Package ‘micEconDistRay’

March 23, 2026

Version 0.1-4

Date 2026-03-23

Title Econometric Production Analysis with Ray-Based Distance Functions

Depends R (>= 4.2.0)

Imports sfaR (>= 0.1.1), stats (>= 4.2.0)

Suggests micEcon (>= 0.6-18), quadprog (>= 1.5-8)

Description Econometric analysis of multiple-input-multiple-output production technologies with ray-based input distance functions as suggested by Price and Henningsen (2023): ``A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application", Journal of Productivity Analysis 60, p. 179-188, <doi:10.1007/s11123-023-00684-1>.

License GPL (>= 2)

Encoding UTF-8

URL `https://github.com/micEcon/micEconDistRay`

BugReports `https://github.com/micEcon/micEconDistRay/issues`

NeedsCompilation no

Author Arne Henningsen [aut, cre],
Juan José Price [aut]

Maintainer Arne Henningsen `<arne.henningsen@gmail.com>`

Repository CRAN

Date/Publication 2026-03-23 15:30:02 UTC

Contents

distRayCalc *Calculate the Dependent Variable of a Ray-Based Input Distance Function*

Description

Calculate the dependent variable (logarithmic distance) of a ray-based input distance function (Price & Henningsen, forthcoming).

Usage

```
distRayCalc( xNames, yNames, zNames = NULL, sNames = NULL,
  coef, data, form = "tl", conDummy = NULL, fixThetas = FALSE )
```

Arguments

`xNames` a vector of character strings containing the names of the variables that indicate the input quantities.

`yNames` a vector of two or more character strings containing the names of the variables that indicate the output quantities.

`zNames` an optional vector of character strings containing the names of 'environmental' variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should be interacted with the input quantities and the angles of the output vector.

`sNames` an optional vector of character strings containing the names of 'environmental' variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should not be interacted with the input quantities and the angles of the output vector.

`coef` numeric vector containing the coefficients.

`data` data frame containing the data.

`form` a character string that indicates the functional form; currently, `"cd"` for the Cobb-Douglas functional form and `"tl"` for the Translog functional form are available.

`conDummy` an optional numeric vector indicating the positions in argument `zNames` that are dummy variables so that quadratic terms of these variables are omitted.

`fixThetas` logical value that indicates whether undefined angles of the output should be 'fixed' if the last two or more output quantities are zero for some of the observations.

Value

A vector.

Author(s)

Arne Henningsen and Juan José Price

References

Price, J.J. & Henningsen, A. (forthcoming): A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application. Journal of Productivity Analysis.

Examples

```
# load and prepare data set
data( appleProdFr86, package = "micEcon" )
appleProdFr86$qCap <- appleProdFr86$vCap / appleProdFr86$pCap
appleProdFr86$qLab <- appleProdFr86$vLab / appleProdFr86$pLab
appleProdFr86$qMat <- appleProdFr86$vMat / appleProdFr86$pMat

# Cobb-Douglas ray-based input distance function (with manually set parameters)
appleProdFr86$logDistCD <- distRayCalc( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ), data = appleProdFr86,
  coef = c( "(Intercept)" = -11.116, alpha_1 = 0.082, alpha_2 = 0.615,
  beta_1 = -0.031, beta_2 = -0.388 ), form = "cd" )
summary( appleProdFr86$logDistCD )

# Translog ray-based input distance function (with estimated parameters)
estTL <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86 )
appleProdFr86$logDistTL <- distRayCalc( xNames = c( "qCap", "qLab", "qMat" ),
  yNames= c( "qApples", "qOtherOut" ), data = appleProdFr86,
  coef = coef( estTL ) )
summary( appleProdFr86$logDistTL )
```

`distRayEla` *Elasticities and Derivatives of Ray-Based Input Distance Functions*

Description

Calculate distance elasticities and derivatives based on ray-based input distance functions (Price & Henningsen, forthcoming).

Usage

```
distRayEla( xNames, yNames, zNames = NULL, sNames = NULL,
  coef, data, form = "tl", conDummy = NULL, fixThetas = FALSE, ... )

distRayDeriv( xNames, yNames, zNames = NULL, sNames = NULL,
  coef, data, form = "tl", conDummy = NULL, fixThetas = FALSE,
  numDeriv = FALSE, eps = 1e-6 )
```

Arguments

`xNames` a vector of character strings containing the names of the variables that indicate the input quantities.

`yNames` a vector of two or more character strings containing the names of the variables that indicate the output quantities.

`zNames` an optional vector of character strings containing the names of 'environmental' variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should be interacted with the input quantities and the angles of the output vector.

`sNames` an optional vector of character strings containing the names of 'environmental' variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should not be interacted with the input quantities and the angles of the output vector.

`coef` numeric vector containing the coefficients.

`data` data frame containing the data.

`form` a character string that indicates the functional form; currently, `"cd"` for the Cobb-Douglas functional form and `"tl"` for the Translog functional form are available.

`conDummy` an optional numeric vector indicating the positions in argument `zNames` that are dummy variables so that quadratic terms of these variables are omitted.

`fixThetas` logical value that indicates whether undefined angles of the output should be 'fixed' if the last two or more output quantities are zero for some of the observations.

`numDeriv` logical value that indicates whether derivatives (and elasticities) should be calculated by numerical finite-difference differentiation.

`eps` small positive value that is used as change in the variables when calculating derivatives (and elasticities) by numerical finite-difference differentiation.

`...` further arguments of `distRayEla` are passed to `distRayDeriv`.

Value

A list that will be described here later.

Author(s)

Arne Henningsen and Juan José Price

References

Price, J.J. & Henningsen, A. (forthcoming): A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application. Journal of Productivity Analysis.

Examples

```
# load and prepare data set
data( appleProdFr86, package = "micEcon" )
appleProdFr86$qCap <- appleProdFr86$vCap / appleProdFr86$pCap
appleProdFr86$qLab <- appleProdFr86$vLab / appleProdFr86$pLab
appleProdFr86$qMat <- appleProdFr86$vMat / appleProdFr86$pMat

# estimate Translog ray-based input distance function
estTL <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86 )
summary( estTL )

# calculate elasticities
ela <- distRayEla( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  coef = coef( estTL ), data = appleProdFr86 )
summary( ela )

# calculate derivatives
deriv <- distRayDeriv( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  coef = coef( estTL ), data = appleProdFr86 )
summary( deriv )
```

`distRayEst` *Estimate a Ray-Based Input Distance Function*

Description

Empirically analyse multiple-input-multiple-output production technologies by estimating a ray-based input distance function (Price & Henningsen, forthcoming).

Usage

```
distRayEst( xNames, yNames, zNames = NULL, sNames = NULL,
  data, form = "tl", method = "sfa", fixThetas = FALSE, ... )
```

Arguments

`xNames` a vector of character strings containing the names of the variables that indicate the input quantities.

`yNames` a vector of two or more character strings containing the names of the variables that indicate the output quantities.

`zNames` an optional vector of character strings containing the names of ‘environmental’ variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should be interacted with the input quantities and the angles of the output vector.

`sNames` an optional vector of character strings containing the names of ‘environmental’ variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should not be interacted with the input quantities and the angles of the output vector.

`data` data frame containing the data.

`form` a character string that indicates the functional form; currently, `"cd"` for the Cobb-Douglas functional form and `"tl"` for the Translog functional form are available.

`method` a character string that indicates the estimation method; currently, `"lm"` for the Ordinary Least-Squares method (using the `lm()` function of the `stats` package) and `"sfa"` for stochastic frontier estimation (using the `sfacross()` function of the `sfaR` package) are available.

`fixThetas` logical value that indicates whether undefined angles of the output should be ‘fixed’ if the last two or more output quantities are zero for some of the observations.

`...` further arguments of `distRayEst` are passed to `lm` or `sfacross`.

Value

A list that will be described here later.

Author(s)

Arne Henningsen and Juan José Price

References

Price, J.J. & Henningsen, A. (forthcoming): A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application. Journal of Productivity Analysis.

Examples

```
# load and prepare data set
data( appleProdFr86, package = "micEcon" )
appleProdFr86$qCap <- appleProdFr86$vCap / appleProdFr86$pCap
appleProdFr86$qLab <- appleProdFr86$vLab / appleProdFr86$pLab
appleProdFr86$qMat <- appleProdFr86$vMat / appleProdFr86$pMat

# Cobb-Douglas ray-based input distance function
estCD <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86, form = "cd" )
summary( estCD )

# Translog ray-based input distance function
estTL <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86 )
summary( estTL )
```

distRayMonoRestr *Imposing Monotonicity on a Ray-Based Input Distance Function*

Description

Create a matrix and vector for imposing monotonicity on a ray-based input distance function (Price & Henningsen, forthcoming).

Usage

```
distRayMonoRestr( xNames, yNames, zNames = NULL, sNames = NULL,
  data, form = "tl", conDummy = NULL )
```

Arguments

`xNames` a vector of character strings containing the names of the variables that indicate the input quantities.

`yNames` a vector of two or more character strings containing the names of the variables that indicate the output quantities.

`zNames` an optional vector of character strings containing the names of ‘environmental’ variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should be interacted with the input quantities and the angles of the output vector.

`sNames` an optional vector of character strings containing the names of ‘environmental’ variables, i.e., variables that affect the production possibility set (i.e., the feasible combinations of input-output quantities) that—in the case of a Translog functional form—should not be interacted with the input quantities and the angles of the output vector.

`data` data frame containing the data.

`form` a character string that indicates the functional form; currently, `"cd"` for the Cobb-Douglas functional form and `"tl"` for the Translog functional form are available.

`conDummy` an optional numeric vector indicating the positions in argument `zNames` that are dummy variables so that quadratic terms of these variables are omitted.

Value

A list that contains a matrix (`RMat`) and a vector (`rVec`).

Author(s)

Arne Henningsen and Juan José Price

References

Price, J.J. & Henningsen, A. (forthcoming): A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application. Journal of Productivity Analysis.

Examples

```
# load and prepare data set
data( appleProdFr86, package = "micEcon" )
appleProdFr86$qCap <- appleProdFr86$vCap / appleProdFr86$pCap
appleProdFr86$qLab <- appleProdFr86$vLab / appleProdFr86$pLab
appleProdFr86$qMat <- appleProdFr86$vMat / appleProdFr86$pMat

# Cobb-Douglas ray-based input distance function (with manually set parameters)
estCD <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86, form = "cd" )
summary( estCD )

# the vector of unrestricted coefficients and their covariance matrix
nCoefCD <- length( coef( estCD ) ) - 2
uCoefCD <- coef( estCD )[ 1:nCoefCD ]
uCovInvCD <- solve( vcov( estCD )[ 1:nCoefCD, 1:nCoefCD ] )

# obtain the matrix and vector to impose monotonicity
restrCD <- distRayMonoRestr( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ), data = appleProdFr86,
  form = "cd" )

# obtain the restricted coefficients
library( "quadprog" )
minDistCD <- solve.QP( Dmat = uCovInvCD, dvec = rep( 0, nCoefCD ),
  Amat = t( restrCD$RMat ), bvec = - restrCD$RMat %*% uCoefCD + restrCD$rVec )
rCoefCD <- minDistCD$solution + uCoefCD


# Translog ray-based input distance function (with estimated parameters)
estTL <- distRayEst( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ),
  data = appleProdFr86 )
appleProdFr86$logDistTL <- distRayCalc( xNames = c( "qCap", "qLab", "qMat" ),
  yNames= c( "qApples", "qOtherOut" ), data = appleProdFr86,
  coef = coef( estTL ) )
summary( appleProdFr86$logDistTL )

# the vector of unrestricted coefficients and their covariance matrix
nCoefTL <- length( coef( estTL ) ) - 2
uCoefTL <- coef( estTL )[ 1:nCoefTL ]
uCovInvTL <- solve( vcov( estTL )[ 1:nCoefTL, 1:nCoefTL ] )

# obtain the matrix and vector to impose monotonicity
restrTL <- distRayMonoRestr( xNames = c( "qCap", "qLab", "qMat" ),
  yNames = c( "qApples", "qOtherOut" ), data = appleProdFr86 )

# obtain the restricted coefficientslibrary( "quadprog" )
minDistTL <- solve.QP( Dmat = uCovInvTL, dvec = rep( 0, nCoefTL ),
  Amat = t( restrTL$RMat ), bvec = - restrTL$RMat %*% uCoefTL + restrTL$rVec )
rCoefTL <- minDistTL$solution + uCoefTL
```

MuseumsDk *Data on Museums in Denmark*

Description

The MuseumsDk data set is a balanced panel data set of 93 state-recognized museums in Denmark over a six years (2012 and 2014-2018; 2013 is unavailable).

Usage

```
data( "MuseumsDk" )
```

Format

This data frame contains the following columns:

museum Name of the museum.

type Type of museum (Kulturhistorisk museum = cultural history museum; Kunstmuseer = arts museum; Naturhistorisk museum = natural history museum; Blandet museum = mixed museum).

munic Municipality, in which the museum is located.

yr Year of the observation.

units Number of visit sites.

resp Whether or not the museum has special responsibilities (0 = no special responsibilities; 1 = at least one special responsibility).

vis Number of (physical) visitors.

aarc Number of articles published (archeology).

ach Number of articles published (cultural history).

aah Number of articles published (art history).

anh Number of articles published (natural history).

exh Number of temporary exhibitions.

edu Number of primary school classes on educational visits to the museum.

ev Number of events other than exhibitions.

ftesc Scientific labor (full-time equivalents).

ftensc Non-scientific labor (full-time equivalents).

expProperty Running and maintenance costs [1,000 DKK].

expCons Conservation expenditure [1,000 DKK].

ipc Consumer Price Index in Denmark (the value for year 2014 is set to 1).

Source

A subset of this data set is used for the empirical analysis in Price & Henningsen (forthcoming). It has been obtained from Statistics Denmark and the Danish Ministry of Culture.

References

Price, J.J. & Henningsen, A. (forthcoming): A Ray-Based Input Distance Function to Model Zero-Valued Output Quantities: Derivation and an Empirical Application. Journal of Productivity Analysis.

Index