# Package ‘CobbDouglas’

April 19, 2021

**Type** Package

**Title** Efficiency Analysis using the Cobb-Douglas Production Frontier

**Version** 0.1

**Date** 2021-04-19

**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. It is possible to set a lower bound for the elasticities and/or the desired returns to scale, to predict the maximum producible output or technical efficiency, and to obtain boostrap confidence intervals for parameters and technical efficiencies.

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

**Imports** graphics, stats

**License** GPL-2

**NeedsCompilation** no

## R topics documented:

---

## `CobbDouglas-package` *Efficiency analysis using the Cobb-Douglas production frontier*

---

### Description

Functionalities to estimate a Cobb-Douglas production frontier from sample data through constrained least squares. It is possible to set a lower bound for the elasticities and/or the desired returns to scale, to predict the maximum producible output or technical efficiency, and to obtain boostrap confidence intervals for parameters and technical efficiencies.

### Details

| | |
|---|---|
| Package: | CobbDouglas |
| Type: | Package |
| Version: | 0.1 |
| Date: | 2021-04-19 |
| 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 technical efficiencies.

### 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, beta.min=0)
```

### Arguments

| | |
|---|---|
| `y.name` | The name of the output variable. |
| `x.names` | The names of the input variables. If `NULL` (the default), it is set equal to the name of the variables in `data` besides `y.name`. |
| `data` | A data.frame containing the output and the input variables. |
| `beta.sum` | Constraint on the sum of beta parameters. If `NULL` (the default), beta parameters are freely estimated (subjected only to non-negative constraint). |
| `beta.min` | Constraint on the minimum of each beta parameter. Default is 0. |

## Details

Consider a sample of $n$ production units, for which the quantity of the output $Y$ and of $H$ input variables $X_1, \ldots, X_H$ 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 $i$, $\tau > 0$ is a parameter representing the total factor productivity for a technically efficient unit, and $\beta_h \geq 0$ $(h = 1, \ldots, H)$ is a parameter representing the elasticity of the output with respect to the $h$-th input. Note that the Cobb-Douglas production frontier is linear on the logarithmic scale:

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

Constant returns to scale hold if $\sum_{h=1}^{H} \beta_h = 1$, and they can be obtained by 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.

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

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

$$= \mathtt{argmin}_{\tau, \beta_1, \ldots, \beta_H} \sum_{i=1}^{n} (\log y_i - \log \tau - \sum_{h=1}^{H} \beta_h \log x_{hi})^2$$

subjected to:

$$\log y_i^* \geq \log y_i \qquad i = 1, \ldots, n$$

$$\beta_h \geq 0 \qquad h = 1, \ldots, H$$

S3 methods available for class `CobbDouglas` are:

- `print`: to get essential information.
- `summary`: to get summaries of estimation.
- `plot`: to display the estimated frontier. See plot.CobbDouglas.
- `predict`: to predict the maximum producible output or technical efficiency. See predict.CobbDouglas.

Also, the method CobbDouglas_boot is available to approximate confidence intervals for parameters and technical efficiencies.

## Value

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

| | |
|---|---|
| `parameters` | Parameter estimates. |
| `efficiency` | Output-side (`y.side`) and input-side (`x.side`) technical efficiencies of the sample units. |
| `fitted` | Fitted values, equal to the logarithm of the maximum producible output for each unit. |

| | |
|---|---|
| residuals | Residuals, equal to the logarithm of output-side technical efficiencies. |
| beta.sum | Value passed to argument beta.sum. |
| beta.min | Value passed to argument beta.min. |
| y.name | Value passed to argument y.name. |
| x.names | Value passed to argument x.names. |
| data | 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

plot.CobbDouglas; predict.CobbDouglas; CobbDouglas_boot.

## Examples

```
data(production)


### one input variable: 'labour'

m1 <- CobbDouglas("output", "labour", data=production)
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),]

# setting beta=1 (constant returns to scale) seems to fit worse
m1c <- CobbDouglas("output", "labour", data=production, 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)


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

# no constraints on the sum of beta parameters
m2 <- CobbDouglas("output", c("labour","capital"), data=production)
summary(m2)
m2$efficiency[which(m2$efficiency$y.side==1),]

# beta.sum=1 (constant returns to scale)
m2c <- CobbDouglas("output", c("labour","capital"), data=production, 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=production, 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

Boostrap resampling to approximate confidence intervals for parameters and technical efficiencies 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. |

### Details

Non-parameteric bootstrap resampling is performed (Efron, 1979) and bias-corrected boostrap confidence intervals are computed (Efron, 1987).

### Value

An object of class CobbDouglas_boot, that is a list with three components: parameters, y.side and x.side. Each component is a data.frame containing the estimated value, the estimated standard error and the bootstrap confidence interval at level conf for each parameter (component parameters), output-side technical efficiencies (component y.side) and input-side technical efficiencies (component x.side).

### References

B. Efron (1987). Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association*, 82(397): 171-185.

B. Efron (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1): 1-26.

### See Also

CobbDouglas.

## Examples

```
data(production)

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

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

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

---

# plot.CobbDouglas *Graphic for an estimated Cobb-Douglas frontier*

---

## Description

Display the margin of an estimated Cobb-Douglas frontier with respect to a selected input variable.

## Usage

```
## S3 method for class 'CobbDouglas'
plot(x, x.name=NULL, x.fixed=NULL, add.grid=TRUE, add.points=TRUE, add.legend=TRUE,
  cex.legend=0.9, digits=3, xlab=NULL, ylab=NULL, col="red", ...)
```

## Arguments

| | |
|---|---|
| x | Object of class CobbDouglas. |
| x.name | Character string indicating the name of the input variable to consider. If NULL (the default), the first input variable is selected. |
| x.fixed | Named numerical vector including the values at which input variables besides the considered one should be held constant. If NULL (the default) or invalid, the empirical means are used. Ignored if there is a single input variable. |
| add.grid | Logical value indicating whether a grid should be added to the graphic. Default is TRUE. |
| add.points | Logical value indicating whether the observed points should be added to the graphic. Default is TRUE. Ignored if there are more than one input variable. |
| add.legend | Logical value indicating whether a legend showing the values of the input variables held constant should be added to the graphic. Default is TRUE. |
| cex.legend | Size of the legend. Default is 0.9. |
| digits | Integer non-negative value indicating the number of decimal places to be used in the legend. Default is 3. Ignored if add.legend=FALSE. |
| ylab | Text for y-axis (optional). |
| xlab | Text for x-axis (optional). |
| col | The color for the frontier. Default is "red". |
| ... | Further graphical parameters. |

### See Also

CobbDouglas.

### Examples

```
data(production)


### one input variable: 'labour'

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

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


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

m2 <- CobbDouglas("output", c("labour","capital"), data=production)
summary(m2)

## plot the estimated frontier from the side of 'labour'
# 'capital' fixed at its empirical mean
plot(m2, x.name="labour", cex.axis=1.1, cex.lab=1.2)
# 'capital' fixed at value 1
plot(m2, x.name="labour", x.fixed=c(capital=1), cex.axis=1.1, cex.lab=1.2)

## plot the estimated frontier from the side of 'capital'
# 'labour' fixed at its empirical mean
plot(m2, x.name="capital", cex.axis=1.1, cex.lab=1.2)
# 'labour' fixed at value 1
plot(m2, x.name="capital", x.fixed=c(labour=1), cex.axis=1.1, cex.lab=1.2)
```

---

## predict.CobbDouglas *Prediction based on an estimated Cobb-Douglas production frontier*

---

### Description

Predict the maximum producible output or technical efficiency based on an estimated 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"). |

`type` The type of prediction: `"output"` (maximum producible output) or `"efficiency"` (technical efficiency). default is `"output"`. It can be abbreviated.

`...` Further arguments passed to the generic `predict` method.

### Value

An object of class `data.frame`.

### References

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

### See Also

CobbDouglas.

### Examples

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

# 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=11.5,labour=20,capital=5), type="eff")
```

---

## `production` *Simulated production data*

---

### Description

Simulated data for production processes with one output variable and two input variables.

### Usage

```
data(production)
```

### Format

A data.frame with a total of 56 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` *Rice production data*

## Description

Data on fictitious processes of rice production.

## Usage

```
data(rice)
```

## Format

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

`rice` The amount of rice produced (tonnes).

`land` The amount of land 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