` function, which is somewhat more flexible.

Author(s)

John Fox <jfox@mcmaster.ca>

See Also

[unfold](#)

Index

- *Topic **datasets**
 - Dialysis, [3](#)
 - Rossi, [7](#)
- *Topic **dplot**
 - mfrow, [4](#)
- *Topic **hplot**
 - plot.coxph, [5](#)
- *Topic **htest**
 - anova.coxph, [3](#)
- *Topic **manip**
 - unfold, [10](#)
- *Topic **package**
 - RcmdrPlugin.survival-package, [2](#)
- *Topic **survival**
 - anova.coxph, [3](#)
 - plot.coxph, [5](#)
 - quantile.survfit, [6](#)
 - SurvivalData, [10](#)

anova.coxph, [3](#), [3](#)

coxph, [6](#)

Dialysis, [3](#)

mfrow, [4](#)

plot.coxph, [5](#)

plot.survfit, [5](#), [6](#)

quantile.survfit, [6](#)

RcmdrPlugin.survival
(RcmdrPlugin.survival-package),
[2](#)

RcmdrPlugin.survival-package, [2](#)

Rossi, [7](#)

survfit, [6](#), [7](#)

survfit.coxph, [6](#)

SurvivalData, [10](#)

Unfold (Unfold-dialog), [12](#)

unfold, [10](#), [12](#)

Unfold-dialog, [12](#)