Package 'spduration'

August 29, 2016

Title Split-Population Duration (Cure) Regression

Version 0.15.1

Date 2016-05-11

Description Functions for estimating split-duration regression models and various associated generic function methods.

Depends R (>= 3.1.2)

License GPL-3

LazyData true

Imports corpcor, graphics, plyr, MASS, separationplot, stats, Rcpp (>= 0.11.0), xtable

Suggests testthat

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

NeedsCompilation yes

Author Andreas Beger [aut, cre],
Daina Chiba [aut],
Daniel Hill [aut],
Nils Metternich [aut],
Shahryar Minhas [aut],
Michael Ward [cph]

Maintainer Andreas Beger <adbeger@gmail.com>

Repository CRAN

Date/Publication 2016-05-12 15:15:16

R topics documented:

accessors 2
add_duration 3
AIC.spdur 5
as.data.frame.spdur 6
BIC.spdur 7

bscoup 7
coups 8
forecast 9
forecast.default 10
forecast.spdur 10
model.coups 11
panel_lag 12
plot.spdur 13
plot_hazard 13
plot_hazard1 14
plot_hazard2 15
predict.spdur 15
print.summary.spdur 17
separationplot.spdur 17
spdur 18
spduration 20
summary.spdur 20
xtable.spdur 21

Index **23**

`accessors` *Accessor methods for spdur Objects*

Description

Several standard accessor methods for a `spdur` class object.

Usage

```
## S3 method for class 'spdur'
logLik(object, ...)

## S3 method for class 'spdur'
nobs(object, ...)

## S3 method for class 'spdur'
coef(object, model = c("full", "duration", "risk", "distr"),
  ...)

## S3 method for class 'spdur'
vcov(object, model = c("full", "duration", "risk", "distr"),
  ...)

## S3 method for class 'spdur'
model.matrix(object, model = c("duration", "risk"), ...)

## S3 method for class 'spdur'
terms(x, model = c("duration", "risk"), ...)
```

Arguments

object	an object inheriting from class spdur.
model	return full model, or only duration or risk equations, or distribution parameters.
x	spdur class object for terms
...	not used

See Also

AIC.spdur, BIC.spdur

Examples

```
data(model.coups)

logLik(model.coups)

nobs(model.coups)

coef(model.coups)

vcov(model.coups)

head(model.matrix(model.coups))

terms(model.coups)
```

---

add_duration *Add duration variables to panel data*

Description

Builds a duration version of a data frame representing panel data.

Usage

```
add_duration(data, y, unitID, tID, freq = "month", sort = FALSE,
  ongoing = TRUE, slice.last = FALSE)
```

Arguments

data	Data frame representing panel data.
y	A binary indicator of the incidence of some event, e.g. a coup.
unitID	Name of the variable in the data frame identifying the cross-sectional units, e.g. "country".
tID	Name of the variable in the data frame identifying the time unit, preferably as class Date. E.g. "year".

freq	Frequency at which units are measured in tID. Currently yearly, monthly, and daily data are supported, i.e. "year", "month", or "day".
sort	Sort data by unit and time? Default is FALSE, i.e. return data in original order.
ongoing	If TRUE, successive 1's are considered ongoing events and treated as NA after the first 1. If FALSE, successive 1's are all treated as failures.
slice.last	Set to TRUE to create a slice of the last time period; used with forecast.spdur. For compatibility with CRISP and ICEWS projects.

Details

This function processes a panel data frame by creating a failure variable from y and corresponding duration counter, as well as risk/immunity indicators. Supported time resolutions are year, month, and day, and input data should be (dis-)aggregated to one of these levels.

The returned data frame should have the same number of rows at the original. If y is an indicator of the incidence of some event, rather than an onset indicator, then ongoing spells of failure beyond the initial event are coded as NA (e.g. 000111 becomes a spell of 0001 NA NA). This is to preserve compatability with the base dataset. Note that the order of rows may be different though.

There cannot be missing values ("NA") in any of the key variables y, unitID, or tID; they will stop the function.

Furthermore, series that start with an event, e.g. (100), are treated as experiencing failure in the first time period. If those events are in fact ongoing, e.g. the last year of a war that started before the start time of the dataset, they should be dropped manually before using buildDuration().

t.0 is the starting time of the period of observation at tID. It is by default set as duration - 1 and currenlty only serves as a placeholder to allow future expansion for varying observation times.

Value

Returns the original data frame with 8 duration-specific additional variables:

failure	Binary indicator of an event.
ongoing	Binary indicator for ongoing events, not counting the inital failure time.
end.spell	Binary indicator for the last observation in a spell, either due to censoring or failure.
cured	Binary indicator for spells that are coded as cured, or immune from failure. Equal to 1 - atrisk.
atrisk	Binary indicator for spells that are coded as at risk for failure. Equal to 1 - cured.
censor	Binary indicator for right-censored spells.
duration	t, counter for how long a spell has survived without failure.
t.0	Starting time for period observed during t, by default equals duration - 1.

See Also

panel_lag for lagging variables in a panel data frame before building duration data.

Examples

```
# Yearly data
data <- data.frame(y=c(0,0,0,1,0),
                   unitID=c(1,1,1,1,1),
                   tID=c(2000, 2001, 2002, 2003, 2004))
dur.data <- add_duration(data, "y", "unitID", "tID", freq="year")
dur.data
```

`AIC.spdur` *AIC method for spdur*

Description

Computes the Akaike Information Criterion for an `spdur` class object.

Usage

```
## S3 method for class 'spdur'
AIC(object, ..., k = 2)
```

Arguments

`object`	An object of class `spdur`.
`k`	The penalty parameter, by default 2. For `BIC.spdur`, the penalty parameter equals `log(N)`.
`...`	Optional arguments.

See Also

`link{AIC}`, `link{BIC.spdur}`

Examples

```
data(model.coups)
AIC(model.coups)
```

`as.data.frame.spdur` *Convert spdur results to summary data frame*

Description

`table`-like function for class "`spdur`".

Usage

```
## S3 method for class 'spdur'
as.data.frame(x, row.names = TRUE, optional = FALSE, ...)
```

Arguments

`x`	An object with class `spdur`.
`row.names`	Indicates whether parameter names should be added as row names to the data frame returned, or as a separate column with blank row row names.
`optional`	Not used
`...`	Not used.

Details

This will create a data frame containing the estimated coefficients and standard errors for the risk and duration equations of a split-population duration model. It's intented purpose is to help create larger tables combining several model results.

Value

An data frame with model coefficients and p-values.

See Also

`xtable.spdur` for formatting a single model to Latex output.

Examples

```
data(model.coups)
data.frame(model.coups)
```

BIC.spdur *BIC method for spdur*

Description

Computes the Bayesian Information Criterion for an spdur class object.

Usage

```
## S3 method for class 'spdur'
BIC(object, ...)
```

Arguments

object	An object of class spdur.
...	Optional arguments.

Details

Computed as AIC(object, k = log(nobs(object))).

See Also

BIC, AIC.spdur

Examples

```
data(model.coups)
BIC(model.coups)
```

bscoup *B&S 2003 coup data*

Description

Replication data from Belkin and Schofer's 2003 paper on coups.

Usage

```
bscoups
```

Format

A data frame with 5828 observations of 9 variables:

`countryid` Gleditsch and Ward country codes.

`year` Year

`couprisk` Structural coup risk index, see paper for details.

`recentcoups` Alternative coup risk measure, running count of coups in past 10 years.

`rwar` Country participated in war in past 10 years.

`milreg` 1=Military regime, 0=other

`wealth` log of GDP per capita

`instab` Domestic instability and violence.

`coup` Indicator for successful coup.

`africa` Indicator for countries in Africa.

`eurnam` Indicator for countries in Europe and N. America.

`samerica` Indicator for countries in South America.

`camerica` Indicator for countries in Central America.

`regconf` Regional conflict.

Source

Belkin, Aaron and Evan Schofer. 2003. "Toward a structural understanding of coup risk." Journal of Conflict Resolution Vol. 47 No. 5.

Examples

```
data(bscoup)
table(bscoup$coup)
range(bscoup$year)
```

---

coups *Global coups, 1979 to 2010*

Description

Data on global coups from 1979 to 2010 from Powell & Thyne

Usage

```
coups
```

Format

A data frame with 5828 observations of 9 variables:

`gwcode` Gleditsch and Ward country codes.

`year` Year, in date format.

`coup1`

`succ.coup` Successful coup, 0/1.

`democ` Polity democracy score (0-10).

`autoc` Polity autocracy score (0-10).

`polity` Polity score (democ-autoc).

`polity2` Polity score with correction for regime transitions.

`regtrans` Regime transitions.

Source

Powell, Jonathan M. and Clayton L. Thyne. "Global instances of coups from 1950 to 2010: A new dataset." Journal of Peace Research Vol. 48 No. 2.

Gleditsch, Kristian S. and Michael D. Ward. 1999. "Interstate System Membership: A Revised List of the Independent States since 1816." International Interactions 25.

Examples

```
data(coups)
table(coups$succ.coup)
```

`forecast` *Model forecasts*

Description

`forecast` is a generic function for creating out-of-sample predictions. It invokes particular *methods* which depend on the `class` of the first argument

Usage

```
forecast(object, ...)
```

Arguments

`object`	A model object from which forecasts are created.
`...`	additional arguments affecting the forecasts produced.

Details

This generic is implemented in the `spduration` package with a `forecast.spdur` method.

forecast.default *Default forecast method*

Description

The default forecast method, currently undefined.

Usage

```
## Default S3 method:
forecast(object, ...)
```

Arguments

object	A model object.
...	Additional arguments

forecast.spdur *Plot spdur object predictions*

Description

forecast method for spdur class objects.

Usage

```
## S3 method for class 'spdur'
forecast(object, ..., pred.data = NULL,
  stat = "conditional hazard", n.ahead = 6)
```

Arguments

object	A spdur class model object.
pred.data	Data on which to base forecasts, i.e. slice of last time unit's observations for all cross-sectional units.
stat	Which statistic to forecast, see predict.spdur for possible options
n.ahead	How many time periods to predict ahead. Default is 6.
...	Optional aguments, not used.

Details

This function will create out-of-sample predictions of "stat" using model estimates and the prediction data provided. It is assumed that prediction data consist of a slice of the last time period observed for the data used to estimate the model in `object`. For each row, `forecast.spdur` will estimate the model predictions for that time point and then extrapolate the resulting probability to `n.ahead` time periods using appropriate probability theory.

For situations in which the covariate values are known for future time periods, e.g. in a test sample use `predict.spdur` instead.

Examples

```
data(coups)
data(model.coups)

coups.dur <- add_duration(coups, "succ.coup", "gwcode", "year", freq="year")
pred.data <- coups.dur[coups.dur$year==max(coups.dur$year), ]
pred.data <- pred.data[complete.cases(pred.data), ]
fcast <- forecast(model.coups, pred.data=pred.data)
```

`model.coups` *Model of global coups from 1979 to 2010*

Description

This is a model object for a split-duration model of the Powell & Thyne coups. It is used in several example code sections to speed up package testing by elminiating the need to re-estimate a model each time.

Usage

```
model.coups
```

Format

An object of class `spdur`.

Source

For information on the data used in this model, see the data documentation, `coups`.

Examples

```
data(model.coups)
str(model.coups)
```

panel_lag *Lag panel data*

Description

A function that correctly lags panel data where units are identified by `id` and time periods are identified with `t`. Results are in same order as `data` and are padded with `NA` as needed.

Usage

```
panel_lag(x, id, t, lag = 1, data = NULL)
```

Arguments

`x`	String identifying the vectors to be lagged in `data`.
`id`	String identifying the unit (e.g. country) identifier in `data`.
`t`	String identifying the time identifier in `data`.
`lag`	Lag order, i.e. by how many time periods should `x` be lagged? Unlike the default `lag`, positive values indicate that past data is used for the current time period.
`data`	A data frame. If not provided, a new one will be constructed with the vectors supplied for the other parameters.

Value

A vector of same length as `x` representing lagged values with leading `NA`'s.

Examples

```
data(coups)
# No need to order before using panelLag, just do it here so we can compare results below.
coups <- coups[order(coups$gwcode, coups$year), ]
test <- panel_lag("polity2", "gwcode", "year", data=coups)

# Compare output
head(coups$polity2)
head(test)
```

plot.spdur *Plot split-duration model results.*

Description

Plot results from a spduration model. Two types are currently implemented: a separation plot for evaluating model predictions ("sepplot"), and a plot of the conditional hazard rate ("hazard"), with or without simulation-based confidence intervals.

Usage

```
## S3 method for class 'spdur'
plot(x, type = "sepplot", ci = TRUE, ...)
```

Arguments

x	An object of class "spdur".
type	What kind of plot? "sepplot" or "hazard".
ci	For plots of the hazard rate, should a confidence interval be included?
...	Optional parameters passed to separationplot.spdur or plot_hazard.

See Also

separationplot.spdur, plot_hazard

Examples

```
# get model estimates
data(model.coups)

# plot
plot(model.coups)
plot(model.coups, type = "hazard")
```

plot_hazard *Plot hazard function*

Description

plot_hazard plots the shape of estimated hazard function in respect to duration, given a set of values for the duration and risk equations covariates. Confidence intervals are provided through simulation.

Usage

```
plot_hazard(x, t = NULL, ci = TRUE, n = 1000, xvals = NULL,
  zvals = NULL, ...)
```

Arguments

x	An object of class spdur
t	Time values at which to evaluate hazard function, e.g. c(1:50). Defaults to 1 through 1.2 * maximum duration value in data.
ci	Compute simulation-based confidence interval?
n	Number of simulations to use for CI, defaults to 1,000.
xvals	A vector of values for the duration equation variables, in the same order as the duration equation in x. Defaults to means.
zvals	A vector of values for the risk equation varialbes, in teh same order as the risk equation in x. Defaults to means.
...	Additional parameters passed to plot.

See Also

separationplot.spdur

Examples

```
# Get model estimates
data(model.coups)

# Plot
plot_hazard(model.coups, ci = FALSE)
plot_hazard(model.coups, ci = TRUE)
```

---

plot_hazard1 *Plot conditional hazard rate*

Description

Plot hazard function without simulated confidence intervals. See plot_hazard instead.

Usage

```
plot_hazard1(x, ...)
```

Arguments

x	class "spdur" object
...	passed to plot_hazard

Value

NULL, plots.

`plot_hazard2` *Simulate and plot hazard function*

Description

Plot hazard function with simulated confidence intervals. See `plot_hazard` instead.

Usage

```
plot_hazard2(x, ...)
```

Arguments

`x`	class "spdur" object
`...`	passed to `plot_hazard`

Value

NULL, plots.

`predict.spdur` *Predict methods for spdur Objects*

Description

`predict` and related methods for class "`spdur`".

Usage

```
## S3 method for class 'spdur'
predict(object, newdata = NULL, type = c("response"),
  truncate = TRUE, ...)

## S3 method for class 'spdur'
fitted(object, ...)

## S3 method for class 'spdur'
residuals(object, type = c("response"), ...)
```

Arguments

`object`	Object of class "spdur".
`newdata`	Optional data for which to calculate fitted values, defaults to training data.
`type`	Quantity of interest to calculate. Default conditional hazard, i.e. conditioned on observed survival up to time `t`. See below for list of values. For `residuals`, the type of residual to calculate
`truncate`	For conditional hazard, truncate values greater than 1.
`...`	not used, for compatability with generic function.

Details

Calculates various types of probabilities, where "conditional" is used in reference to conditioning on the observed survival time of a spell up to time t, in addition to conditioning on any variables included in the model (which is always done). Valid values for the `type` option include:

- "conditional risk": $Pr(Cure = 0|Z\gamma, T > t)$
- "conditional cure": $Pr(Cure = 1|Z\gamma, T > t)$
- "hazard": $Pr(T = t|T > t, C = 0, X\beta) * Pr(Cure = 0|Z\gamma)$
- "failure": $Pr(T = t|T > t-1, C = 0, X\beta) * Pr(Cure = 0|Z\gamma)$
- "unconditional risk": $Pr(Cure = 0|Z\gamma)$
- "unconditional cure": $Pr(Cure = 1|Z\gamma)$
- "conditional hazard" or "response": $Pr(T = t|T > t, C = 0, X\beta) * Pr(Cure = 0|Z\gamma, T > t)$
- "conditional failure": $Pr(T = t|T > t-1, C = 0, X\beta) * Pr(Cure = 0|Z\gamma, T > t)$

The vector $Z\gamma$ indicates the cure/at risk equation covariate vector, while $X\beta$ indicates the duration equation covariate vector.

Value

Returns a data frame with 1 column corresponding to `type`, in the same order as the data frame used to estimate `object`.

Note

See `forecast.spdur` for producing forecasts when future covariate values are unknown.

Examples

```
# get model estimates
data(model.coups)
ch <- predict(model.coups)

head(fitted(model.coups))

head(residuals(model.coups))
```

print.summary.spdur *Print a split-population duration model results summary*

Description

print method for class "summary.spdur".

Usage

```
## S3 method for class 'summary.spdur'
print(x, ...)
```

Arguments

x	An object with class spdur.
...	Further arguments passed to or from other methods.

Details

Formats spdur summaries for printing.

See Also

The model fitting function is spdur, and see summary.spdur for associated summary method.

Examples

```
data(model.coups)
s <- summary(model.coups)
class(s)
print(s)
```

separationplot.spdur *Generate a Separation Plot*

Description

separationplot method for class "spdur".

Usage

```
## S3 method for class 'spdur'
separationplot(x, pred_type = "conditional hazard",
  obs = NULL, endSpellOnly = FALSE, lwd1 = 5, lwd2 = 2,
  shuffle = TRUE, heading = "", show.expected = TRUE, newplot = FALSE,
  type = "line", ...)
```

Arguments

x	An object of class "spdur".
pred_type	Which statistic to plot, i.e. "conditional hazard" or "conditional risk".
obs	Variable that captures observed outcomes. If NULL (default), it is chosen based on pred_type: "fail" for (conditional) hazard, and "atrisk" for (conditional) risk.
endSpellOnly	Should only the last observation in each spell be kept? FALSE by default.
lwd1	See separationplot.
lwd2	See separationplot.
shuffle	See separationplot.
heading	See separationplot.
show.expected	See separationplot.
newplot	See separationplot
type	See separationplot.
...	Optional parameters passed to separationplot, e.g. type of statistic to calculate.

Details

Creates a separationplot of fitted values from split-duration model results using predict.spdur.

See Also

separationplot, predict.spdur

Examples

```
# get model estimates
data(model.coups)

# plot
p <- plot(model.coups)
```

---

spdur *Split-population duration (cure) regression*

Description

This function estimates a split-population duration model and returns a object of class spdur.

Usage

```
spdur(duration, atrisk, data = NULL, last = "end.spell", t.0 = "t.0",
  fail = "failure", distr = c("weibull", "loglog"), max.iter = 300,
  na.action, silent = FALSE, ...)
```

Arguments

duration	A formula of the form Y ~ X1 + X2 ..., where Y is duration until failure or censoring.
atrisk	A formula of the form C ~ Z1 + Z2 ..., where C is a binary indicator of risk (1 - cure).
data	A data frame containing the variables in formula and formula2.
last	A string identifying the vector in data that indicates when a spell ends due to failure or right-censoring.
t.0	The starting point for time-varying covariate intervals, by default duration-1 when using add_duration.
fail	Name of the variable indicating that a spell ended in failure.
distr	The type of distribution to use in the hazard rate. Valid options are "weibull" or "loglog".
max.iter	Maximum number of iterations to use in the likelihood maximization.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset.
silent	Supress optimization output, FALSE by default.
...	Optional arguments, see details.

Details

See summary.spdur, predict.spdur , and plot.spdur for post-estimation options.

Optional arguments:

base.inits Initial values for the base duration model that is estimates to get initial values for the full split-population model. This needs to be a vector with starting values for the constant, coefficients in the duration equation, and an additional value for the shape parameter of the density used, e.g. Weibull. By default they are 0 for all coefficients and 0 or 1 for the Weibull and LogLog shape parameters respectively.

Value

Returns an object of class spdur, with attributes:

coefficients	A named vector of coefficient point estimates.
vcv	Estimated covariance matrix.
se	Standard error estimates.
zstat	Z-statistic values.
pval	P-values.
mf.dur	Model frame for the duration equation.
mf.risk	Model frame for the risk equation.
Y	Matrix of duration variables: risk, duration, end of spell, and t.0.
na.action	What action was taken for missing values in data.
call	The original, unevaluated spdur call.
distr	Distribution used for the hazard rate.

Examples

```
# Prepare data
data(coups)
dur.coups <- add_duration(coups, "succ.coup", unitID="gwcode", tID="year",
                          freq="year")

# Estimate model
model.coups <- spdur(duration ~ polity2, atrisk ~ polity2, data=dur.coups)
model.coups <- spdur(duration ~ polity2, atrisk ~ polity2, data=dur.coups,
                     distr="loglog")
```

`spduration` *Split-Population Duration (Cure) Regression Models*

Description

The spduration package provides functions to estimate split-population duration regression models in which only a subset of the population is at risk for failure, while the remainder is immune, or cured, from the possibility of experiencing a failure event. In practice, this class of models also may produce better performance in sparse data with few actual failure events.

Details

The main function `spdur` is used to estimate the model objects with class `spdur`.

Postestimation tools include `predict.spdur`, for calculating fitted values with arbitrary data and for several probabilities that might be of interest, as well as `plot.spdur` for visual display of model fit.

References

Leisch, Friedrich. 2009. "Creating R Packages: A Tutorial."

Svolik, Milan. 2008. "Authoritarian Reversals and Democratic Consolidation." American Political Science Review.

`summary.spdur` *Summarize split-population duration results*

Description

`summary` method for class "spdur".

Usage

```
## S3 method for class 'spdur'
summary(object, ...)
```

Arguments

object	An object with class spdur.
...	Further arguments passed to or from other methods.

Details

This will list the estimated coefficients and standard errors for the risk and duration equations of a split-population duration model.

Value

An object with class summary.spdur.

See Also

The model fitting function is spdur, and see summary for the generic function.

For print formatting, see print.summary.spdur.

Examples

```
data(model.coups)
s <- summary(model.coups)
class(s)
print(s)
```

---

xtable.spdur *Create export table for a split-duration model*

Description

xtable-like function for class "spdur".

Usage

```
## S3 method for class 'spdur'
xtable(x, ...)
```

Arguments

x	An object with class spdur.
...	Further arguments passed to xtable.

Details

Format a split-duration model for export to Latex or html.

Value

An object with class `xtable`.

See Also

`xtable`, or `as.data.frame.spdur` for a simpler alternative that will convert a `spdur` object to a data frame containing model parameter estimates.

For print formatting, see `print.xtable`.

Examples

```
library(xtable)
data(model.coups)
xtable(model.coups)
print(xtable(model.coups), include.rownames=FALSE)
```

Index

∗Topic **datasets**
bscoup, 7
coups, 8

accessors, 2
add_duration, 3, *19*
AIC.spdur, *3*, 5, *7*
as.data.frame.spdur, 6, *22*

BIC, *7*
BIC.spdur, *3*, *5*, 7
bscoup, 7

class, *9*
coef.spdur *(*accessors*)*, 2
coups, 8, *11*

Date, *3*

fitted.spdur *(*predict.spdur*)*, 15
forecast, 9, *10*
forecast.default, 10
forecast.spdur, *4*, *9*, 10, *16*

lag, *12*
logLik.spdur *(*accessors*)*, 2

model.coups, 11
model.matrix.spdur *(*accessors*)*, 2

na.fail, *19*
nobs.spdur *(*accessors*)*, 2

panel_lag, *4*, 12
plot, *14*
plot.spdur, 13, *19*, *20*
plot_hazard, *13*, 13, *14*, *15*
plot_hazard1, 14
plot_hazard2, 15
predict.spdur, *10*, *11*, 15, *18–20*
print.summary.spdur, 17, *21*
print.xtable, *22*

residuals.spdur *(*predict.spdur*)*, 15

separationplot, *17*, *18*
separationplot.spdur, *13*, *14*, 17
spdur, *10*, *17*, 18, *20*, *21*
spduration, *9*, 20
spduration-package *(*spduration*)*, 20
summary, *21*
summary.spdur, *17*, *19*, 20

terms.spdur *(*accessors*)*, 2

vcov.spdur *(*accessors*)*, 2

xtable, *21*, *22*
xtable.spdur, *6*, 21