

# Package ‘riskscores’

May 30, 2025

**Title** Optimized Integer Risk Score Models

**Version** 1.2.1

**Description** Implements an optimized approach to learning risk score models, where sparsity and integer constraints are integrated into the model-fitting process.

**URL** <https://github.com/hjeglington/riskscores>

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** dplyr, foreach, ggplot2, magrittr, pROC, stats

**Suggests** knitr, kableExtra, rmarkdown, doParallel

**VignetteBuilder** knitr, kableExtra

**Depends** R (>= 2.10)

**LazyData** true

**NeedsCompilation** no

**Author** Hannah Eglinton [aut, cre],  
Seehanah Tang [aut, aut],  
Alice Paul [aut, cph],  
Oscar Yan [aut],  
R Core Team [ctb, cph] (Copyright holder of Rinternals.h, R.h, lm.c,  
Applic.h, statsR.h, glm package),  
Robert Gentleman [ctb, cph] (Author and copyright holder of  
Rinternals.h),  
Ross Ihaka [ctb, cph] (Author and copyright holder of Rinternals.h),  
Simon Davies [ctb] (Author of glm.fit function (modified in  
cv\_risk\_mod.R)),  
Thomas Lumley [ctb] (Author of glm.fit function (modified in  
cv\_risk\_mod.R))

**Maintainer** Hannah Eglinton <eglingtonh@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-05-30 21:00:02 UTC

Contents

breastcancer . . . . .	2
clip_exp_vals . . . . .	3
coef.risk_mod . . . . .	3
cv_risk_mod . . . . .	4
get_metrics . . . . .	6
get_metrics_internal . . . . .	7
get_risk . . . . .	8
get_score . . . . .	8
plot.cv_risk_mod . . . . .	9
plot.risk_mod . . . . .	9
plot_accuracy.cv_risk_mod . . . . .	10
plot_deviance.cv_risk_mod . . . . .	11
predict.risk_mod . . . . .	11
randomized_rounding . . . . .	12
risk_mod . . . . .	12
risk_mod_random_start . . . . .	15
stratify_folds . . . . .	16
summary.risk_mod . . . . .	16
<b>Index</b>	<b>18</b>

---

breastcancer	<i>Breast tissue biopsy data</i>
--------------	----------------------------------

---

Description

The Breast Cancer Wisconsin dataset from the UCI machine learning repository records the measurements from breast tissue biopsies. The outcome of interest is whether the sample was benign or malignant.

Usage

breastcancer

Format

- breastcancer:
- A data frame with 683 rows and 10 columns:
- Benign** 1 for malignant, 0 for benign
  - ClumpThickness** Clump thickness on an integer scale from 1 to 10
  - UniformityOfCellSize** Uniformity of cell size on an integer scale from 1 to 10
  - UniformityOfCellShape** Uniformity of cell shape on an integer scale from 1 to 10
  - MarginalAdhesion** Marginal adhesion on an integer scale from 1 to 10
  - SingleEpithelialCellSize** Single epithelial cell size on an integer scale from 1 to 10
  - BareNuclei** Bare nuclei on an integer scale from 1 to 10

**BlandChromatin** Bland chromatin on an integer scale from 1 to 10

**NormalNucleoli** Normal nucleoli on an integer scale from 1 to 10

**Mitosis** Mitosis on an integer scale from 1 to 10

### Source

<https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original>

---

clip_exp_vals	<i>Clip Values</i>
---------------	--------------------

---

### Description

Clip values prior to exponentiation to avoid numeric errors.

### Usage

```
clip_exp_vals(x)
```

### Arguments

x                      Numeric vector.

### Value

Input vector x with all values between -709.78 and 709.78.

### Examples

```
clip_exp_vals(710)
```

---

coef.risk_mod	<i>Extract Model Coefficients</i>
---------------	-----------------------------------

---

### Description

Extracts a vector of model coefficients (both nonzero and zero) from a "risk\_mod" object. Equivalent to accessing the beta attribute of a "risk\_mod" object.

### Usage

```
## S3 method for class 'risk_mod'  
coef(object, ...)
```

**Arguments**

object            An object of class "risk\_mod", usually a result of a call to `risk_mod()`.  
 ...              Additional arguments.

**Value**

Numeric vector with coefficients.

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod <- risk_mod(X, y, lambda0 = 0.01)
coef(mod)
```

---

cv\_risk\_mod

*Run Cross-Validation to Tune Lambda0*


---

**Description**

Runs k-fold cross-validation on a grid of  $\lambda_0$  values. Records class accuracy and deviance for each  $\lambda_0$ . Returns an object of class "cv\_risk\_mod".

**Usage**

```
cv_risk_mod(
  X,
  y,
  weights = NULL,
  beta = NULL,
  a = -10,
  b = 10,
  max_iters = 10000,
  tol = 1e-05,
  nlambdas = 25,
  lambda_min_ratio = ifelse(nrow(X) < ncol(X), 0.01, 1e-04),
  lambda0 = NULL,
  nfolds = 10,
  foldids = NULL,
  parallel = FALSE,
  shuffle = TRUE,
  seed = NULL,
  method = "annealscore"
)
```

**Arguments**

<code>X</code>	Input covariate matrix with dimension $n \times p$ ; every row is an observation.
<code>y</code>	Numeric vector for the (binomial) response variable.
<code>weights</code>	Numeric vector of length $n$ with weights for each observation. Unless otherwise specified, default will give equal weight to each observation.
<code>beta</code>	Starting numeric vector with $p$ coefficients. Default starting coefficients are rounded coefficients from a logistic regression model.
<code>a</code>	Integer lower bound for coefficients (default: -10).
<code>b</code>	Integer upper bound for coefficients (default: 10).
<code>max_iters</code>	Maximum number of iterations (default: 10000).
<code>tol</code>	Tolerance for convergence (default: 1e-5).
<code>nlambda</code>	Number of lambda values to try (default: 25).
<code>lambda_min_ratio</code>	Smallest value for lambda, as a fraction of lambda_max (the smallest value for which all coefficients are zero). The default depends on the sample size ( $n$ ) relative to the number of variables ( $p$ ). If $n > p$ , the default is 0.0001, close to zero. If $n < p$ , the default is 0.01.
<code>lambda0</code>	Optional sequence of lambda values. By default, the function will derive the lambda0 sequence based on the data (see lambda_min_ratio).
<code>nfolds</code>	Number of folds, implied if foldids provided (default: 10).
<code>foldids</code>	Optional vector of values between 1 and nfolds.
<code>parallel</code>	If TRUE, parallel processing (using <a href="#">foreach</a> ) is implemented during cross-validation to increase efficiency (default: FALSE). User must first register parallel backend with a function such as <a href="#">registerDoParallel</a> .
<code>shuffle</code>	Whether order of coefficients is shuffled during coordinate descent (default: TRUE).
<code>seed</code>	An integer that is used as argument by <code>set.seed()</code> for offsetting the random number generator. Default is to not set a particular randomization seed.
<code>method</code>	A string that specifies which method ("riskcd" or "annealscore") to run (default: "annealscore")

**Value**

An object of class "cv\_risk\_mod" with the following attributes:

<code>results</code>	Dataframe containing a summary of deviance, accuracy, and auc for each value of lambda0 (mean and SD). Also includes the number of nonzero coefficients that are produced by each lambda0 when fit on the full data.
<code>lambda_min</code>	Numeric value indicating the lambda0 that resulted in the highest mean auc
<code>lambda_1se</code>	Numeric value indicating the largest lambda0 that had a mean auc within one standard error of lambda_min.

get\_metrics

*Get Model Metrics***Description**

Calculates a risk model's accuracy, sensitivity, and specificity given a set of data.

**Usage**

```
get_metrics(
  mod,
  X = NULL,
  y = NULL,
  weights = NULL,
  threshold = NULL,
  threshold_type = c("response", "score")
)
```

**Arguments**

mod	An object of class <code>risk_mod</code> , usually a result of a call to <code>risk_mod()</code> .
X	Input covariate matrix with dimension $n \times p$ ; every row is an observation.
y	Numeric vector for the (binomial) response variable.
weights	Numeric vector of length $n$ with weights for each observation. Unless otherwise specified, default will give equal weight to each observation.
threshold	Numeric vector of classification threshold values used to calculate the accuracy, sensitivity, and specificity of the model. Defaults to a range of risk probability thresholds from 0.1 to 0.9 by 0.1.
threshold_type	Defines whether the threshold vector contains risk probability values ("response") or threshold values expressed as scores from the risk score model ("score"). Default: "response".

**Value**

Data frame with accuracy, sensitivity, and specificity for each threshold.

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod <- risk_mod(X, y)
get_metrics(mod, X, y)

get_metrics(mod, X, y, threshold = c(150, 175, 200), threshold_type = "score")
```

---

get\_metrics\_internal     *Get Model Metrics for a Single Threshold*


---

## Description

Calculates a risk model's deviance, accuracy, sensitivity, and specificity given a set of data and a threshold value.

## Usage

```
get_metrics_internal(
  mod,
  X = NULL,
  y = NULL,
  weights = NULL,
  threshold = 0.5,
  threshold_type = c("response", "score")
)
```

## Arguments

mod	An object of class <code>risk_mod</code> , usually a result of a call to <code>risk_mod()</code> .
X	Input covariate matrix with dimension $n \times p$ ; every row is an observation.
y	Numeric vector for the (binomial) response variable.
weights	Numeric vector of length $n$ with weights for each observation. Unless otherwise specified, default will give equal weight to each observation.
threshold	Numeric vector of classification threshold values used to calculate the accuracy, sensitivity, and specificity of the model. Defaults to a range of risk probability thresholds from 0.1 to 0.9 by 0.1.
threshold_type	Defines whether the threshold vector contains risk probability values ("response") or threshold values expressed as scores from the risk score model ("score"). Default: "response".

## Value

List with deviance (dev), accuracy (acc), sensitivity (sens), specificity (spec), and auc.

---

get_risk	<i>Calculate Risk Probability from Score</i>
----------	--

---

**Description**

Returns the risk probabilities for the provided score value(s).

**Usage**

```
get_risk(object, score)
```

**Arguments**

object	An object of class "risk_mod", usually a result of a call to <a href="#">risk_mod()</a> .
score	Numeric vector with score value(s).

**Value**

Numeric vector with the same length as score.

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod <- risk_mod(X, y)
get_risk(mod, score = c(1, 10, 20))
```

---

get_score	<i>Calculate Score from Risk Probability</i>
-----------	--

---

**Description**

Returns the score(s) for the provided risk probabilities.

**Usage**

```
get_score(object, risk)
```

**Arguments**

object	An object of class "risk_mod", usually a result of a call to <a href="#">risk_mod()</a> .
risk	Numeric vector with probability value(s).



**Value**

Numeric vector with the same length as risk.

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod <- risk_mod(X, y)
get_score(mod, risk = c(0.25, 0.50, 0.75))
```

---

plot.cv_risk_mod	<i>Plot Risk Score Cross-Validation Results</i>
------------------	---

---

**Description**

Plots the mean auc for each  $\lambda_{0}$  tested during cross-validation.

**Usage**

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

**Arguments**

x	An object of class "cv_risk_mod", usually a result of a call to <code>cv_risk_mod()</code> .
...	Additional arguments affecting the plot produced

**Value**

Object of class "ggplot".

---

plot.risk_mod	<i>Plot Risk Score Model Curve</i>
---------------	------------------------------------

---

**Description**

Plots the linear regression equation associated with the integer risk score model. Plots the scores on the x-axis and risk on the y-axis.

**Usage**

```
## S3 method for class 'risk_mod'
plot(x, score_min = NULL, score_max = NULL, ...)
```

**Arguments**

x	An object of class "risk_mod", usually a result of a call to <a href="#">risk_mod()</a> .
score_min	The minimum score displayed on the x-axis. The default is the minimum score predicted from model's training data.
score_max	The maximum score displayed on the x-axis. The default is the maximum score predicted from model's training data.
...	Additional arguments affecting the plot produced

**Value**

Object of class "ggplot".

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])
mod <- risk_mod(X, y, lambda0 = 0.01)

plot(mod)
```

---

```
plot_accuracy.cv_risk_mod
```

*Plot Risk Score Cross-Validation Results*

---

**Description**

Plots the mean accuracy for each  $\lambda_{0}$  tested during cross-validation.

**Usage**

```
plot_accuracy.cv_risk_mod(x, ...)
```

**Arguments**

x	An object of class "cv_risk_mod", usually a result of a call to <a href="#">cv_risk_mod()</a> .
...	Additional arguments affecting the plot produced

**Value**

Object of class "ggplot".

---

plot\_deviance.cv\_risk\_mod

*Plot Risk Score Cross-Validation Results*


---

**Description**

Plots the mean deviance for each  $\lambda_{0}$  tested during cross-validation.

**Usage**

```
plot_deviance.cv_risk_mod(x, ...)
```

**Arguments**

x	An object of class "cv_risk_mod", usually a result of a call to <code>cv_risk_mod()</code> .
...	Additional arguments affecting the plot produced

**Value**

Object of class "ggplot".

---

predict.risk\_mod

*Predict Method for Risk Model Fits*


---

**Description**

Obtains predictions from risk score models.

**Usage**

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

**Arguments**

object	An object of class "risk_mod", usually a result of a call to <code>risk_mod()</code> .
newx	Optional matrix of new values for X for which predictions are to be made. If omitted, the fitted values are used.
type	The type of prediction required. The default ("link") is on the scale of the predictors (i.e. log-odds); the "response" type is on the scale of the response variable (i.e. risk probabilities); the "score" type returns the risk score calculated from the integer model.
...	Additional arguments.

**Value**

Numeric vector of predicted values.

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])
mod <- risk_mod(X, y, lambda0 = 0.01)
predict(mod, type = "link")[1]
predict(mod, type = "response")[1]
predict(mod, type = "score")[1]
```

---

randomized_rounding	<i>Randomly round the initialized coefficients before coordinate descent</i>
---------------------	--

---

**Description**

Round each LR coefficient based on its decimal value. The decimal is the probability of rounding the coefficient up to the next integer

**Usage**

```
randomized_rounding(beta)
```

**Arguments**

beta	Numeric vector or logistic regression coefficients initialized before cyclical coordinate descent in <code>risk_mod()</code> . The first element is the intercept and is not modified.
------	--

**Value**

A numeric vector with randomized rounding (apart from the first element).

---

risk_mod	<i>Fit an Integer Risk Score Model</i>
----------	--

---

**Description**

Fits an optimized integer risk score model using a heuristic algorithm. Returns an object of class "risk\_mod".

**Usage**

```

risk_mod(
  X,
  y,
  gamma = NULL,
  beta = NULL,
  weights = NULL,
  n_train_runs = 1,
  lambda0 = 0,
  a = -10,
  b = 10,
  max_iters = 10000,
  tol = 1e-05,
  shuffle = TRUE,
  seed = NULL,
  method = "annealscore"
)

```

**Arguments**

<code>X</code>	Input covariate matrix with dimension $n \times p$ ; every row is an observation.
<code>y</code>	Numeric vector for the (binomial) response variable.
<code>gamma</code>	Starting value to rescale coefficients for prediction (optional).
<code>beta</code>	Starting numeric vector with $p$ coefficients. Default starting coefficients are rounded coefficients from a logistic regression model.
<code>weights</code>	Numeric vector of length $n$ with weights for each observation. Unless otherwise specified, default will give equal weight to each observation.
<code>n_train_runs</code>	A positive integer representing the number of times to initialize and train the model, returning the run with the lowest objective function for the training data.
<code>lambda0</code>	Penalty coefficient for L0 term (default: 0). See <code>cv_risk_mod()</code> for <code>lambda0</code> tuning.
<code>a</code>	Integer lower bound for coefficients (default: -10).
<code>b</code>	Integer upper bound for coefficients (default: 10).
<code>max_iters</code>	Maximum number of iterations (default: 10000).
<code>tol</code>	Tolerance for convergence (default: 1e-5).
<code>shuffle</code>	Whether order of coefficients is shuffled during coordinate descent (default: TRUE).
<code>seed</code>	An integer that is used as argument by <code>set.seed()</code> for offsetting the random number generator. Default is to not set a particular randomization seed.
<code>method</code>	A string that specifies which method ("riskcd" or "annealscore") to run (default: "annealscore")

## Details

This function uses either a cyclical coordinate descent algorithm or simulated annealing algorithm to solve the following optimization problem.

$$\min_{\alpha, \beta} \frac{1}{n} \sum_{i=1}^n (\gamma y_i x_i^T \beta - \log(1 + \exp(\gamma x_i^T \beta))) + \lambda_0 \sum_{j=1}^p 1(\beta_j \neq 0)$$

$$l \leq \beta_j \leq u \quad \forall j = 1, 2, \dots, p$$

$$\beta_j \in \mathbb{Z} \quad \forall j = 1, 2, \dots, p$$

$$\beta_0, \gamma \in \mathbb{R}$$

These constraints ensure that the model will be sparse and include only integer coefficients.

## Value

An object of class "risk\_mod" with the following attributes:

gamma	Final scalar value.
beta	Vector of integer coefficients.
glm_mod	Logistic regression object of class "glm" (see <a href="#">stats::glm</a> ).
X	Input covariate matrix.
y	Input response vector.
weights	Input weights.
lambda0	Input lambda0 value.
model_card	Dataframe displaying the nonzero integer coefficients (i.e. "points") of the risk score model.
score_map	Dataframe containing a column of possible scores and a column with each score's associated risk probability.

## Examples

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod1 <- risk_mod(X, y)
mod1$model_card

mod2 <- risk_mod(X, y, lambda0 = 0.01,)
mod2$model_card

mod3 <- risk_mod(X, y, lambda0 = 0.01, a = -5, b = 5, method = "riskcd")
mod3$model_card
```

---

`risk_mod_random_start` *Run risk model with random start*

---

## Description

Runs `nstart` iterations of `risk_mod()`, each with a different warm start, and selects the best model. Each coefficient start is randomly selected as -1, 0, or 1.

## Usage

```
risk_mod_random_start(
  X,
  y,
  weights = NULL,
  lambda0 = 0,
  a = -10,
  b = 10,
  max_iters = 100,
  tol = 1e-05,
  seed = NULL,
  nstart = 5
)
```

## Arguments

<code>X</code>	Input covariate matrix with dimension $n \times p$ ; every row is an observation.
<code>y</code>	Numeric vector for the (binomial) response variable.
<code>weights</code>	Numeric vector of length $n$ with weights for each observation. Unless otherwise specified, default will give equal weight to each observation.
<code>lambda0</code>	Penalty coefficient for L0 term (default: 0). See <a href="#">cv_risk_mod()</a> for <code>lambda0</code> tuning.
<code>a</code>	Integer lower bound for coefficients (default: -10).
<code>b</code>	Integer upper bound for coefficients (default: 10).
<code>max_iters</code>	Maximum number of iterations (default: 10000).
<code>tol</code>	Tolerance for convergence (default: 1e-5).
<code>seed</code>	An integer that is used as argument by <code>set.seed()</code> for offsetting the random number generator. Default is to not set a particular randomization seed.
<code>nstart</code>	Number of different random starts to try (default: 5).

---

stratify_folds	<i>Generate Stratified Fold IDs</i>
----------------	-------------------------------------

---

**Description**

Returns a vector of fold IDs that preserves class proportions.

**Usage**

```
stratify_folds(y, nfolds = 10, seed = NULL)
```

**Arguments**

y	Numeric vector for the (binomial) response variable.
nfolds	Number of folds (default: 10).
seed	An integer that is used as argument by <code>set.seed()</code> for offsetting the random number generator. Default is to not set a particular randomization seed.

**Value**

Numeric vector with the same length as y.

**Examples**

```
y <- rbinom(100, 1, 0.3)
foldids <- stratify_folds(y, nfolds = 5)
table(y, foldids)
```

---

summary.risk_mod	<i>Summarize Risk Model Fit</i>
------------------	---------------------------------

---

**Description**

Prints text that summarizes "risk\_mod" objects.

**Usage**

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

**Arguments**

object	An object of class "risk_mod", usually a result of a call to <code>risk_mod()</code> .
...	Additional arguments affecting the summary produced.



**Value**

Printed text with intercept, nonzero coefficients, gamma, lambda, and deviance

**Examples**

```
y <- breastcancer[[1]]
X <- as.matrix(breastcancer[,2:ncol(breastcancer)])

mod <- risk_mod(X, y, lambda0 = 0.01)
summary(mod)
```

# Index

## \* datasets

breastcancer, [2](#)

breastcancer, [2](#)

clip\_exp\_vals, [3](#)

coef.risk\_mod, [3](#)

cv\_risk\_mod, [4](#)

cv\_risk\_mod(), [9–11](#), [13](#), [15](#)

foreach, [5](#)

get\_metrics, [6](#)

get\_metrics\_internal, [7](#)

get\_risk, [8](#)

get\_score, [8](#)

plot.cv\_risk\_mod, [9](#)

plot.risk\_mod, [9](#)

plot\_accuracy.cv\_risk\_mod, [10](#)

plot\_deviance.cv\_risk\_mod, [11](#)

predict.risk\_mod, [11](#)

randomized\_rounding, [12](#)

registerDoParallel, [5](#)

risk\_mod, [12](#)

risk\_mod(), [4](#), [6–8](#), [10](#), [11](#), [16](#)

risk\_mod\_random\_start, [15](#)

stats::glm, [14](#)

stratify\_folds, [16](#)

summary.risk\_mod, [16](#)