

FEAR 1.11 Command Reference

PAUL W. WILSON
Department of Economics
222 Surrine Hall
Clemson University
Clemson, South Carolina 29634 USA
pww@clemson.edu

21 April 2008

This manual is for FEAR version 1.11, a library for estimating productive efficiency, *etc.* using *R*.
Copyright © 2008 Paul W. Wilson. All rights reserved.

Contents

ap.plot	1
ap	2
boot.sw98	4
bootstrap.ci	8
ccr	10
cost.min	11
cquan	13
dea.resample	15
dea	17
eff.bw	19
fdh	21
gen.unif	23
genxy	24
hquan	26
malmquist.components	28
malmquist	31
orderm	35
profit.max	37
revenue.max	39
rnorm.trunc	41
sexton	42
show.dens	43
trunc.reg	45
wood	47

Description

This routine produces log-ratio plots as in Wilson (1993) using data returned by the command `ap`.

Usage

```
ap.plot(RATIO, NLEN = 25, plot.options = list(pch = 19),
        ylim = NULL, ylab = "log-ratio", xlab = "i", main = "",
        err.check = TRUE)
```

Arguments

<code>RATIO</code>	n by NDEL matrix returned by <code>ap</code> ;
<code>NLEN</code>	number of rows in <code>RATIO</code> to be used in plot; default value is 25, the number used in Wilson (1993);
<code>plot.options</code>	list of plotting options to be used;
<code>ylim</code>	vector of length 2 specifying minimum and maximum values on the vertical axis; default values are obtained when <code>ylim=NULL</code> ;
<code>ylab</code>	label for vertical axis;
<code>xlab</code>	label for horizontal axis;
<code>main</code>	an overall title for the plot;
<code>err.check</code>	equals <code>TRUE</code> (default) for error-checking, or <code>FALSE</code> for no error-checking.

Author(s)

Paul W. Wilson

References

Wilson, P.W. (1993), Detecting outliers in deterministic nonparametric frontier models with multiple outputs, *Journal of Business and Economic Statistics* 11, 319-323.

See Also

[ap](#), [par](#), [plot](#).

Examples

```
x=matrix(rnorm(25),nrow=1)
y=2+2*x+1.5*rnorm(25)
tmp=ap(X=x,Y=y,NDEL=4)
ap.plot(RATIO=tmp$ratio)
```

Description

This routine implements the Wilson (1993) outlier detection method.

Usage

```
ap(X, Y, NDEL = 3, err.check = TRUE)
```

Arguments

<code>X</code>	$p \times n$ matrix of observations on p inputs of n firms whose efficiency is to be estimated;
<code>Y</code>	$q \times n$ matrix of observations on q outputs of n firms whose efficiency is to be estimated;
<code>NDEL</code>	total number of observations to be deleted; i in Wilson(1993);
<code>err.check</code>	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

In terms of the notation in Wilson (1993), this routine computes $R_{min}^{(i)}$ for $i = 1, \dots, NDEL$. The QR decomposition is computed using Householder transformations. See Wilson (1993) for further details.

Value

<code>r0</code>	a vector of length NDEL containing values $R_{min}^{(i)}$; the first element of <code>r0</code> contains the value of $R_{min}^{(1)}$, the second element contains the value of $R_{min}^{(2)}$, etc.
<code>imat</code>	an NDEL by NDEL matrix containing indices of observations; indices in the i -th row of <code>imat</code> determine $R_{min}^{(i)}$.
<code>ratio</code>	an n by NDEL matrix containing data that can be used to produce log-ratio plots as in Wilson (1993); see the command <code>ap.plot</code> .

Author(s)

Paul W. Wilson

References

Wilson, P.W. (1993), Detecting outliers in deterministic nonparametric frontier models with multiple outputs, *Journal of Business and Economic Statistics* 11, 319-323.

See Also

[ap.plot](#)

Examples

```
x=matrix(rnorm(25),nrow=1)
y=2+2*x+1.5*rnorm(25)
ap(X=x,Y=y,NDEL=4)
```

Description

This routine implements the bootstrap method of Simar and Wilson (1998) for estimating confidence intervals for Shephard (1970) input and output distance functions as well as for hyperbolic graph distance functions.

Usage

```
boot.sw98(XOBS, YOBS, NREP = 2000, DHAT = NULL,
          RTS = 1, ORIENTATION = 1, alpha = 0.05, CI.TYPE=2,
          XREF = NULL, YREF = NULL, DREF = NULL,
          OUTPUT.FARRELL = FALSE, NOPRINT = FALSE, errchk = TRUE)
```

Arguments

XOBS	$p \times n$ matrix of observations on p inputs of n firms for which confidence intervals are to be estimated.
YOBS	$q \times n$ matrix of observations on q outputs of n firms for which confidence intervals are to be estimated.
NREP	number of bootstrap replications to be performed.
DHAT	(optional) vector of n estimates of Shephard distance functions corresponding to (XOBS, YOBS) for which confidence intervals are to be estimated; these efficiency estimates are relative to the technology supported by observations in (XREF, YREF) if different from the technology supported by (XOBS, YOBS).
RTS	indicator for returns to scale (equal 1 for variable returns to scale, 2 for non-increasing returns to scale, or 3 for constant returns to scale).
ORIENTATION	indicator input/output orientation (equals 1 for input orientation, 2 for output orientation, or 3 for hyperbolic graph orientation).
alpha	either a scalar or a vector length n_a of values giving statistical sizes of the confidence intervals to be estimated.
CI.TYPE	logical variable indicating method to be used to construct bootstrap confidence interval estimates; CI.TYPE can be: 1 - Efron percentile intervals; 2 - Hall percentile intervals based on differences; 3 - Efron's bias-corrected intervals; or 4 - percentile intervals based on ratios.
XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology.
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology.

DREF	(optional) vector of n_r Shephard distance function estimates corresponding to (XREF, YREF) giving estimates of technical efficiency of each observation in (XREF, YREF) relative to the technology defined by (XREF, YREF).
OUTPUT.FARRELL	logical flag; if TRUE and ORIENTATON=2, Farrell output efficiency estimates are returned (these will be greater than or equal to 1) along with corresponding bias estimates, bias-corrected efficiency estimates, and confidence interval estimates. If FALSE and ORIENTATION=2, Shephard output distance function estimates are returned (these will be less than or equal to 1) along with corresponding bias estimates, bias-corrected efficiency estimates, and confidence interval estimates. If ORIENTATION=1 then the value of this argument has no effect.
NOPRINT	logical flag; if TRUE, warning messages are suppressed. If FALSE, warning messages are printed on the console as appropriate.
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

The homogeneous bootstrap method described by Simar and Wilson (1998) is used to estimate confidence intervals for Shephard (1970) input or output distance functions, or for hyperbolic graph distance functions, corresponding to each observation in (XOBS, YOBS). Distance function estimates from previous computations may be supplied in DHAT (or DREF); if these are not supplied, they are computed automatically. Efficiency estimates are computed using the routine `dea`. Note that distance function estimates are biased; consequently, confidence intervals obtained using Efron's percentile method (`CI.TYPE=1`) should be corrected by subtracting 2 times the estimated bias from both the lower and upper bounds of confidence interval estimates when this method is used; see Simar and Wilson (1998) for details. This problem can be avoided by using `CI.TYPE=2, 3, or 4`. The method described in Simar and Wilson (2000b) corresponds to `CI.TYPE=2`.

Value

A list containing the following items is returned:

bias	vector of bootstrap bias estimates corresponding to observations in (XOBS,YOBS).
var	vector of bootstrap variance estimates corresponding to observations in (XOBS,YOBS).
conf.int	An $(n \times (2n_a))$ matrix containing confidence interval estimates corresponding to observations in (XOBS,YOBS).
dhat	efficiency estimates for observations in (XOBS, YOBS) relative to the technology supported by (XREF, YREF), or the technology supported by (XOBS, YOBS) if XREF and YREF are not supplied or are not different from (XOBS, YOBS).
dhat.bc	bias-corrected efficiency estimates for observations in (XOBS, YOBS) relative to the technology supported by (XREF, YREF), or the technology

	supported by (XOBS, YOBS) if XREF and YREF are not supplied or are not different from (XOBS, YOBS); equals dhat-bias.
<code>dref</code>	efficiency estimates for observations in (XREF, YREF) relative to the technology supported by (XREF, YREF).
<code>bias.flag</code>	logical variable; equals TRUE if any bias-corrected distance function estimates are less than or equal to zero, or FALSE otherwise.
<code>boot</code>	an $n \times NREP$ matrix containing bootstrap estimates corresponding to the n observations in (XOBS, YOBS).

If alpha is scalar, columns 1:2 of conf.int contain bootstrap estimates of lower and upper bounds of $(1 - \alpha)100$ -percent confidence intervals for each observation in (XOBS, YOBS). If alpha is a vector, columns 1:2 contain similar information corresponding to the first element of alpha; columns 3:4 contain similar information corresponding to the second element of alpha; *etc.*

Efficiency estimates returned in dhat and dref are estimated Shephard (1970) input or output distance functions, depending on whether ORIENTATION equals 1 or 2. Note if DHAT or DREF are passed as arguments when `boot.sw98` is called, then the same values are returned in dhat and dref.

Note

In some cases, use of Shephard output distance functions can result in bias-corrected distance function estimates that are negative. This will occur whenever the estimated bias is larger than the distance function estimate. In such cases, a warning message is printed (provided `NOPRINT=FALSE`), and the user is advised to use the Farrell output efficiency measure (which is the reciprocal of the Shephard output distance function) by setting `OUTPUT.FARRELL=TRUE` in the argument list. The problem is avoided since the Farrell output measures are weakly greater than one.

Author(s)

Paul W. Wilson

References

- Hadley, G. (1962), *Linear Programming*, Reading, MA: Addison-Wesley Publishing Company, Inc.
- Shephard, R.W. (1970), *Theory of Cost and Production Function*, Princeton: Princeton University Press.
- Simar, L. and P.W. Wilson (1998), Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models, *Management Science* 44, 49–61.
- Simar, L. and Wilson, P.W. (2000a), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.
- Simar, L. and P.W. Wilson (2000b), A general methodology for bootstrapping in non-parametric frontier models, *Journal of Applied Statistics* 27, 779-802.

See Also

dea.

Examples

```
data(ccr)
x=matrix(c(ccr$x1, ccr$x2, ccr$x3, ccr$x4, ccr$x5), nrow=5, ncol=70)
y=matrix(c(ccr$y1, ccr$y2, ccr$y3), nrow=3, ncol=70)
nrep=100 # this is only an example; 100 replications are insufficient
        # to obtain reliable estimates of confidence intervals!

#
# here, estimate 70 CIs:
result=boot.sw98(XOBS=x, YOBS=y, NREP=nrep)
print(result)
#
# here, estimate CIs for only the first 5 observations:
x0=x[,1:5]
y0=y[,1:5]
dhat=result$dhat[1:5]
dref=result$dref
boot.sw98(XOBS=x0, YOBS=y0, DHAT=dhat, XREF=x, YREF=y, DREF=dref, NREP=nrep)
```

`bootstrap.ci`

Compute Bootstrap Confidence Interval Estimates from a Matrix of Bootstrap Estimates

Description

This function computes several types of bootstrap confidence interval estimates, and is designed for use with efficiency estimates which may be bounded either from above or from below at one. The function can also be used to estimate confidence intervals for Malmquist productivity indices and components, as well as other quantities of interest.

Usage

```
bootstrap.ci(BOOT, alpha =c(0.1,0.05,0.01), BHAT = NULL, DEA = TRUE,  
            METHOD = 2, errchk = TRUE)
```

Arguments

<code>BOOT</code>	$n \times B$ matrix of B bootstrap estimates corresponding to n (original) estimates.
<code>alpha</code>	vector of sizes for confidence intervals.
<code>BHAT</code>	optional) a vector of length n containing the original estimates of the quantities of interest.
<code>DEA</code>	equals TRUE if the quantities of interest are distance functions, Farrell-type efficiency scores, or other quantities bounded either above or below by 1.
<code>METHOD</code>	logical variable indicating method to be used to construct bootstrap confidence interval estimates; <code>METHOD</code> can be: 1 - Efron percentile intervals; 2 - Hall percentile intervals based on differences; 3 - Efron's bias-corrected intervals; or 4 - percentile intervals based on ratios.
<code>errchk</code>	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

Typically, values of `alpha` will be between 0 and 0.1; the default is to use three values: 0.1, 0.05, and 0.01. Ordinary percentile intervals (`METHOD==1`) should be avoided when estimating confidence intervals based on Shephard (1970) distance functions or Farrell (1957) efficiency measures due to bias problems. Intervals based on ratios (`METHOD==4`) are used by Kneip et al. (2003).

Value

`bootstrap.ci` returns an $(n \times 2k)$ matrix, with each row giving confidence interval estimates corresponding to the n rows in `BOOT`, where k is the length of `alpha`. The first k columns of the returned matrix give lower bounds corresponding to the k sizes given in `alpha`, while the second group of k columns gives corresponding upper bounds.

Author(s)

Paul W. Wilson

References

Farrell, M.J. (1957), The measurement of productive efficiency, *Journal of the Royal Statistical Society Series A* 120, 253-281.

Hall, P. (1992), *The Bootstrap and Edgeworth Expansion*, New York: Springer-Verlag.

Kneip, A., L. Simar, and P.W. Wilson (2003), Asymptotics for DEA Estimators in Non-parametric Frontier Models, Discussion paper #0317, Institut de Statistique, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

Shephard, R.W. (1970), *Theory of Cost and Production Function*, Princeton: Princeton University Press.

Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

Examples

```
n=10
nrep=2000
xbar=rnorm(10)
boot=matrix(rnorm(n*nrep),nrow=n,ncol=nrep) #NOTE: not real bootstrap values!
bootstrap.ci(BOOT=boot,BHAT=xbar,DEA=FALSE,METHOD=1)
```

```
n=100
nrep=2000
dhat=rnorm(n)+1.5
dhat=ifelse(dhat<=1,1,dhat)
boot=rnorm(n*nrep)+1.3
boot=ifelse(boot<=1,1,boot)
boot=matrix(boot,nrow=n,ncol=nrep)
bootstrap.ci(BOOT=boot,BHAT=dhat,METHOD=4)
```

Description

The `ccr` data frame has 70 rows and 9 columns. The data are from Charnes *et al.* (1982).

Usage

```
data(ccr)
```

Format

This data frame contains the following columns:

dmu decision-making unit number

x1 input no. 1: education level of mother as measured in terms of percentage of high school graduates among female parents;

x2 input no. 2: highest occupation of a family member according to a pre-arranged rating scale;

x3 input no. 3: parental visit index representing the number of visits to the school site;

x4 input no. 4: parent counseling index calculated from data on time spent with child on school-related topic such as reading together, etc.;

x5 input no. 5: number of teachers at a given site.

y1 output no. 1: total reading score as measured by the Metropolitan Achievement Test;

y2 output no. 2: total mathematics score as measured by the Metropolitan Achievement Test;

y3 output no. 3: Coopersmith Self-Esteem Inventory, intended as a measure of self-esteem.

Source

Charnes, A., W.W. Cooper, and E. Rhodes (1981), "Evaluating program and managerial efficiency: An application of data envelopment analysis to Program Follow Through, *Management Science* 27, 668-97.

Examples

```
data(ccr)
```

Description

Given matrices of input and output vectors, and a matrix of input prices, this routine computes an estimate of optimal input levels for each of the given input price vectors.

Usage

```
cost.min(XREF, YREF, XPRICE, YOBS = NULL, RTS = 1, errchk = TRUE)
```

Arguments

XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology.
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology.
XPRICE	$p \times n_p$ matrix of input-price vectors, where n_p equals either 1 or n_0 (see description of YOBS below).
YOBS	(optional) $q \times n_0$ matrix of observations on q outputs of n_0 firms for which cost-minimizing inputs are to be estimated; if this argument is not passed, cost-minimizing inputs for the output vectors in YREF are estimated.
RTS	indicator for returns to scale (equal 1 for variable returns to scale, 2 for non-increasing returns to scale, or 3 for constant returns to scale).
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

See Fare *et al.* (1985, pp. 104-105, eqn 4.7.7) for details. Linear programs are solved using the simplex method described by Hadley (1962).

Value

`cost.min` returns a matrix with estimates of cost-minimizing input vectors for each output vector in YOBS (or in YREF if YOBS is not passed as an argument to `cost.min`).

Author(s)

Paul W. Wilson

References

- Fare, R., S. Grosskopf, and C.A.K. Lovell (1985), *The Measurement of Efficiency of Production*, Boston: Kluwer-Nijoff Publishing, Inc.
- Hadley, G. (1962), *Linear Programming*, Reading, MA: Addison-Wesley Publishing Company, Inc.
- Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[dea profit.max revenue.max](#)

Examples

```
input.prices=matrix(1,nrow=2,ncol=1)
y=matrix(2,nrow=1,ncol=7)
x=matrix(c(1,2,2,2,2,1,1,3,1,4,3,1.25,4,1.25),nrow=2,ncol=7)
cost.min(XREF=x,YREF=y,XPRICE=input.prices)
```

cquan	<i>Nonparametric Conditionl and Unconditional Alpha-Quantile Estimates</i>
-------	--

Description

This function computes nonparametric, conditional input or output alpha-quantile estimates, or nonparametric, unconditional, hyperbolic alpha-quantile estimates.

Usage

```
cquan(XOBS, YOBS, XREF = NULL, YREF = NULL, alpha = 0.95,  
      ORIENTATION = 1, errchk = TRUE)
```

Arguments

XOBS	$p \times n$ matrix of observations on p inputs of n firms whose efficiency is to be estimated;
YOBS	$q \times n$ matrix of observations on q outputs of n firms whose efficiency is to be estimated;
XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology;
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology;
alpha	quantile, in decimal form (must be greater than zero and less than or equal to one);
ORIENTATION	equals 1 for input orientation, 2 for output orientation, or 3 for hyperbolic graph direction;
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

For either the input- or output-oriented estimates (ORIENTATION=1 or ORIENTATION=2), an exact solution is computed as described by Daouia and Simar (2007). For the unconditional hyperbolic measure (ORIENTATION=3), a numerical solution is computed as described by Wheelock and Wilson (2008), who note that this method is faster than the corresponding exact solution for the hyperbolic orientation.

Value

cquan returns a vector of length n containing estimates of Shephard-type distance functions. Note that these are the reciprocals of the corresponding Farrell measures. For a point contained within the convex hull of the reference observations, the returned value will be weakly greater than one for ORIENTATION=1 (input orientation) when alpha=1, or weakly less than one for ORIENTATION=2 (output orientation) when alpha=1. For the hyperbolic

graph direction, estimates will be weakly greater than one when $\alpha=1$. In this case, a point of interest can be projected onto an estimated quantile by dividing input quantities by the distance function estimate while multiplying output quantities by the same distance function estimate. When $\alpha < 1$, returned values may be greater than, equal to, or less than one.

Author(s)

Paul W. Wilson

References

Daouia, A. and L. Simar (2007), Nonparametric efficiency analysis: A multivariate conditional quantile approach, *Journal of Econometrics* 140, 375–400.

Wheelock, D.C. and P.W. Wilson (2008), Non-parametric, Unconditional Quantile Estimation for Efficiency Analysis with an Application to Federal Reserve Check Processing Operations, *Journal of Econometrics*, forthcoming.

See Also

[orderm](#)

Examples

```
x=matrix(c(1:100),nrow=1)
y=matrix(c(1:100),nrow=1)
tmp1=cquan(XOBS=x,YOBS=y,ORIENTATION=1)
summary(tmp1)
tmp2=cquan(XOBS=x,YOBS=y,ORIENTATION=2)
summary(tmp2)
tmp3=cquan(XOBS=x,YOBS=y,ORIENTATION=3)
summary(tmp3)
```

Description

Given a set of n efficiency estimates (bounded either above or below at unity) and a bandwidth, this function returns a vector of n values drawn from a kernel estimate of the (bounded) density of the efficiency estimates. The returned values are needed for the homogeneous bootstrap method described by Simar and Wilson (1998). The reflection method is used to avoid the well-known problems of bias and inconsistency of kernel density estimates near boundaries of support.

Usage

```
dea.resample(dist, bw, m = NULL, omit.ones=FALSE)
```

Arguments

<code>dist</code>	vector of n efficiency estimates, bounded either above or below at 1;
<code>bw</code>	bandwidth for the kernel density estimate.
<code>m</code>	number of draws to be taken from elements of <code>dist</code> ; if NULL (the default), then the number of draws taken is n , the length of <code>dist</code> .
<code>omit.ones</code>	if FALSE (the default), and elements of <code>dist</code> that are identically equal to one are not removed; if TRUE, such values are removed before sampling from <code>dist</code> .

Details

See Simar and Wilson (1998) for details.

Value

A vector of n random efficiency scores is returned.

Note

A suitable bandwidth may be obtained by a call to [eff.bw](#).

Author(s)

Paul W. Wilson

References

Simar, L. and P.W. Wilson (1998), Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models, *Management Science* 44, 49–61.

Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[eff.bw](#).

Examples

```
dhat=rnorm(100)+1
dhat=ifelse(dhat>=1,dhat,1)
h=eff.bw(dhat)
dstar=dea.resample(dhat,h)
```

dea

Compute DEA Efficiency Estimates (general version)

Description

This function computes Shephard (1970) input or output distance functions, under variable, non-increasing, or constant returns to scale.

Usage

```
dea(XOBS, YOBS, RTS = 1, ORIENTATION = 1, XREF = NULL, YREF = NULL,  
    IS.EFF = NULL, errchk = TRUE)
```

Arguments

XOBS	$p \times n$ matrix of observations on p inputs of n firms whose efficiency is to be estimated.
YOBS	$q \times n$ matrix of observations on q outputs of n firms whose efficiency is to be estimated.
RTS	indicator for returns to scale (equal 1 for variable returns to scale, 2 for non-increasing returns to scale, or 3 for constant returns to scale).
ORIENTATION	indicates direction in which efficiency is to be evaluated (equals 1 for input orientation, 2 for output orientation, or 3 for hyperbolic graph orientation).
XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology.
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology.
IS.EFF	(optional) vector of length n indicating which columns of (XOBS, YOBS) or (XREF, YREF) are to be used to define the frontier estimate.
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

For ORIENTATION=1 or ORIENTATION=2, compute data envelopment analysis (DEA) efficiency estimates for each of n DMUs relative to the convex hull of the free-disposal hull of the same n DMUs. Efficiency is measured in terms of Shephard (1970) input or output distance functions, which are the reciprocals of the Farell (1957) input or output efficiency measures. Distance function estimates are computed using the simplex method described by Hadley (1962). For ORIENTATION=3, compute the hyperbolic graph measure of technical efficiency using a bisection method.

Value

For **ORIENTATION=1** or **ORIENTATION=2**, **dea** returns a vector of length n containing estimates of Shephard input or output distance functions. Note that these are the reciprocals of the corresponding Farrell measures. For a point contained within the convex hull of the reference observations, the returned value will be weakly greater than one for **ORIENTATION=1** (input orientation) or weakly less than one for **ORIENTATION=2** (output orientation). For **ORIENTATION=3**, distance function values will be greater than one for a point contained within the convex hull of the reference observations. In cases where distance function estimates cannot be computed, NA is returned.

Author(s)

Paul W. Wilson

References

- Charnes, A., Cooper, W.W. and Rhodes E. (1978), Measuring the inefficiency of decision making units, *European Journal of Operational Research* 2, 429-444.
- Farrell, M.J. (1957), The measurement of productive efficiency, *Journal of the Royal Statistical Society Series A* 120, 253-281.
- Hadley, G. (1962), *Linear Programming*, Reading, MA: Addison-Wesley Publishing Company, Inc.
- Shephard, R.W. (1970), *Theory of Cost and Production Function*, Princeton: Princeton University Press.
- Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[fdh](#), [orderm](#), [genxy](#).

Examples

```
data(ccr)
x=matrix(c(ccr$x1,ccr$x2,ccr$x3,ccr$x4,ccr$x5),nrow=5,ncol=70)
y=matrix(c(ccr$y1,ccr$y2,ccr$y3),nrow=3,ncol=70)
dea(XOBS=x,YOBS=y)
```

Description

This function computes a bandwidth suitable for use in kernel estimates of densities of efficiency estimates that are bounded either above or below at one.

Usage

```
eff.bw(dist, method = 1)
```

Arguments

<code>dist</code>	vector of efficiency estimates, bounded (above or below) at 1;
<code>method</code>	if equal to 1 (default), use two-stage plug-in method (Sheather and Jones, 1991), or if equal to 2, use unbaised cross-validation.

Details

It is anticipated that bandwidths selected using `eff.bw` will be used in kernel density estimates that rely on the reflection method to avoid problems of bias and inconsistency at the boundary of support. Consequently, the data are reflected around unity before applying either of the bandwidth-selection methods. The resulting bandwidth is then scale up to reflect the fact that only n of the $2n$ observations after reflection are real. The distance function values passed to `eff.bw` are first rounded to four decimal places; the rounded values are then checked to see whether at least one value is less than 1 and at least one value is greater than 1. If so, an error message is printed on the console and execution stops.

Value

A scalar bandwidth is returned.

Author(s)

Paul W. Wilson

References

- Scott, D. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons, Inc.
- Sheather, S.J., and M.C. Jones (1991), A reliable data-based bandwidth selection method for kernel density estimation, *Journal of the Royal Statistical Society B*, 53, 684–690.
- Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*,

See Also

[bw.ucv](#), [dpik](#).

Examples

```
dhat=rnorm(100)+1
dhat=ifelse(dhat>=1,dhat,1)
h=eff.bw(dhat)
```

`fdh`

Compute FDH efficiency estimates

Description

This function computes technical efficiency for a set of observed input/output data relative to the Free Disposal Hull (FDH) of a set of reference observations. The set of reference observations may be the same as the observations for which efficiency estimates are desired, or a different set of observations.

Usage

```
fdh(XOBS, YOBS, ORIENTATION = 1, XREF = NULL, YREF = NULL, errchk = TRUE)
```

Arguments

<code>XOBS</code>	$p \times n$ matrix of observations on inputs of n firms whose efficiency is to be estimated;
<code>YOBS</code>	$q \times n$ matrix of observations on outputs of n firms whose efficiency is to be estimated;
<code>ORIENTATION</code>	indicator input/output orientation (equals 1 for input orientation or 2 for output orientation);
<code>XREF</code>	(optional) $p \times n_r$ matrix of observations on inputs of n_r firms that serve to define the technology;
<code>YREF</code>	(optional) $q \times n_r$ matrix of observations on outputs of n_r firms that serve to define the technology;
<code>errchk</code>	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

`fdh` computes efficiency estimates for each of n DMUs relative to the free-disposal hull of the same n DMUs if `XREF` and `YREF` are not passed as arguments, or alternatively compute efficiency estimates for each of the n DMUs represented in `XOBS` and `YOBS` relative to the free-disposal hull of the observations in `XREF` and `YREF` when these are passed as arguments. When `XREF` and `YREF` are not passed as arguments, the efficiency estimates will be weakly greater than one in the case of input orientation, or weakly less than one in the case of output orientation.

Value

When `ORIENTATION` equals 1 or 2, `fdh` returns a $2 \times n$ matrix; the first row contains the input or output oriented efficiency estimates for the n observations, and the second row contains the number of observations in the reference set that weakly dominate each of the n observations in `XOBS`, `YOBS`. When `ORIENTATION` equals 3, `fdh` returns a $3 \times n$ matrix; the first row contains the input-oriented efficiency estimates for the n observations, the second

row contains the output-oriented efficiency estimates, and the third row contains the number of observations in the reference set that weakly dominate each of the n observations in $XOBS$, $YOBS$.

Author(s)

Paul W. Wilson

References

Deprins, D., L. Simar, and H. Tulkens (1984), Measuring labor inefficiency in post offices, in *The Performance of Public Enterprises: Concepts and measurements*, ed. by M. Marchand, P. Pestieau and H. Tulkens, Amsterdam, North-Holland, 243-267.

Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[dea](#), [orderm](#).

Examples

```
tmp=genxy(90001,100,2,2)
x=tmp$x
y=tmp$y
rm(tmp)
dhat=fdh(XOBS=x,YOBS=y)
```

Description

This function generates pseudo-random uniform numbers on the interval (0,1) using the multiplicative congruential method.

Usage

```
gen.unif(seed, n)
```

Arguments

<code>seed</code>	seed for generator (must be a number in the open interval (1,2147483647));
<code>n</code>	number of random deviates to be generated

Details

`gen.unif` generates uniform (0,1) deviates using the multiplicative congruential method with modulus $2^{31} - 1$ and multiplier 7^5 . This routine is provided as an alternative to R's generators in the base package in order to make it easy to retain the seed value. This is required in order to replicate experiments exactly.

Value

A LIST is returned containing the following elements:

<code>ran</code>	vector of n generated deviates;
<code>seed</code>	value of the generator's seed to be used on the next call.

Author(s)

Paul W. Wilson

References

Lewis, P.A., A.S. Goodman, and J.M. Miller (1969), A pseudo-random number generator for the System/360, *IBM Systems Journal* 8, 136–146.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling (1986), *Numerical Recipes*, Cambridge University Press, Cambridge.

Examples

```
seed=900001
n=20
gen.unif(seed,n)
```

genxy

Generate artificial input-output observations

Description

This function generates artificial observations on inputs and outputs from a known, simulate data-generating process based on microeconomic theory of the firm.

Usage

```
genxy(seed, n, p, q)
```

Arguments

seed	seed value for random number generator (must be a number in the open interval (1,2147483647))
n	number of observations to be generated
p	number of inputs
q	number of outputs

Details

The p inputs for each observation are generated as independent uniform deviates on the interval (10,20) by a call to `gen.unif`. Next, the inputs x_1, \dots, x_p for a given observation are used to compute $Y = \prod_{j=1}^p x_j^{0.8/p}$. In the case of one output, Y is the output value. In the case of multiple outputs ($q > 1$), $(q - 1)$ uniform $(0, \pi/2)$ deviates $\phi_1, \dots, \phi_{q-1}$ are generated. Then, y_q is set equal to $\sqrt{Y^2 / (\sum_{j=1}^{q-1} \tan^2(\phi_j) + 1)}$, and $y_j = y_q \tan \phi_j$ for $j = 1, \dots, q-1$. In other words, in the case of multiple outputs, the outputs are assumed to be randomly distributed on the surface contained in the positive orthant of a (hyper)sphere centered at the origin and with radius y .

Value

A list is returned, containing the following elements:

x	a $p \times n$ matrix of simulated input values
y	a $q \times n$ matrix of simulated output values
seed	seed value for random number generator

Author(s)

Paul W. Wilson

Examples

```
n=10  
p=3  
q=2  
tmp=genxy(9001,n,p,q)
```

hquan	<i>Nonparametric Conditional and Unconditional Alpha-Quantile Estimates</i>
-------	---

Description

This function computes nonparametric, conditional input or output alpha-quantile estimates, or nonparametric, unconditional, hyperbolic alpha-quantile estimates.

Usage

```
hquan(XOBS, YOBS, XREF = NULL, YREF = NULL, alpha = 0.95,  
      ORIENTATION = 1, errchk = TRUE)
```

Arguments

XOBS	$p \times n$ matrix of observations on p inputs of n firms whose efficiency is to be estimated;
YOBS	$q \times n$ matrix of observations on q outputs of n firms whose efficiency is to be estimated;
XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology;
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology;
alpha	quantile, in decimal form (must be greater than zero and less than or equal to one);
ORIENTATION	equals 1 for input orientation, 2 for output orientation, or 3 for hyperbolic graph direction;
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

This function is identical to the `cquan` function.

For either the input- or output-oriented estimates (ORIENTATION=1 or ORIENTATION=2), an exact solution is computed as described by Daouia and Simar (2007). For the unconditional hyperbolic measure (ORIENTATION=3), a numerical solution is computed as described by Wheelock and Wilson (2008), who note that this method is faster than the corresponding exact solution for the hyperbolic orientation.

Value

`hquan` returns a vector of length n containing estimates of Shephard-type distance functions. Note that these are the reciprocals of the corresponding Farrell measures. For a point contained within the convex hull of the reference observations, the returned value will be weakly greater than one for ORIENTATION=1 (input orientation) when alpha=1, or weakly less

than one for ORIENTATION=2 (output orientation) when alpha=1. For the hyperbolic graph direction, estimates will be weakly greater than one when alpha=1. In this case, a point of interest can be projected onto an estimated quantile by dividing input quantities by the distance function estimate while multiplying output quantities by the same distance function estimate. When alpha<1, returned values may be greater than, equal to, or less than one.

Author(s)

Paul W. Wilson

References

Daouia, A. and L. Simar (2007), Nonparametric efficiency analysis: A multivariate conditional quantile approach, *Journal of Econometrics* 140, 375–400.

Wheelock, D.C. and P.W. Wilson (2008), Non-parametric, Unconditional Quantile Estimation for Efficiency Analysis with an Application to Federal Reserve Check Processing Operations, *Journal of Econometrics*, forthcoming.

See Also

[orderm](#)

Examples

```
x=matrix(c(1:100),nrow=1)
y=matrix(c(1:100),nrow=1)
tmp1=hquan(XOBS=x,YOBS=y,ORIENTATION=1)
summary(tmp1)
tmp2=hquan(XOBS=x,YOBS=y,ORIENTATION=2)
summary(tmp2)
tmp3=hquan(XOBS=x,YOBS=y,ORIENTATION=3)
summary(tmp3)
```

Description

This function computes distance function estimates needed to construct Malmquist productivity indices, using either balanced or unbalanced panel data.

Usage

```
malmquist.components(X1, Y1, ID1, X2, Y2, ID2, ORIENTATION = 1, NREP = 0,  
                    errchk = TRUE)
```

Arguments

X1	$p \times n_1$ matrix of observations on p inputs of n_1 firms operating in period 1.
Y1	$q \times n_1$ matrix of observations on q outputs of n_1 firms operating in period 1.
ID1	a vector of length n_1 containing unique, identifying labels for the n firms represented by columns of X1 and Y1 .
X2	$p \times n_2$ matrix of observations on p inputs of n_2 firms operating in period 2.
Y2	$q \times n_2$ matrix of observations on q outputs of n_2 firms operating in period 2.
ID2	a vector of length n_2 containing unique, identifying labels for the n firms represented by columns of X2 and Y2 .
ORIENTATION	indicates direction in which efficiency is to be evaluated for purposes of constructing Malmquist indices (equals 1 for input orientation, 2 for output orientation, or 3 for hyperbolic graph orientation).
NREP	(optional) Number of bootstrap replications to be performed for inference-making purposes.
errchk	(optional) Perform error checking if TRUE; do not check for errors if FALSE.

Details

Calls are made to `dea` to compute estimates of efficiency under both constant as well as variable returns to scale for (i) each of the n_1 firms in period 1, relative to estimates of the technology at time 1; (ii) each of the n_1 firms in period 1, relative to estimates of the technology at time 2; (iii) each of the n_2 firms in period 2, relative to estimates of the technology at time 1; and (iv) each of the n_2 firms in period 2, relative to estimates of the technology at time 2. For each firm, as many as 8 estimates are computed (fewer estimates will be available in some cases, as distance function estimates may not be feasible for some of

the cross-period cases. In all cases, all of the n_1 observations available for period 1 are used to estimate the technology in period 1. Similarly, all of the n_2 observations available for period 2 are used to estimate the technology in period 2. Note that some firms that appear in period 1 may be absent in period 2, and some that appear in period 2 may be absent in period 1; for such firms, Malmquist indices cannot be estimated. If $NREP > 0$, bootstrap estimates are computed using the algorithm described by Simar and Wilson (1999), with some small modifications (see below).

Value

Let S be the set of firms that appear in both periods 1 and 2. If $NREP = 0$, a list containing the following elements is returned:

id	a vector of length n containing unique identifiers from input arguments ID1 and ID2 representing those n firms that appear in both periods 1 and 2.
v11	a vector of length n containing distance function estimates (under varying returns to scale) for the n firms in period 1 and listed in ID relative to estimated technology in period 1.
v22	a vector of length n containing distance function estimates (under varying returns to scale) for the n firms in period 2 and listed in ID relative to estimated technology in period 2.
v12	a vector of length n containing distance function estimates (under varying returns to scale) for the n firms in period 1 and listed in ID relative to estimated technology in period 2.
v21	a vector of length n containing distance function estimates (under varying returns to scale) for the n firms in period 2 and listed in ID relative to estimated technology in period 1.
c11	a vector of length n containing distance function estimates (under constant returns to scale) for the n firms in period 1 and listed in ID relative to estimated technology in period 1.
c22	a vector of length n containing distance function estimates (under constant returns to scale) for the n firms in period 2 and listed in ID relative to estimated technology in period 2.
c12	a vector of length n containing distance function estimates (under constant returns to scale) for the n firms in period 1 and listed in ID relative to estimated technology in period 2.
c21	a vector of length n containing distance function estimates (under constant returns to scale) for the n firms in period 2 and listed in ID relative to estimated technology in period 1.
bv11	an $n \times NREP$ matrix of bootstrap estimates corresponding to the n estimates in v11 .
bv22	an $n \times NREP$ matrix of bootstrap estimates corresponding to the n estimates in v22 .
bv12	an $n \times NREP$ matrix of bootstrap estimates corresponding to the n estimates in v12 .

bv21	an $n \times \text{NREP}$ matrix of bootstrap estimates corresponding to the n estimates in v21.
bc11	an $n \times \text{NREP}$ matrix of bootstrap estimates corresponding to the n estimates in c11.
bc22	an $n \times \text{NREP}$ matrix of bootstrap estimates corresponding to the n estimates in c22.
bc12	an $n \times \text{NREP}$ matrix of bootstrap estimates corresponding to the n estimates in c12.
bc21	an $n \times \text{NREP}$ matrix of bootstrap estimates corresponding to the n estimates in c21.

Note

In Simar and Wilson (1999), data are projected onto constant-returns estimates of production-set boundaries and then projected away from the boundaries to obtain bootstrap pseudo-data. In `malmquist.components`, data are first projected onto variable-returns estimates of production-set boundaries, and then projected away from the boundaries using draws from a nonparametric estimate of the joint density of variable-returns distance function estimates.

Author(s)

Paul W. Wilson

References

Simar, L. and Wilson, P.W. (1999), Estimating and bootstrapping Malmquist indices, *European Journal of Operational Research* 115, 459-471.

See Also

[dea malmquist](#)

Examples

```
tmp=genxy(seed=900001,n=100,p=1,q=1)
x1=matrix(tmp$x*exp(0.2*abs(rnorm(100))),nrow=1)
y1=matrix(tmp$y,nrow=1)
tmp=genxy(seed=900001,n=90,p=1,q=1)
x2=matrix(tmp$x*exp(0.2*abs(rnorm(90))),nrow=1)
y2=matrix(tmp$y*(1+rnorm(90)*0.2),nrow=1)
id1=c(1:100)
id2=c(11:80,101:120)
m1=malmquist.components(X1=x1,Y1=y1,ID1=id1,X2=x2,Y2=y2,ID2=id2,
                        ORIENTATION=1)
```

`malmquist`

Malmquist Productivity Indices

Description

This function computes Malmquist productivity indices and various decompositions from component distance function estimates.

Usage

```
malmquist(LIST, alpha = c(0.1,0.05,0.01), CI.TYPE = 2)
```

Arguments

<code>LIST</code>	a list of estimates for n firms returned by malmquist.components .
<code>alpha</code>	vector of length K containing sizes of confidence intervals to be estimated from bootstrap estimates included in <code>LIST</code> .
<code>CI.TYPE</code>	logical variable indicating method to be used to construct bootstrap confidence interval estimates; <code>CI.TYPE</code> can be: 1 - Efron percentile intervals; 2 - Hall percentile intervals based on differences; 3 - Efron's bias-corrected intervals; or 4 - percentile intervals based on ratios.

Details

This routine processes the results returned by [malmquist.components](#) to computed estimates of Malmquist indices of productivity change, as well as components of productivity change under several decompositions that have been proposed in the literature. If the list returned by [malmquist.components](#) includes bootstrap estimates (i.e., if `NREP > 0` in the call to [malmquist.components](#)), then confidence intervals for the various quantities are also estimated. If the list returned by [malmquist.components](#) does not include bootstrap estimates, then the arguments `alpha` and `CI.TYPE` are not used.

For a given firm, define the following terms:

- V11 variable returns efficiency at time 1 relative to technology at time 1;
- V12 variable returns efficiency at time 1 relative to technology at time 2;
- V21 variable returns efficiency at time 2 relative to technology at time 1;
- V22 variable returns efficiency at time 2 relative to technology at time 2;
- C11 constant returns efficiency at time 1 relative to technology at time 1;
- C12 constant returns efficiency at time 1 relative to technology at time 2;
- C21 constant returns efficiency at time 2 relative to technology at time 1;

C22 constant returns efficiency at time 2 relative to technology at time 2.

The Malmquist productivity index is defined by $(\frac{C_{21}}{C_{11}} \times \frac{C_{22}}{C_{12}})^{1/2}$. Fare et al. (1992) (FGLR) decomposed the Malmquist productivity index into a measure of efficiency change given by $\frac{C_{22}}{C_{11}}$ and change in technology given by $((\frac{C_{21}}{C_{22}} \times (\frac{C_{11}}{C_{12}}))^{1/2}$. Fare et al. (1994) (FGNZ) decomposed the Malmquist productivity index into three parts, representing (i) change in pure efficiency given by $\frac{V_{22}}{V_{11}}$, (ii) change in scale efficiency given by $\frac{C_{22}/V_{22}}{C_{11}/V_{11}}$, and (iii) change in technology (this term is identical to the one proposed by FGLR). Ray and Desli (1997) decomposed the Malmquist productivity index into (i) change in pure efficiency, (ii) pure technical change given by $(\frac{V_{21}}{V_{22}} \times \frac{V_{11}}{V_{12}})^{1/2}$ and a term they labelled SCH, given by $(\frac{C_{21}/V_{21}}{C_{12}/V_{12}} \times \frac{C_{22}/V_{22}}{C_{11}/V_{11}})^{1/2}$. Simar and Wilson (1998) (SW) and Wheelock and Wilson (1999) (WW) decomposed the Malmquist productivity index into four parts, including (i) change in pure efficiency, (ii) pure technical change, (iii) change in scale efficiency (identical to the term appearing in the FGLR and FGNZ decompositions), and (iv) a term labelled change in scale of technology, given by $(\frac{C_{21}/V_{21}}{C_{22}/V_{22}} \times \frac{C_{11}/V_{11}}{C_{12}/V_{12}})^{1/2}$ the square root of $((C_{21}/V_{21})/(C_{22}/V_{22}))/((C_{11}/V_{11})/(C_{12}/V_{12}))$. This term is the product of the FGLR change in efficiency term and pure technical change, while SCH is the product of change in scale efficiency and change in scale of technology.

Value

A list is returned. If input argument LIST was obtained by a call to [malmquist.components](#) with NREP=0, then the returned list includes the following items:

<code>id</code>	a vector of length n containing identifiers for each firm; these are the same as in LIST.
<code>malm</code>	a vector of length n containing estimates of the Malmquist productivity index for each firm.
<code>eff</code>	a vector of length n containing estimates of efficiency change as defined by FGLR for each firm.
<code>tech</code>	a vector of length n containing estimates of technical change as defined by FGLR for each firm.
<code>pure.eff</code>	a vector of length n containing estimates of pure efficiency change as defined by FGNZ for each firm.
<code>scale</code>	a vector of length n containing estimates of change in scale efficiency as defined by FGNZ for each firm.
<code>pure.tech</code>	a vector of length n containing estimates of pure technical change as defined by RD, SW, and WW for each firm.
<code>scale.tech</code>	a vector of length n containing estimates of changes in scale of technology as defined by SW and WW for each firm.
<code>sch</code>	a vector of length n containing estimates of the residual SCH term as defined by RD for each firm.
<code>ci.malm</code>	an $(n \times 2K)$ matrix of confidence interval estimates for the Malmquist productivity indices for each of n firms.
<code>ci.eff</code>	an $(n \times 2K)$ matrix of confidence interval estimates for efficiency change (FGLR definition) for each of n firms.

<code>ci.tech</code>	an $(n \times 2K)$ matrix of confidence interval estimates for technical change (FGLR definition) for each of n firms.
<code>ci.pure.eff</code>	an $(n \times 2K)$ matrix of confidence interval estimates for pure efficiency change (FGNZ definition) for each of n firms.
<code>ci.scale</code>	an $(n \times 2K)$ matrix of confidence interval estimates for change in scale efficiency (FGNZ definition) for each of n firms.
<code>ci.pure.tech</code>	an $(n \times 2K)$ matrix of confidence interval estimates for pure technical change (RD, SW, WW definition) for each of n firms.
<code>ci.scale.tech</code>	an $(n \times 2K)$ matrix of confidence interval estimates for change in scale of technology (SW, WW definition) for each of n firms.
<code>ci.sch</code>	an $(n \times 2K)$ matrix of confidence interval estimates for residual term defined by RD.

for each of n firms.

Note

Bootstrap confidence interval estimates are obtained by calls to [bootstrap.ci](#). Typically, values of `alpha` will be between 0 and 0.1; the default is to use three values: 0.1, 0.05, and 0.01. Returned matrices containing confidence interval estimates will have dimension $n \times 2k$; the first k columns give lower bounds corresponding to the k sizes given in `alpha`, while the second group of k columns gives corresponding upper bounds.

Author(s)

Paul W. Wilson

References

- Fare, R., S. Grosskopf, B. Lindgren, and P. Roos (1992), Productivity changes in Swedish pharmacies 1980–1989: A non-parametric Malmquist approach, *Journal of Productivity Analysis* 3, 85-101.
- Fare, R., S. Grosskopf, M. Norris, and Z. Zhang (1994), Productivity growth, technical progress, and efficiency change in industrialized countries, *American Economic Review* 84, 66-83.
- Ray, S.C. and E. Desli (1997), Productivity growth, technical progress, and efficiency change in industrialized countries: Comment, *American Economic Review* 87, 1033-1039.
- Simar, L. and P.W. Wilson (1998), “Productivity Growth in Industrialized Countries,” Discussion Paper #9810, Institut de Statistique, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.
- Simar, L. and Wilson, P.W. (1999), Estimating and bootstrapping Malmquist indices, *European Journal of Operational Research* 115, 459-471.
- Wheelock, D.C. and P.W. Wilson (1999), Technical progress, inefficiency, and productivity change in U.S. banking, 1984–1993, *Journal of Money, Credit, and Banking* 31, 212-234.

See Also

[dea malmquist.components](#)

Examples

```
#
# input orientation:
tmp=genxy(seed=900001,n=100,p=1,q=1)
x1=matrix(tmp$x*exp(0.2*abs(rnorm(100))),nrow=1)
y1=matrix(tmp$y,nrow=1)
tmp=genxy(seed=900001,n=90,p=1,q=1)
x2=matrix(tmp$x*exp(0.2*abs(rnorm(90))),nrow=1)
y2=matrix(tmp$y*(1+runif(90)*0.2),nrow=1)
id1=c(1:100)
id2=c(11:80,101:120)
m1=malmquist.components(X1=x1,Y1=y1,ID1=id1,X2=x2,Y2=y2,ID2=id2,
                        ORIENTATION=1,NREP=200)
tmp1=malmquist(LIST=m1,alpha=c(0.1,0.05,0.01),CI.TYPE=1)
tmp2=malmquist(LIST=m1,alpha=c(0.1,0.05,0.01),CI.TYPE=2)
tmp3=malmquist(LIST=m1,alpha=c(0.1,0.05,0.01),CI.TYPE=3)
tmp4=malmquist(LIST=m1,alpha=c(0.1,0.05,0.01),CI.TYPE=4)
#
# output orientation:
tmp=genxy(seed=900001,n=100,p=1,q=1)
x1=matrix(tmp$x,nrow=1)
y1=matrix(tmp$y*exp(-0.2*abs(rnorm(100))),nrow=1)
tmp=genxy(seed=900001,n=90,p=1,q=1)
x2=matrix(tmp$x,nrow=1)
y2=matrix(tmp$y*(1+runif(90)*0.2)*exp(-0.2*abs(rnorm(90))),nrow=1)
id1=c(1:100)
id2=c(11:80,101:120)
m2=malmquist.components(X1=x1,Y1=y1,ID1=id1,X2=x2,Y2=y2,ID2=id2,
                        ORIENTATION=2,NREP=20)
tmp=malmquist(LIST=m2,alpha=c(0.1,0.05,0.01))
#
# hyperbolic orientation:
tmp=genxy(seed=900001,n=100,p=1,q=1)
d1=exp(0.2*abs(rnorm(100)))
x1=matrix(tmp$x*d1,nrow=1)
y1=matrix(tmp$y/d1,nrow=1)
tmp=genxy(seed=900001,n=90,p=1,q=1)
d2=exp(0.2*abs(rnorm(90)))
x2=matrix(tmp$x*d2,nrow=1)
y2=matrix(tmp$y*(1+runif(90)*0.2)/d2,nrow=1)
id1=c(1:100)
id2=c(11:80,101:120)
m3=malmquist.components(X1=x1,Y1=y1,ID1=id1,X2=x2,Y2=y2,ID2=id2,
                        ORIENTATION=3,NREP=20)
tmp=malmquist(LIST=m3,alpha=c(0.1,0.05,0.01))
```

Description

This function computes estimates of the Cazals *et al.* (2002) order- m efficiency measure for a set of observed input/output data relative to the order- m frontier based on a set of reference observations. The set of reference observations may be the same as the observations for which efficiency estimates are desired, or a different set of observations.

Usage

```
orderm(XOBS, YOBS, ORIENTATION = 1, M = 25, NREP = 200,
       XREF = NULL, YREF = NULL, errchk = TRUE)
```

Arguments

XOBS	$p \times n$ matrix of observations on p inputs of n firms whose efficiency is to be estimated;
YOBS	$q \times n$ matrix of observations on q outputs of n firms whose efficiency is to be estimated;
ORIENTATION	indicator for input or output orientation (equals 1 for input orientation, 2 for output orientation, or 3 for both orientations);
M	order of the reference frontier;
NREP	number of Monte Carlo replications used in computing the order- m estimates;
XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology;
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology;
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

See Cazals *et al.* (2002) for detailed description of this estimator. `orderm` computes efficiency estimates for each of n DMUs relative to the order- m frontier estimated from a reference set of observations. If `XREF`, `YREF` are not passed as arguments, the reference set of observations are those in `XOBS`, `YOBS`. Alternatively, a different reference set of observations may be specified in `XREF`, `YREF`.

Value

When `ORIENTATION` equals 1 or 2, `orderm` returns a $2 \times n$ matrix; the first row contains the input or output oriented order- m efficiency estimates for the n observations, and the second row contains corresponding estimates of standard error. When `ORIENTATION` equals 3, `orderm` returns a $4 \times n$ matrix; the first row contains the input-oriented order- m efficiency estimates for the n observations, the second row contains the output-oriented efficiency estimates, the third row contains the standard error estimates corresponding to the input-oriented efficiency estimates, and the fourth row contains the the standard error estimates corresponding to the output-oriented efficiency estimates.

Author(s)

Paul W. Wilson

References

Cazals, C., J.P. Florens, and L. Simar (2002), Nonparametric frontier estimation: A robust approach, *Journal of Econometrics* 106, 1–25.

Wheelock, D.C. and P.W. Wilson (2003), “Robust Nonparametric Estimation of Efficiency and Technical Change in U.S. Commercial Banking,” unpublished working paper, Department of Economics, University of Texas, Austin, Texas 78712 USA.

See Also

[dea](#), [fdh](#).

Examples

```
tmp=genxy(90001,100,2,2)
x=tmp$x
y=tmp$y
rm(tmp)
dhat=orderm(XOBS=x,YOBS=y)
```

Description

Given matrices of input and output vectors, and matrices of input and output prices, this routine computes estimates of optimal input and output levels for each of given pair of input and output price vectors.

Usage

```
profit.max(XREF, YREF, XPRICE, YPRICE, RTS = 1, errchk = TRUE)
```

Arguments

XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology.
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology.
XPRICE	$p \times n_p$ matrix of input-price vectors, where n_p equals either 1 or n_0 .
YPRICE	$q \times n_p$ matrix of output-price vectors, where n_p equals either 1 or n_0 .
RTS	indicator for returns to scale (equal 1 for variable returns to scale, 2 for non-increasing returns to scale, or 3 for constant returns to scale).
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

See Fare *et al.* (1985, pp. 129-130, eqn 5.8.7) for details. Linear programs are solved using the simplex method described by Hadley (1962).

Value

profit.max returns a list:

x	matrix of profit-maximizing input vectors;
y	matrix of profit-maximizing output vectors.

Author(s)

Paul W. Wilson

References

- Fare, R., S. Grosskopf, and C.A.K. Lovell (1985), *The Measurement of Efficiency of Production*, Boston: Kluwer-Nijoff Publishing, Inc.
- Hadley, G. (1962), *Linear Programming*, Reading, MA: Addison-Wesley Publishing Company, Inc.
- Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[cost.min dea revenue.max](#)

Examples

```
input.prices=matrix(3,nrow=1,ncol=1)
output.prices=matrix(c(2,1),nrow=2,ncol=1)
x=matrix(c(2,2,4,6,7,8,9),nrow=1,ncol=7)
y=matrix(c(1.5,1,2,2,3,2,6,6,6,6,7,4,7,4),nrow=2,ncol=7)
profit.max(XREF=x,YREF=y,XPRICE=input.prices,YPRICE=output.prices)
```

Description

Given matrices of input and output vectors, and a matrix of output prices, this routine computes an estimate of optimal output levels for each of the given output price vectors.

Usage

```
revenue.max(XREF, YREF, XOBS = NULL, YPRICE, RTS = 1, errchk = TRUE)
```

Arguments

XREF	(optional) $p \times n_r$ matrix of observations on p inputs of n_r firms that serve to define the technology.
YREF	(optional) $q \times n_r$ matrix of observations on q outputs of n_r firms that serve to define the technology.
YPRICE	$q \times n_p$ matrix of output-price vectors, where n_p equals either 1 or n_0 (see description of XOBS below).
XOBS	(optional) $p \times n_0$ matrix of observations on p inputs of n_0 firms for which revenue-maximizing outputs are to be estimated; if this argument is not passed, revenue-maximizing outputs for the input vectors in XREF are estimated.
RTS	indicator for returns to scale (equal 1 for variable returns to scale, 2 for non-increasing returns to scale, or 3 for constant returns to scale).
errchk	equals TRUE (default) for error-checking, or FALSE for no error-checking.

Details

See Fare *et al.* (1985, pp. 104-105, eqn 4.7.7) for details. Linear programs are solved using the simplex method described by Hadley (1962).

Value

`revenue.max` returns a matrix with estimates of revenue-maximizing output vectors for each input vector in XOBS (or in XREF if XOBS is not passed as an argument to `revenue.max`).

Author(s)

Paul W. Wilson

References

- Fare, R., S. Grosskopf, and C.A.K. Lovell (1985), *The Measurement of Efficiency of Production*, Boston: Kluwer-Nijoff Publishing, Inc.
- Hadley, G. (1962), *Linear Programming*, Reading, MA: Addison-Wesley Publishing Company, Inc.
- Simar, L. and Wilson, P.W. (2000), Statistical inference in nonparametric frontier models: The state of the art, *Journal of Productivity Analysis* 13, 49-78.

See Also

[cost.min dea revenue.max](#)

Examples

```
output.prices=matrix(c(1,2),nrow=2,ncol=1)
x=matrix(5,nrow=1,ncol=7)
y=matrix(c(2,5,3,5,4,6,5,4,6,5,6,4,6,2),nrow=2,ncol=7)
revenue.max(XREF=x,YREF=y,YPRICE=output.prices)
```

Description

This function generates random deviates from a normal distribution with mean `mean` and standard deviation `sigma`, and with left-truncation at `t.left` and right-truncation at `t.right`.

Usage

```
rnorm.trunc(n, t.left = 0, t.right = Inf, mean = 0, sigma = 1)
```

Arguments

<code>n</code>	number of deviates to be generated.
<code>t.left</code>	left truncation point(s); may be either a scalar or a vector of length <code>n</code> .
<code>t.right</code>	right truncation point(s); may be either a scalar or a vector of length <code>n</code> .
<code>mean</code>	mean parameter of the original distribution before truncation.
<code>sigma</code>	standard deviation of the original distribution before truncation.

Details

If `mean` or `sigma` are not specified, they assume default values of 0 and 1, respectively. If `t.left` and `t.right` are not specified, they assume values of 0 and positive infinity, respectively; in this case, draws are taken from a half-normal distribution with non-negative support. Deviates are generated using a transformation method, which is implemented with a call to [runif](#).

Value

Random deviates are returned in a vector of length `n`.

Author(s)

Paul W. Wilson

See Also

[runif](#), [set.seed](#).

Examples

```
rnorm.trunc(n=20)
```

sexton

*Modified Program Follow-Through Data from Sexton et al.
(1986)*

Description

The `sexton` data frame has 70 rows and 9 columns. The data are identical to those in `ccr`, except that observation 28 for the third output has been deliberately miscoded as discussed by Sexton *et al.*

Usage

```
data(sexton)
```

Format

This data frame contains the following columns:

dmu decision-making unit number

x1 input no. 1: education level of mother as measured in terms of percentage of high school graduates among female parents;

x2 input no. 2: highest occupation of a family member according to a pre-arranged rating scale;

x3 input no. 3: parental visit index representing the number of visits to the school site;

x4 input no. 4: parent counseling index calculated from data on time spent with child on school-related topic such as reading together, etc.;

x5 input no. 5: number of teachers at a given site.

y1 output no. 1: total reading score as measured by the Metropolitan Achievement Test;

y2 output no. 2: total mathematics score as measured by the Metropolitan Achievement Test;

y3 output no. 3: Coopersmith Self-Esteem Inventory, intended as a measure of self-esteem.

Source

Chames, A., W.W. Cooper, and E. Rhodes (1981), "Evaluating program and managerial efficiency: An application of data envelopment analysis to Program Follow Through, *Management Science* 27, 668-97.

Sexton, T.R., R.H. Silkman, and A.J. Hogan (1986), Data envelopment analysis: Critique and extensions, in *Measuring Efficiency: An Assessment of Data Envelopment Analysis*, R.H. Silkman (ed.), Jossey-Bass Co., San Francisco.

Examples

```
data(sexton)
```

`show.dens`

Plot estimated density of bounded efficiency estimates

Description

This function plots a kernel estimate of the density of a set of efficiency estimates bounded either above or below at unity.

Usage

```
show.dens(dist, bw, XLIM = NULL, YLIM = NULL, show.plot = TRUE,  
          XLAB = "distance function estimate", YLAB = "density")
```

Arguments

<code>dist</code>	vector of efficiency estimates, bounded either above or below at 1;
<code>bw</code>	bandwidth for the kernel density estimate;
<code>XLIM</code>	(optional) range of efficiency estimates over which the density estimate is to be plotted (defaults to the range of the estimates in <code>dist</code>);
<code>YLIM</code>	(optional) range for density that is to be plotted (defaults to range of the estimated density);
<code>show.plot</code>	display plot on screen if true; otherwise, return values that can be passed to a plotting command;
<code>XLAB</code>	label for the horizontal axis;
<code>YLAB</code>	lable for the vertical axis.

Details

Given a set of (bounded at unity) estimates in `dist` and a bandwidth, `show.dens` computes a kernel density estimate and plots the estimate. It is well known that kernel density estimates are biased and inconsistent near boundaries of support; to avoid this problem, the `show.dens` employs the reflection method described by Silverman (1986) and Scott (1992).

Value

A list (x,y) is returned, with both equal to NULL if `show.dens` is TRUE. Otherwise, x is a vector of length 256 giving abscissa values and y is a vector of length 256 giving ordinate values that may be passed to a plotting routine to render a plot of the kernel density estimate computed by `show.dens`.

Author(s)

Paul W. Wilson

References

Scott, D. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons, Inc.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London.

See Also

[dea](#), [fdh](#), [eff.bw](#), [plot](#).

Examples

```
dhat=rnorm(100)+1
dhat=ifelse(dhat>=1,dhat,1)
h=eff.bw(dhat)
show.dens(dhat,h)
```

Description

Estimate a truncated normal regression equation using the method of maximum likelihood.

Usage

```
trunc.reg(Y, X, TPOINT = 1, tol = 1e-05, maxit = 100, iovar = 0,  
          err.check = TRUE)
```

Arguments

<code>Y</code>	vector of length n containing observations on the truncated dependent variable.
<code>X</code>	matrix of regressors with n rows.
<code>TPOINT</code>	scalar truncation point such that $Y > TPOINT$.
<code>tol</code>	convergence tolerance.
<code>maxit</code>	maximum number of iterations.
<code>iovar</code>	if 0, information about each iteration is suppressed; if 1, values of the gradient and step size are written to the console on each iteration.
<code>err.check</code>	if TRUE, check for errors in arguments; if FALSE, do not check for errors.

Details

The log-likelihood is maximized using a Newton method. The right-hand side data in X should be scaled so that the data do not differ by too many orders of magnitude from 1; otherwise, achieving convergence may be difficult.

Value

A list is returned containing the following components:

<code>bhat</code>	vector of estimates of intercept and slope parameters; length is same as number of columns in X .
<code>sighat</code>	estimate of square-root of the variance parameter.
<code>cov</code>	estimated variance-covariance matrix.
<code>se</code>	estimated standard errors of (bhat, sighat).
<code>grad</code>	gradient vector for the log-likelihood at the last iteration.
<code>iter</code>	number of iterations performed.
<code>ier</code>	error code; equals 0 if estimation was successful; equals 1 otherwise.

Author(s)

Paul W. Wilson

References

Olsen, R. (1978), A note on the uniqueness of the maximum likelihood estimator in the tobit model, *Econometrica* 46, 1211–1215.

Simar, L. and P.W. Wilson (2004), "Estimation and Inference in Two-Stage, Semi-Parametric Models of Production Processes," unpublished working paper, Department of Economics, University of Texas at Austin, Austin, Texas, USA.

Examples

```
set.seed(900001)
n=100
k=5
km1=4
z=matrix(c(rep(1,n),rnorm(km1*n)),nrow=n,ncol=k)
z[,5]=100 + 10*z[,5]
b=matrix(c(rep(1,k)),nrow=k)
zb=z %*% b
t1=1-zb
eps=rnorm.trunc(n,t.left=t1)
y=zb+eps
tmp=trunc.reg(Y=y,X=z,TPOINT=1)
```

wood

Wood (1973) refinery data

Description

The wood data frame contains 82 observations from a process variable study of an oil refinery unit; see Wood (1973) for details.

Usage

```
data(wood)
```

Format

This data frame contains the following variables:

dmu observation number

x1 input no. 1

x2 input no. 2

x3 input no. 3

x4 input no. 4

y output; octane number of the product produced.

The first three inputs represent feed compositions, while the fourth input is the log of a combination of process conditions.

Source

Wood, F.S. (1973), The use of individual effects and residuals in fitting equations to data, *Technometrics* 15, 677–694.

Examples

```
data(wood)
```