Package 'acepack'

July 22, 2025

Version 1.6.1

Description Two nonparametric methods for multiple regression transform selection are provided.
The first, Alternating Conditional Expectations (ACE),
is an algorithm to find the fixed point of maximal
correlation, i.e. it finds a set of transformed response variables that maximizes R^2
using smoothing functions [see Breiman, L., and J.H. Friedman. 1985. ``Estimating Optimal Transformations
for Multiple Regression and Correlation". Journal of the American Statistical Association.
80:580-598. <doi:10.1080/01621459.1985.10478157>].
Also included is the Additivity Variance Stabilization (AVAS) method which works better than ACE when
correlation is low [see Tibshirani, R. 1986. ``Estimating Transformations for Regression via Additivity
and Variance Stabilization". Journal of the American Statistical Association. 83:394-405.
<doi:10.1080/01621459.1988.10478610>]. A good introduction to these two methods is in chapter 16 of
Frank Harrell's ``Regression Modeling Strategies" in the Springer Series in Statistics.
A permutation independence test is included from [Holzmann, H., Klar, B. 2025. ``Lancaster correlation - a new dependence measure
linked to maximum correlation". Scandinavian Journal of Statistics.
52(1):145-169 <doi:10.1111/sjos.12733>].

Title ACE and AVAS for Selecting Multiple Regression Transformations

License MIT + file LICENSE

Suggests testthat, roxygen2

Repository CRAN

NeedsCompilation yes

RoxygenNote 7.3.2

Encoding UTF-8

Author Phil Spector [aut],
Jerome Friedman [aut],
Robert Tibshirani [aut],
Thomas Lumley [aut],
Shawn Garbett [cre, aut] (ORCID:

<https://orcid.org/0000-0003-4079-5621>),
Jonathan Baron [aut],
Bernhard Klar [aut],
Scott Chasalow [aut]

Maintainer Shawn Garbett <shawn.garbett@vumc.org>

Date/Publication 2025-02-13 05:30:10 UTC

Contents

---

ace *Alternating Conditional Expectations*

Description

Uses the alternating conditional expectations algorithm to find the transformations of y and x that maximize the proportion of variation in y explained by x. When x is a matrix, it is transformed so that its columns are equally weighted when predicting y.

Usage

```
ace(...)

## Default S3 method:
ace(
  x,
  y,
  wt = NULL,
  cat = NULL,
  mon = NULL,
  lin = NULL,
  circ = NULL,
  delrsq = 0.01,
  control = NULL,
  on.error = warning,
  ...
)

## S3 method for class 'formula'
ace(
  formula,
```

```
  data = NULL,
  subset = NULL,
  na.action = getOption("na.action"),
  ...
)

## S3 method for class 'ace'
summary(object, ...)

## S3 method for class 'ace'
print(x, ..., digits = 4)

## S3 method for class 'ace'
plot(
  x,
  ...,
  which = 1:(x$p + 1),
  caption = c(list("Response Y ACE Transformation"), as.list(paste("Carrier",
    rownames(x$x), "ACE Transformation"))),
  xlab = "Original",
  ylab = "Transformed",
  ask = prod(par("mfcol")) < length(which) && dev.interactive()
)
```

Arguments

...	additional arguments which go ignored for ace call. Included for S3 dispatch consistency. They are utilized when using print as they get passed to cat. Also when plotting an ace object they are passed to plot.
x	matrix; A matrix containing the independent variables.
y	numeric; A vector containing the response variable.
wt	numeric; An optional vector of weights.
cat	integer; An optional integer vector specifying which variables assume categorical values. Positive values in cat refer to columns of the x matrix and zero to the response variable. Variables must be numeric, so a character variable should first be transformed with as.numeric() and then specified as categorical.
mon	integer; An optional integer vector specifying which variables are to be transformed by monotone transformations. Positive values in mon refer to columns of the x matrix and zero to the response variable.
lin	integer; An optional integer vector specifying which variables are to be transformed by linear transformations. Positive values in lin refer to columns of the x matrix and zero to the response variable.
circ	integer; An integer vector specifying which variables assume circular (periodic) values. Positive values in circ refer to columns of the x matrix and zero to the response variable.
delrsq	numeric(1); termination threshold. Iteration stops when R-squared changes by less than delrsq in 3 consecutive iterations (default 0.01).

control	named list; control parameters to set. Documented at set_control.
on.error	function; call back for when ierr is not equal to zero. Defaults to warning.
formula	formula; an object of class "formula": a symbolic description of the model to be smoothed.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which ace is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process. Only used when a formula is specified.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
object	an S3 ace object
digits	rounding digits for summary/print
which	when plotting an ace object which plots to produce.
caption	a list of captions for a plot.
xlab	the x-axis label when plotting.
ylab	the y-axis label when plotting.
ask	when plotting should the terminal be asked for input between plots.

Value

A structure with the following components:

x	the input x matrix.
y	the input y vector.
tx	the transformed x values.
ty	the transformed y values.
rsq	the multiple R-squared value for the transformed values.
l	the codes for cat, mon, ...

References

Breiman and Friedman, Journal of the American Statistical Association (September, 1985).

The R code is adapted from S code for avas() by Tibshirani, in the Statlib S archive; the FORTRAN is a double-precision version of FORTRAN code by Friedman and Spector in the Statlib general archive.

Examples

```
TWOPI <- 8*atan(1)
x <- runif(200,0,TWOPI)
y <- exp(sin(x)+rnorm(200)/2)
a <- ace(x,y)
par(mfrow=c(3,1))
plot(a$y,a$ty)  # view the response transformation
plot(a$x,a$tx)  # view the carrier transformation
plot(a$tx,a$ty) # examine the linearity of the fitted model

# example when x is a matrix
X1 <- 1:10
X2 <- X1^2
X <- cbind(X1,X2)
Y <- 3*X1+X2
a1 <- ace(X,Y)
par(mfrow=c(1,1))
plot(rowSums(a1$tx),a1$y)
(lm(a1$y ~ a1$tx)) # shows that the colums of X are equally weighted

# From D. Wang and M. Murphy (2005), Identifying nonlinear relationships
# regression using the ACE algorithm.  Journal of Applied Statistics,
# 32, 243-258.
X1 <- runif(100)*2-1
X2 <- runif(100)*2-1
X3 <- runif(100)*2-1
X4 <- runif(100)*2-1

# Original equation of Y:
Y <- log(4 + sin(3*X1) + abs(X2) + X3^2 + X4 + .1*rnorm(100))

# Transformed version so that Y, after transformation, is a
# linear function of transforms of the X variables:
# exp(Y) = 4 + sin(3*X1) + abs(X2) + X3^2 + X4

a1 <- ace(cbind(X1,X2,X3,X4),Y)

# For each variable, show its transform as a function of
# the original variable and the of the transform that created it,
# showing that the transform is recovered.
par(mfrow=c(2,1))

plot(X1,a1$tx[,1])
plot(sin(3*X1),a1$tx[,1])

plot(X2,a1$tx[,2])
plot(abs(X2),a1$tx[,2])

plot(X3,a1$tx[,3])
plot(X3^2,a1$tx[,3])

plot(X4,a1$tx[,4])
```

```
plot(X4,a1$tx[,4])

plot(Y,a1$ty)
plot(exp(Y),a1$ty)
```

---

acetest *ACE permutation test of independence*

Description

Performs a permutation test of independence or association. The alternative hypothesis is that x and y are dependent.

Code authored by Bernhard Klar, Shawn Garbett.

Usage

```
acetest(x, y = NULL, nperm = 999, ...)

## S3 method for class 'acetest'
summary(object, ..., digits)

## S3 method for class 'acetest'
print(x, ...)

## S3 method for class 'acetest'
plot(
  x,
  acol = "blue",
  xlim = c(min(x$tp), max(c(x$tp, ceiling(x$ace * 10)/10))),
  col = "black",
  breaks = 100,
  main = "ACE Correlation Permutations",
  xlab = bquote(rho(.(x$xname), .(x$yname))),
  lwd = 2,
  ...
)
```

Arguments

x	a numeric vector, or a matrix or data frame with two columns. The first column is the 'y' and the second column is the 'x' when calling ace.
y	a vector with same length as x. Default is NULL.
nperm	number of permutations. Default is 999.
...	additional arguments to pass to cor.
object	S3 object of test results to dispatch.

digits	Number of significant digits to round for summary.
acol	for plot; color of the point estimate of correlation
xlim	for plot;xlimit of histogram
col	for plot;color of histogram bars
breaks	for plot;number of breaks. Default to 100.
main	for plot; main title of plot
xlab	for plot; x-axis label
lwd	for plot; line width of point estimate

Value

a list containing the following:

- ace The value of the test statistic.
- pval The *p*-value of the test.

References

Holzmann, H., Klar, B. 2025. "Lancaster correlation - a new dependence measure linked to maximum correlation". Scandinavian Journal of Statistics. 52(1):145-169 <doi:10.1111/sjos.12733>

See Also

cor

Examples

```
n <- 200
z <- matrix(rnorm(2*n), n) / sqrt(rchisq(n, 2)/2)
x <- z[,2]; y <- z[,1]
cor.test(x, y, method="spearman")
acetest(x, y)

plot(acetest(z))
```

---

avas *Additivity and variance stabilization for regression*

Description

Estimate transformations of x and y such that the regression of y on x is approximately linear with constant variance

Usage

```
avas(...)

## Default S3 method:
avas(
  x,
  y,
  wt = NULL,
  cat = NULL,
  mon = NULL,
  lin = NULL,
  circ = NULL,
  delrsq = 0.01,
  yspan = 0,
  control = NULL,
  ...
)

## S3 method for class 'formula'
avas(
  formula,
  data = NULL,
  subset = NULL,
  na.action = getOption("na.action"),
  ...
)

## S3 method for class 'avas'
summary(object, ...)

## S3 method for class 'avas'
print(x, ..., digits = 4)

## S3 method for class 'avas'
plot(
  x,
  ...,
  which = 1:(x$p + 1),
  caption = c(list("Response Y AVAS Transformation"), as.list(paste("Carrier",
    rownames(x$x), "AVAS Transformation"))),
  xlab = "Original",
  ylab = "Transformed",
  ask = prod(par("mfcol")) < length(which) && dev.interactive()
)
```

Arguments

...	additional arguments which go ignored for avas call. Included for S3 dispatch consistency. They are utilized when using print as they get passed to cat. Also when plotting an ace object they are passed to plot.
x	matrix containing the independent variables.
y	a vector containing the response variable.
wt	an optional vector of weights.
cat	an optional integer vector specifying which variables assume categorical values. Positive values in cat refer to columns of the x matrix and zero to the response variable. Variables must be numeric, so a character variable should first be transformed with as.numeric() and then specified
mon	an optional integer vector specifying which variables are to be transformed by monotone transformations. Positive values in mon refer to columns of the x matrix and zero to the response variable.
lin	an optional integer vector specifying which variables are to be transformed by linear transformations. Positive values in lin refer to columns of the x matrix and zero to the response variable.
circ	an integer vector specifying which variables assume circular (periodic) values. Positive values in circ refer to columns of the x matrix and zero to the response variable.
delrsq	numeric(1); Termination threshold for iteration. Stops when R-squared changes by less than delrsq in 3 consecutive iterations (default 0.01).
yspan	yspan Optional window size parameter for smoothing the variance. Range is $[0, 1]$. Default is 0 (cross validated choice). .5 is a reasonable alternative to try.
control	named list; control parameters to set. Documented at set_control.
formula	formula; an object of class "formula": a symbolic description of the model to be smoothed.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which ace is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process. Only used when a formula is specified.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
object	an S3 ace object
digits	rounding digits for summary/print
which	when plotting an ace object which plots to produce.
caption	a list of captions for a plot.
xlab	the x-axis label when plotting.
ylab	the y-axis label when plotting.
ask	when plotting should the terminal be asked for input between plots.

Value

A structure with the following components:

x	the input x matrix.
y	the input y vector.
tx	the transformed x values.
ty	the transformed y values.
rsq	the multiple R-squared value for the transformed values.
l	the codes for cat, mon, ...
m	not used in this version of avas
yspan	span used for smoothing the variance
iters	iteration number and rsq for that iteration
niters	number of iterations used

References

Rob Tibshirani (1987), "Estimating optimal transformations for regression". *Journal of the American Statistical Association* **83**, 394ff.

Examples

```
TWOPI <- 8*atan(1)
x <- runif(200,0,TWOPI)
y <- exp(sin(x)+rnorm(200)/2)
a <- avas(x,y)
plot(a) # View response and carrier transformations
plot(a$tx,a$ty) # examine the linearity of the fitted model

# From D. Wang and M. Murphy (2005), Identifying nonlinear relationships
# regression using the ACE algorithm.  Journal of Applied Statistics,
# 32, 243-258, adapted for avas.
X1 <- runif(100)*2-1
X2 <- runif(100)*2-1
X3 <- runif(100)*2-1
X4 <- runif(100)*2-1

# Original equation of Y:
Y <- log(4 + sin(3*X1) + abs(X2) + X3^2 + X4 + .1*rnorm(100))

# Transformed version so that Y, after transformation, is a
# linear function of transforms of the X variables:
# exp(Y) = 4 + sin(3*X1) + abs(X2) + X3^2 + X4

a1 <- avas(cbind(X1,X2,X3,X4),Y)

par(mfrow=c(2,1))

# For each variable, show its transform as a function of
```

```
# the original variable and the of the transform that created it,
# showing that the transform is recovered.
plot(X1,a1$tx[,1])
plot(sin(3*X1),a1$tx[,1])

plot(X2,a1$tx[,2])
plot(abs(X2),a1$tx[,2])

plot(X3,a1$tx[,3])
plot(X3^2,a1$tx[,3])

plot(X4,a1$tx[,4])
plot(X4,a1$tx[,4])

plot(Y,a1$ty)
plot(exp(Y),a1$ty)
```

`set_control` *Set internal parameters that control ACE and AVAS algorithms*

Description

These parameters are used in the smoothing routines of ACE and AVAS. ACE and AVAS both have their own smoothing implementations. This sets them globally for the package.

The default values are good for the vast majority of cases. This routine is included to provide complete control to the user, but is rarely needed.

Usage

```
set_control(
  alpha = NULL,
  big = NULL,
  span = NULL,
  sml = NULL,
  eps = NULL,
  spans = NULL,
  maxit = NULL,
  nterm = NULL
)
```

Arguments

`alpha`	numeric(1); AVAS; Controls high frequency (small span) penalty used with automatic span selection (base tone control). An alpha < 0.0 or alpha > 10.0 results in no effect. Default is 5.0.
`big`	numeric(1); ACE and AVAS; a large floating point number. Default is 1.0e30.

`span` numeric(1); ACE and AVAS; Span to use in smoothing represents the fraction of observations in smoothing window. Automatic span selection is performed if set to 0.0. Default is 0.0 (automatic).

For small samples (n < 40) or if there are substantial serial correlations between observations close in x - value, then a specified fixed span smoother (span > 0) should be used. Reasonable span values are 0.3 to 0.5.

`sml` numeric(1); AVAS; A small number. Should be set so that '(sml)**(10.0)' does not cause floating point underflow. Default is 1e-30.

`eps` numeric(1); AVAS; Used to numerically stabilize slope calculations for running linear fits.

`spans` numeric(3); AVAS; span values for the three running linear smoothers.

"spans(1)" Tweeter span. Default is 0.05.

"spans(2)" Midrange span. Default is 0.2.

"spans(3)" Woofer span. Default is 0.5.

Warning: These span values should be changed only with great care.

`maxit` integer(1); ACE and AVAS; Maximum number of iterations. Default is 20.

`nterm` integer(1); ACE and AVAS; Number of consecutive iterations for which rsq must change less than delcor for convergence. Default is 3.

Examples

```
set_control(maxit=40)
set_control(maxit=20)
set_control(alpha=5.0)
set_control(big=1e30, sml=1e-30)
set_control(eps=1e-3)
set_control(span=0.0, spans=c(0.05, 0.2, 0.5))
set_control(maxit=20, nterm=3)
```

Index