

Package ‘spduration’

April 7, 2014

Title Split-Population Duration (Cure) Regression

Version 0.11

Date 2014-03-07

Author

Andreas Beger, Daina Chiba, Daniel W. Hill, Nils Metternich, Shahryar Minhas, Michael D. Ward

Maintainer Andreas Beger <andreas.beger@duke.edu>

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

Depends R (>= 2.15.0)

Imports corpcor, ggplot2, plyr, separationplot

Suggests wicews, CRISP

License GPL-3

R topics documented:

spduration-package	2
AIC.spdur	2
attemptDate	3
BIC.spdur	4
buildDuration	4
countryplot	6
coups	7
forecast	8
forecast.default	8
forecast.spdur	9
logLik.spdur	10
model.coups	10
nobs.spdur	11
panelLag	11
plot.spdur	12
predict.spdur	13
print.summary.spdur	14
spdur	14
spdurCrisp	16
summary.spdur	17

Index**19**

spduration-package	<i>Split-Population Duration (Cure) Regression Models</i>
--------------------	---

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

Package:	spdur
Type:	Package
Version:	0.11 (pre-release)
Date:	2014-03-07
License:	GPL 3.0

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.

Author(s)

Andreas Beger, Daina Chiba, Daniel W. Hill, Nils W. Metternich, and Michael D. Ward
 Maintainer: Andreas Beger <andreas.beger@duke.edu>

References

Leisch, Friedrich. 2009. "Creating R Packages: A Tutorial." Svolik, Milan. 2008. "Authoritarian Reversals and Democratic Consolidation." American Political Science Review.

Examples

```
library(spduration)
demo(coups)
```

AIC.spdur	<i>AIC method for spdur</i>
-----------	-----------------------------

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 <code>spdur</code> .
...	Optional arguments.
k	The penalty parameter, by default 2. For BIC.spdur , the penalty parameter equals $\log(N)$.

author Andreas Beger

See Also

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

Examples

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

attemptDate	<i>Attempt to convert to R date format</i>
-------------	--

Description

For internal use only

Usage

```
attemptDate(date, by)
```

Arguments

date	character or numeric
by	character, "year", "month", or "day"

BIC.spdur	<i>BIC method for spdur</i>
-----------	-----------------------------

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)))`.

Author(s)

Andreas Beger

See Also

[BIC](#), [AIC.spdur](#)

Examples

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

buildDuration	<i>Build duration version of panel data</i>
---------------	---

Description

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

Usage

```
buildDuration(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 compatibility 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 currently only serves as a placeholder to allow future expansion for varying observation times.

Value

Returns the original data frame with 9 duration-specific additional variables:

spellID	Unique key for spells.
failure	Binary indicator of an event.
ongoing	Binary indicator for ongoing events, not counting the initial 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.

Author(s)

Andreas Beger

See Also

[panellag](#) 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 <- buildDuration(data, "y", "unitID", "tID", freq="year")
dur.data
```

countryplot

Plot predicted values for a country

Description

For use with CRISP/ICEWS.

Usage

```
countryplot(eoi = "insurgency", model, country, lastMonth, fcastMonths = 24,
            cumul = T, yext = T)
```

Arguments

eoi	Name of variable of interest as quoted string.
model	An object of class <code>spdur</code> .
country	Name of the country to plot.
lastMonth	Month from which to forecast from, as a date object e.g. as <code>.Date("2001-03-01")</code>
fcastMonths	Number of months to forecast ahead.
cumul	Show cumulative or absolute probabilities of an event? (T/F)
yext	Extend y-axis to theoretical maximum of 1? (T/F)

Author(s)

Shahryar Minhas, Andreas Beger

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	<i>Model forecasts</i>
----------	------------------------

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 [plot.spdur](#) method.

forecast.default	<i>Default forecast method</i>
------------------	--------------------------------

Description

The default forecast method, currently undefined.

Usage

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

Arguments

object	A model object.
...	Additional arguments

forecast.spdur	<i>Plot spdur object predictions</i>
----------------	--------------------------------------

Description

`plot` method for `spdur` class objects.

Usage

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

Arguments

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

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.

Author(s)

Andreas Beger, Daina Chiba

Examples

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

coups.dur <- buildDuration(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)
```

logLik.spdur	<i>Log-Likelihood of an spdur Object</i>
--------------	--

Description

Returns the log-likelihood of a spdur class object.

Usage

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

Arguments

object	an object inheriting from class spdur.
...	not used

Examples

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

model.coups	<i>Model of global coups from 1979 to 2010</i>
-------------	--

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 eliminating 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)
```

nobs.spdur	<i>Number of Observations in a spdur Object</i>
------------	---

Description

Extract the number of observations in a spdur class model object.

Usage

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

Arguments

object	an object inheriting from class spdur.
...	not used.

See Also

[AIC.spdur](#), [BIC.spdur](#)

panelLag	<i>Lag panel data</i>
----------	-----------------------

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

```
panelLag(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.

Author(s)

Andreas Beger

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 <- panelLag("polity2", "gwcode", "year", data=coups)

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

plot.spdur

Plot split-duration model fit.

Description

plot method for class "spdur".

Usage

```
## S3 method for class 'spdur'
plot(x, ..., failure = "failure", endSpellOnly = TRUE)
```

Arguments

x	An object of class "spdur".
...	Optional parameters passed to <code>predict.spdur</code> , e.g. type of statistic to calculate.
failure	The variable indicating that a failure event has occurred at time t.
endSpellOnly	Should only the last observation in each spell be kept? TRUE by default.

Details

Creates a [separationplot](#) of fitted values from split-duration model results using [predict.spdur](#).

Author(s)

Andreas Beger

See Also

[separationplot](#), [predict.spdur](#)

Examples

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

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

predict.spdur	<i>Predict fitted values for a split-population duration model</i>
---------------	--

Description

predict method for class “spdur”.

Usage

```
## S3 method for class 'spdur'
predict(object, data = NULL, stat = "conditional risk", ...)
```

Arguments

object	Object of class “spdur”.
data	Optional data for which to calculate fitted values, defaults to training data.
stat	Quantity of interest to calculate. Default conditional probability of being at risk, i.e. conditioned on observed survival up to time t . See below for list of values.
...	Optional arguments to pass to predict 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 `stat` 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”: $Pr(T = t|T > t, C = 0, X\beta) * Pr(Cure = 0|Z\gamma, T > t)$
- “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 `stat`, 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.

Author(s)

Andreas Beger, Daina Chiba, Daniel W. Hill, Nils Metternich

Examples

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

```
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 = "weibull", max.iter = 300, na.action,
  silent = FALSE, ...)
```

Arguments

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

Details

See `summary.spdur`, `predict.spdur`, `plot.spdur`, and `countryplot` 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:

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

Author(s)

Andreas Beger

Examples

```
## Not run:
# Prepare data
data(coups)
dur.coups <- buildDuration(coups, "succ.coup", unitID="gwcode", tID="year",
                           freq="year")

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

## End(Not run)
```

spdurCrisp

*Split-Pop Duration Model Wrapper for CRISP/ICEWS***Description**

This is a wrapper for the CRISP and ICEWS packages that replicated the original functionality of the `spdur` command by estimating a model, calculating training and validation set fitted values and 6-month forecasts.

Usage

```
spdurCrisp(duration, atrisk, train = train, test = test, pred = pred,
            last = "end.spell", t.0 = "t.0", distr = "weibull",
            stat = "conditional risk", iter = 100, npred = 6, ...)
```

Arguments

<code>duration</code>	A formula of the form $Y \sim X1 + X2 \dots$, where Y is duration until failure or censoring.
<code>atrisk</code>	A formula of the form $C \sim Z1 + Z2 \dots$, where C is a binary indicator of risk (1 - cure).
<code>train</code>	Object containing the training data used to estimate the model.
<code>test</code>	Object containing the test data used to validate the model.
<code>pred</code>	Object containing the data to use for forecasting.
<code>last</code>	A string identifying the vector in data that indicates when a spell ends due to failure or right-censoring.
<code>t.0</code>	The starting point for time-varying covariate intervals, by default <code>duration-1</code> when using <code>buildDuration</code> .
<code>distr</code>	The type of distribution to use in the hazard rate. Valid options are "weibull" or "loglog".
<code>stat</code>	See <code>predict.spdur</code>
<code>iter</code>	Maximum number of iterations to use in the likelihood maximization.
<code>npred</code>	Number of months to predict out.
<code>...</code>	Optional arguments; not used currently.

Value

Returns an object of class `crisp`, which inherits from `spdur`.

Author(s)

Andreas Beger

See Also

[spdur](#)

Examples

```
## Not run:
# Prepare data
data(coups)
dur.coups <- buildDuration(coups, "succ.coup", unitID="gwcode", tID="year",
                           freq="year")

# Estimate model
model3 <- spdurCrisp(
  duration ~ polity2,
  atrisk ~ polity2,
  train=dur.coups, test=dur.coups[1,],
  pred=dur.coups[1,])

## End(Not run)
```

summary.spdur

Summarize split-population duration results

Description

summary method for class “`spdur`”.

Usage

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

Arguments

<code>object</code>	An object with class <code>spdur</code> .
<code>...</code>	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)
```

Index

AIC.spdur, [2](#), [4](#), [11](#)
attemptDate, [3](#)

BIC, [4](#)
BIC.spdur, [3](#), [4](#), [11](#)
buildDuration, [4](#), [15](#), [16](#)

class, [8](#)
countryplot, [6](#), [15](#)
coups, [7](#), [10](#)

Date, [5](#)

forecast, [8](#)
forecast.default, [8](#)
forecast.spdur, [5](#), [9](#), [13](#)

lag, [11](#)
logLik.spdur, [10](#)

model.coups, [10](#)

na.fail, [15](#)
nobs.spdur, [11](#)

panelLag, [6](#), [11](#)
plot, [9](#)
plot.spdur, [2](#), [8](#), [12](#), [15](#)
predict.spdur, [2](#), [9](#), [12](#), [13](#), [15](#), [16](#)
print.summary.spdur, [14](#), [18](#)

separationplot, [12](#)
spdur, [2](#), [9](#), [14](#), [14](#), [17](#), [18](#)
spduration, [8](#)
spduration (spduration-package), [2](#)
spduration-package, [2](#)
spdurCrisp, [16](#)
summary, [18](#)
summary.spdur, [14](#), [15](#), [17](#)