

# Package ‘CobbDouglas’

April 4, 2020

**Type** Package

**Title** Estimation and Efficiency Analysis for the Cobb-Douglas Production Frontier

**Version** 1.0

**Date** 2020-04-04

**Author** Alessandro Magrini

**Maintainer** Alessandro Magrini <alessandro.magrini@unifi.it>

**Description** Functionalities to estimate a Cobb-Douglas production frontier from sample data through constrained least squares (possibly setting the desired returns to scale), to predict the maximum producible output or technical efficiency, and to obtain bootstrap confidence intervals for parameters and fitted values.

**Depends** R (>= 3.5.0), quadprog

**Imports** stats

**License** GPL-2

**NeedsCompilation** no

## R topics documented:

CobbDouglas-package	1
CobbDouglas	2
CobbDouglas_boot	5
predict.CobbDouglas	6
proc	7
rice	7

<b>Index</b>	<b>8</b>
--------------	----------

---

CobbDouglas-package	<i>Estimation and efficiency analysis for the Cobb-Douglas production frontier</i>
---------------------	--

---

## Description

Functionalities to estimate a Cobb-Douglas production frontier from sample data through constrained least squares (possibly setting the desired returns to scale), to predict the maximum producible output or technical efficiency, and to obtain bootstrap confidence intervals for parameters and fitted values.

## Details

Package: CobbDouglas  
 Type: Package  
 Version: 1.0  
 Date: 2020-04-04  
 License: GPL-2

The main functions of the package are:

- [CobbDouglas](#), to estimate the frontier from sample data;
- [predict.CobbDouglas](#), to predict the maximum producible output or technical efficiency;
- [CobbDouglas\\_boot](#), to obtain bootstrap confidence intervals for parameters and fitted values.

## Author(s)

Alessandro Magrini <alessandro.magrini@unifi.it>

## References

C. W. Cobb and P. H. Douglas (1928). A theory of production. *American Economic Review*, 18: 139-165.

---

CobbDouglas

*Estimation of a Cobb-Douglas production frontier*

---

## Description

Estimation of a Cobb-Douglas production frontier from sample data.

## Usage

```
CobbDouglas(y.name, x.names=NULL, data, beta.sum=NULL)
```

## Arguments

<code>y.name</code>	The name of the output variable.
<code>x.names</code>	The names of the input variables. If NULL (the default), it is set equal to the name of the variables in data besides <code>y.name</code> .
<code>data</code>	A data.frame containing the output and the input variables.
<code>beta.sum</code>	Constraint on the sum of beta parameters. If NULL (the default), beta parameters are freely estimated (subjected only to positive constraint).

## Details

Consider a sample of  $n$  production units, for which the quantity of the output  $Y$  and of  $H$  input variables  $X_1, \dots, X_H$  is measured. Let  $y_i$  be the quantity of the output for unit  $i$ , and  $x_{hi}$  be the quantity of the  $h$ -th input for unit  $i$ . A Cobb-Douglas production frontier is defined as:

$$y_i^* = \tau \prod_{h=1}^H x_{hi}^{\beta_h}$$

where  $y_i^*$  is the maximum producible output for unit  $i$ ,  $\tau$  is a parameter representing the total factor productivity for a technically efficient unit, and  $\beta_h$  ( $h = 1, \dots, H$ ) is a parameter representing the elasticity of the output with respect to the  $h$ -th input.

Constant returns to scale holds if  $\sum_{h=1}^H \beta_h = 1$  (obtained setting the argument `beta.sum` to value 1). Instead,  $\sum_{h=1}^H \beta_h < 1$  implies decreasing returns to scale, while  $\sum_{h=1}^H \beta_h > 1$  implies increasing returns to scale. For  $i = 1, \dots, n$ , quantities  $y_i^*$  are denoted as *fitted values*, while quantities  $e_i = y_i - y_i^*$  are denoted as *residuals*.

Estimation of the Cobb-Douglas production frontier is performed through constrained least squares on the logarithmic scale:

$$(\hat{\tau}, \hat{\beta}_1, \dots, \hat{\beta}_H) = \operatorname{argmin}_{\tau, \beta_1, \dots, \beta_H} \sum_{i=1}^n (\log y_i - \log y_i^*)^2$$

where:

$$\log y_i^* = \log \tau + \sum_{h=1}^H \beta_h \log x_{hi}$$

subjected to constraints:

$$\begin{aligned} \log y_i^* &\geq \log y_i & i = 1, \dots, n \\ \beta_h &\geq 0 & h = 1, \dots, H \end{aligned}$$

S3 methods available for class `CobbDouglas` are:

- `print`: to get essential information.
- `summary`: to get summaries of estimation.
- `plot`: to display the scatterplot with the estimated frontier (only for frontiers with a single input).
- `predict`: to predict the maximum producible output or technical efficiency. See [predict.CobbDouglas](#).

Also, the method [CobbDouglas\\_boot](#) is available to approximate confidence intervals of parameters and fitted values.

## Value

An object of class `CobbDouglas`, that is a list with the following components:

<code>parameters</code>	Parameter estimates.
<code>efficiency</code>	Technical efficiencies of the sample units.
<code>fitted</code>	Fitted values on both logarithmic and original scale.
<code>residuals</code>	Residuals on both logarithmic and original scale.
<code>beta.sum</code>	Value passed to argument <code>beta.sum</code> .
<code>y.name</code>	Value passed to argument <code>y.name</code> .
<code>x.names</code>	Value passed to argument <code>x.names</code> .
<code>data</code>	Data used in the estimation.

## References

C. W. Cobb and P. H. Douglas (1928). A theory of production. *American Economic Review*, 18: 139-165.

## See Also

[predict.CobbDouglas](#); [CobbDouglas\\_boot](#).

## Examples

```
data(proc)

### 1 input variable: 'labour'

m1 <- CobbDouglas("output", "labour", data=proc)
summary(m1)

# plot the estimated frontier
plot(m1, cex.axis=1.1, cex.lab=1.2)

# technical efficiencies
m1_eff <- m1$efficiency
## NOT RUN:
# m1_eff

# efficient units
m1_eff[which(m1_eff$y.side==1),]

### 1 input variable: 'labour' + constraint on beta

# beta=1 (constant returns to scale)
m1c <- CobbDouglas("output", "labour", data=proc, beta.sum=1)
m1c$parameters
m1c$efficiency[which(m1c$efficiency$y.side==1),]
plot(m1c, cex.axis=1.1, cex.lab=1.2, main="beta = 1", cex.main=1.6)

# beta=1.25 (increasing returns to scale)
m1i <- CobbDouglas("output", "labour", data=proc, beta.sum=1.25)
m1i$parameters
m1i$efficiency[which(m1i$efficiency$y.side==1),]
plot(m1i, cex.axis=1.1, cex.lab=1.2, main="beta = 1.25", cex.main=1.6)

# beta=0.3 (decreasing returns to scale)
m1d <- CobbDouglas("output", "labour", data=proc, beta.sum=0.3)
m1d$parameters
m1d$efficiency[which(m1d$efficiency$y.side==1),]
plot(m1d, cex.axis=1.1, cex.lab=1.2, main="beta = 0.3", cex.main=1.6)

### 2 input variables: 'labour' and 'capital'

# no constraints on the sum of beta parameters
m2 <- CobbDouglas("output", c("labour","capital"), data=proc)
summary(m2)
```

```

m2$efficiency[which(m2$efficiency$y.side==1),]

# beta.sum=1 (constant returns to scale)
m2c <- CobbDouglas("output", c("labour","capital"), data=proc, beta.sum=1)
summary(m2c)
m2c$efficiency[which(m2c$efficiency$y.side==1),]

# beta.sum=0.7 (decreasing returns to scale)
m2d <- CobbDouglas("output", c("labour","capital"), data=proc, beta.sum=0.7)
summary(m2d)
m2d$efficiency[which(m2d$efficiency$y.side==1),]

```

CobbDouglas\_boot

*Bootstrap confidence intervals for a Cobb-Douglas frontier***Description**

Bootstrap resampling to approximate confidence intervals for parameters and fitted values of a Cobb-Douglas production frontier.

**Usage**

```
CobbDouglas_boot(x, nboot=500, conf=0.95)
```

**Arguments**

x	An object of class CobbDouglas.
nboot	The number of bootstrap replications. It must be at least 50.
conf	The confidence level. Default is 0.95.

**Value**

An object of class CobbDouglas\_boot, that is a list with the following components:

parameters	Bootstrap confidence intervals at level conf for the parameters.
fitted	Bootstrap confidence intervals at level conf for the fitted values.

**See Also**

[CobbDouglas](#).

**Examples**

```

data(proc)

m2 <- CobbDouglas("output", c("labour","capital"), data=proc)
set.seed(123)
CobbDouglas_boot(m2, nboot=150)

m2c <- CobbDouglas("output", c("labour","capital"), data=proc, beta.sum=1)
set.seed(123)
CobbDouglas_boot(m2c, nboot=150)

```

```
m2d <- CobbDouglas("output", c("labour","capital"), data=proc, beta.sum=0.7)
set.seed(123)
CobbDouglas_boot(m2d, nboot=150)
```

---

predict.CobbDouglas	<i>Prediction using a Cobb-Douglas production frontier</i>
---------------------	--

---

## Description

Prediction of the maximum producible output or of technical efficiency using a Cobb-Douglas production frontier.

## Usage

```
## S3 method for class 'CobbDouglas'
predict(object, newdata=NULL, type="output", ...)
```

## Arguments

object	An object of class CobbDouglas.
newdata	A data.frame in which to look for variables with which to predict the maximum producible output (if type="output") or technical efficiency (if type="efficiency"). If NULL (the default), fitted values or technical efficiencies of the sample units are returned.
type	The type of prediction: "output" (maximum producible output) or "efficiency" (technical efficiency). It can be abbreviated.
...	Further arguments passed to the generic predict method.

## Value

An object of class data.frame.

## See Also

[CobbDouglas](#).

## Examples

```
data(proc)
m2 <- CobbDouglas("output", c("labour","capital"), data=proc)

# prediction of the maximum producible output
predict(m2, newdata=data.frame(labour=20, capital=5))

# prediction of technical efficiency
predict(m2, newdata=data.frame(output=15, labour=20, capital=5), type="eff")
```

---

proc	<i>Production data</i>
------	------------------------

---

**Description**

Data on fictitious production processes.

**Usage**

```
data(proc)
```

**Format**

A data frame with a total of 60 observations on the following 3 variables:

output The amount of output produced.

capital The amount of capital utilized.

labour The amount of labour employed.

---

rice	<i>Rice production data</i>
------	-----------------------------

---

**Description**

Data on several fictitious rice production processes.

**Usage**

```
data(rice)
```

**Format**

A data frame with a total of 100 observations on the following 5 variables:

prod The amount of rice produced (tonnes).

area The amount of area planted (hectares).

labour The amount of labour employed (man-days).

fertil The amount of fertilizer used (kilograms).

machinery The amount of machinery utilized (index, firm 41=100).

# Index

CobbDouglas, [2](#), [2](#), [5](#), [6](#)  
CobbDouglas-package, [1](#)  
CobbDouglas\_boot, [2–4](#), [5](#)  
  
predict.CobbDouglas, [2–4](#), [6](#)  
proc, [7](#)  
  
rice, [7](#)