

# Package ‘mars’

April 12, 2022

**Title** Fitting a Multivariate Adaptive Regression Splines (MARS)

**Version** 0.0.0.9000

**Description** This package provides function to fit a Multivariate adaptive regression splines (MARS) model. Four methods are also provided to apply on the mars object.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Imports** stats, ISLR, grDevices, graphics, AmesHousing, devtools

**Suggests** rmarkdown,  
knitr,  
testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**LazyData** true

## R topics documented:

mars . . . . .	1
mars.control . . . . .	3
marstestdata . . . . .	3
plot.mars . . . . .	4
predict.mars . . . . .	5
print.mars . . . . .	5
summary.mars . . . . .	6
<b>Index</b>	7

---

`mars`*Fit a MARS model*

---

## Description

Build a regression model using the techniques in Friedman's papers "Multivariate Adaptive Regression Splines (MARS)".

## Usage

```
mars(formula, data, control)
```

## Arguments

<code>formula</code>	An R formula
<code>data</code>	A data frame containing the data for the model
<code>control</code>	An object of class <code>mars.control</code>

## Details

The algorithm will search for, and discover, nonlinearities in the data that help maximize predictive accuracy. Multivariate adaptive regression splines (MARS), an algorithm that essentially creates a piecewise linear model which provides an intuitive stepping block into nonlinearity after grasping the concept of linear regression and other intrinsically linear models.

## Value

An object of class `mars` that includes the final regression and a description of the basis functions. There are `plot`, `predict`, `summary` and `print` methods for `mars` object.

## Author(s)

Tauseef Kashtwari, Promit Chowdhury, Ibraheem Azad

## References

Jerome H. Friedman. "Multivariate Adaptive Regression Splines." *Ann. Statist.* 19(1) 1 - 67, March, 1991. <https://doi.org/10.1214/aos/1176347963>.

## See Also

`mars.control()` for constructing the control object  
`plot.mars()` for plotting the results  
`predict.mars()` for predictions  
`summary.mars()` for summarizing mars objects  
`print.mars()` for printing mars objects  
Other method: `plot.mars()`, `predict.mars()`, `print.mars()`, `summary.mars()`

## Examples

```
mm <- mars(y~., data=mars::marstestdata)
```

---

<code>mars.control</code>	<i>Mars Control Object</i>
---------------------------	----------------------------

---

**Description**

Constructor for `mars.control` objects

This function constructs a `mars.control` object that specifies parameters used in the model fitting procedure

**Usage**

```
mars.control(Mmax = 2, d = 3, trace = FALSE)
```

**Arguments**

<code>Mmax</code>	Maximum number of basis functions. Should be an even integer. Default is 2.
<code>d</code>	The coefficient in the penalty term of the generalized cross validation measure. Default is 3.
<code>trace</code>	Should we print status information about the fitting? Default is FALSE.

**Value**

a `mars.control` object

**Examples**

```
mc <- mars.control(Mmax=10)
```

---

<code>marstestdata</code>	<i>A test dataset for the MARS package</i>
---------------------------	--

---

**Description**

A dataset of size  $N=100$  with  $n=10$  explanatory variables, and a response variable that depends on only the first two explanatory variables.

**Usage**

```
marstestdata
```

**Format**

A data frame with 100 rows and 11 variables:

**y** response variable

**x1** explanatory variable

**x2** explanatory variable

**x3** explanatory variable

**x4** explanatory variable  
**x5** explanatory variable  
**x6** explanatory variable  
**x7** explanatory variable  
**x8** explanatory variable  
**x9** explanatory variable  
**x10** explanatory variable

---

plot.mars

*Plot method for the mars object*


---

### Description

Plots the fitted basis functions that depend on explanatory variable(main effects) or two explanatory variables (two-way interactions).

### Usage

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

### Arguments

x	A mars object
...	Additional arguments for plotting

### Value

Four diagnose plots:

- Residuals vs Fitted value
- Normal Q-Q plot
- Squared standardized residuals vs. Fitted value
- Cook's distance.

### See Also

Other method: [mars\(\)](#), [predict.mars\(\)](#), [print.mars\(\)](#), [summary.mars\(\)](#)

### Examples

```
mm <- mars(y~., data=mars::marstestdata)
plot(mm)
```

---

predict.mars	<i>Predict method for the MARS object</i>
--------------	---

---

**Description**

Provides predictions from the results of the MARS object's basis functions.

**Usage**

```
## S3 method for class 'mars'  
predict(object, newdata, ...)
```

**Arguments**

object	A mars object
newdata	New data <i>*optional</i>
...	Additional arguments

**Value**

If newdata is missing, fitted values are returned; otherwise, predicted values on the new data are returned

**See Also**

Other method: [mars\(\)](#), [plot.mars\(\)](#), [print.mars\(\)](#), [summary.mars\(\)](#)

**Examples**

```
mm <- mars(y~x1+x2, data=mars::marstestdata)  
pred <- predict(mm, newdata=data.frame(x1=rnorm(100), x2=rnorm(100)))
```

---

print.mars	<i>Print method for the MARS object</i>
------------	---

---

**Description**

Prints the function call of MARS object, then each coefficient of selected variables.

**Usage**

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

**Arguments**

x	A mars object
...	Additional arguments

**Value**

Function call and coefficients of the MARS object

**See Also**

Other method: [mars\(\)](#), [plot.mars\(\)](#), [predict.mars\(\)](#), [summary.mars\(\)](#)

**Examples**

```
mm <- mars(y~., data=mars::marstestdata)
print(mm)
```

---

summary.mars	<i>Summary for MARS object</i>
--------------	--------------------------------

---

**Description**

Prints a summary statistic of the MARS object, from the data and fitted model.

**Usage**

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

**Arguments**

object	A mars object
...	Additional arguments

**Value**

A summary statistic of the MARS object including:

- Function call
- Quarterly summary of residual values
- Coefficients of variables
- Residual standard error

**See Also**

Other method: [mars\(\)](#), [plot.mars\(\)](#), [predict.mars\(\)](#), [print.mars\(\)](#)

**Examples**

```
mm <- mars(y~., data=mars::marstestdata)
summary(mm)
```

# Index

## \* datasets

marstestdata, 3

## \* method

mars, 1

plot.mars, 4

predict.mars, 5

print.mars, 5

summary.mars, 6

mars, 1, 4-6

mars.control, 3

mars.control(), 2

marstestdata, 3

plot.mars, 2, 4, 5, 6

plot.mars(), 2

predict.mars, 2, 4, 5, 6

predict.mars(), 2

print.mars, 2, 4, 5, 5, 6

print.mars(), 2

summary.mars, 2, 4, 5, 6, 6

summary.mars(), 2