

Package ‘tram’

June 22, 2018

Title Transformation Models

Version 0.2-2

Date 2018-06-22

Description Formula-based user-interfaces to specific transformation models implemented in package 'mlt'. Available models include Cox models, some parametric survival models (Weibull, etc.), models for ordered categorical variables, normal and non-normal (Box-Cox type) linear models, and continuous outcome logistic regression (Lohse et al., 2017, <DOI:10.12688/f1000research.12934.1>). The underlying theory is described in Hothorn et al. (2018) <DOI:10.1111/sjos.12291>.

Depends mlt (>= 1.0-2)

Imports Formula, multcomp, variables (>= 1.0-0), basefun (>= 1.0-0), sandwich, stats, survival, graphics

Suggests MASS, TH.data, trtf, mlbench, knitr, quantreg, ordinal, colorspace, ATR

VignetteBuilder knitr

Encoding UTF-8

License GPL-2

NeedsCompilation no

Author Torsten Hothorn [aut, cre] (<<https://orcid.org/0000-0001-8301-0471>>)

Maintainer Torsten Hothorn <Torsten.Hothorn@R-project.org>

Repository CRAN

Date/Publication 2018-06-22 18:26:36 UTC

R topics documented:

BoxCox	2
Colr	3
Coxph	4
Lm	6
Polr	7

Survreg	8
tram	10
tram-methods	12

Index	15
--------------	-----------

BoxCox	<i>(Similar to) Box-Cox Models</i>
--------	------------------------------------

Description

Non-normal linear regression inspired by Box-Cox models

Usage

```
BoxCox(formula, data, subset, weights, offset, cluster, na.action = na.omit, ...)
```

Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under tram and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	a function which indicates what should happen when the data contain NAs. The default is set to <code>na.omit</code> .
...	additional arguments to tram .

Details

A normal model for transformed responses, where the transformation is estimated from the data simultaneously with the regression coefficients. This is similar to a Box-Cox transformation, but the technical details differ. Examples can be found in the package vignette.

Value

An object of class `BoxCox`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("BostonHousing2", package = "mlbench")

lm(cmedv ~ crim + zn + indus + chas + nox + rm + age + dis +
    rad + tax + ptratio + b + lstat, data = BostonHousing2)

BoxCox(cmedv ~ chas + crim + zn + indus + nox +
    rm + age + dis + rad + tax + ptratio + b + lstat,
    data = BostonHousing2)
```

Colr

Continuous Outcome Logistic Regression

Description

A proportional-odds model for continuous variables

Usage

```
Colr(formula, data, subset, weights, offset, cluster, na.action = na.omit, ...)
```

Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under tram and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	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.
...	additional arguments to tram .

Details

Simultaneous estimation of all possible binary logistic models obtained by dichotomisation of a continuous response. The regression coefficients can be constant allowing for an interpretation as log-odds ratios.

Value

An object of class `Colr`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

Tina Lohse, Sabine Rohrmann, David Faeh and Torsten Hothorn (2017), Continuous Outcome Logistic Regression for Analyzing Body Mass Index Distributions, *F1000Research*, **6**(1933), doi: [10.12688/f1000research.12934.1](https://doi.org/10.12688/f1000research.12934.1).

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("BostonHousing2", package = "mlbench")

lm(cmedv ~ crim + zn + indus + chas + nox + rm + age + dis +
    rad + tax + ptratio + b + lstat, data = BostonHousing2)

Colr(cmedv ~ chas + crim + zn + indus + nox +
    rm + age + dis + rad + tax + ptratio + b + lstat,
    data = BostonHousing2)
```

Coxph

Cox Proportional Hazards Model

Description

Cox model with fully parameterised baseline hazard function

Usage

```
Coxph(formula, data, subset, weights, offset, cluster, na.action = na.omit, ...)
```

Arguments

`formula` an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under `tram` and in the package vignette.

<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> .
<code>subset</code>	an optional vector specifying a subset of observations to be used in the fitting process.
<code>weights</code>	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If present, the weighted log-likelihood is maximised.
<code>offset</code>	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector of length equal to the number of cases.
<code>cluster</code>	optional factor with a cluster ID employed for computing clustered covariances.
<code>na.action</code>	a function which indicates what should happen when the data contain NAs. The default is set to <code>na.omit</code> .
<code>...</code>	additional arguments to <code>tram</code> .

Details

The original implementation of Cox models via the partial likelihood, treating the baseline hazard function as a nuisance parameter, is available in `coxph`. This function allows simultaneous estimation of the log-hazard ratios and the log-cumulative baseline hazard, the latter parameterised by a Bernstein polynomial. The model can be fitted under stratification (time-varying coefficients), all types of random censoring and truncation. An early reference to this parameterisation is McLain and Ghosh (2013).

The responses is bounded (`bounds = c(0, Inf)`) when specified as a `Surv` object. Otherwise, bounds can be specified via `...`

Value

An object of class `Coxph`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

- Alexander C. McLain and Sujit K. Ghosh (2013). Efficient Sieve Maximum Likelihood Estimation of Time-Transformation Models, *Journal of Statistical Theory and Practice*, **7**(2), 285–303, doi: [10.1080/15598608.2013.772835](https://doi.org/10.1080/15598608.2013.772835).
- Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("GBSG2", package = "TH.data")

library("survival")
coxph(Surv(time, cens) ~ horTh, data = GBSG2)
```

```
Coxph(Surv(time, cens) ~ horTh, data = GBSG2)
```

Lm

*Normal Linear Model***Description**

Normal linear model with benefits

Usage

```
Lm(formula, data, subset, weights, offset, cluster, na.action = na.omit, ...)
```

Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under tram and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	a function which indicates what should happen when the data contain NAs. The default is set to <code>na.omit</code> .
...	additional arguments to tram .

Details

A normal linear model with simultaneous estimation of regression coefficients and scale parameter(s). This function also allows for stratum-specific intercepts and variances as well as censoring and truncation in the response.

Note that the scale of the parameters is different from what is reported by [lm](#); the discrepancies are explained in the package vignette.

Value

An object of class `BoxCox`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("BostonHousing2", package = "mlbench")

lm(cmedv ~ crim + zn + indus + chas + nox + rm + age + dis +
    rad + tax + ptratio + b + lstat, data = BostonHousing2)

Lm(cmedv ~ chas + crim + zn + indus + nox +
    rm + age + dis + rad + tax + ptratio + b + lstat,
    data = BostonHousing2)
```

Polr

Ordered Categorical Regression

Description

Some regression models for ordered categorical responses

Usage

```
Polr(formula, data, subset, weights, offset, cluster, na.action = na.omit,
      method = c("logistic", "probit", "cloglog"), ...)
```

Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under tram and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.

<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>method</code>	a character describing the link function.
<code>...</code>	additional arguments to <code>tram</code> .

Details

Models for ordered categorical responses reusing the interface of `polr`. Allows for stratification, censoring and truncation.

Value

An object of class `Polr`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("wine", package = "ordinal")

library("MASS")
polr(rating ~ temp + contact, data = wine)

Polr(rating ~ temp + contact, data = wine)
```

Survreg

Parametric Survival Models

Description

Weibull, log-normal, log-logistic and other parametric models (not exclusively) for survival analysis

Usage

```
Survreg(formula, data, subset, weights, offset, cluster, na.action = na.omit,
         dist = c("weibull", "logistic", "gaussian", "exponential", "rayleigh",
                 "loggaussian", "lognormal", "loglogistic"), scale = 0, ...)
```


Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under tram and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	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.
dist	character defining the conditional distribution of the (not necessarily positive) response, current choices include Weibull, logistic, normal, exponential, Rayleigh, log-normal (same as log-gaussian), or log-logistic.
scale	a fixed value for the scale parameter(s).
...	additional arguments to tram .

Details

Parametric survival models reusing the interface of [survreg](#). The parameterisation is, however, a little different, see the package vignette.

Value

An object of class `Survreg`, with corresponding `coef`, `vcov`, `logLik`, `estfun`, `summary`, `print`, `plot` and `predict` methods.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Examples

```
data("GBSG2", package = "TH.data")

library("survival")
survreg(Surv(time, cens) ~ horTh, data = GBSG2)
```

```
Survreg(Surv(time, cens) ~ horTh, data = GBSG2)
```

tram

Stratified Linear Transformation Models

Description

Likelihood-inference for stratified linear transformation models

Usage

```
tram(formula, data, subset, weights, offset, cluster, na.action = na.omit,
     distribution = c("Normal", "Logistic", "MinExtrVal"),
     transformation = c("discrete", "linear", "logarithmic", "smooth"),
     LRtest = TRUE, prob = c(0.1, 0.9), support = NULL,
     bounds = NULL, add = c(0, 0), order = 6,
     negative = TRUE, scale = TRUE, model_only = FALSE, ...)
tram_data(formula, data, subset, weights, offset, cluster, na.action = na.omit)
```

Arguments

formula	an object of class "formula": a symbolic description of the model structure to be fitted. The details of model specification are given under Details and in the package vignette.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> .
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> or a numeric vector. If present, the weighted log-likelihood is maximised.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be <code>NULL</code> or a numeric vector of length equal to the number of cases.
cluster	optional factor with a cluster ID employed for computing clustered covariances.
na.action	a function which indicates what should happen when the data contain NAs. The default is set to <code>na.omit</code> .
distribution	character specifying how the transformation function is mapped into probabilities. Available choices include the cumulative distribution functions of the standard normal, the standard logistic and the standard minimum extreme value distribution.

transformation	character specifying the complexity of the response-transformation. For discrete responses, one parameter is assigned to each level (except the last one), for continuous responses linear, log-linear and smooth (parameterised as a Bernstein polynomial) function are implemented.
LRtest	logical specifying if a likelihood-ratio test for the null of all coefficients in the linear predictor being zero shall be performed.
prob	two probabilities giving quantiles of the response defining the support of a smooth Bernstein polynomial (if transformation = "smooth").
support	a vector of two elements; the support of a smooth Bernstein polynomial (if transformation = "smooth").
bounds	an interval defining the bounds of a real sample space.
add	add these values to the support before generating a grid via <code>mkgrid</code> .
order	integer ≥ 1 defining the order of the Bernstein polynomial (if transformation = "smooth").
negative	logical defining the sign of the linear predictor.
scale	logical defining if variables in the linear predictor shall be scaled. Scaling is internally used for model estimation, rescaled coefficients are reported in model output.
model_only	logical, if TRUE the unfitted model is returned.
...	additional arguments.

Details

The model formula is of the form $y \mid s \sim x$ where y is an at least ordered response variable, s are the variables defining strata and x defines the linear predictor. $y \sim x$ defines a model without strata (but response-varying intercept function) and $y \mid s \sim \emptyset$ sets-up response-varying coefficients for all variables in s .

The two functions `tram` and `tram_data` are not intended to be called directly by users. Instead, functions `Coxph` (Cox proportional hazards models), `Survreg` (parametric survival models), `Polr` (models for ordered categorical responses), `Lm` (normal linear models), `BoxCox` (non-normal linear models) or `Colr` (continuous outcome logistic regression) allow direct access to the corresponding models.

The model class and the specific models implemented in **tram** are explained in the package vignette of package **tram**. The underlying theory of most likely transformations is presented in Hothorn et al. (2018), computational and modelling aspects in more complex situations are discussed by Hothorn (2018).

Value

An object of class `tram` inheriting from `mlt`.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

Torsten Hothorn (2018), Most Likely Transformations: The `mlt` Package, *Journal of Statistical Software*, forthcoming. URL: <https://cran.r-project.org/package=mlt.docreg>

Description

Methods for objects inheriting from class tram

Usage

```
## S3 method for class 'tram'
as.mlt(object)
## S3 method for class 'tram'
model.frame(formula, ...)
## S3 method for class 'tram'
model.matrix(object, with_baseline = FALSE, ...)
## S3 method for class 'tram'
coef(object, with_baseline = FALSE, ...)
## S3 method for class 'Lm'
coef(object, as.lm = FALSE, ...)
## S3 method for class 'Survreg'
coef(object, as.survreg = FALSE, ...)
## S3 method for class 'tram'
vcov(object, with_baseline = FALSE, ...)
## S3 method for class 'tram'
logLik(object, parm = coef(as.mlt(object), fixed = FALSE), ...)
## S3 method for class 'tram'
estfun(object, parm = coef(as.mlt(object), fixed = FALSE), ...)
## S3 method for class 'tram'
predict(object, newdata = model.frame(object),
        type = c("lp", "trafo", "distribution", "survivor", "density",
                "logdensity", "hazard", "loghazard", "cumhazard", "quantile"),
        ...)
## S3 method for class 'tram'
plot(x, newdata = model.frame(x),
     which = c("QQ-PIT", "baseline only", "distribution"),
     confidence = c("none", "interval", "band"), level = 0.95,
     K = 50, cheat = K, col = "black", fill = "lightgrey", lwd = 1, ...)
```

Arguments

`object`, `formula`, `x`
a fitted stratified linear transformation model inheriting from class tram.

`with_baseline` logical, if TRUE all model parameters are returned, otherwise parameters describing the baseline transformation are ignored.

`as.lm` logical, return parameters in the `lm` parameterisation if TRUE.

`as.survreg` logical, return parameters in the `survreg` parameterisation if TRUE.

parm	model parameters, including baseline parameters.
newdata	an optional data frame of new observations.
type	type of prediction, current options include linear predictors ("lp", of x variables in the formula $y s \sim x$), transformation functions ("trafo") or distribution functions on the scale of the cdf ("distribution"), survivor function, density function, log-density function, hazard function, log-hazard function, cumulative hazard function or quantile function.
which	type of plot, either a QQ plot of the probability-integral transformed observations ("QQ-PIT"), of the baseline transformation of the whole distribution.
confidence	type of uncertainty assessment.
level	confidence level.
K	number of grid points in the response, see plot.ctm .
cheat	reduced number of grid points for the computation of confidence bands, see confband .
col	line color.
fill	fill color.
lwd	line width.
...	additional arguments to the underlying methods for class <code>mlt</code> , see mlt-methods .

Details

`coef` can be used to get (and set) model parameters, `logLik` evaluates the log-likelihood (also for parameters other than the maximum likelihood estimate); `vcov` returns the estimated variance-covariance matrix (possibly taking cluster into account) and `estfun` gives the score contribution by each observation. `predict` and `plot` can be used to inspect the model on different scales.

References

Torsten Hothorn, Lisa Moest, Peter Buehlmann (2018), Most Likely Transformations, *Scandinavian Journal of Statistics*, **45**(1), 110–134, doi: [10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291).

See Also

[mlt-methods](#), [plot.ctm](#)

Examples

```
data("BostonHousing2", package = "mlbench")

### fit non-normal Box-Cox type linear model with two
### baseline functions (for houses near and off Charles River)
BC_BH_2 <- BoxCox(cmedv | 0 + chas ~ crim + zn + indus + nox +
                 rm + age + dis + rad + tax + ptratio + b + lstat,
                 data = BostonHousing2)

logLik(BC_BH_2)
```

```
### classical likelihood inference
summary(BC_BH_2)

### coefficients of the linear predictor
coef(BC_BH_2)

### plot linear predictor (mean of _transformed_ response)
### vs. observed values
plot(predict(BC_BH_2, type = "lp"), BostonHousing2$cmedv)

### all coefficients
coef(BC_BH_2, with_baseline = TRUE)

### compute predicted median along with 10% and 90% quantile for the first
### observations
predict(BC_BH_2, newdata = BostonHousing2[1:3,], type = "quantile",
       prob = c(.1, .5, .9))

### plot the predicted density for these observations
plot(BC_BH_2, newdata = BostonHousing2[1:3, -1],
     which = "distribution", type = "density", K = 1000)

### evaluate the two baseline transformations, with confidence intervals
nd <- model.frame(BC_BH_2)[1:2, -1]
nd$chas <- factor(c("0", "1"))
library("colorspace")
col <- diverge_hcl(2, h = c(246, 40), c = 96, l = c(65, 90))
fill <- diverge_hcl(2, h = c(246, 40), c = 96, l = c(65, 90), alpha = .3)
plot(BC_BH_2, which = "baseline only", newdata = nd, col = col,
     confidence = "interval", fill = fill, lwd = 2,
     xlab = "Median Value", ylab = expression(h[Y]))
legend("bottomright", lty = 1, col = col,
     title = "Near Charles River", legend = c("no", "yes"), bty = "n")
```

Index

*Topic **models**

BoxCox, 2
Colr, 3
Coxph, 4
Lm, 6
Polr, 7
Survreg, 8
tram, 10

*Topic **regression**

BoxCox, 2
Colr, 3
Coxph, 4
Lm, 6
Polr, 7
Survreg, 8
tram, 10

*Topic **smooth**

BoxCox, 2
Colr, 3
Coxph, 4
tram, 10

*Topic **survival**

Coxph, 4
Survreg, 8
tram, 10

as.mlt.tram (tram-methods), 12

BoxCox, 2, 11

coef.Lm (tram-methods), 12

coef.Survreg (tram-methods), 12

coef.tram (tram-methods), 12

Colr, 3, 11

confband, 13

Coxph, 4, 11

coxph, 5

estfun.tram (tram-methods), 12

Lm, 6, 11

lm, 6, 12

logLik.tram (tram-methods), 12

mkgrid, 11

model.frame.tram (tram-methods), 12

model.matrix.tram (tram-methods), 12

plot.ctm, 13

plot.tram (tram-methods), 12

Polr, 7, 11

polr, 8

predict.tram (tram-methods), 12

Survreg, 8, 11

survreg, 9, 12

tram, 2–9, 10

tram-methods, 12

tram_data (tram), 10

vcov.tram (tram-methods), 12