

# Package ‘islasso’

May 16, 2023

**Type** Package

**Title** The Induced Smoothed Lasso

**Version** 1.5.1

**Date** 2023-05-12

**Depends** glmnet (>= 4.0), Matrix (>= 1.0-6), R (>= 4.0.0)

**Imports** stats, utils, graphics

**Suggests** knitr, lars, xfun, rmarkdown

## Description

An implementation of the induced smoothing (IS) idea to lasso regularization models to allow estimation and inference on the model coefficients (currently hypothesis testing only). Linear, logistic, Poisson and gamma regressions with several link functions are implemented. The algorithm is described in the original paper; see <doi:10.1177/0962280219842890> and discussed in a tutorial <doi:10.13140/RG.2.2.16360.11521>.

**License** GPL (>= 2)

**Encoding** UTF-8

**NeedsCompilation** yes

**Author** Gianluca Sottile [aut, cre],  
Giovanna Cilluffo [aut, ctb],  
Vito MR Muggeo [aut, ctb]

**Maintainer** Gianluca Sottile <gianluca.sottile@unipa.it>

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2023-05-15 23:40:02 UTC

## R topics documented:

islasso-package . . . . .	2
aic.islasso . . . . .	3
anova.islasso . . . . .	5
breast . . . . .	7
confint.islasso . . . . .	8

diabetes . . . . .	9
GoF.islasso.path . . . . .	10
is.control . . . . .	11
islasso . . . . .	12
islasso.path . . . . .	16
plot.islasso . . . . .	20
plot.islasso.path . . . . .	21
predict.islasso . . . . .	22
predict.islasso.path . . . . .	24
Prostate . . . . .	25
simulXy . . . . .	26
summary.islasso . . . . .	27
summary.islasso.path . . . . .	28

<b>Index</b>	<b>30</b>
--------------	-----------

---

islasso-package	<i>The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression</i>
-----------------	--

---

## Description

This package implements an induced smoothed approach for hypothesis testing in lasso regression.

## Details

Package: islasso  
 Type: Package  
 Version: 1.5.1  
 Date: 2023-05-12  
 License: GPL-2

[islasso](#) is used to fit generalized linear models with a L1-penalty on (some) regression coefficients. Along with point estimates, the main advantage is to return the full covariance matrix of estimate. The resulting standard errors can be used to make inference in the lasso framework. The main function is [islasso](#) and the corresponding fitter function [islasso.fit](#), and many auxiliary functions are implemented to summarize and visualize results: [summary.islasso](#), [predict.islasso](#), [logLik.islasso](#), [deviance.islasso](#), [residuals.islasso](#).

[islasso.path](#) is used to fit a generalized linear model via the Induced Smoothed Lasso. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda. Along with coefficients profile, the main advantage is to return also the standard errors profile. The resulting standard errors can be used to make inference in the lasso framework. The main function is [islasso.path](#) and the corresponding fitter function [islasso.path.fit](#), and many auxiliary functions are implemented to summarize and visualize results: [summary.islasso.path](#), [predict.islasso.path](#), [logLik.islasso.path](#), [deviance.islasso.path](#), [residuals.islasso.path](#), [coef.islasso.path](#), [fitted.islasso.path](#).

**Author(s)**

Gianluca Sottile based on some preliminary functions by Vito Muggeo.

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**References**

Cilluffo, G, Sottile, G, S, La Grutta, S and Muggeo, VMR (2019). *The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression*. Statistical Methods in Medical Research, DOI: 10.1177/0962280219842890.

Sottile, G, Cilluffo, G, Muggeo, VMR (2019). *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report on ResearchGate. doi:10.13140/RG.2.2.16360.11521.

**Examples**

```
set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = gaussian())
temp <- GoF.islasso.path(o)

lambda <- temp$lambda.min["BIC"]
o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = gaussian(), lambda = lambda)
o
summary(o, pval = .05)
```

**Description**

This function performs a minimization of the AIC/BIC criterion for selecting the tuning parameter in “islasso”.

**Usage**

```
aic.islasso(object, method = c("AIC", "BIC", "AICc", "GCV", "GIC"),
  interval, g = 0, y, X,
  intercept = FALSE, family = gaussian(), alpha = 1, offset,
  weights, unpenalized, control = is.control(), trace = TRUE)
```

**Arguments**

object	a fitted model object of class "islasso".
method	the criterion to optimize, AIC, BIC, AICc, GCV, GIC.
interval	the lower and upper limits of $\lambda$ wherein the AIC/BIC criterion should be optimized. Can be missing, if object has been obtained via cross-validation (and therefore includes the range of lambdas)
g	a value belonging to the interval [0, 1]. Classical BIC is returned by letting $g = 0$ (default value), whereas extended BIC corresponds to the case $g = 0.5$ .
y	if object is missing, the response vector of length n.
X	if object is missing, the design matrix of dimension $n * p$ .
intercept	if object is missing, if TRUE the intercept is added to the model matrix.
family	if object is missing, a description of the error distribution, family=gaussian, family=binomial and family=poisson are implemented with canonical link.
alpha	The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$ . The penalty is defined as $(1 - \alpha)/2 \ \beta\ _2^2 + \alpha \ \beta\ _1.$ alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.
offset	this can be used to specify an a priori 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.
weights	observation weights. Default is 1 for each observation.
unpenalized	a vector used to specify the unpenalized estimators; unpenalized has to be a vector of logicals.
control	a list of parameters for controlling the fitting process (see islasso.control for more details).
trace	Should the iterative procedure be printed? TRUE is the default value.

**Details**

Minimization of the Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC) or several other criteria are sometimes employed to select the tuning parameter as an alternative to the cross validation. The model degrees of freedom (not necessarily integers as in the plain lasso) used in all methods are computed as trace of the hat matrix at convergence.

**Value**

the optimal lambda value is returned

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#) and [deviance.islasso](#) methods.

**Examples**

```
set.seed(1)
n <- 100
p <- 100
p1 <- 20 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

o <- islasso(y ~ ., data = data.frame(y = y, X))

## Not run:
#use the evaluation interval of the fit
lambda_aic <- aic.islasso(o, method = "AIC")

#overwrites the evaluation interval for lambda
lambda_bic <- aic.islasso(o, interval = c(.1, 30), method = "BIC")

#overwrites the evaluation interval for lambda using eBIC criterion
lambda_ebic <- aic.islasso(o, interval = c(.1, 30), method = "BIC", g = .5)

## End(Not run)
```

---

anova.islasso

*General Linear Combination method for islasso objects*


---

**Description**

General linear hypotheses and confidence intervals estimation for linear combinations of the regression coefficients in islasso fits

**Usage**

```
## S3 method for class 'islasso'
anova(object, A, b = NULL, ci, ...)
```

**Arguments**

object	a fitted model object of class "islasso".
A	matrix (or vector) giving linear combinations of coefficients by rows, or a character vector giving the hypothesis in symbolic form (see Details).
b	right-hand-side vector for hypothesis, with as many entries as rows in the hypothesis matrix A; can be omitted, in which case it defaults to a vector of zeroes.
ci	optionally, a two columns matrix of estimated confidence intervals for the estimated coefficients.
...	not used.

**Details**

For the islasso regression model with coefficients  $\beta$ , the null hypothesis is

$$H_0 : A\beta = b$$

where A and b are known matrix and vector.

The hypothesis matrix A can be supplied as a numeric matrix (or vector), the rows of which specify linear combinations of the model coefficients, which are tested equal to the corresponding entries in the right-hand-side vector b, which defaults to a vector of zeroes.

Alternatively, the hypothesis can be specified symbolically as a character vector with one or more elements, each of which gives either a linear combination of coefficients, or a linear equation in the coefficients (i.e., with both a left and right side separated by an equals sign). Components of a linear expression or linear equation can consist of numeric constants, or numeric constants multiplying coefficient names (in which case the number precedes the coefficient, and may be separated from it by spaces or an asterisk); constants of 1 or -1 may be omitted. Spaces are always optional. Components are separated by plus or minus signs. Newlines or tabs in hypotheses will be treated as spaces. See the examples below.

**Value**

An object of class "anova.islasso" which contains the estimates, the standard errors, the Wald statistics and corresponding p value of each linear combination and of the restricted model.

**Author(s)**

The main function of the same name was inspired by the R function previously implemented by Vito MR Muggeo.

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**Examples**

```
set.seed(1)
n <- 100
p <- 100
p1 <- 10 #number of nonzero coefficients
coef.true <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
```

```

coef <- c(coef.true, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X %*% coef)
mu <- eta
y <- mu + rnorm(n, 0, sigma)

o <- islasso(y ~ . - 1, data = data.frame(y = y, X),
            family = gaussian())
anova(o, A = diag(p), b = coef)
anova(o, A = c("X1 + X2 + X3 + X4 + X5 = -7.5"))
anova(o, A = c("X1 + X2 + X3 + X4 + X5 = 0"))
anova(o, A = c("X6 + X7 + X8 + X9 + X10"), b = 8.75)
anova(o, A = c("X6 + X7 + X8 + X9 + X10"), b = 0)
anova(o, A = c("X1 + X2 + X3 + X4 + X5 = -7.5",
              "X6 + X7 + X8 + X9 + X10 = 8.75"))
anova(o, A = c("X1 + X2 + X3 + X4 + X5",
              "X6 + X7 + X8 + X9 + X10"), b = c(-7.5, 8.75))
anova(o, A = c("X1 + X2 + X3 + X4 + X5",
              "X6 + X7 + X8 + X9 + X10"))

```

---

breast

*Breast Cancer microarray experiment*


---

## Description

This data set details microarray experiment for 52 breast cancer patients. The binary variable `status` is used to indicate whether or not the patient has died of breast cancer (`status = 0` = did not die of breast cancer, `status = 1` = died of breast cancer). The other variables contain the amplification or deletion of the considered genes.

Rather than measuring gene expression, this experiment aims to measure gene amplification or deletion, which refers to the number of copies of a particular DNA sequence within the genome. The aim of the experiment is to find out the key genomic factors involved in aggressive and non-aggressive forms of breast cancer.

The experiment was conducted by the Dr. John Bartlett and Dr. Caroline Witton in the Division of Cancer Sciences and Molecular Pathology of the University of Glasgow at the city's Royal Infirmary.

## Usage

```
data(breast)
```

## Source

Dr. John Bartlett and Dr. Caroline Witton, Division of Cancer Sciences and Molecular Pathology, University of Glasgow, Glasgow Royal Infirmary.

## References

Augugliaro L., Mineo A.M. and Wit E.C. (2013) *dgLARS: a differential geometric approach to sparse generalized linear models*, *Journal of the Royal Statistical Society. Series B.*, Vol 75(3), 471-498.

Wit E.C. and McClure J. (2004) "Statistics for Microarrays: Design, Analysis and Inference" Chichester: Wiley.

---

confint.islasso      *confint method for islasso objects*

---

## Description

confint method for islasso objects

## Usage

```
## S3 method for class 'islasso'
confint(object, parm, level = 0.95,
        type.ci = "wald", trace = TRUE, ...)
```

## Arguments

object	a fitted model object of class "islasso".
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
type.ci	Only Wald-type confidence intervals are implemented yet! type.ci = "wald" estimates and standard errors are used to build confidence interval
trace	if TRUE (default) a bar shows the iterations status.
...	additional argument(s) for methods.

## Author(s)

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

## See Also

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#) and [deviance.islasso](#) methods.



**Examples**

```

set.seed(1)
n <- 100
p <- 100
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X%%coef)

##### gaussian #####
mu <- eta
y <- mu + rnorm(n, 0, sigma)

o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = gaussian())

ci <- confint(o, type.ci = "wald", parm = 1:10)
ci
plot(ci)

```

---

diabetes

*Blood and other measurements in diabetics*


---

**Description**

The diabetes data frame has 442 rows and 3 columns. These are the data used in the Efron et al "Least Angle Regression" paper.

**Format**

This data frame contains the following columns:

- x** a matrix with 10 columns
- y** a numeric vector
- x2** a matrix with 64 columns

**Details**

The x matrix has been standardized to have unit L2 norm in each column and zero mean. The matrix x2 consists of x plus certain interactions.

**Source**

[https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle\\_2002.ps](https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.ps)

**References**

Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) *Annals of Statistics*

---

GoF.islasso.path      *Optimization for the selection of the tuning parameter*

---

**Description**

This function extracts the value of the tuning parameter which minimizes the AIC/BIC/AICc/eBIC/GCV/GIC criterion in "islasso.path".

**Usage**

```
GoF.islasso.path(object, plot = TRUE, ...)
```

**Arguments**

object	a fitted model object of class "islasso.path".
plot	a logical flag indicating if each criterion have to be plotted
...	further arguments passed to or from other methods.

**Details**

Minimization of the Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC) or several other criteria are sometimes employed to select the tuning parameter as an alternative to the cross validation. The model degrees of freedom (not necessarily integers as in the plain lasso) used in all methods are computed as trace of the hat matrix at convergence.

**Value**

A list of

gof	the goodness of fit measures
minimum	the position of the optimal lambda values
lambda.min	the optimal lambda values

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.path](#), [islasso.path.fit](#), [coef.islasso.path](#), [residuals.islasso.path](#), [summary.islasso.path](#), [logLik.islasso.path](#), [fitted.islasso.path](#), [predict.islasso.path](#) and [deviance.islasso.path](#) methods.

**Examples**

```

set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

o <- islasso.path(y ~ ., data = data.frame(y = y, X))
GoF.islasso.path(o)

```

---

is.control

*Auxiliary for controlling islasso model fitting*


---

**Description**

Auxiliary function for controlling the islasso model fitting.

**Usage**

```

is.control(sigma2 = -1, tol = 1E-05, itmax = 1E+3, stand = TRUE,
  trace = 0, nfolds = 5, seed = NULL, adaptive = FALSE, g = .5,
  b0 = NULL, V0 = NULL, c = .5)

```

**Arguments**

sigma2	optional. The fixed value of dispersion parameter. If -1 (default) it is estimated from the data
tol	tolerance value to declare convergence, default to 1e-5
itmax	maximum number of iterations, default to 1000
stand	if TRUE (default), the covariates are standardized prior to fitting the model. However the coefficients are always returned on the original scale.
trace	Should the iterative procedure be printed? 0: no printing, 1 = compact printing, 2 = enlarged printing, 3 = compact printing including Fisher scoring information (only used in glm family).
nfolds	if lambda is unspecified in islasso, the number of folds to be used to perform cross validation. Default to 5, and nfolds>2 is allowed. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. nfolds is ignored if lambda is supplied.

seed	optional, the seed to be used to split the dataframe and to perform cross validation. Useful to make reproducible the results.
adaptive	experimental, if TRUE the adaptive LASSO is implemented.
g	a value belonging to the interval [0, 1]. Classical BIC is returned by letting g = 0 (default value), whereas extended BIC corresponds to the case g = 0.5.
b0	optional, starting values for the regression coefficients. If NULL, the point estimates from <code>glmnet</code> are used.
V0	optional, starting value for the estimates covariance matrix, If NULL, the identity matrix is used.
c	the weight of the mixture in the induced smoothed lasso, the default is c = .5. c = -1 means to compute it at each step of the iterative algorithm.

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

---

islasso	<i>The Induced Smoothed lasso</i>
---------	-----------------------------------

---

**Description**

islasso is used to fit lasso regression models wherein the nonsmooth  $L_1$  norm penalty is replaced by a smooth approximation justified under the induced smoothing paradigm. Simple lasso-type or elastic-net penalties are permitted and Linear, Logistic, Poisson and Gamma responses are allowed.

**Usage**

```
islasso(formula, family = gaussian, lambda, alpha = 1, data, weights, subset,
        offset, unpenalized, contrasts = NULL, control = is.control())
```

**Arguments**

formula	an object of class “formula” (or one that can be coerced to that class): the ‘usual’ symbolic description of the model to be fitted.
family	the assumed response distribution. Gaussian, (quasi) Binomial, (quasi) Poisson, and Gamma are allowed. <code>family=gaussian</code> is implemented with identity link, <code>family=binomial</code> is implemented with logit or probit links, <code>family=poisson</code> is implemented with log link, and <code>family=Gamma</code> is implemented with inverse, log and identity links.
lambda	Value of the tuning parameter in the objective. If missing, the optimal lambda is computed using <code>cv.glmnet</code> .
alpha	The elastic-net mixing parameter, with $0 \leq \alpha \leq 1$ . The penalty is defined as

$$(1 - \alpha)/2 \|\beta\|_2^2 + \alpha \|\beta\|_1.$$

alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.

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> , typically the environment from which <code>islasso</code> is called.
weights	observation weights. Default is 1 for each observation.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
offset	this can be used to specify an a priori 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.
unpenalized	optional. A vector of integers or characters indicating any covariate (in the formula) with coefficients not to be penalized. The intercept, if included in the model, is <i>always</i> unpenalized.
contrasts	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .
control	a list of parameters for controlling the fitting process (see <code>islasso.control</code> for more details).

### Details

`islasso` estimates regression models by imposing a lasso-type penalty on some or all regression coefficients. However the nonsmooth  $L_1$  norm penalty is replaced by a smooth approximation justified under the induced smoothing paradigm. The advantage is that reliable standard errors are returned as model output and hypothesis testing on linear combinations of the regression parameters can be carried out straightforwardly via the Wald statistic. Simulation studies provide evidence that the proposed approach controls type-I errors and exhibits good power in different scenarios.

### Value

A list of

coefficients	a named vector of coefficients
se	a named vector of standard errors
residuals	the working residuals
fitted.values	the fitted values
rank	the estimated degrees of freedom
family	the family object used
linear.predictors	the linear predictors
deviance	the family deviance
aic	the Akaike Information Criterion
null.deviance	the family null deviance
iter	the number of iterations of IWLS used
weights	the working weights, that is the weights in the final iteration of the IWLS fit
df.residual	the residual degrees of freedom

df.null	the degrees of freedom of a null model
converged	logical. Was the IWLS algorithm judged to have converged?
model	if requested (the default), the model frame used.
call	the matched call
formula	the formula supplied
terms	the terms object used
data	the data argument.
offset	the offset vector used.
control	the value of the control argument used
xlevels	(where relevant) a record of the levels of the factors used in fitting.
lambda	the lambda value used in the islasso algorithm
alpha	the elasticnet mixing parameter
dispersion	the estimated dispersion parameter
internal	internal elements
contrasts	(only where relevant) the contrasts used.

### Author(s)

The main function of the same name was inspired by the R function previously implemented by Vito MR Muggeo.

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

### References

Cilluffo, G, Sottile, G, S, La Grutta, S and Muggeo, VMR (2019). *The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression*. Statistical Methods in Medical Research, DOI: 10.1177/0962280219842890.

Sottile, G, Cilluffo, G, Muggeo, VMR (2019). *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report on ResearchGate. doi:10.13140/RG.2.2.16360.11521.

### See Also

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#) and [deviance.islasso](#) methods.

### Examples

```
set.seed(1)
n <- 100
p <- 100
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
```

```

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X%%coef)

##### gaussian #####
mu <- eta
y <- mu + rnorm(n, 0, sigma)

o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = gaussian())
o
summary(o)
coef(o)
fitted(o)
predict(o, type="response")
plot(o)
residuals(o)
deviance(o)
AIC(o)
logLik(o)

## Not run:
# for the interaction
o <- islasso(y ~ X1 * X2, data = data.frame(y = y, X),
            family = gaussian())

##### binomial #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- binomial()$linkinv(eta)
y <- rbinom(n, 100, mu)
y <- cbind(y, 100-y)

o <- islasso(cbind(y1, y2) ~ .,
            data = data.frame(y1 = y[,1], y2 = y[,2], X),
            family = binomial())
summary(o, pval = .05)

##### poisson #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(1, coef))
mu <- poisson()$linkinv(eta)
y <- rpois(n, mu)

o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = poisson())
summary(o, pval = .05)

##### Gamma #####
coef <- c(c(1,1,1), rep(0, p-3))

```

```

X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- Gamma(link="log")$linkinv(eta)
shape <- 10
phi <- 1 / shape
y <- rgamma(n, scale = mu / shape, shape = shape)

o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = Gamma(link = "log"))
summary(o, pval = .05)

## End(Not run)

```

---

islasso.path

*The Induced Smoothed lasso path*


---

### Description

islasso.path is used to fit a generalized linear model via induced smoothed lasso method. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda. Fits linear, logistic, poisson and gamma regression models.

### Usage

```

islasso.path(formula, family = gaussian(), lambda = NULL, nlambdas = 100,
             lambda.min.ratio = ifelse(nobs < nvars, 1E-2, 1E-03), alpha = 1, data,
             weights, subset, offset, contrasts = NULL, unpenalized, control = is.control())

```

### Arguments

formula	an object of class “formula” (or one that can be coerced to that class): the ‘usual’ symbolic description of the model to be fitted.
family	the assumed response distribution. Gaussian, (quasi) Binomial, (quasi) Poisson, and Gamma are allowed. family=gaussian is implemented with identity link, family=binomial is implemented with logit or probit links, family=poisson is implemented with log link, and family=Gamma is implemented with inverse, log and identity links.
lambda	A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambdas and lambda.min.ratio. Supplying a value of lambda overrides this.
nlambdas	The number of lambda values - default is 100.
lambda.min.ratio	Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size nobs relative to the number of variables nvars. If nobs > nvars, the default is 0.00001, close to zero. If nobs < nvars, the default is 0.001. A very small value of lambda.min.ratio will lead to a saturated fit in the nobs < nvars case.



alpha	The elastic-net mixing parameter, with $0 \leq \alpha \leq 1$ . The penalty is defined as $(1 - \alpha)/2\ \beta\ _2^2 + \alpha\ \beta\ _1.$ alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which islasso is called.
weights	observation weights. Default is 1 for each observation.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
offset	this can be used to specify an a priori 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.
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
control	a list of parameters for controlling the fitting process (see islasso.control for more details).
unpenalized	optional. A vector of integers or characters indicating any covariate (in the formula) with coefficients not to be penalized. The intercept, if included in the model, is <i>always</i> unpenalized.

## Details

The sequence of models implied by lambda is fit the islasso method. islasso estimates regression models by imposing a lasso-type penalty on some or all regression coefficients. However the nonsmooth  $L_1$  norm penalty is replaced by a smooth approximation justified under the induced smoothing paradigm. The advantage is that reliable standard errors are returned as model output and hypothesis testing on linear combinations of the regression parameters can be carried out straightforwardly via the Wald statistic. Simulation studies provide evidence that the proposed approach controls type-I errors and exhibits good power in different scenarios.

## Value

A list of	
call	the matched call.
Info	a named matrix containing information about lambda values, estimated degrees of freedom, estimated dispersion parameters, deviance, loglikelihood, number of iterations and convergence criteria.
GoF	a named matrix containing information criteria, i.e., AIC, BIC, AICc, eBIC, GCV, GIC.
Coef	a $\text{length}(\text{lambda}) \times \text{nvars}$ matrix of coefficients.
SE	a $\text{length}(\text{lambda}) \times \text{nvars}$ matrix of standard errors.
Weights	a $\text{length}(\text{lambda}) \times \text{nvars}$ matrix of the weight of the mixture in the induced smoothed lasso.

Linear.predictors	a <code>length(lambda) x nvars</code> matrix of linear predictors
Fitted.values	a <code>length(lambda) x nvars</code> matrix of fitted values
Residuals	a <code>length(lambda) x nvars</code> matrix of working residuals
Input	a named list containing several input arguments, i.e., the numbers of observations and predictors, if an intercept has to be estimated, the model matrix and the response vector, the observation weights, the offset, the family object used, The elasticnet mixing parameter and the vector used to specify the unpenalized estimators.
control	the value of the control argument used.
formula	the formula supplied.
model	if requested (the default), the model frame used.
terms	the terms object used.
data	the data argument.
xlevels	(where relevant) a record of the levels of the factors used in fitting.
contrasts	(only where relevant) the contrasts used.

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**References**

Cilluffo, G, Sottile, G, S, La Grutta, S and Muggeo, VMR (2019). *The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression*. Statistical Methods in Medical Research, DOI: 10.1177/0962280219842890.

Sottile, G, Cilluffo, G, Muggeo, VMR (2019). *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report on ResearchGate. doi:10.13140/RG.2.2.16360.11521.

**See Also**

[islasso.path.fit](#), [coef.islasso.path](#), [summary.islasso.path](#), [residuals.islasso.path](#), [GoF.islasso.path](#), [logLik.islasso.path](#), [fitted.islasso.path](#), [predict.islasso.path](#) and [deviance.islasso.path](#) methods.

**Examples**

```
set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))
```

```

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X%*%coef)

##### gaussian #####
mu <- eta
y <- mu + rnorm(n, 0, sigma)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = gaussian(), nlambda = 30L)
o
summary(o, lambda = 10)
coef(o, lambda = 10)
fitted(o, lambda = 10)
predict(o, type="response", lambda = 10)
plot(o, xvar = "coef")
residuals(o, lambda = 10)
deviance(o, lambda = 10)
logLik(o, lambda = 10)
GoF.islasso.path(o)

## Not run:
##### binomial #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- binomial()$linkinv(eta)
y <- rbinom(n, 100, mu)
y <- cbind(y, 100-y)

o <- islasso.path(cbind(y1, y2) ~ .,
                 data = data.frame(y1 = y[,1], y2 = y[,2], X),
                 family = binomial(), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])

##### poisson #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(1, coef))
mu <- poisson()$linkinv(eta)
y <- rpois(n, mu)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = poisson(), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])

##### Gamma #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- Gamma(link="log")$linkinv(eta)
shape <- 10

```

```

phi <- 1 / shape
y <- rgamma(n, scale = mu / shape, shape = shape)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = Gamma(link = "log"), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])

## End(Not run)

```

---

plot.islasso

*Diagnostics plots for Induced Smoothing Lasso Model*


---

## Description

Diagnostics plots for Induced Smoothing Lasso Model

## Usage

```

## S3 method for class 'islasso'
plot(x, ...)

```

## Arguments

x                    an object of class `islasso`, usually, a result of a call to `islasso`.  
...                    other graphical parameters for the plot

## Details

The plot on the top left is a plot of the standard deviance residuals against the fitted values. The plot on the top right is a normal QQ plot of the standardized deviance residuals. The red line is the expected line if the standardized residuals are normally distributed, i.e. it is the line with intercept 0 and slope 1. The bottom two panels are plots of link and variance functions. On the left is squared standardized Pearson residuals against the fitted values. On the right working vector against the linear predictor.

## Author(s)

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

## See Also

`islasso.fit`, `summary.islasso`, `residuals.islasso`, `logLik.islasso`, `predict.islasso` and `deviance.islasso` methods.

**Examples**

```
## Not run:
set.seed(1)
n <- 100
p <- 100
p1 <- 20 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

lambda <- 2
o <- islasso(y ~ ., data = data.frame(y = y, X),
             family = gaussian(), lambda = lambda)
plot(o)

## End(Not run)
```

---

plot.islasso.path      *plot coefficient profile from a fitted "islasso.path" object.*

---

**Description**

Produces a coefficient profile plot of the coefficient paths for a fitted "islasso.path" object.

**Usage**

```
## S3 method for class 'islasso.path'
plot(x,
     yvar = c("coefficients", "se", "gradient", "weight", "gof"),
     gof = c("none", "AIC", "BIC", "AICc", "eBIC", "GCV", "GIC"),
     label = FALSE, legend = FALSE, ...)
```

**Arguments**

x	an object of class islasso, usually, a result of a call to <a href="#">islasso.path</a> .
yvar	What is on the Y-axis. "coef" plot the log-lambda sequence against the coefficients; "se" plot the log-lambda sequence against the standard deviations; "gradient" plot the log-lambda sequence against the gradient; "weight" plot the log-lambda sequence against the mixture weight of the islasso method; "gof" plot the log-lambda sequence against the chosen criterion.
gof	the chosen criterion to highlight the active variables. "none" doesn't highlight active variables.

label	a logical flag indicating if some labels have to be added.
legend	a logical flag indicating if legend has to be shown.
...	other graphical parameters for the plot, i.e., main, xlab, ylab, xlim, ylim, lty, col, lwd, cex.axis, cex.lab, cex.main, gof_lty, gof_col and gof_lwd. The last three parameters are used to modify aspects of the legend, and of the goodness of fit measure used.

### Details

A coefficient profile plot is produced for Induced Smoothing Lasso Model path.

### Examples

```
## Not run:
set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = gaussian())
par(mfrow = c(2, 2))
plot(o, yvar = "coefficients", gof = "AICc", label = TRUE)
plot(o, yvar = "se", gof = "AICc")
plot(o, yvar = "gradient", gof = "AICc")
plot(o, yvar = "gof", gof = "AICc")

## End(Not run)
```

---

predict.islasso

*Prediction method for islasso fitted objects*

---

### Description

Prediction method for islasso fitted objects

**Usage**

```
## S3 method for class 'islasso'
predict(object, newdata = NULL,
        type = c("link", "response", "coefficients", "class", "terms"),
        se.fit = FALSE, ci = NULL, type.ci = "wald",
        level = .95, terms = NULL, na.action = na.pass, ...)
```

**Arguments**

object	a fitted object of class "islasso".
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The coefficients option returns coefficients. Type "class" applies only to "binomial" models, and produces the class label. The "terms" option returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale.
se.fit	logical switch indicating if confidence intervals are required.
ci	optionally, a two columns matrix of estimated confidence intervals for the estimated coefficients.
type.ci	Only Wald-type confidence intervals are implemented yet! type.ci = "wald" estimates and standard errors are used to build confidence interval
level	the confidence level required.
terms	with type = "terms" by default all terms are returned. A character vector specifies which terms are to be returned.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
...	further arguments passed to or from other methods.

**Value**

An object depending on the type argument

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#) and [deviance.islasso](#) methods.

**Examples**

```

set.seed(1)
n <- 100
p <- 100
p1 <- 20 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)
lambda <- 2
o <- islasso(y ~ ., data = data.frame(y = y, X), lambda = lambda)
predict(o, type = "response")

```

---

predict.islasso.path *Prediction method for islasso.path fitted objects*

---

**Description**

Prediction method for islasso fitted objects

**Usage**

```

## S3 method for class 'islasso.path'
predict(object, newdata, type = c("link", "response",
  "coefficients", "class"), lambda, ...)

```

**Arguments**

object	a fitted object of class "islasso.path".
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The coefficients option returns coefficients. Type "class" applies only to "binomial" models, and produces the class label.
lambda	Value(s) of the penalty parameter lambda at which predictions are required. Default is the entire sequence used to create the model.
...	further arguments passed to or from other methods.

**Value**

An object depending on the type argument



**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.path](#), [islasso.path.fit](#), [coef.islasso.path](#), [residuals.islasso.path](#), [GoF.islasso.path](#), [logLik.islasso.path](#), [fitted.islasso.path](#), [summary.islasso.path](#) and [deviance.islasso.path](#) methods.

**Examples**

```
## Not run:
set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%%coef)
y <- mu + rnorm(n, 0, sigma)

o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = gaussian())
temp <- GoF.islasso.path(o)
predict(o, type = "response", lambda = temp$lambda.min)

## End(Not run)
```

---

Prostate

*Prostate Cancer Data*

---

**Description**

These data come from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. It is data frame with 97 rows and 9 columns.

**Usage**

```
data(Prostate)
```

**Format**

The data frame has the following components:

```
lcavol log(cancer volume)
lweight log(prostate weight)
age age
lbph log(benign prostatic hyperplasia amount)
svi seminal vesicle invasion
lcp log(capsular penetration)
gleason Gleason score
pgg45 percentage Gleason scores 4 or 5
lpsa log(prostate specific antigen)
```

**Source**

Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A. and Yang, N. (1989) Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical prostatectomy treated patients, *Journal of Urology* **141**(5), 1076–1083.

---

 simulXy

---

*Simulate model matrix and response*


---

**Description**

Simulate model matrix and response from a specified distribution.

**Usage**

```
simulXy(n, p, interc = 0, beta, family = gaussian(), prop =
  0.1, lim.b = c(-3, 3), sigma = 1, size = 1, rho = 0,
  scale = TRUE, seed, X)
```

**Arguments**

n	number of observations.
p	total number of covariates in the model matrix.
interc	the model intercept.
beta	the vector of p coefficients in the linear predictor.
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. Only gaussian, binomial or poisson are allowed.

prop	if beta is missing, prop represent the quote of non-null coefficients out of p. The default is $0.10 p$ .
lim.b	if beta is missing, the coefficients come from uniform variates in lim.b. The default is (-3,3).
sigma	if family is 'gaussian', the standard deviation of the response. The default is 1.
size	if family is 'binomial', the number of trials to build the response vector. The default is 1.
rho	correlation value to define the variance covariance matrix to build the model matrix, i.e., $\rho^{ i-j }$ $i, j = 1, \dots, p$ and $i$ different from $j$ . The default is 0.
scale	Should the columns of the mdoel matrix be scaled? The default is TRUE
seed	optional, the seed to generate the data.
X	optional, the model matrix.

### Examples

```
n <- 100
p <- 100
beta <- c(runif(10, -3, 3), rep(0, p-10))
dat <- simulXy(n, p, beta = beta, seed=1234)
```

---

summary.islasso

*summary method for islasso fitted objects*


---

### Description

summary method for islasso fitted objects

### Usage

```
## S3 method for class 'islasso'
summary(object, pval = 1, which, use.t = FALSE,
        type.pval = "wald", ...)
```

### Arguments

object	fitted "islasso" object
pval	a threshold p-value value indicating which coefficients should be printed. If $pval = 0.10$ , say, only the variables/coefficients with $p - value \leq 0.10$ are printed. Possible unpenalized coefficients (including the intercept if in the model) are always printed, regardless of their p-value.
which	a specification of which parameters are to be given p-values. If missing, all parameters are considered.
use.t	if TRUE, the p-values are computed using the t-distribution with residual model degrees of freedom

type.pval	Only Wald-type confidence intervals are implemented yet! type.pval = "wald" (default) estimates and standard errors are used to build confidence interval
...	not used

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#) and [deviance.islasso](#) methods.

**Examples**

```
## Not run:
#continues example from ?islasso
summary(o, pval = .1) #print just the "borderline" significant coefficients

## End(Not run)
```

---

summary.islasso.path *summary method for islasso.path fitted objects*

---

**Description**

summary method for islasso.path fitted objects

**Usage**

```
## S3 method for class 'islasso.path'
summary(object, pval = 1, use.t = FALSE, lambda, ...)
```

**Arguments**

object	fitted "islasso.path" object
pval	a threshold p-value value indicating which coefficients should be printed. If $pval = 0.10$ , say, only the variables/coefficients with $p - value \leq 0.10$ are printed. Possible unpenalized coefficients (including the intercept if in the model) are always printed, regardless of their p-value.
use.t	if TRUE, the p-values are computed using the t-distribution with residual model degrees of freedom
lambda	Value of the penalty parameter lambda at which summary are required.
...	not used

**Author(s)**

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

**See Also**

[islasso.path](#), [islasso.path.fit](#), [coef.islasso.path](#), [residuals.islasso.path](#), [GoF.islasso.path](#), [logLik.islasso.path](#), [fitted.islasso.path](#), [predict.islasso.path](#) and [deviance.islasso.path](#) methods.

**Examples**

```
## Not run:  
#continues example from ?islasso.path  
summary(o, pval = .1, lambda = 5) #print just the "borderline" significant coefficients  
  
## End(Not run)
```

# Index

- \* **datasets**
  - breast, 7
  - diabetes, 9
  - Prostate, 25
- \* **models**
  - aic.islasso, 3
  - GoF.islasso.path, 10
  - islasso, 12
  - islasso.path, 16
- \* **package**
  - islasso-package, 2
- \* **regression**
  - aic.islasso, 3
  - GoF.islasso.path, 10
  - islasso, 12
  - islasso.path, 16
- aic.islasso, 3
- anova.islasso, 5
- breast, 7
- coef.islasso.path, 2, 10, 18, 25, 29
- coef.islasso.path(islasso.path), 16
- confint.islasso, 8
- cooks.distance.islasso(islasso), 12
- cv.glmnet, 12
- deviance.islasso, 2, 5, 8, 14, 20, 23, 28
- deviance.islasso(islasso), 12
- deviance.islasso.path, 2, 10, 18, 25, 29
- deviance.islasso.path(islasso.path), 16
- diabetes, 9
- extractAIC.islasso(islasso), 12
- family.islasso(islasso), 12
- fitted.islasso.path, 2, 10, 18, 25, 29
- fitted.islasso.path(islasso.path), 16
- formula.islasso(islasso), 12
- GoF.islasso.path, 10, 18, 25, 29
- influence.islasso(islasso), 12
- is.control, 11
- islasso, 2, 12, 20
- islasso-package, 2
- islasso.fit, 2, 5, 8, 14, 20, 23, 28
- islasso.path, 2, 10, 16, 21, 25, 29
- islasso.path.fit, 2, 10, 18, 25, 29
- logLik.islasso, 2, 5, 8, 14, 20, 23, 28
- logLik.islasso(islasso), 12
- logLik.islasso.path, 2, 10, 18, 25, 29
- logLik.islasso.path(islasso.path), 16
- model.frame.islasso(islasso), 12
- model.matrix.islasso(islasso), 12
- model.matrix.islasso.path(islasso.path), 16
- modelX(simulXy), 26
- nobs.islasso(islasso), 12
- plot.confint.islasso(confint.islasso), 8
- plot.islasso, 20
- plot.islasso.path, 21
- predict.islasso, 2, 5, 8, 14, 20, 22, 23, 28
- predict.islasso.path, 2, 10, 18, 24, 29
- print.anova.islasso(anova.islasso), 5
- print.confint.islasso(confint.islasso), 8
- print.islasso(islasso), 12
- print.islasso.path(islasso.path), 16
- print.logLik.islasso.path(islasso.path), 16
- print.summary.islasso(summary.islasso), 27
- print.summary.islasso.path(summary.islasso.path), 28
- Prostate, 25

residuals.islasso, [2](#), [5](#), [8](#), [14](#), [20](#), [23](#), [28](#)  
residuals.islasso (islasso), [12](#)  
residuals.islasso.path, [2](#), [10](#), [18](#), [25](#), [29](#)  
residuals.islasso.path (islasso.path),  
[16](#)  
rstandard.islasso (islasso), [12](#)  
rstudent.islasso (islasso), [12](#)

simulXy, [26](#)  
summary.islasso, [2](#), [5](#), [8](#), [14](#), [20](#), [23](#), [27](#), [28](#)  
summary.islasso.path, [2](#), [10](#), [18](#), [25](#), [28](#)

variable.names.islasso (islasso), [12](#)  
vcov.islasso (islasso), [12](#)

weights.islasso (islasso), [12](#)