

Package: spduration (via r-universe)

September 21, 2026

Title Split-Population Duration (Cure) Regression

Version 0.17.3.9000

Description An implementation of split-population duration regression models. Unlike regular duration models, split-population duration models are mixture models that accommodate the presence of a sub-population that is not at risk for failure, e.g. cancer patients who have been cured by treatment. This package implements Weibull and Loglogistic forms for the duration component, and focuses on data with time-varying covariates. These models were originally formulated in Boag (1949) and Berkson and Gage (1952), and extended in Schmidt and Witte (1989).

Encoding UTF-8

Depends R (>= 3.1.2)

License GPL-3

LazyData true

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

Suggests covr, devtools, testthat, knitr, rmarkdown, tibble

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

VignetteBuilder knitr

URL <https://github.com/andybega/spduration>,
<https://www.andybeger.com/spduration/>,
<http://www.andybeger.com/spduration/>

BugReports <https://github.com/andybega/spduration/issues>

Config/pak/sysreqs cmake make libicu-dev libuv1-dev

Repository <https://andybega.r-universe.dev>

Date/Publication 2025-07-28 10:18:16 UTC

RemoteUrl <https://github.com/andybega/spduration>

RemoteRef HEAD
RemoteSha c4baf0ab9791a569c09c19fbb347b937e0a9d1ed

Contents

accessors	2
add_duration	3
AIC.spdur	5
as.data.frame.spdur	6
BIC.spdur	7
bscoup	7
coups	8
forecast.spdur	9
model.coups	10
panel_lag	11
plot.spdur	12
plot_hazard	12
plot_hazard1	13
plot_hazard2	14
predict.spdur	14
print.summary.spdur	16
sepplot	17
spdur	18
spduration	20
summary.spdur	21
xtable.spdur	22
Index	23

accessors	<i>Accessor methods for spdur Objects</i>
-----------	---

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 <code>spdur</code> .
...	not used
model	return full model, or only duration or risk equations, or distribution parameters.
x	<code>spdur</code> class object for terms

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	<i>Add duration variables to panel data</i>
--------------	---

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

<code>data</code>	Data frame representing panel data.
<code>y</code>	A binary indicator of the incidence of some event, e.g. a coup.
<code>unitID</code>	Name of the variable in the data frame identifying the cross-sectional units, e.g. "country".
<code>tID</code>	Name of the variable in the data frame identifying the time unit, preferably as class Date . E.g. "year".
<code>freq</code>	Frequency at which units are measured in <code>tID</code> . Currently yearly, monthly, and daily data are supported, i.e. "year", "month", or "day".
<code>sort</code>	Sort data by unit and time? Default is FALSE, i.e. return data in original order.
<code>ongoing</code>	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.
<code>slice.last</code>	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 8 duration-specific additional variables:

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

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

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

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

<code>x</code>	An object with class <code>spdur</code> .
<code>row.names</code>	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.
<code>optional</code>	Not used
<code>...</code>	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 intended 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	<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)))`.

See Also

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

Examples

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

bscoup	<i>B&S 2003 coup data</i>
--------	-------------------------------

Description

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

Usage

```
bscoup
```

Format

A data frame with 5463 observations of 14 variables:

countryid Gleditsch and Ward country codes.

year Year

coup risk 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:

gocode 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.spdur	<i>Forecast from a spdur model</i>
----------------	------------------------------------

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.
...	Optional arguments, not used.
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.

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

```
library(forecast)
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 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)
```

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

<code>x</code>	String identifying the vectors to be lagged in data.
<code>id</code>	String identifying the unit (e.g. country) identifier in data.
<code>t</code>	String identifying the time identifier in data.
<code>lag</code>	Lag order, i.e. by how many time periods should <code>x</code> be lagged? Unlike the default lag , positive values indicate that past data is used for the current time period.
<code>data</code>	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	<i>Plot split-duration model results.</i>
------------	---

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 sepplot or plot_hazard .

See Also

[sepplot](#), [plot_hazard](#)

Examples

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

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

plot(model.coups)
```

plot_hazard	<i>Plot hazard function</i>
-------------	-----------------------------

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

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

See Also

[sepplot](#)

Examples

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

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

<code>plot_hazard1</code>	<i>Plot conditional hazard rate</i>
---------------------------	-------------------------------------

Description

Plot hazard function without simulated confidence intervals. See [plot_hazard](#) instead.

Usage

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

Arguments

<code>x</code>	class "spdur" object
<code>...</code>	passed to <code>plot_hazard</code>

Value

NULL, plots.

plot_hazard2	<i>Simulate and plot hazard function</i>
--------------	--

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	<i>Predict methods for spdur Objects</i>
---------------	--

Description

predict and related methods for class "spdur".

Usage

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

## 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.
na.action	Function determining what should be done with missing values in newdata. The default is to predict NA (na.exclude).
...	not used, for compatibility 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	<i>Print a split-population duration model results summary</i>
---------------------	--

Description

print method for class “summary.spdur”.

Usage

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

Arguments

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

sepplot

*Generate a Separation Plot***Description**

A [separationplot](#) wrapper for class “spdur”.

Usage

```
sepplot(
  x,
  pred_type = "conditional hazard",
  obs = NULL,
  endSpellOnly = FALSE,
  lwd1 = 5,
  lwd2 = 2,
  shuffle = TRUE,
  heading = "",
  show.expected = TRUE,
  newplot = FALSE,
  type = "line",
  ...
)
```

Arguments

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

Details

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

See Also

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

Examples

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

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

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

<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>add_duration</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”; defaults to “weibull”.
<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>	Suppress optimization output, FALSE by default.
<code>...</code>	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 estimated 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.

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.

Author(s)

Maintainer: Andreas Beger <adbeger@gmail.com> ([ORCID](#))

Authors:

- Daina Chiba <dchiba@essex.ac.uk> ([ORCID](#))
- Daniel W. Hill, Jr. <dwhill@uga.edu>
- Nils W. Metternich <n.metternich@ucl.ac.uk> ([ORCID](#))
- Shahryar Minhas <hermes829@gmail.com>
- Michael D. Ward <michael.don.ward@gmail.com> ([ORCID](#)) [copyright holder]

References

Boag, J.W. 1949. "Maximum Likelihood Estimates of the Proportion of Patients Cured by Cancer Therapy." <<https://www.jstor.org/stable/2983694>>

Berkson, J. and Gage, R.P. "Survival Curve for Cancer Patients Following Treatment." <<https://www.jstor.org/stable/2281318>>

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

Schmidt, Peter and Witte, Ann Dryden. 1989. "Predicting Criminal Recidivism Using "Split Population" Survival Time Models." <[doi:10.1016/0304-4076\(89\)90034-1](https://doi.org/10.1016/0304-4076(89)90034-1)>

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

See Also

Useful links:

- <https://github.com/andybega/spduration>
- <https://www.andybeger.com/spduration/>
- <http://www.andybeger.com/spduration/>
- Report bugs at <https://github.com/andybega/spduration/issues>

summary.spdur	<i>Summarize split-population duration results</i>
---------------	--

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

<code>x</code>	An object with class <code>spdur</code> .
<code>...</code>	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

* 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

coef.spdur (accessors), 2

coups, 8, 11

Date, 4

fitted.spdur (predict.spdur), 14

forecast, 9

forecast.spdur, 4, 9, 15

lag, 11

logLik.spdur (accessors), 2

model.coups, 10

model.matrix.spdur (accessors), 2

na.fail, 19

nobs.spdur (accessors), 2

panel_lag, 5, 11

plot, 13

plot.spdur, 12, 19, 20

plot_hazard, 12, 12–14

plot_hazard1, 13

plot_hazard2, 14

predict.spdur, 10, 14, 17–20

print.summary.spdur, 16, 21

print.xtable, 22

residuals.spdur (predict.spdur), 14

separationplot, 17, 18

sepplot, 12, 13, 17

spdur, 9, 10, 16, 18, 20, 21

spduration, 20

spduration-package (spduration), 20

summary, 21

summary.spdur, 16, 19, 21

terms.spdur (accessors), 2

vcov.spdur (accessors), 2

xtable, 22

xtable.spdur, 6, 22