

# Package ‘mars’

April 11, 2022

**Version** 0.0.0.88888888

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

**Author** Siyul Sam Byun, So Yeon Park, Dawu Liu

**Maintainer** Siyul Sam Byun <siyulb@sfu.ca>

So Yeon Park <syp7@sfu.ca>

Dawu Liu <dla189@sfu.ca>

**Description** Build regression models using the techniques in Friedman's paper ``Multivariate Adaptive Regression Splines".

**References** Jerame H. Friedman. ``Multivariate Adaptive Regression Splines." Ann, Statist. 19 (1) 1 - 67, March, 1991 \url{https://doi.org/10.1214/aos/1176347963}

**Example** test <- mars(y~x1+x2, data=mars::marstestdata, control=mars.control(Mmax=2))

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Imports** stats,

graphics

**Suggests** knitr,

rmarkdown,

testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Depends** R (>= 3.5.0)

## R topics documented:

mars . . . . .	2
mars.control . . . . .	3
marstestdata . . . . .	4
plot.mars . . . . .	4
predict.mars . . . . .	5
print.mars . . . . .	5
summary.mars . . . . .	6
<b>Index</b>	7

mars

*Multivariate Adaptive Regression Splines (MARS)***Description**

MARS (Multivariate Adaptive Regression Splines) is a regression analysis that automatically splits and better linear fit into non-linear models.

**Usage**

```
mars(formula, data, control=NULL, ...)
```

**Arguments**

formula	an R formula
data	a data frame containing the data
control	an object of class 'mars.control'
...	further arguments

**Details**

MARS - forward stepwise algorithm: Forward stepwise fits the data nicely by adjusting the coefficient values as well as deriving a proper set of basis functions. Forward stepwise produces basis functions which do not have zero pairwise product expectations. The advantage of this is that basis functions, except for  $B_0$ , can be removed without leaving a hole in the predictor space.

MARS - backward stepwise algorithm: Jstar(J\*) backward stepwise is all of the basis function set and derived from forward stepwise. One basis function is deleted at each iteration of the outer loop, while the inner loop chooses which one to be deleted. The chosen basis function to be deleted is either the one improving the model by its removal or degrading the model the least by the removal. Constant basis function  $B_0(x) = 1$  is not considered for elimination. A sequence of (Mmax - 1) models is what is constructed from backward stepwise, and the current sequence has one less basis function than that of the previous sequence. Once the iteration is terminated, the users are left with the best model.

**Value**

an S3 object of class mars that includes the final regression model and a description of the basis functions which are constructed by the hinge functions

**Author(s)**

Siyul Sam Byun, So Yeon Park, Dawu Liu

**References**

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

**See Also**

[plot.mars](#) for plotting the basis functions of the mars object  
[predict.mars](#) for making predictions on a new data using the mars object  
[summary.mars](#) for summarizing the mars object  
[print.mars](#) for printing the mars object

**Examples**

```
test <- mars(y~x1+x2, data=mars::marstestdata, control=mars.control(Mmax=2))
```

---

mars.control	<i>Constructor for 'mars.control' objects</i>
--------------	---

---

**Description**

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**

Mmax	A maximum even integer of basis functions. Default value is 2.
d	A smoothing parameter for the generalized cross-validation. Default value is 3.
trace	A logical value

**Value**

a mars.control object

**Examples**

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

---

marstestdata	<i>marstestdata</i>
--------------	---------------------

---

**Description**

A dataset for testing purpose

**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	<i>Plots a mars object</i>
-----------	----------------------------

---

**Description**

Plots basis functions that are constructed with a single explanatory variable or two explanatory variables

**Usage**

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

**Arguments**

<b>x</b>	a mars object
<b>...</b>	further arguments

**See Also**

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

**Examples**

```
mar <- mars(y~x1+x2+x3, data=mars::marstestdata, control=mars.control(Mmax=6))
plot(x=mar)
```

---

predict.mars	<i>Predicted values based on mars object.</i>
--------------	---

---

**Description**

Predicted values based on mars object.

**Usage**

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

**Arguments**

object	a mars object
newdata	an optional data frame in which to look for variables with which to predict (if omitted, the fitted values are used)
...	further arguments

**Value**

predicted values of the response variable

**See Also**

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

**Examples**

```
mar <- mars(y~x1+x2+x3, data=mars::marstestdata, control=mars.control(Mmax=6))
predict(object=mar, newdata=mars::marstestdata)
```

---

print.mars	<i>Prints out the function call and coefficients of a mars object</i>
------------	---

---

**Description**

Prints out the function call and coefficients of a mars object

**Usage**

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

**Arguments**

x	a mars object
...	further arguments

**See Also**

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

**Examples**

```
mar <- mars(y~x1+x2+x3, data=mars::marstestdata, control=mars.control(Mmax=6))
print(mar)
```

---

summary.mars

*Summary method for class "mars".*


---

**Description**

Prints function call, five-number summary, summary of hinge functions for each basis function, and the coefficients of each basis function of a mars object

**Usage**

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

**Arguments**

object	a mars object for which a summary is desired.
...	further arguments

**See Also**

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

**Examples**

```
mar <- mars(y~x1+x2+x3, data=mars::marstestdata, control=mars.control(Mmax=6))
summary(mar)
```

# Index

- \* **datasets**
  - marstestdata, [4](#)
- \* **methods for mars object**
  - mars, [2](#)
- \* **methods**
  - plot.mars, [4](#)
  - predict.mars, [5](#)
  - print.mars, [5](#)
  - summary.mars, [6](#)

mars, [2](#)  
mars.control, [3](#)  
marstestdata, [4](#)

plot.mars, [3](#), [4](#), [5](#), [6](#)  
predict.mars, [3](#), [4](#), [5](#), [6](#)  
print.mars, [3–5](#), [5](#), [6](#)

summary.mars, [3–6](#), [6](#)