

Package ‘grpss’

January 3, 2016

Type Package

Title Group Screening and Selection

Version 3.0.1

Date 2016-01-02

Author Debin Qiu, Jeongyoun Ahn

Maintainer Debin Qiu <debinqiu@uga.edu>

Description Contains the tools to screen grouped variables, and select screened grouped variables afterwards. The main function `grpss()` can perform the grouped variables screening as well as selection for ultra-high dimensional data with group structure. The screening step is primarily used to reduce the dimensions of data so that the selection procedure can easily handle the moderate or low dimensions instead of ultra-high dimensions.

License GPL-2 | GPL-3

LazyData TRUE

Imports MASS, doParallel, foreach, grpreg

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-03 22:28:01

R topics documented:

<code>distcor</code>	2
<code>grp.criValues</code>	3
<code>grpss</code>	5
<code>importance</code>	9
<code>norm_vec</code>	10
<code>predict.cv.grpreg</code>	11
<code>predict.grpreg</code>	12
<code>print.grpss</code>	14
<code>summary.cv.grpreg</code>	15
<code>summary.grpreg</code>	16

Index	18
--------------	-----------

`distcor`*Compute the distance correlation*

Description

Computes the distance correlation between two random variables.

Usage

```
distcor(X, y)
```

Arguments

<code>X</code>	A numeric vector or matrix.
<code>y</code>	A numeric vector or matrix.

Details

Distance correlation measures the statistical dependence between two random variables or two random vectors. The important property is that the distance correlation is zero if and only if two random variables are independent. The details of computing the distance correlation can be seen in Szekely, Rizzo and Bakirov (2007) or https://en.wikipedia.org/wiki/Distance_correlation. This function is the same as `dcor` in `energy` package.

Value

A sample distance correlation.

Author(s)

Debin Qiu, Jeongyoun Ahn

References

Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007), Measuring and Testing Dependence by Correlation of Distances, *Annals of Statistics*, Vol. 35 No. 6, pp. 2769-2794.

Examples

```
X <- matrix(rnorm(200), ncol = 2)
y <- X*%matrix(c(1.5,4), ncol = 1) + rnorm(100) # univariate y
distcor(X,y)

X <- iris[1:50, 1:4]
Y <- iris[51:100, 1:4] # multiple response y
distcor(X,Y)
```

grp.criValues *Compute group screening criterion values*

Description

Computes the screening criterion values for each group.

Usage

```
grp.criValues(X, y, group, criterion = c("gSIS", "gHOLP", "gAR2", "gDC"),
  family = c("gaussian", "binomial", "poisson"), scale = c("standardize",
  "normalize", "none"), norm = c("L1", "L2", "Linf"))
```

Arguments

X	A matrix of grouped predictors.
y	A numeric vector of response.
group	A vector of group indices for each predictor. Numeric and consecutive group indices are recommended.
criterion	The group screening criterion. The default is gSIS.
family	A description of the error distribution and link function to be used in the model. The default is gaussian.
scale	The type of scaling of the predictors. The default is "standardize".
norm	The type of norm for "gSIS" or "gHOLP" screening criterion. See norm_vec for details. The default is L1.

Details

In the group screening procedure, we first have to calculate the values which measure the strength of relationship between entire predictors of each group and response. These values can be used to screen out the important grouped variables (equivalently, remove the unimportant grouped variables) so that we can reduce the dimension of data from high or ultra-high to moderate or even small one.

In greater details, let $X = (x_{11}, x_{12}, \dots, x_{1p_1}, \dots, x_{j1}, x_{j2}, \dots, x_{jp_j}, \dots, x_{J1}, x_{J2}, \dots, x_{Jp_J})$ be the grouped predictors, where J is the number of groups and p_j is the number of predictors in the j -th group.

For the case in which family = "gaussian", four approaches are applied to calculate such criterion values.

The first criterion is "gSIS" that is the grouped version of sure independence screening [SIS, Fan and Lv (2008)] and defined as

$$\hat{w} = X^T y = (w_{11}, w_{12}, \dots, w_{1p_1}, \dots, w_{j1}, w_{j2}, \dots, w_{jp_j}, \dots, w_{J1}, w_{J2}, \dots, w_{Jp_J}).$$

Then we take the norm of the vector $(w_{j1}, w_{j2}, \dots, w_{jp_j})$ from the j -th group divided by its size p_j , defined as W_j and thus we obtain the criterion values for the whole groups defined as

$$\hat{W} = (W_1, \dots, W_J).$$

The details of norm type can be seen in [norm_vec](#).

The second criterion is "gHOLP" that is a grouped version of High-dimensional Ordinary Least-squares Projector [HOLP, Wang and Leng (2015)] and defined as

$$\hat{\beta} = X^T (X X^T)^{-1} y = (\beta_{11}, \beta_{12}, \dots, \beta_{1p_1}, \dots, \beta_{j1}, \beta_{j2}, \dots, \beta_{jp_j}, \dots, \beta_{J1}, \beta_{J2}, \dots, \beta_{Jp_J})$$

and then we proceed the same way as "gSIS" to incorporate the group structure.

The third criterion is "gAR2" which is called groupwise adjusted r.squared. The basic idea is that we fit a linear model for each group separately and compute the adjusted r.squared that measures the correlation between each group and response. Note that in order to calculate the adjusted r.squared, the maximum group size $\max(p_j), j = 1, \dots, J$ should not be larger than sample size n .

The last criterion is "gDC" which is called grouped distance correlation. The distance correlation [Szekely, Rizzo and Bakirov (2007)] measures the dependence between two random variables or two random vectors. Thus, similar to the idea of "gAR2", we compute the distance correlation between each group and response. It is worthwhile pointing out that distance correlation can not only measure the linear relationship, but also nonlinear relationship. However, it may take longer time in computation due to the three steps of calculating distance correlation. The distance correlation has been applied to screen the individual variables as in Li, Zhong and Zhu (2012).

For the case in which family = "binomial" and family = "poisson", a different screening criterion is used for computing the relationship between response and predictors in each group. To measure the strength of relationship between predictors and response, the Akaike's Information Criterion (AIC) is utilized and defined as

$$AIC = -2 * \text{LogLikelihood} + 2 * npar,$$

where *LogLikelihood* is the log-likelihood for a fitted generalized linear model, and *npar* is the number of parameters in the fitted model. In this case, *npar* is the number of variables within each group, i.e., $npar = p_j, j = 1, \dots, J$.

Note that the individual "SIS", "HOLP" can be regarded as a special case of "gSIS", and "gHOLP" when each group has only one predictor.

Value

A numeric matrix with two columns: the first column is the group index, and the second column is the grouped screening criterion values corresponding to the first column.

Author(s)

Debin Qiu, Jeongyoun Ahn

References

- Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space (with discussion). *Journal of the Royal Statistical Society B*, 70, 849-911.
- Li, R., Zhong, W., and Zhu, L. (2012). Feature screening via distance correlation learning. *Journal of American Statistical Association*, 107, 1129-1139.
- Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007), Measuring and Testing Dependence by Correlation of Distances, *Annals of Statistics*, Vol. 35 No. 6, pp. 2769-2794.
- Wang, X. and Leng, C. (2015). High-dimensional Ordinary Least-squares Projector for screening variables. *Journal of the Royal Statistical Society: Series B*. To appear.

See Also

[grpss](#)

Examples

```
library(MASS)
n <- 30 # sample size
p <- 3 # number of predictors in each group
J <- 50 # number of groups
group <- rep(1:J,each = 3) # group indices
Sigma <- diag(p*J) # covariance matrix
X <- mvrnorm(n,seq(0,5,length.out = p*J),Sigma)
beta <- runif(12,-2,5) # coefficients
y <- X%*%matrix(c(beta,rep(0,p*J-12)),ncol = 1) + rnorm(n)

grp.criValues(X,y,group) # gSIS
grp.criValues(X,y,group,"gHOLP") # gHOLP
grp.criValues(X,y,group,"gAR2") # gAR2
grp.criValues(X,y,group,"gDC") # gDC
```

grpss

Group screening and selection

Description

Performs the grouped variable screening and selection.

Usage

```
grpss(...)
```

Default S3 method:

```
grpss(X, y, group, threshold = NULL,
      scale = c("standardize", "normalize", "none"), criterion = c("gSIS",
      "gHOLP", "gAR2", "gDC"), family = c("gaussian", "binomial", "poisson"),
      select = FALSE, penalty = c("grSCAD", "grLasso", "grMCP", "gel", "cMCP"),
```

```
cross.validation = FALSE, norm = c("L1", "L2", "Linf"), q = 1,
perm.seed = 1, nfolds = 10, cv.seed = NULL, parallel = FALSE,
cl = NULL, cores = NULL, ...)
```

```
## S3 method for class 'formula'
grps(formula, data, group, ...)
```

Arguments

...	Optional arguments passed to grpreg .
X	A matrix of grouped predictors.
y	A numeric vector of response.
group	A vector of describing the grouping of the predictors. Numeric and consecutive group indices are recommended.
threshold	A threshold meaning how many groups are screened out. The default is NULL. See details.
scale	The type of scaling of the predictors. The default is "standardize".
criterion	The screening criterion. The default is "gSIS".
family	A description of the error distribution and link function to be used in the model. The default is "gaussian".
select	A logical value indicating whether to perform the grouped variable selection. The default is FALSE.
penalty	The penalty to be applied to the screened model. The default is "grSCAD". Only valid when <code>select = TRUE</code> .
cross.validation	A logical value indicating whether to perform the k-fold cross-validation when conducting the grouped variable selection. Only valid when <code>select = TRUE</code> . The default is FALSE.
norm	The type of norm applied to <code>criterion = gSIS</code> and <code>criterion = gHOLP</code> . The default is L1 norm.
q	A quantile for calculating the data-driven threshold in the permutation-based grouped screening. The default is 1. (i.e., the maximum absolute value of the permuted estimates). See details for more information.
perm.seed	A seed of the random number generator used for the permutation-based screening to obtain the threshold. See details.
nfolds	The number of folds to perform the cross-validation. The default is 10.
cv.seed	A seed of the random number generator used for the cross-validation.
parallel	A logical value indicating whether to use the parallel computing. The default is FALSE.
cl	A cluster object as returned by <code>makeCluster</code> , or the number of nodes to be created in the cluster.
cores	The number of cores to use for parallel execution. If not specified, the number of core is set to be 3.
formula	An object of class " formula ".
data	An optional data frame.

Details

The grouped variable selection will have big challenges or even fail in the presence of ultra-high dimension of groups. To solve these issues, we implement a two-stage procedure. At the first stage, a grouped screening procedure is applied to reduce the dimensions of groups from ultra-high to moderate or even small one, then we can use the grouped variable selection for the screened data without facing the big challenges at the second stage. At the first stage, the sure screening property ensures that the screening procedure can retain all important groups with overwhelming probability. This function is used to accomplish this two-stage procedure. At the first stage, we apply different screening criteria for grouped variables by calculating the grouped screening values that measures the strength of relationship between response and entire predictors of each group. See [grp.criValues](#) for the details of calculating the grouped screening values. For the family = "gaussian" case, we select the groups which have the largest threshold values of screening criterion indices. On the contrary, for the family = "binomial" or "poisson" case, we keep the groups which have the smallest threshold values of screening criterion values.

If threshold = NULL, we use the random permutation strategy to gain the threshold (threshold), which is called the data-driven threshold. The details can be seen in Fan, Feng and Song (2011). Larger threshold (threshold) will lead to larger probability of containing the true important groups, but may result in more intense computation in grouped variable selection and larger false positive rate.

At the second stage, we use the function [grpreg](#) in [grpreg](#) package developed by Patrick Breheny to fit the penalized regression model for the grouped variables that are screened out at the first stage. More details of the grouped variable selection can be referred to the details of [grpreg](#).

Also, we use the parallel computation in this function by importing the [doParallel](#) package to improve the computation efficiency.

Value

If select = FALSE, a list with class "grpss" containing the following components:

call	The function call.
y	The response.
X	The screened predictors.
group.screen	The indices of screened groups.
threshold	The threshold.
criterion	The screening criterion.

If select = TRUE, a list with class "grpreg" or "cv.grpreg" (when cross.validation = TRUE) containing the similar components as in function [grpreg](#) or [cv.grpreg](#), plus the following three elements:

call	Same as above.
group.screen	Same as above.
criterion	Same as above.

Note

The missing values are removed before the analysis.

Author(s)

Debin Qiu, Jeongyoun Ahn

References

Fan J, Feng Y, Song R (2011). Nonparametric independence screening in sparse ultra-high dimensional additive models. *Journal of the American Statistical Association*. 106:544-557.

See Also

[grpreg](#), [cv.grpreg](#), [grp.criValues](#)

Examples

```
library(MASS)
set.seed(23)
n <- 30 # sample size
p <- 3 # number of predictors in each group
J <- 50 # group size
group <- rep(1:J,each = 3) # group indices
##autoregressive correlation
Sigma <- 0.6^abs(matrix(1:(p*J),p*J,p*J) - t(matrix(1:(p*J),p*J,p*J)))
X <- mvrnorm(n,seq(0,5,length.out = p*J),Sigma)
betaTrue <- runif(12,-2,5)
mu <- X%%matrix(c(betaTrue,rep(0,p*J-12)),ncol = 1)

# normal distribution
y <- mu + rnorm(n)

# only conduct screening procedure
(gss01 <- grpss(X,y,group)) # gSIS

# perform both screening and selection procedures
## use grpss.default with cross-validation
gss11 <- grpss(X,y,group,select = TRUE,cross.validation = TRUE)
summary(gss11)
## without cross-validation
gss12 <- grpss(X,y,threshold = 10,group,select = TRUE,criterion = "gHOLP")
summary(gss12)

## binomial distribution
y1 <- rbinom(n,1,1/(1 + exp(-mu)))
(gss21 <- grpss(X,y1,group, criterion = "gAR2")) # use AIC
(gss22 <- grpss(X,y1,group, criterion = "gDC")) # use gDC

## poisson distribution
y2 <- rpois(n,lambda = exp(mu))
(gss31 <- grpss(X,y2,group, criterion = "gAR2"))
(gss22 <- grpss(X,y2,group, criterion = "gDC"))
```

importance	<i>Arrange and visualize the importance of groups</i>
------------	---

Description

Arranges and visualizes the importance of groups for the results of [grp.criValues](#).

Usage

```
importance(grp.values, n = 10, plot = TRUE)
```

Arguments

grp.values	The result from grp.criValues .
n	Number of important groups to display. The default is the top 10 groups.
plot	A logical value indicating whether to make a barplot of the importance of groups.

Details

This function arranges the values of screening criterion from the most important to the least important, and then make a barplot to visualize the importance.

Value

A matrix containing the first n important group indices and values of screening criterion.

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also

[grp.criValues](#)

Examples

```
library(MASS)
n <- 30 # sample size
p <- 3 # number of predictors in each group
J <- 50 # number of groups
group <- rep(1:J,each = 3) # group indices
Sigma <- diag(p*J) # covariance matrix
X <- mvrnorm(n,seq(0,5,length.out = p*J),Sigma)
beta <- runif(12,-2,5) # coefficients
y <- X%%matrix(c(beta,rep(0,p*J-12)),ncol = 1) + rnorm(n)

crivalues <- grp.criValues(X,y,group) # gSIS
importance(crivalues, n = 20)
```

norm_vec	<i>Compute the norm of a vector</i>
----------	-------------------------------------

Description

Computes the norm of a vector with size adjustment.

Usage

```
norm_vec(x, norm = c("L1", "L2", "Linf"))
```

Arguments

x	A numeric vector.
norm	The type of norm. See details.

Details

This function calculates the norm of a vector. It is slightly different from the ordinary norm function in that L1-norm and L2-norm are adjusted by the length of vector, n . To be specific, define a vector $x = (x_1, x_2, \dots, x_n)$. The L1-norm is defined as

$$\|x\|_1 = (|x_1| + |x_2| + \dots + |x_n|)/n.$$

The L2-norm is defined as

$$\|x\|_2 = (\sqrt{(x_1)^2 + (x_2)^2 + \dots + (x_n)^2})/n.$$

The Linf-norm is defined as

$$\|x\|_\infty = \max(|x_1|, |x_2|, \dots, |x_n|).$$

Note that a matrix X will be coerced to a vector.

Value

A non-negative number.

Author(s)

Debin Qiu, Jeongyoun Ahn

Examples

```
x <- 1:10
# "L1" norm, same as norm(as.matrix(x), "1")*length(x)
norm_vec(x)

# "L2" norm
norm_vec(x, "L2")

# "Linf" norm, same as norm(as.matrix(x), "I")
norm_vec(x, "Linf")
```

predict.cv.grpreg *Model predictions for a fitted cv.grpreg object*

Description

Similar to usual predict methods and predict.cv.grpreg in grpreg package.

Usage

```
## S3 method for class 'cv.grpreg'
predict(object, newdata, type = c("response", "class",
  "probability"), ...)
```

Arguments

object	A fitted "grpreg" object from grpss , or cv.grpreg function.
newdata	Optionally, a matrix or data frame where to predict. If omits, the fitted predictors are used.
type	The type of prediction: "response" gives the fitted values; "class" returns the predicted class for the binomial outcome; "probability" returns the predicted probabilities for the logistic regression.
...	Not used.

Details

This function gives the predictions at newdata or all predictors if the argument newdata is not supplied. Typically, type = "response" is used for linear or poisson regression, and type = "class" or type = "probability" is used for logistic regression.

Value

The predicted values depending on the type.

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also[grpss](#)**Examples**

```

library(MASS)
set.seed(23)
n <- 30 # sample size
p <- 3 # number of predictors in each group
J <- 50 # group size
group <- rep(1:J,each = 3) # group indices
X <- mvrnorm(n,seq(0,5,length.out = p*J),diag(p*J))
beta <- runif(12,-2,5)
mu <- X%%matrix(c(beta,rep(0,p*J-12)),ncol = 1)

# linear regression with family = "gaussian"
y <- mu + rnorm(n)

## with cross-validation
gss11 <- grpss(X,y,group,select = TRUE,cross.validation = TRUE)
predict(gss11) # fitted values
predict(gss11, newdata = X[1,]) # predicted values at given 'newdata'

# logistic regression with family = "binomial"
set.seed(23)
y1 <- rbinom(n,1,1/(1 + exp(-mu)))
gss21 <- grpss(X,y1,group, criterion = "gDC",select = TRUE,
              family = "binomial")
predict(gss21)

```

`predict.grpreg`*Model predictions for a fitted grpreg object*

Description

Similar to usual predict methods and predict.grpreg in grpreg package.

Usage

```

## S3 method for class 'grpreg'
predict(object, newdata, lambda = NULL,
        type = c("response", "class", "probability"), ...)

```

Arguments

object	A fitted "grpreg" object from grpss , or grpreg function.
newdata	Optionally, a matrix or data frame where to predict. If omits, the fitted predictors are used.

lambda	Value of the regularization parameter lambda at which predictions are requested. See details for the default.
type	The type of prediction: "response" gives the fitted values; "class" returns the predicted class for the binomial outcome; "probability" returns the predicted probabilities for the logistic regression.
...	Not used.

Details

This function gives the predictions at newdata or all predictors if the argument newdata is not supplied. The default lambda for "grpreg" object is the one at which we obtain the minimum loss value, i.e., negative log-likelihood value. Typically, type = "response" is used for linear or poisson regression, and type = "class" or type = "probability" is used for logistic regression.

Value

The predicted values depending on the type.

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also

[grpss](#)

Examples

```
library(MASS)
set.seed(23)
n <- 30 # sample size
p <- 3 # number of predictors in each group
J <- 50 # group size
group <- rep(1:J,each = 3) # group indices
X <- mvrnorm(n,seq(0,5,length.out = p*J),diag(p*J))
beta <- runif(12,-2,5)
mu <- X%%matrix(c(beta,rep(0,p*J-12)),ncol = 1)

# linear regression with family = "gaussian"
y <- mu + rnorm(n)

## without cross-validation
gss12 <- grpss(X,y,ncut = 10,group,select = TRUE)
predict(gss12) # fitted values
predict(gss12,lambda = 0.2) # fitted values at lambda = 0.2

# logistic regression with family = "binomial"
set.seed(23)
y1 <- rbinom(n,1,1/(1 + exp(-mu)))
gss21 <- grpss(X,y1,group, criterion = "gDC",select = TRUE,
              family = "binomial")
```

```
predict(gss21)
```

```
print.grpss
```

```
Print the grpss object
```

Description

Prints the "grpss" object obtained from [grpss](#).

Usage

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

Arguments

x	A fitted "grpss" object.
...	Not used.

Details

This function prints out the group screening results obtained from [grpss](#) when "selection = FALSE".

Value

The call, criterion, threshold and screened groups will be printed out.

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also

[grpss](#)

summary.cv.gprreg	<i>Summarize a fitted cv.gprreg object</i>
-------------------	--

Description

Summarizes a fitted penalized regression model with 'cv.gprreg' class.

Usage

```
## S3 method for class 'cv.gprreg'
summary(object, digits = 4, ...)
```

Arguments

object	A fitted "cv.gprreg" object.
digits	Number of digits past the decimal point to print out. The default is 4.
...	Optional arguments passed to other methods.

Details

This function is similar to `summary.cv.gprreg` in `gprreg` package, but gives several different results. The `lambda` here is only valid for "gprreg" object. See [summary.gprreg](#).

Value

A list with class "summary.cv.gprreg" containing the following components:

n	Number of observations.
p	Number of screened predictors.
penalty	The penalty applied to the model.
model	The type of model.
family	The link function.
criterion	The screening criterion.
lambda	The default or specified regularization parameter.
beta	The estimates of coefficients at the specified lambda.
iter	The number of iterations at the specified lambda.
df	The estimates of effective number of model parameters at the specified lambda.
cve	The average cross-validation error at specified lambda.
call	The function call.

Additional elements are contained for the case in which `family = "gaussian"`:

r.squared	The r.squared.
snr	The signal-to-noise ratio.

scale The scale parameter estimate (sigma).

and following elements for the case in which family = "poisson" or "binomial":

logLik The negative log-likelihood values for the fitted model.

aic Akaike's information criterion (AIC).

bic Bayesian information criterion (BIC).

aicc The AIC with a correction for finite sample sizes (AICC).

pe The prediction error for family = "binomial".

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also

[grpss](#), [summary.grpreg](#)

summary.grpreg *Summarize a fitted grpreg object*

Description

Summarizes a fitted penalized regression model with 'grpreg' class.

Usage

```
## S3 method for class 'grpreg'
summary(object, lambda = NULL, digits = 4, ...)
```

Arguments

object A "summary.grpreg" object.

lambda A regularization parameter at which to summarize.

digits Number of digits past the decimal point to print out. The default is 4.

... Optional arguments passed to other methods.

Details

The default lambda is the one at which we obtain the minimum loss (i.e., negative log-likelihood value), if lambda is not supplied.

Value

A list with class "summary.grpreg" containing the following components:

n	Number of observations.
p	Number of screened predictors.
penalty	The penalty applied to the model.
model	The type of model.
family	The link function.
criterion	The screening criterion.
lambda	The default or specified regularization parameter.
beta	The estimates of coefficients at the specified lambda.
iter	The number of iterations at the specified lambda.
df	The estimates of effective number of model parameters at the specified lambda.
call	The function call.

Additional elements are contained for the case in which family = "gaussian":

r.squared	The r.squared.
snr	The signal-to-noise ratio.
scale	The scale parameter estimate.

and following elements for the case in which family = "poisson" or "binomial":

logLik	The negative log-likelihood values for the fitted model.
aic	Akaike's information criterion (AIC).
bic	Bayesian information criterion (BIC).
aicc	The AIC with a correction for finite sample sizes (AICC).
pe	The prediction error for family = "binomial".

Author(s)

Debin Qiu, Jeongyoun Ahn

See Also

[grpss](#), [summary.cv.grpreg](#)

Index

`cv.grpreg`, [7](#), [8](#), [11](#)

`distcor`, [2](#)

`formula`, [6](#)

`grp.criValues`, [3](#), [7–9](#)

`grpreg`, [6–8](#), [12](#)

`grpss`, [5](#), [5](#), [11–14](#), [16](#), [17](#)

`importance`, [9](#)

`norm_vec`, [3](#), [4](#), [10](#)

`predict.cv.grpreg`, [11](#)

`predict.grpreg`, [12](#)

`print.grpss`, [14](#)

`summary.cv.grpreg`, [15](#), [17](#)

`summary.grpreg`, [15](#), [16](#), [16](#)