

# Package ‘BeSS’

January 23, 2021

**Type** Package

**Title** Best Subset Selection /Ridge Regression in Linear, Logistic,  
Poisson and CoxPH Models

**Version** 2.0.2

**Date** 2021-01-23

**Author** Canhong Wen [aut, cre],  
Aijun Zhang [aut],  
Shijie Quan [aut],  
Liyuan Hu [aut],  
Kangkang Jiang [aut],  
Yanhang Zhang [aut],  
Jin Zhu [aut],  
Xueqin Wang [aut]

**Maintainer** Canhong Wen <wencanhong@gmail.com>

**Description** An implementation of best subset selection in generalized linear model and Cox proportional hazard model via the primal dual active set algorithm proposed by Wen, C., Zhang, A., Quan, S. and Wang, X. (2020) <doi:10.18637/jss.v094.i04>. The algorithm formulates coefficient parameters and residuals as primal and dual variables and utilizes efficient active set selection strategies based on the complementarity of the primal and dual variables.

**License** GPL-3

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**LazyData** true

**Imports** Rcpp (>= 1.0.3), Matrix (>= 1.2-6), MASS, pheatmap, survival

**LinkingTo** Rcpp, RcppEigen

**RoxygenNote** 7.1.1

**Suggests** knitr, HCmodelSets, rmarkdown

**VignetteBuilder** knitr

**URL** <https://github.com/Mamba413/bess>

**BugReports** <https://github.com/Mamba413/bess/issues>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2021-01-23 07:10:02 UTC

## R topics documented:

BeSS-package . . . . .	2
bess . . . . .	3
bess.one . . . . .	10
coef.bess . . . . .	15
deviance.bess . . . . .	16
duke . . . . .	17
gen.data . . . . .	18
logLik.bess . . . . .	20
plot.bess . . . . .	22
predict.bess . . . . .	24
print.bess . . . . .	26
summary.bess . . . . .	27
trim32 . . . . .	28

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

---

BeSS-package	<i>BeSS: Best Subset Selection /Ridge Regression in Linear, Logistic, Poisson and CoxPH Models</i>
--------------	--

---

## Description

An implementation of best subset selection in generalized linear model and Cox proportional hazard model via the primal dual active set algorithm proposed by Wen, C., Zhang, A., Quan, S. and Wang, X. (2020) <doi:10.18637/jss.v094.i04>. The algorithm formulates coefficient parameters and residuals as primal and dual variables and utilizes efficient active set selection strategies based on the complementarity of the primal and dual variables.

## Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

---

bess *Best subset selection*

---

### Description

Best subset selection for generalized linear model and Cox's proportional model.

### Usage

```
bess(
  x,
  y,
  family = c("gaussian", "binomial", "poisson", "cox"),
  type = c("bss", "bsrr"),
  method = c("gsection", "sequential", "pgsection", "psequential"),
  tune = c("gic", "ebic", "bic", "aic", "cv"),
  s.list,
  lambda.list = 0,
  s.min,
  s.max,
  lambda.min = 0.001,
  lambda.max = 100,
  nlambda = 100,
  always.include = NULL,
  screening.num = NULL,
  normalize = NULL,
  weight = NULL,
  max.iter = 20,
  warm.start = TRUE,
  nfolds = 5,
  group.index = NULL,
  seed = NULL
)
```

### Arguments

x	Input matrix, of dimension $n \times p$ ; each row is an observation vector and each column is a predictor/feature/variable.
y	The response variable, of n observations. For family = "binomial" should be a factor with two levels. For family="poisson", y should be a vector with positive integer. For family = "cox", y should be a two-column matrix with columns named time and status.
family	One of the following models: "gaussian", "binomial", "poisson", or "cox". Depending on the response. Any unambiguous substring can be given.
type	One of the two types of problems. type = "bss" for the best subset selection, and type = "bsrr" for the best subset ridge regression.

method	The method to be used to select the optimal model size and $L_2$ shrinkage. For method = "sequential", we solve the best subset selection and the best subset ridge regression problem for each $s$ in $1, 2, \dots, s.max$ and $\lambda$ in <code>lambda.list</code> . For method = "gsection", which is only valid for type = "bss", we solve the best subset selection problem with model size ranged between <code>s.min</code> and <code>s.max</code> , where the specific model size to be considered is determined by golden section. we solve the best subset selection problem with a range of non-continuous model sizes. For method = "pgsection" and "psequential", the Powell method is used to solve the best subset ridge regression problem. Any unambiguous substring can be given.
tune	The criterion for choosing the model size and $L_2$ shrinkage parameters. Available options are "gic", "ebic", "bic", "aic" and "cv". Default is "gic".
s.list	An increasing list of sequential values representing the model sizes. Only used for method = "sequential". Default is <code>1:min(p, round(n/log(n)))</code> .
lambda.list	A lambda sequence for "bsrr". Default is <code>exp(seq(log(100), log(0.01), length.out = 100))</code> .
s.min	The minimum value of model sizes. Only used for method = "gsection", "psequential" and "pgsection". Default is 1.
s.max	The maximum value of model sizes. Only used for method = "gsection", "psequential" and "pgsection". Default is <code>min(p, round(n/log(n)))</code> .
lambda.min	The minimum value of lambda. Only used for method = "powell". Default is <code>0.001</code> .
lambda.max	The maximum value of lambda. Only used for method = "powell". Default is <code>100</code> .
nlambda	The number of $\lambda$ s for the Powell path with sequential line search method. Only valid for method = "psequential".
always.include	An integer vector containing the indexes of variables that should always be included in the model.
screening.num	Users can pre-exclude some irrelevant variables according to maximum marginal likelihood estimators before fitting a model by passing an integer to <code>screening.num</code> and the sure independence screening will choose a set of variables of this size. Then the active set updates are restricted on this subset.
normalize	Options for normalization. <code>normalize = 0</code> for no normalization. Setting <code>normalize = 1</code> will only subtract the mean of columns of $x$ . <code>normalize = 2</code> for scaling the columns of $x$ to have $\sqrt{n}$ norm. <code>normalize = 3</code> for subtracting the means of the columns of $x$ and $y$ , and also normalizing the columns of $x$ to have $\sqrt{n}$ norm. If <code>normalize = NULL</code> , by default, <code>normalize</code> will be set 1 for "gaussian", 2 for "binomial" and "poisson", 3 for "cox".
weight	Observation weights. Default is 1 for each observation.
max.iter	The maximum number of iterations in the bess function. In most of the case, only a few steps can guarantee the convergence. Default is 20.
warm.start	Whether to use the last solution as a warm start. Default is TRUE.
nfolds	The number of folds in cross-validation. Default is 5.

group.index	A vector of integers indicating the which group each variable is in. For variables in the same group, they should be located in adjacent columns of $x$ and their corresponding index in group.index should be the same. Denote the first group as 1, the second 2, etc. If you do not fit a model with a group structure, please set group.index = NULL. Default is NULL.
seed	Seed to be used to divide the sample into $K$ cross-validation folds. Default is NULL.

## Details

The best subset selection problem with model size  $s$  is

$$\min_{\beta} -2 \log L(\beta) \quad \text{s.t.} \quad \|\beta\|_0 \leq s.$$

In the GLM case,  $\log L(\beta)$  is the log-likelihood function; In the Cox model,  $\log L(\beta)$  is the log partial likelihood function.

The best ridge regression problem with model size  $s$  is

$$\min_{\beta} -2 \log L(\beta) + \lambda \|\beta\|_2^2 \quad \text{s.t.} \quad \|\beta\|_0 \leq s.$$

In the GLM case,  $\log L(\beta)$  is the log likelihood function; In the Cox model,  $\log L(\beta)$  is the log partial likelihood function.

For each candidate model size and  $\lambda$ , the best subset selection and the best subset ridge regression problems are solved by the primal-dual active set (PDAS) algorithm, see Wen et al (2020) for details. This algorithm utilizes an active set updating strategy via primal and dual variables and fits the sub-model by exploiting the fact that their support sets are non-overlap and complementary. For the case of method = "sequential" if warm.start = "TRUE", we run the PDAS algorithm for a list of sequential model sizes and use the estimate from the last iteration as a warm start. For the case of method = "gsection" of the best subset selection problem, a golden section search technique is adopted to determine the optimal model size efficiently. And for the case of method = "psequential" and method = "pgsection" of the best ridge regression problem, the Powell method using a sequential line search method or a golden section search technique is used for parameters determination.

## Value

A list with class attribute 'bess' and named components:

beta	The best fitting coefficients.
coef0	The best fitting intercept.
bestmodel	The best fitted model for type = "bss", the class of which is "lm", "glm" or "coxph".
loss	The training loss of the best fitting model.
ic	The information criterion of the best fitting model when model selection is based on a certain information criterion.
cvm	The mean cross-validated error for the best fitting model when model selection is based on the cross-validation.

lambda	The lambda chosen for the best fitting model
beta.all	For bess objects obtained by gsection, pgsection and psequential, beta.all is a matrix with each column be the coefficients of the model in each iterative step in the tuning path. For bess objects obtained by sequential method, A list of the best fitting coefficients of size $s=0, 1, \dots, p$ and $\lambda$ in lambda.list with the smallest loss function. For "bess" objects of "bsrr" type, the fitting coefficients of the $i^{th}$ $\lambda$ and the $j^{th}$ s are at the $i^{th}$ list component's $j^{th}$ column.
coef0.all	For bess objects obtained from gsection, pgsection and psequential, coef0.all contains the intercept for the model in each iterative step in the tuning path. For bess objects obtained from sequential path, coef0.all contains the best fitting intercepts of size $s = 0, 1, \dots, p$ and $\lambda$ in lambda.list with the smallest loss function.
loss.all	For bess objects obtained from gsection, pgsection and psequential, loss.all contains the training loss of the model in each iterative step in the tuning path. For bess objects obtained from sequential path, this is a list of the training loss of the best fitting intercepts of model size $s = 0, 1, \dots, p$ and $\lambda$ in lambda.list. For "bess" object obtained by "bsrr", the training loss of the $i^{th}$ $\lambda$ and the $j^{th}$ s is at the $i^{th}$ list component's $j^{th}$ entry.
ic.all	For bess objects obtained from gsection, pgsection and psequential, ic.all contains the values of the chosen information criterion of the model in each iterative step in the tuning path. For bess objects obtained from sequential path, this is a matrix of the values of the chosen information criterion of model size $s = 0, 1, \dots, p$ and $\lambda$ in lambda.list with the smallest loss function. For "bess" object obtained by "bsrr", the training loss of the $i^{th}$ $\lambda$ and the $j^{th}$ s is at the $i^{th}$ row $j^{th}$ column. Only available when model selection is based on a certain information criterion.
cvm.all	For bess objects obtained from gsection, pgsection and psequential, cvm.all contains the mean cross-validation error of the model in each iterative step in the tuning path. For bess objects obtained from sequential path, this is a matrix of the mean cross-validation error of model size $s = 0, 1, \dots, p$ and $\lambda$ in lambda.list with the smallest loss function. For "bess" object obtained by "bsrr", the training loss of the $i^{th}$ $\lambda$ and the $j^{th}$ s is at the $i^{th}$ row $j^{th}$ column. Only available when model selection is based on the cross-validation.
lambda.all	The lambda chosen for each step in pgsection and psequential.
family	Type of the model.
s.list	The input s.list.
nsample	The sample size.
type	Either "bss" or "bsrr".
method	Method used for tuning parameters selection.
ic.type	The criterion of model selection.

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

## See Also

[plot.bess](#), [summary.bess](#), [coef.bess](#), [predict.bess](#), [bess.one](#).

## Examples

```
#-----linear model-----#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lm.bss <- bess(x, y)
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(x, y, type = "bsrr", method = "pgsection")
coef(lm.bss)
coef(lm.bsrr)
print(lm.bss)
print(lm.bsrr)
summary(lm.bss)
summary(lm.bsrr)
pred.bss <- predict(lm.bss, newx = x_new)
pred.bsrr <- predict(lm.bsrr, newx = x_new)

# generate plots
plot(lm.bss, type = "both", breaks = TRUE)
plot(lm.bsrr)
#-----logistic model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "binomial", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
logi.bss <- bess(x, y, family = "binomial")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
logi.bsrr <- bess(x, y, type = "bsrr", family = "binomial", lambda.list = lambda.list)
```

```

coef(logi.bss)
coef(logi.bsrr)
print(logi.bss)
print(logi.bsrr)
summary(logi.bss)
summary(logi.bsrr)
pred.bss <- predict(logi.bss, newx = x_new)
pred.bsrr <- predict(logi.bsrr, newx = x_new)

# generate plots
plot(logi.bss, type = "both", breaks = TRUE)
plot(logi.bsrr)
#-----poisson model-----#
Data <- gen.data(n, p, k, rho=0.3, family = "poisson", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
poi.bss <- bess(x, y, family = "poisson")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
poi.bsrr <- bess(x, y, type = "bsrr",
                 family = "poisson", lambda.list = lambda.list)

coef(poi.bss)
coef(poi.bsrr)
print(poi.bss)
print(poi.bsrr)
summary(poi.bss)
summary(poi.bsrr)
pred.bss <- predict(poi.bss, newx = x_new)
pred.bsrr <- predict(poi.bsrr, newx = x_new)

# generate plots
plot(poi.bss, type = "both", breaks = TRUE)
plot(poi.bsrr)
#-----coxph model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "cox", scal = 10, beta = Tbeta)

x <- Data$x[1:140, ]
y <- Data$y[1:140, ]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200, ]
cox.bss <- bess(x, y, family = "cox")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
cox.bsrr <- bess(x, y, type = "bsrr", family = "cox", lambda.list = lambda.list)
coef(cox.bss)
coef(cox.bsrr)
print(cox.bss)
print(cox.bsrr)
summary(cox.bss)
summary(cox.bsrr)
pred.bss <- predict(cox.bss, newx = x_new)

```



```

pred.bsrr <- predict(cox.bsrr, newx = x_new)

# generate plots
plot(cox.bss, type = "both", breaks = TRUE)
plot(cox.bsrr)

#-----High dimensional linear models-----#
## Not run:
data <- gen.data(n, p = 1000, k, family = "gaussian", seed = seed)

# Best subset selection with SIS screening
lm.high <- bess(data$x, data$y, screening.num = 100)

## End(Not run)

#-----group selection-----#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, seed =10)
x <- Data$x
y <- Data$y

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.group <- bess(x, y, s.min=1, s.max = 8, type = "bss", group.index = group.index)
lm.groupbsrr <- bess(x, y, type = "bsrr", s.min = 1, s.max = 8, group.index = group.index)
coef(lm.group)
coef(lm.groupbsrr)
print(lm.group)
print(lm.groupbsrr)
#'summary(lm.group)
summary(lm.groupbsrr)
pred.group <- predict(lm.group, newx = x_new)
pred.groupl0l2 <- predict(lm.groupbsrr, newx = x_new)
#-----include specified variables-----#
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess(Data$x, Data$y, always.include = 2)

## Not run:
#-----trim32 data analysis in doi: 10.18637/jss.v094.i04-----#
# import trim32 data by:
load(url('https://github.com/Mamba413/bess/tree/master/data/trim32.RData'))
# or manually downloading trim32.RData in the github page:
# "https://github.com/Mamba413/bess/tree/master/data/" and read it by:
load('trim32.RData')

X <- trim32$x
Y <- trim32$y
dim(X)

# running bess with argument method = "sequential".
fit.seq <- bess(X, Y, method = "sequential")
summary(fit.seq)

```

```

# the bess function outputs an 'lm' type of object bestmodel associated
# with the selected best model
bm.seq <- fit.seq$bestmodel
summary(bm.seq)
pred.seq <- predict(fit.seq, newdata = data$x)
plot(fit.seq, type = "both", breaks = TRUE)

# We now call the function bess with argument method = "gsection"
fit.gs <- bess(X, Y, family = "gaussian", method = "gsection")
bm.gs <- fit.gs$bestmodel
summary(bm.gs)
beta <- coef(fit.gs, sparse = TRUE)
class(beta)
pred.gs <- predict(fit.gs, newdata = X)

## End(Not run)

```

---

bess.one

*Best subset selection/Best subset ridge regression with a specified model size and a shrinkage parameter*

---

### Description

Best subset selection with a specified model size for generalized linear models and Cox's proportional hazard model.

### Usage

```

bess.one(
  x,
  y,
  family = c("gaussian", "binomial", "poisson", "cox"),
  type = c("bss", "bsrr"),
  s,
  lambda = 0,
  always.include = NULL,
  screening.num = NULL,
  normalize = NULL,
  weight = NULL,
  max.iter = 20,
  group.index = NULL
)

```

### Arguments

**x** Input matrix, of dimension  $n \times p$ ; each row is an observation vector and each column is a predictor/feature/variable.

<code>y</code>	The response variable, of $n$ observations. For <code>family = "binomial"</code> should be a factor with two levels. For <code>family="poisson"</code> , <code>y</code> should be a vector with positive integer. For <code>family = "cox"</code> , <code>y</code> should be a two-column matrix with columns named <code>time</code> and <code>status</code> .
<code>family</code>	One of the following models: <code>"gaussian"</code> , <code>"binomial"</code> , <code>"poisson"</code> , or <code>"cox"</code> . Depending on the response.
<code>type</code>	One of the two types of problems. <code>type = "bss"</code> for the best subset selection, and <code>type = "bsrr"</code> for the best subset ridge regression.
<code>s</code>	A specified model size
<code>lambda</code>	A shrinkage parameter for <code>"bsrr"</code> .
<code>always.include</code>	An integer vector containing the indexes of variables that should always be included in the model.
<code>screening.num</code>	Users can pre-exclude some irrelevant variables according to maximum marginal likelihood estimators before fitting a model by passing an integer to <code>screening.num</code> and the sure independence screening will choose a set of variables of this size. Then the active set updates are restricted on this subset.
<code>normalize</code>	Options for normalization. <code>normalize = 0</code> for no normalization. Setting <code>normalize = 1</code> will only subtract the mean of columns of $x$ . <code>normalize = 2</code> for scaling the columns of $x$ to have $\sqrt{n}$ norm. <code>normalize = 3</code> for subtracting the means of the columns of $x$ and $y$ , and also normalizing the columns of $x$ to have $\sqrt{n}$ norm. If <code>normalize = NULL</code> , by default, <code>normalize</code> will be set 1 for <code>"gaussian"</code> , 2 for <code>"binomial"</code> and <code>"poisson"</code> , 3 for <code>"cox"</code> .
<code>weight</code>	Observation weights. Default is 1 for each observation.
<code>max.iter</code>	The maximum number of iterations in the <code>bess</code> function. In most of the case, only a few steps can guarantee the convergence. Default is 20.
<code>group.index</code>	A vector of integers indicating the which group each variable is in. For variables in the same group, they should be located in adjacent columns of $x$ and their corresponding index in <code>group.index</code> should be the same. Denote the first group as 1, the second 2, etc. If you do not fit a model with a group structure, please set <code>group.index = NULL</code> . Default is <code>NULL</code> .

## Details

Given a model size  $s$ , we consider the following best subset selection problem:

$$\min_{\beta} -2 \log L(\beta); s.t. \|\beta\|_0 = s.$$

And given a model size  $s$  and a shrinkage parameter  $\lambda$ , consider the following best subset ridge regression problem:

$$\min_{\beta} -2 \log L(\beta) + \lambda \|\beta\|_2^2; s.t. \|\beta\|_0 = s.$$

In the GLM case,  $\log L(\beta)$  is the log likelihood function; In the Cox model,  $\log L(\beta)$  is the log partial likelihood function.

The best subset selection problem is solved by the primal dual active set algorithm, see Wen et al. (2017) for details. This algorithm utilizes an active set updating strategy via primal and dual variables and fits the sub-model by exploiting the fact that their support set are non-overlap and complementary.

**Value**

A list with class attribute 'bess' and named components:

beta	The best fitting coefficients.
coef0	The best fitting intercept.
bestmodel	The best fitted model for type = "bss", the class of which is "lm", "glm" or "coxph".
loss	The training loss of the fitting model.
s	The model size.
lambda	The shrinkage parameter.
family	Type of the model.
nsample	The sample size.
type	Either "bss" or "bsrr".

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

**See Also**

[bess](#), [summary.bess](#) [coef.bess](#), [predict.bess](#).

**Examples**

```
#-----linear model-----#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lm.bss <- bess.one(x, y, s = 5)
lm.bsrr <- bess.one(x, y, type = "bsrr", s = 5, lambda = 0.01)
```

```

coef(lm.bss)
coef(lm.bsrr)
print(lm.bss)
print(lm.bsrr)
summary(lm.bss)
summary(lm.bsrr)
pred.bss <- predict(lm.bss, newx = x_new)
pred.bsrr <- predict(lm.bsrr, newx = x_new)

#-----logistic model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "binomial", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
logi.bss <- bess.one(x, y, family = "binomial", s = 5)
logi.bsrr <- bess.one(x, y, type = "bsrr", family = "binomial", s = 5, lambda = 0.01)
coef(logi.bss)
coef(logi.bsrr)
print(logi.bss)
print(logi.bsrr)
summary(logi.bss)
summary(logi.bsrr)
pred.bss <- predict(logi.bss, newx = x_new)
pred.bsrr <- predict(logi.bsrr, newx = x_new)

#-----poisson model-----#
Data <- gen.data(n, p, k, rho=0.3, family = "poisson", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
poi.bss <- bess.one(x, y, family = "poisson", s=5)
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
poi.bsrr <- bess.one(x, y, type = "bsrr", family = "poisson", s = 5, lambda = 0.01)
coef(poi.bss)
coef(poi.bsrr)
print(poi.bss)
print(poi.bsrr)
summary(poi.bss)
summary(poi.bsrr)
pred.bss <- predict(poi.bss, newx = x_new)
pred.bsrr <- predict(poi.bsrr, newx = x_new)

#-----coxph model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "cox", beta = Tbeta, scal = 10)

x <- Data$x[1:140, ]
y <- Data$y[1:140, ]

```

```

x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200, ]
cox.bss <- bess.one(x, y, family = "cox", s = 5)
cox.bsrr <- bess.one(x, y, type = "bsrr", family = "cox", s = 5, lambda = 0.01)
coef(cox.bss)
coef(cox.bsrr)
print(cox.bss)
print(cox.bsrr)
summary(cox.bss)
summary(cox.bsrr)
pred.bss <- predict(cox.bss, newx = x_new)
pred.bsrr <- predict(cox.bsrr, newx = x_new)
#-----High dimensional linear models-----#
## Not run:
data <- gen.data(n, p = 1000, k, family = "gaussian", seed = seed)

# Best subset selection with SIS screening
fit <- bess.one(data$x, data$y, screening.num = 100, s = 5)

## End(Not run)

#-----group selection-----#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, seed =10)
x <- Data$x
y <- Data$y

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.group <- bess.one(x, y, s = 5, type = "bss", group.index = group.index)
lm.groupbsrr <- bess.one(x, y, type = "bsrr", s = 5, lambda = 0.01, group.index = group.index)
coef(lm.group)
coef(lm.groupbsrr)
print(lm.group)
print(lm.groupbsrr)
summary(lm.group)
summary(lm.groupbsrr)
pred.group <- predict(lm.group, newx = x_new)
pred.group1012 <- predict(lm.groupbsrr, newx = x_new)
#-----include specified variables-----#
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess.one(Data$x, Data$y, s = 5, always.include = 2)

## Not run:
#-----code demonstration in doi: 10.18637/jss.v094.i04-----#
Tbeta <- rep(0, 20)
Tbeta[c(1, 2, 5, 9)] <- c(3, 1.5, -2, -1)

data <- gen.data(n = 200, p = 20, family = "gaussian", beta = Tbeta,
rho = 0.2, seed = 123)
fit.one <- bess.one(data$x, data$y, s = 4, family = "gaussian")
print(fit.one)
summary(fit.one)

```

```
coef(fit.one, sparse = FALSE)
pred.one <- predict(fit.one, newdata = data$x)
bm.one <- fit.one$bestmodel
summary(bm.one)

## End(Not run)
```

---

`coef.bess`*Provides estimated coefficients from a fitted "bess" object.*

---

## Description

This function provides estimated coefficients from a fitted "bess" object.

## Usage

```
## S3 method for class 'bess'
coef(object, sparse = TRUE, ...)
```

## Arguments

<code>object</code>	A "bess" project.
<code>sparse</code>	Logical or NULL, specifying whether the coefficients should be presented as sparse matrix or not.
<code>...</code>	Other arguments.

## Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

## See Also

[bess](#), [print.bess](#).

## Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess(Data$x, Data$y, method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(Data$x, Data$y, type = "bsrr", method = "pgsection")
coef(lm.bss)
coef(lm.bsrr)
```

---

deviance.bess

*Extract the deviance from a "bess.one" object.*

---

## Description

Similar to other deviance methods, which returns deviance from a fitted "bess.one" object.

## Usage

```
## S3 method for class 'bess'
deviance(object, best.model = TRUE, ...)
```

## Arguments

object	A "bess" object.
best.model	Whether only return the loglikelihood of the best model. Default is TRUE. If best.model = FALSE, the loglikelihood of the best models with model size and $\lambda$ in the original s.list and lambda.list (for method = "sequential") or in the iteration path (for method = "gsection", method = "pgsection", and method = "psequential") is returned.
...	additional arguments

## Value

A matrix or vector containing the deviance for each model is returned. For bess object fitted by sequential method, values in each row in the returned matrix corresponding to the model size in s.list, and each column the shrinkage parameters in lambda.list.

For bess object fitted by gsection, pgsection and psequential, the returned vector contains deviance for fitted models in each iteration. The coefficients of those model can be extracted from beta.all and coef0.all in the bess object.



**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

**See Also**

[bess](#), [summary.bess](#).

**Examples**

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", seed = seed)
lm.bss <- bess(Data$x, Data$y, method = "sequential")

deviance(lm.bss)
deviance(lm.bss, best.model = FALSE)
```

---

duke

*Duke breast cancer data*

---

**Description**

This data set details microarray experiment for breast cancer patients.

**Format**

A data frame with 86 rows and 501 variables, where the first variable is the label of estrogen receptor-positive/negative, and the remaining 500 variables are 500 gene.

**Details**

The binary variable Status is used to classify the patients into estrogen receptor-positive ( $y = 0$ ) and estrogen receptor-negative ( $y = 1$ ). The other variables contain the expression level of the considered genes.

## References

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, Jr., J.R. Marks and Joseph R. Nevins (2001) <doi:10.1073/pnas.201162998> Predicting the clinical status of human breast cancer by using gene expression profiles, Proceedings of the National Academy of Sciences of the USA, Vol 98(20), 11462-11467.

---

gen.data	<i>Generate simulated data</i>
----------	--------------------------------

---

## Description

Generate data for simulations under the generalized linear model and Cox model.

## Usage

```
gen.data(
  n,
  p,
  k = NULL,
  rho = 0,
  family = c("gaussian", "binomial", "poisson", "cox"),
  beta = NULL,
  cortype = 1,
  snr = 10,
  censoring = TRUE,
  c = 1,
  scal,
  sigma = 1,
  seed = 1
)
```

## Arguments

n	The number of observations.
p	The number of predictors of interest.
k	The number of nonzero coefficients in the underlying regression model. Can be omitted if beta is supplied.
rho	A parameter used to characterize the pairwise correlation in predictors. Default is 0.
family	The distribution of the simulated data. "gaussian" for gaussian data. "binomial" for binary data. "poisson" for count data. "cox" for survival data.
beta	The coefficient values in the underlying regression model.

cortype	The correlation structure. cortype = 1 denotes the exponential structure, where the covariance matrix has $(i, j)$ entry equals $\rho^{ i-j }$ . codecortype = 2 denotes the constant structure, where the $(i, j)$ entry of covariance matrix is $\rho$ for every $i \neq j$ and 1 elsewhere. cortype = 3 denotes the moving average structure. Details can be found below.
snr	A numerical value controlling the signal-to-noise ratio (SNR). The SNR is defined as the variance of $x\beta$ divided by the variance of a gaussian noise: $\frac{Var(x\beta)}{\sigma^2}$ . The gaussian noise $\epsilon$ is set with mean 0 and variance. The noise is added to the linear predictor $\eta = x\beta$ . Default is snr = 10. This option is invalid for cortype = 3.
censoring	Whether data is censored or not. Valid only for family = "cox". Default is TRUE.
c	The censoring rate. Default is 1.
scal	A parameter in generating survival time based on the Weibull distribution. Only used for the "cox" family.
sigma	A parameter used to control the signal-to-noise ratio. For linear regression, it is the error variance $\sigma^2$ . For logistic regression and Cox's model, the larger the value of sigma, the higher the signal-to-noise ratio. Valid only for cortype = 3.
seed	seed to be used in generating the random numbers.

## Details

We generate an  $n \times p$  random Gaussian matrix  $X$  with mean 0 and a covariance matrix with an exponential structure or a constant structure. For the exponential structure, the covariance matrix has  $(i, j)$  entry equals  $\rho^{|i-j|}$ . For the constant structure, the  $(i, j)$  entry of the covariance matrix is  $\rho$  for every  $i \neq j$  and 1 elsewhere. For the moving average structure, For the design matrix  $X$ , we first generate an  $n \times p$  random Gaussian matrix  $\bar{X}$  whose entries are i.i.d.  $\sim N(0, 1)$  and then normalize its columns to the  $\sqrt{n}$  length. Then the design matrix  $X$  is generated with  $X_j = \bar{X}_j + \rho(\bar{X}_{j+1} + \bar{X}_{j-1})$  for  $j = 2, \dots, p-1$ .

For family = "gaussian", the data model is

$$Y = X\beta + \epsilon.$$

The underlying regression coefficient  $\beta$  has uniform distribution  $[m, 100m]$ ,  $m = 5\sqrt{2\log(p)/n}$ .

For family= "binomial", the data model is

$$Prob(Y = 1) = \exp(X\beta + \epsilon)/(1 + \exp(X\beta + \epsilon)).$$

The underlying regression coefficient  $\beta$  has uniform distribution  $[2m, 10m]$ ,  $m = 5\sigma\sqrt{2\log(p)/n}$ .

For family = "poisson", the data is modeled to have an exponential distribution:

$$Y = Exp(\exp(X\beta + \epsilon)).$$

For family = "cox", the data model is

$$T = (-\log(S(t))/\exp(X\beta))^{1/scal}.$$

The centering time is generated from uniform distribution  $[0, c]$ , then we define the censor status as  $\delta = I\{T \leq C\}$ ,  $R = \min\{T, C\}$ . The underlying regression coefficient  $\beta$  has uniform distribution  $[2m, 10m]$ ,  $m = 5\sigma\sqrt{2\log(p)/n}$ . In the above models,  $\epsilon \sim N(0, \sigma^2)$ , where  $\sigma^2$  is determined by the snr.

**Value**

x	Design matrix of predictors.
y	Response variable.
Tbeta	The coefficients used in the underlying regression model.

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

**See Also**

[bess](#), [predict.bess](#).

**Examples**

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
SNR <- 10
cortype <- 1
seed <- 10
Data <- gen.data(n, p, k, rho, family = "gaussian", cortype = cortype, snr = SNR, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lm.bss <- bess(x, y, method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(x, y, type = "bsrr", method = "pgsection")
```

---

logLik.bess

*Extract the log-likelihood from a "bess.one" object.*

---

**Description**

This function returns the log-likelihood for the fitted models.

## Usage

```
## S3 method for class 'bess'  
logLik(object, best.model = TRUE, ...)
```

## Arguments

object	A "bess" object.
best.model	Whether only return the log-likelihood of the best model. Default is TRUE. If best.model = FALSE, the log-likelihood of the best models with model size and $\lambda$ in the original <code>s.list</code> and <code>lambda.list</code> (for method = "sequential") or in the iteration path (for method = "gsection", method = "pgsection", and method = "psequential") is returned.
...	additional arguments

## Details

The log-likelihood for the best model chosen by a certain information criterion or cross-validation corresponding to the call in `bess` or the best models with model size and  $\lambda$  in the original `s.list` and `lambda.list` (or the in the iteration path) can be returned. For "lm" fits it is assumed that the scale has been estimated (by maximum likelihood or REML), and all the constants in the log-likelihood are included.

## Value

A matrix or vector containing the log-likelihood for each model is returned. For `bess` objects fitted by `sequential` method, values in each row in the returned matrix corresponding to the model size in `s.list`, and each column the shrinkage parameters in `lambda.list`.

For `bess` objects fitted by `gsection`, `pgsection` and `psequential`, the returned vector contains log-likelihood for fitted models in each iteration. The coefficients of those model can be extracted from `beta.all` and `coef0.all` in the `bess` object.

## Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

## See Also

[bess](#), [summary.bess](#).

**Examples**

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
SNR <- 10
cortype <- 1
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", cortype = cortype, snr = SNR, seed = seed)
lm.bss <- bess(Data$x, Data$y, method = "sequential")

logLik(lm.bss)
logLik(lm.bss, best.model = FALSE)
```

---

plot.bess	<i>Produces a coefficient profile plot of the coefficient or loss function paths</i>
-----------	--

---

**Description**

Produces a coefficient profile plot of the coefficient or loss function paths

**Usage**

```
## S3 method for class 'bess'
plot(
  x,
  type = c("loss", "tune", "coefficients", "both"),
  breaks = TRUE,
  K = NULL,
  sign.lambda = 0,
  ...
)
```

**Arguments**

x	A "bess" object.
type	One of "loss", "tune", "coefficients", "both". This option is only valid for "bess" object obtained from "bss". If type = "loss" (type = "tune"), a path of loss function (corresponding information criterion or cross-validation loss) is provided. If type = "coefficients", it provides a coefficient profile plot of the coefficient. If type = "both", it combines the path of corresponding information criterion or cross-validation loss with the coefficient profile plot.

breaks	If TRUE, a vertical line is drawn at a specified break point in the coefficient paths.
K	Which break point should the vertical line be drawn at. Default is the optimal model size.
sign.lambda	A logical value indicating whether to show lambda on log scale. Default is 0. Valid for "bess" object obtained from "bsrr".
...	Other graphical parameters to plot

### Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

### References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

### See Also

[bess.](#)

### Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess(Data$x, Data$y, method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(Data$x, Data$y, type = "bsrr", method = "pgsection")

# generate plots
plot(lm.bss, type = "both", breaks = TRUE)
plot(lm.bsrr)
```

---

predict.bess            *make predictions from a "bess" object.*

---

### Description

Returns predictions from a fitted "bess" object.

### Usage

```
## S3 method for class 'bess'  
predict(object, newx, type = c("link", "response"), ...)
```

### Arguments

object	Output from the bess function.
newx	New data used for prediction. If omitted, the fitted linear predictors are used.
type	type = "link" gives the linear predictors for "binomial", "poisson" or "cox" models; for "gaussian" models it gives the fitted values. type = "response" gives the fitted probabilities for "binomial", fitted mean for "poisson" and the fitted relative-risk for "cox"; for "gaussian", type = "response" is equivalent to type = "link"
...	Additional arguments affecting the predictions produced.

### Value

The object returned depends on the types of family.

### Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

### References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

### See Also

[bess.](#)



**Examples**

```

#-----linear model-----#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lm.bss <- bess(x, y, method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(x, y, type = "bsrr", method = "pgsection")

pred.bss <- predict(lm.bss, newx = x_new)
pred.bsrr <- predict(lm.bsrr, newx = x_new)

#-----logistic model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "binomial", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
logi.bss <- bess(x, y, family = "binomial", method = "sequential", tune = "cv")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
logi.bsrr <- bess(x, y, type = "bsrr", tune="cv",
                  family = "binomial", lambda.list = lambda.list, method = "sequential")

pred.bss <- predict(logi.bss, newx = x_new)
pred.bsrr <- predict(logi.bsrr, newx = x_new)

#-----coxph model-----#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "cox", beta = Tbeta, scal = 10)

x <- Data$x[1:140, ]
y <- Data$y[1:140, ]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200, ]
cox.bss <- bess(x, y, family = "cox", method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
cox.bsrr <- bess(x, y, type = "bsrr", family = "cox", lambda.list = lambda.list)

pred.bss <- predict(cox.bss, newx = x_new)

```

```

pred.bsrr <- predict(cox.bsrr, newx = x_new)

#-----group selection-----#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, seed =10)
x <- Data$x
y <- Data$y

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.group <- bess(x, y, s.min=1, s.max = 8, type = "bss", group.index = group.index)
lm.groupbsrr <- bess(x, y, type = "bsrr", s.min = 1, s.max = 8, group.index = group.index)

pred.group <- predict(lm.group, newx = x_new)
pred.groupbsrr <- predict(lm.groupbsrr, newx = x_new)

```

---

```

print.bess          print method for a "bess" object

```

---

### Description

Print the primary elements of the "bess" object.

### Usage

```

## S3 method for class 'bess'
print(x, digits = max(5, getOption("digits") - 5), nonzero = FALSE, ...)

```

### Arguments

x	A "bess" object.
digits	Minimum number of significant digits to be used.
nonzero	Whether the output should only contain the non-zero coefficients.
...	additional print arguments

### Details

prints the fitted model and returns it invisibly.

### Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

### References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

**See Also**

[bess](#), [coef.bess](#).

**Examples**

```
# Generate simulated data
n = 200
p = 20
k = 5
rho = 0.4
seed = 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data = gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed=seed)
lm.bss = bess(Data$x, Data$y, method = "sequential")
lambda.list = exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr = bess(Data$x, Data$y, type = "bsrr", lambda.list = lambda.list, method = "sequential")

print(lm.bss)
print(lm.bsrr)
```

---

summary.bess

*summary method for a "bess.one" object*

---

**Description**

Print a summary of the "bess.one" object.

**Usage**

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

**Arguments**

object	A "bess" object.
...	additional print arguments

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

**See Also**

[bess.](#)

**Examples**

```
#-----linear model-----#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess(Data$x, Data$y, method = "sequential")
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bess(Data$x, Data$y, type = "bsrr", method = "pgsection")

summary(lm.bss)
summary(lm.bsrr)

#-----group selection-----#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, snr = 100, seed =10)

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
               rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.group <- bess(Data$x, Data$y, s.min=1, s.max = 8, type = "bss", group.index = group.index)
lm.groupbsrr <- bess(Data$x, Data$y, type = "bsrr", s.min = 1, s.max = 8, group.index = group.index)

summary(lm.group)
summary(lm.groupbsrr)

#-----summary for bess.one-----#
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bss <- bess.one(Data$x, Data$y, s = 5)
lm.bsrr <- bess.one(Data$x, Data$y, type = "bsrr", s = 5, lambda = 0.01)

summary(lm.bss)
summary(lm.bsrr)
```

**Description**

Gene expression data (500 gene probes for 120 samples) from the microarray experiments of mammalian eye tissue samples of Scheetz et al. (2006).

**Format**

A data frame with 120 rows and 501 variables, where the first variable is the expression level of TRIM32 gene, and the remaining 500 variables are 500 gene probes.

**Details**

In this study, laboratory rats (*Rattus norvegicus*) were studied to learn about gene expression and regulation in the mammalian eye. Inbred rat strains were crossed and tissue extracted from the eyes of 120 animals from the F2 generation. Microarrays were used to measure levels of RNA expression in the isolated eye tissues of each subject. Of the 31,000 different probes, 18,976 were detected at a sufficient level to be considered expressed in the mammalian eye. For the purposes of this analysis, we treat one of those genes, Trim32, as the outcome. Trim32 is known to be linked with a genetic disorder called Bardet-Biedl Syndrome (BBS): the mutation (P130S) in Trim32 gives rise to BBS.

**Note**

This data set contains 120 samples with 500 predictors. The 500 predictors are features with maximum marginal correlation to Trim32 gene.

**References**

T. Scheetz, k. Kim, R. Swiderski, A. Philp, T. Braun, K. Knudtson, A. Dorrance, G. DiBona, J. Huang, T. Casavant, V. Sheffield, E. Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. Proceedings of the National Academy of Sciences of the United States of America, 2006.

# Index

BeSS (BeSS-package), [2](#)  
bess, [3](#), [12](#), [15](#), [17](#), [20](#), [21](#), [23](#), [24](#), [27](#), [28](#)  
BeSS-package, [2](#)  
bess.one, [7](#), [10](#)  
  
coef.bess, [7](#), [12](#), [15](#), [27](#)  
  
deviance.bess, [16](#)  
duke, [17](#)  
  
gen.data, [18](#)  
  
logLik.bess, [20](#)  
  
plot.bess, [7](#), [22](#)  
predict.bess, [7](#), [12](#), [20](#), [24](#)  
print.bess, [15](#), [26](#)  
  
summary.bess, [7](#), [12](#), [17](#), [21](#), [27](#)  
  
trim32, [28](#)