

# Package: binst (via r-universe)

January 8, 2025

**Type** Package

**Title** Data Preprocessing, Binning for Classification and Regression

**Version** 0.2.1

**Date** 2018-01-05

**Author** Chapman Siu

**Maintainer** Chapman Siu <chpmn.siu@gmail.com>

**Description** Various supervised and unsupervised binning tools including using entropy, recursive partition methods and clustering.

**LazyData** TRUE

**Imports** stats, rpart

**Suggests** discretization, Formula, testthat, BMMtools, earth

**RoxygenNote** 5.0.1

**License** MIT + file LICENSE

**URL** <https://github.com/jules-and-dave/binst>

**Repository** <https://sourdoughcat.r-universe.dev>

**RemoteUrl** <https://github.com/sourdoughcat/binst>

**RemoteRef** HEAD

**RemoteSha** 00e5e2a78f37160bd51cc9acdcf5c8e9f3c02705

**RemoteSubdir** R

## Contents

create_bins	2
create_breaks	2
create_dtbreks	3
create_earthbreaks	4
create_jenksbreaks	5
create_kmeansbreaks	5
create_mdldbks	6
get_control	7

**Index****8**


---

create_bins	<i>Creates bins given breaks</i>
-------------	----------------------------------

---

**Description**

Creates bins given breaks

**Usage**

```
create_bins(x, breaks, method = "cuts")
```

**Arguments**

x	X is a numeric vector which is to be discretized
breaks	Breaks are the breaks for the vector X to be broken at. This excludes endpoints
method	the approach to bin the variable, can either be cuts or hinge.

**Value**

A vector same length as X is returned with the numeric discretization

**See Also**

[create\\_breaks](#)

**Examples**

```
create_bins(1:10, c(3, 5))
```

---

create_breaks	<i>A convenience functon for creating breaks with various methods.</i>
---------------	--

---

**Description**

A convenience functon for creating breaks with various methods.

**Usage**

```
create_breaks(x, y = NULL, method = "kmeans", control = NULL, ...)
```

**Arguments**

x	X is a numeric vector to be discretized
y	Y is the response vector used for calculating metrics for discretization
method	Method is the type of discretization approach used. Possible methods are: "dt", "entropy", "kmeans", "jenks"
control	Control is used for optional parameters for the method. It is a list of optional parameters for the function
...	instead of passing a list into control, arguments can be parsed as is.

**Value**

A vector containing the breaks

**See Also**

[get\\_control](#), [create\\_bins](#)

**Examples**

```
kmeans_breaks <- create_breaks(1:10)
create_bins(1:10, kmeans_breaks)

# passing the k means parameter "centers" = 4
kmeans_breaks <- create_breaks(1:10, list(centers=4))
create_bins(1:10, kmeans_breaks)

entropy_breaks <- create_breaks(1:10, rep(c(1,2), each = 5), method="entropy")
create_bins(1:10, entropy_breaks)

dt_breaks <- create_breaks(iris$Sepal.Length, iris$Species, method="dt")
create_bins(iris$Sepal.Length, dt_breaks)
```

---

create_dtbreaks	<i>Create breaks using decision trees (recursive partitioning)</i>
-----------------	--

---

**Description**

Create breaks using decision trees (recursive partitioning)

**Usage**

```
create_dtbreaks(x, y, control = NULL)
```

**Arguments**

x	X is a numeric vector to be discretized
y	Y is the response vector used for calculating metrics for discretization
control	Control is used for optional parameters for the method

**Value**

A vector containing the breaks

**See Also**

[create\\_breaks](#)

**Examples**

```
dt_breaks <- create_breaks(iris$Sepal.Length, iris$Species, method="dt")
create_bins(iris$Sepal.Length, dt_breaks)
```

---

create\_earthbreaks      *Create breaks using earth (i.e. MARS)*

---

**Description**

Create breaks using earth (i.e. MARS)

**Usage**

```
create_earthbreaks(x, y, control = NULL)
```

**Arguments**

x	X is a numeric vector to be discretized
y	Y is the response vector used for calculating metrics for discretization
control	Control is used for optional parameters for the method

**Value**

A vector containing the breaks

**See Also**

[create\\_breaks](#)

**Examples**

```
earth_breaks <- create_breaks(x=iris$Sepal.Length, y=iris$Sepal.Width, method="earth")
create_bins(iris$Sepal.Length, earth_breaks)
```

---

create\_jenksbreaks     *Create Jenks breaks*

---

**Description**

Create Jenks breaks

**Usage**

```
create_jenksbreaks(x, control = NULL)
```

**Arguments**

x                    X is a numeric vector to be discretized  
control              Control is used for optional parameters for the method

**Value**

A vector containing the breaks

**See Also**

[create\\_breaks](#)

**Examples**

```
jenks_breaks <- create_breaks(1:10, method="jenks")  
create_bins(1:10, jenks_breaks)
```

---

create\_kmeansbreaks     *Create kmeans breaks.*

---

**Description**

Create kmeans breaks.

**Usage**

```
create_kmeansbreaks(x, control = NULL)
```

**Arguments**

x                    X is a numeric vector to be discretized  
control              Control is used for optional parameters for the method

**Value**

A vector containing the breaks

**See Also**

[create\\_breaks](#)

**Examples**

```
kmeans_breaks <- create_breaks(1:10)
create_bins(1:10, kmeans_breaks)
```

---

create_mdldbks	<i>Create breaks using mdlp</i>
----------------	---------------------------------

---

**Description**

Create breaks using mdlp

**Usage**

```
create_mdldbks(x, y)
```

**Arguments**

x	X is a numeric vector to be discretized
y	Y is the response vector used for calculating metrics for discretization

**Value**

A vector containing the breaks

**See Also**

[create\\_breaks](#)

**Examples**

```
entropy_breaks <- create_breaks(1:10, rep(c(1,2), each = 5), method="entropy")
create_bins(1:10, entropy_breaks)
```

---

`get_control`                    *gets the default parameters for each method.*

---

**Description**

gets the default parameters for each method.

**Usage**

```
get_control(method = "kmeans", control = NULL)
```

**Arguments**

<code>method</code>	Method is the type of discretization approach used
<code>control</code>	Control are the controls for the algorithm

**Value**

List of default parameters

# Index

`create_bins`, [2](#), [3](#)  
`create_breaks`, [2](#), [2](#), [4-6](#)  
`create_dtbreaks`, [3](#)  
`create_earthbreaks`, [4](#)  
`create_jenksbreaks`, [5](#)  
`create_kmeansbreaks`, [5](#)  
`create_mdldbks`, [6](#)  
  
`get_control`, [3](#), [7](#)