

Package: RcmdrPlugin.survival (via r-universe)

October 16, 2024

Type Package

Title R Commander Plug-in for the 'survival' Package

Version 1.3-2

Date 2023-08-19

Author John Fox

Maintainer John Fox <jfox@mcmaster.ca>

Depends survival, date, stats

Imports Rcmdr (>= 2.8-0), car

Description 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

RcmdrModels coxph, survreg, coxph.penal

NeedsCompilation no

Date/Publication 2023-08-21 08:52:38 UTC

Repository <https://john-d-fox.r-universe.dev>

RemoteUrl <https://github.com/cran/RcmdrPlugin.survival>

RemoteRef HEAD

RemoteSha 79d7437ab89f616282d858a931fb17d975e50ee2

Contents

RcmdrPlugin.survival-package	2
Dialysis	3

mfrow	4
plot.coxph	4
Rossi	6
SurvivalData	8
SurvivalDiagnostics	9
unfold	11
Unfold-dialog	12
Index	14

RcmdrPlugin.survival-package

*Rcmdr Plug-In Package for the **survival** Package*

Description

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

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), Matjaz Jeran (Slovenian), Anton Korobeinikov (Russian), Manuel Munoz Marquez (Spanish), and Marilia Sa Carvalho (Portuguese).

Author(s)

John Fox

Maintainer: John Fox <jfox@mcmaster.ca>

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32, doi:[10.18637/jss.v049.i07](https://doi.org/10.18637/jss.v049.i07).

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.

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

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

<code>n</code>	number of plots
<code>max.plots</code>	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>.

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

See Also

`coxph`, `survfit.coxph`, `plot.survfit`.

Examples

```
require(survival)
cancer$sex <- factor(ifelse(cancer$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)

```

 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.

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

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>

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

SurvivalDiagnostics *Diagnostics for Survival Regression Models*

Description

These are primarily convenience functions for the **RcmdrPlugin.survival** package, to produce diagnostics for `coxph` and `survreg` models in a convenient form for plotting via the package's GUI.

Usage

```
crPlots(model, ...)
## S3 method for class 'coxph'
crPlots(model, ...)

## S3 method for class 'coxph'
dfbeta(model, ...)
## S3 method for class 'dfbeta.coxph'
plot(x, ...)

## S3 method for class 'coxph'
dfbetas(model, ...)
## S3 method for class 'dfbetas.coxph'
plot(x, ...)

## S3 method for class 'survreg'
dfbeta(model, ...)
## S3 method for class 'dfbeta.survreg'
plot(x, ...)

## S3 method for class 'survreg'
dfbetas(model, ...)
## S3 method for class 'dfbetas.survreg'
plot(x, ...)
```

```

MartingalePlots(model, ...)
## S3 method for class 'coxph'
MartingalePlots(model, ...)

testPropHazards(model, test.terms = FALSE, plot.terms = FALSE, ...)
## S3 method for class 'coxph'
testPropHazards(model, test.terms = FALSE, plot.terms = FALSE, ...)

```

Arguments

<code>model, x</code>	a Cox regression or parametric survival regression model, as appropriate.
<code>test.terms</code>	test proportional hazards by terms in the Cox model, rather than by coefficients (default is FALSE).
<code>plot.terms</code>	diagnostic plots of proportional hazards by terms in the Cox model, rather than by coefficients (default is FALSE).
<code>...</code>	arguments to be passed down.

Details

- `crPlots.coxph` is a method for the `crPlots` function in the **car** package, to create component+residual (partial-residual) plots, using `residuals.coxph` and `predict.coxph` in the **survival** package.
- `testPropHazards` is essentially a wrapper for the `cox.zph` function in the **survival** package.
- `MartingalePlots` creates null-model Martingale plots for Cox regression models, using the `residuals.coxph` function in the **survival** package.
- `dfbeta.coxph` and `dfbetas.coxph` provide methods for the standard `dfbeta` and `dfbetas` functions, using the `residuals.coxph` function in the **survival** package for computation. `plot.dfbeta.coxph` and `plot.dfbetas.coxph` are plot methods for the objects produced by these functions.
- `dfbeta.survreg`, `dfbetas.survreg`, `plot.dfbeta.survreg` and `plot.dfbetas.survreg` are similar methods for **survreg** objects.

Value

Most of these function create graphs and don't return useful values; the `dfbeta` and `dfbetas` methods create matrices of `dfbeta` and `dfbetas` values.

Author(s)

John Fox <jfox@mcmaster.ca>

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

See Also

[coxph](#), [survreg](#), [crPlots](#), [residuals.coxph](#), [residuals.survreg](#), [predict.coxph](#), [cox.zph](#)

unfold	<i>Convert a Survival Data Set from "Wide" to "Long" Format</i>
--------	---

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

data	a data frame to be "unfolded" from wide to long.
time	the column number or quoted name of the event/censoring-time variable in data.
event	the column number or quoted name of the event/censoring-indicator variable in data.
cov	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.
cov.names	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.
suffix	the suffix to be attached to the name of the time-to-event variable in the output data set; defaults to '.time'.
cov.times	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 cov). (2) An arbitrary numerical vector with one more entry than the length of each vector in cov. (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 cov.
common.times	a logical value indicating whether the times of observation are the same for all individuals; defaults to TRUE.

lag number of observation periods to lag each value of the time-varying covariate(s); defaults to 0.

show.progress if TRUE, the default, show a progress bar as the observations are processed.

... arguments to be passed down.

Value

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

Author(s)

John Fox <jfox@mcmaster.ca>

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

Examples

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

Unfold-dialog

Dialog to 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() # 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>

References

John Fox, Marilia Sa Carvalho (2012). The RcmdrPlugin.survival Package: Extending the R Commander Interface to Survival Analysis. *Journal of Statistical Software*, 49(7), 1-32. doi:10.18637/jss.v049.i07.

See Also

`unfold`

Index

- * **datasets**
 - Dialysis, [3](#)
 - Rossi, [6](#)
- * **dplot**
 - mfrow, [4](#)
- * **hplot**
 - plot.coxph, [4](#)
- * **manip**
 - unfold, [11](#)
- * **package**
 - RcmdrPlugin.survival-package, [2](#)
- * **survival**
 - plot.coxph, [4](#)
 - SurvivalData, [8](#)
- cox.zph, [10](#), [11](#)
- coxph, [5](#), [9](#), [11](#)
- crPlots, [10](#), [11](#)
- crPlots (SurvivalDiagnostics), [9](#)
- dfbeta, [10](#)
- dfbeta.coxph (SurvivalDiagnostics), [9](#)
- dfbeta.survreg (SurvivalDiagnostics), [9](#)
- dfbetas, [10](#)
- dfbetas.coxph (SurvivalDiagnostics), [9](#)
- dfbetas.survreg (SurvivalDiagnostics), [9](#)
- Dialysis, [3](#)
- MartingalePlots (SurvivalDiagnostics), [9](#)
- mfrow, [4](#)
- plot.coxph, [4](#)
- plot.dfbeta.coxph (SurvivalDiagnostics), [9](#)
- plot.dfbeta.survreg (SurvivalDiagnostics), [9](#)
- plot.dfbetas.coxph (SurvivalDiagnostics), [9](#)
- plot.dfbetas.survreg (SurvivalDiagnostics), [9](#)
- plot.survfit, [5](#)
- predict.coxph, [10](#), [11](#)
- RcmdrPlugin.survival (RcmdrPlugin.survival-package), [2](#)
- RcmdrPlugin.survival-package, [2](#)
- residuals.coxph, [10](#), [11](#)
- residuals.survreg, [11](#)
- Rossi, [6](#)
- survfit.coxph, [5](#)
- SurvivalData, [8](#)
- SurvivalDiagnostics, [9](#)
- survreg, [9–11](#)
- testPropHazards (SurvivalDiagnostics), [9](#)
- Unfold (Unfold-dialog), [12](#)
- unfold, [11](#), [13](#)
- Unfold-dialog, [12](#)