

Package ‘CobbDouglas’

April 8, 2020

Type Package

Title Efficiency Analysis using the Cobb-Douglas Production Frontier

Version 1.0

Date 2020-04-08

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 the desired returns to scale, to predict the maximum producible output or technical efficiency, and to obtain bootstrap 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	1
CobbDouglas	2
CobbDouglas_boot	5
predict.CobbDouglas	6
prod	7
rice	7
Index	9

CobbDouglas-package	<i>Efficiency analysis using the Cobb-Douglas production frontier</i>
---------------------	---

Description

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

Details

Package: CobbDouglas
 Type: Package
 Version: 1.0
 Date: 2020-04-08
 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	<i>Estimation of a Cobb-Douglas production frontier</i>
-------------	---

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 , τ is a parameter representing the total factor productivity for a technically efficient unit, and β_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 for parameters.

Value

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

<code>parameters</code>	Parameter estimates.
<code>efficiency</code>	Output-side (<code>y.side</code>) and input-side (<code>x.side</code>) technical efficiencies of the sample units.
<code>fitted</code>	Fitted values on logarithmic (<code>log.scale</code>) and original (<code>orig.scale</code>) scale.
<code>residuals</code>	Residuals on logarithmic (<code>log.scale</code>) and original (<code>orig.scale</code>) 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.

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.

See Also

[predict.CobbDouglas](#); [CobbDouglas_boot](#).

Examples

```
data(prod)

### 1 input variable: 'labour'

m1 <- CobbDouglas("output", "labour", data=prod)
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 not so good
m1c <- CobbDouglas("output", "labour", data=prod, 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)

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

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

# beta.sum=1 (constant returns to scale)
m2c <- CobbDouglas("output", c("labour","capital"), data=prod, 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=prod, beta.sum=0.7)
summary(m2d)
m2d$efficiency[which(m2d$efficiency$y.side==1),]
```

CobbDouglas_boot	<i>Bootstrap confidence intervals for a Cobb-Douglas frontier</i>
------------------	---

Description

Bootstrap 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 bootstrap 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 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).

Author(s)

Alessandro Magrini <alessandro.magrini@unifi.it>

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(prod)

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

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

m2d <- CobbDouglas("output", c("labour","capital"), data=prod, 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.

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.

See Also[CobbDouglas](#).**Examples**

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

# 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")
```

prod	<i>Simulated production data</i>
------	----------------------------------

Description

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

Usage

```
data(prod)
```

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.

References

T. Coelli (1996). A Guide to FRONTIER Version 4.1: A Computer Program for Stochastic Frontier Production and Cost Function Estimation. CEPA Working Paper 96/08, University of New England.

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

Description

Data on fictitious rice production processes.

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

CobbDouglas, [2](#), [2](#), [5](#), [7](#)
CobbDouglas-package, [1](#)
CobbDouglas_boot, [2–4](#), [5](#)

predict.CobbDouglas, [2–4](#), [6](#)
prod, [7](#)

rice, [7](#)