Package 'Renvlp'

January 20, 2025

Type Package

Title Computing Envelope Estimators

Version 3.4.5

Date 2023-09-11

Author Minji Lee, Zhihua Su

Maintainer Minji Lee <`minjilee101@gmail.com`>

Description Provides a general routine, envMU, which allows estimation of the M envelope of span(U) given root n consistent estimators of M and U. The routine envMU does not presume a model. This package implements response envelopes, partial response envelopes, envelopes in the predictor space, heteroscedastic envelopes, simultaneous envelopes, scaled response envelopes, scaled envelopes in the predictor space, groupwise envelopes, weighted envelopes, envelopes in logistic regression, envelopes in Poisson regression envelopes in function-on-function linear regression, envelope-based Partial Partial Least Squares, envelopes with non-constant error covariance, envelopes with t-distributed errors, reduced rank envelopes and reduced rank envelopes with non-constant error covariance. For each of these model-based routines the package provides inference tools including bootstrap, cross validation, estimation and prediction, hypothesis testing on coefficients are included except for weighted envelopes. Tools for selection of dimension include AIC, BIC and likelihood ratio testing. Background is available at Cook, R. D., Forzani, L. and Su, Z. (2016) <doi:10.1016/j.jmva.2016.05.006>. Optimization is based on a clockwise coordinate descent algorithm.

License GPL-2

NeedsCompilation no

Imports Rsolnp, stats, orthogonalsplinebasis, pls, matrixcalc, Matrix

Suggests MASS

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-10-10 21:00:02 UTC

Contents

Renvlp-package 4

amitriptyline 6
Berkeley 6
boot.env 8
boot.env.apweights 9
boot.env.tcond 10
boot.eppls 11
boot.genv 12
boot.henv 13
boot.logit.env 14
boot.penv 15
boot.pois.env 16
boot.rrenv 17
boot.rrenv.apweights 18
boot.senv 19
boot.stenv 20
boot.sxenv 21
boot.xenv 22
concrete 23
cv.env 24
cv.env.apweights 25
cv.env.tcond 26
cv.eppls 27
cv.genv 28
cv.henv 29
cv.logit.env 30
cv.penv 31
cv.pois.env 32
cv.rrenv 33
cv.rrenv.apweights 34
cv.senv 35
cv.stenv 36
cv.sxenv 38
cv.xenv 39
d.select 40
env 40
env.apweights 42
env.tcond 44
eppls 46
felmdir 48
felmKL 49
fiberpaper 51
genv 52
henv 54
horseshoecrab 56
logit.env 57
NJdata 58
penv 60
pois.env 62

pred.env . . . 63
pred.env.apweights . . . 65
pred.env.tcond . . . 66
pred.eppls . . . 67
pred.felmdir . . . 68
pred.felmKL . . . 69
pred.genv . . . 71
pred.henv . . . 72
pred.logit.env . . . 73
pred.penv . . . 74
pred.pois.env . . . 75
pred.rrenv . . . 76
pred.rrenv.apweights . . . 77
pred.senv . . . 78
pred.stenv . . . 79
pred.sxenv . . . 80
pred.xenv . . . 81
pred2.env . . . 82
rrenv . . . 83
rrenv.apweights . . . 85
sales . . . 87
senv . . . 88
stenv . . . 89
sxenv . . . 91
testcoef.env . . . 93
testcoef.env.apweights . . . 94
testcoef.env.tcond . . . 96
testcoef.genv . . . 97
testcoef.henv . . . 98
testcoef.logit.env . . . 99
testcoef.penv . . . 101
testcoef.pois.env . . . 102
testcoef.rrenv . . . 103
testcoef.rrenv.apweights . . . 104
testcoef.senv . . . 106
testcoef.stenv . . . 107
testcoef.sxenv . . . 108
testcoef.xenv . . . 109
u.env . . . 111
u.env.apweights . . . 112
u.env.tcond . . . 113
u.eppls . . . 114
u.felmdir . . . 115
u.felmKL . . . 116
u.genv . . . 118
u.henv . . . 119
u.logit.env . . . 120
u.penv . . . 121

u.pois.env 122
u.pred2.env 123
u.rrenv 124
u.rrenv.apweights 125
u.senv 126
u.stenv 126
u.sxenv 127
u.xenv 128
vehicles 129
waterstrider 130
weighted.env 131
weighted.penv 133
weighted.pred.env 135
weighted.xenv 136
wheatprotein 137
xenv 138

Index **141**

`Renvlp-package` *Computing Envelope Estimators*

Description

This package contains functions for estimating envelope models including response envelopes, partial response envelopes, envelopes in the predictor space, heteroscedastic envelopes, simultaneous envelopes, scaled response envelopes, scaled envelopes in the predictor space, groupwise envelopes, weighted envelopes, envelopes in logistic regression, envelopes in poisson regression, envelopes in function-on-function linear regression, envelope-based Partial Partial Least Squares, envelopes with non-constant error covariance, envelopes with t-distributed errors, reduced rank envelopes and reduced rank envelopes with non-constant error covariance.

Details

Package:	Renvlp
Type:	Package
Version:	3.4.5
Date:	2023-09-11
License:	GPL-2
Imports:	Rsolnp, orthogonalsplinebasis, pls, matrixcalc, Matrix
Suggests:	Mass, stats

Author(s)

Minji Lee and Zhihua Su.

Maintainer: Minji Lee <minjilee101@gmail.com>

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Su, Z. and Cook, R.D. (2011). Partial envelopes for efficient estimation in multivariate linear regression. Biometrika 98, 133 - 146.

Cook, R. D., Helland, I. S. and Su, Z. (2013). Envelopes and Partial Least Squares Re- gression. Journal of the Royal Statistical Society: Series B 75, 851 - 877.

Su, Z. and Cook, R.D. (2013). Estimation of Multivariate Means with Heteroscedastic Error Using Envelope Models. Statistica Sinica, 23, 213-230.

Cook, R. D., Zhang, X. (2015). Simultaneous Envelopes for Multivariate Linear Regression. Technometrics 57, 11 - 25.

Cook, R. D., Su, Z. (2013). Scaled Envelopes: scale Invariant and Efficient Estimation in Multivariate Linear Regression. Biometrika 100, 939 - 954.

Cook, R. D., Su, Z. (2016). Scaled Predictor Envelopes and Partial Least Squares Regression. Technometrics 58, 155 - 165.

Park, Y., Su, Z. and Zhu, H. (2017) Groupwise envelope models for Imaging Genetic Analysis. Biometrics.

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika.

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Park, Y., Su, Z. and Chung, D. (2022+) Envelope-based Partial Partial Least Squares with Application to Cytokine-based Biomarker Analysis for COVID-19. Statistics in Medicine.

Cook, R. D., Forzani, L. and Zhang, X. (2015). Envelopes and reduced-rank regression. Biometrika 102, 439-456.

Forzani, L. and Su, Z. (2021). Envelopes for elliptical multivariate linear regression. Statist. Sinica 31, 301-332.

`amitriptyline` *Amitriptyline Data*

Description

Patients who were admitted to the hospital after an amitriptyline overdose.

Usage

```
data("amitriptyline")
```

Format

A data frame with 17 observations on the following 7 variables.

`tot`	Total TCAD plasma level.
`ami`	Amount of amitriptyline present in TCAD plasma level.
`gen`	Gender:1 if female, 0 if male.
`amt`	Amount of antidepressants taken at time of overdose.
`pr`	PR wave measurement.
`diap`	Diastolic blood pressure.
`qrs`	QRS wave measurement.

Details

This data set contains measures of side effects of amitriptyline and gender information.

References

Johnson, R.A., Wichern, D.W. (2007). Applied Multivariage Statistical Analysis, 6th edition.

`Berkeley` *Berkeley Guidance Study Data*

Description

Heights of children born in Berkeley

Usage

```
data("Berkeley")
```

Format

A data frame with 93 observations on the following 32 variables.

`V1`	Sex.
`V2`	Age 1.
`V3`	Age 1.25.
`V4`	Age 1.5.
`V5`	Age 1.75.
`V6`	Age 2.
`V7`	Age 3.
`V8`	Age 4.
`V9`	Age 5.
`V10`	Age 6.
`V11`	Age 7.
`V12`	Age 8.
`V13`	Age 8.5.
`V14`	Age 9.
`V15`	Age 9.5.
`V16`	Age 10.
`V17`	Age 10.5.
`V18`	Age 11.
`V19`	Age 11.5.
`V20`	Age 12.
`V21`	Age 12.5.
`V22`	Age 13.
`V23`	Age 13.5.
`V24`	Age 14.
`V25`	Age 14.5.
`V26`	Age 15.
`V27`	Age 15.5.
`V28`	Age 16.
`V29`	Age 16.5.
`V30`	Age 17.
`V31`	Age 17.5.
`V32`	Age 18.

Details

This data set contains measurements of heights of children born in 1928-29 in Berkeley, CA.

References

Tuddenham, R. D. and Snyder, M. M. (1954). Physical growth of California boys and girls from birth to eighteen years. Publications in child developments. University of California, Berkeley, 1(2), 183-364.

`boot.env` *Bootstrap for env*

Description

Compute bootstrap standard error for the response envelope estimator.

Usage

```
boot.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]

u <- u.env(X, Y)
u

B <- 100
bootse <- boot.env(X, Y, 1, B)
bootse
```

`boot.env.apweights` *Bootstrap for env.apweights*

Description

Compute bootstrap standard error for the response envelope estimator that accommodates nonconstant variance.

Usage

```
boot.env.apweights(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model with nonconstant variance by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]

## Not run: u <- u.env.apweights(X, Y)
## Not run: u

B <- 100
## Not run: bootse <- boot.env.apweights(X, Y, 1, B)
## Not run: bootse
```

`boot.env.tcond` *Bootstrap for env.tcond*

Description

Compute bootstrap standard error for the response envelope estimator with t-distributed errors.

Usage

```
boot.env.tcond(X, Y, u, df, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`df`	Degrees of freedom of the t-distribution. A positive number that is greater than 2.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model with t-distributed errors by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]

## Not run: u <- u.env.tcond(X, Y, 6)
## Not run: u

B <- 100
## Not run: bootse <- boot.env.tcond(X, Y, 2, 6, B)
## Not run: bootse
```

`boot.eppls` *Bootstrap for eppls*

Description

Compute bootstrap standard error for the Envelope-based Partial Partial Least Squares estimator.

Usage

```
boot.eppls(X1, X2, Y, u, B)
```

Arguments

`X1`	An n by $p1$ matrix of continuous predictors, where $p1$ is the number of continuous predictors with $p1 < n$.
`X2`	An n by $p2$ matrix of categorical predictors, where $p2$ is the number of categorical predictors with $p2 < n$.
`Y`	An n by r matrix of multivariate responses, where r is the number of responses.
`u`	A given dimension of the Envelope-based Partial Partial Least Squares. It should be an interger between 0 and $p1$.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients beta1 and beta2 in the Envelope-based Partial Partial Least Squares by bootstrapping the residuals.

Value

The output is a list that contains the following components:

`bootse1`	The standard error for elements in beta1 computed by bootstrap. The output is an p1 by r matrix.
`bootse1`	The standard error for elements in beta2 computed by bootstrap. The output is an p2 by r matrix.

Examples

```
data(amitriptyline)

Y <- amitriptyline[ , 1:2]
X1 <- amitriptyline[ , 4:7]
X2 <- amitriptyline[ , 3]

B <- 100
## Not run: bootse <- boot.eppls(X1, X2, Y, 2, B)
## Not run: bootse
```

`boot.genv` *Bootstrap for genv*

Description

Compute bootstrap standard error for the groupwise envelope.

Usage

```
boot.genv(X, Y, Z, u, B)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Z	A group indicator vector of length n, where n denotes the number of observations.
u	Dimension of the groupwise envelope. An integer between 0 and r.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the groupwise envelope model by bootstrapping the residuals.

Value

The output is an p by r matrix.

bootse	The standard error for elements in beta computed by bootstrap.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

## Not run: B <- 100
## Not run: res <- boot.genv(X, Y, Z, 2, B)
## Not run: res$bootse[[1]]
## Not run: res$bootse[[2]]
```

`boot.henv` *Bootstrap for henv*

Description

Compute bootstrap standard error for the heteroscedastic envelope.

Usage

```
boot.henv(X, Y, u, B)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the heteroscedastic envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the heteroscedastic envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

B <- 100
## Not run: res <- boot.henv(X, Y, 2, B)
## Not run: res
```

`boot.logit.env` *Bootstrap for logit.env*

Description

Compute bootstrap standard error for the envelope estimator in logistic regression.

Usage

```
boot.logit.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the coefficients in the logistic regression envelope by the paired bootstrap.

Value

The output is a p by 1 matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

B <- 100
## Not run: bootse <- boot.logit.env(X, Y, 1, B)
## Not run: bootse
```

`boot.penv` *Bootstrap for penv*

Description

Compute bootstrap standard error for the partial envelope estimator.

Usage

```
boot.penv(X1, X2, Y, u, B)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the partial envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients beta1 in the partial envelope model by bootstrapping the residuals.

Value

The output is an r by p1 matrix.

`bootse`	The standard error for elements in beta1 computed by bootstrap.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

B <- 100
## Not run: bootse <- boot.penv(X1, X2, Y, 1, B)
## Not run: bootse
```

`boot.pois.env` *Bootstrap for pois.env*

Description

Compute bootstrap standard error for the envelope estimator in poisson regression.

Usage

```
boot.pois.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be counts.
`u`	Dimension of the envelope. An integer between 0 and p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the coefficients in the poisson regression envelope by the paired bootstrap.

Value

The output is a p by 1 matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

B <- 100
## Not run: bootse <- boot.pois.env(X, Y, 1, B)
## Not run: bootse
```

`boot.rrenv` *Bootstrap for rrenv*

Description

Compute bootstrap standard error for the reduced rank envelope estimator.

Usage

```
boot.rrenv(X, Y, u, d, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the reduced rank envelope model.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

B <- 100
## Not run: bootse <- boot.rrenv(X, Y, 4, 2, B)
## Not run: bootse
```

`boot.rrenv.apweights` *Bootstrap for rrenv.apweights*

Description

Compute bootstrap standard error for the reduced rank envelope estimator with nonconstant error covariance.

Usage

```
boot.rrenv.apweights(X, Y, u, d, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the reduced rank envelope model that accommodates nonconstant error covariance.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

B <- 10
## Not run: bootse <- boot.rrenv.apweights(X, Y, 3, 2, B)
## Not run: bootse
```

`boot.senv` *Bootstrap for senv*

Description

Compute bootstrap standard error for the scaled response envelope estimator.

Usage

```
boot.senv(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the scaled envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u

## Not run: B <- 100
## Not run: bootse <- boot.senv(X, Y, 2, B)
## Not run: bootse
```

`boot.stenv` *Bootstrap for stenv*

Description

Compute bootstrap standard error for the simultaneous envelope estimator.

Usage

```
boot.stenv(X, Y, q, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`q`	Dimension of the X-envelope. An integer between 0 and p.
`u`	Dimension of the Y-envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the residuals.

Value

The output is an p by r matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]
u <- u.stenv(X, Y)
u

## Not run: B <- 100
## Not run: bootse <- boot.stenv(X, Y, 2, 3, B)
## Not run: bootse
```

`boot.sxenv` *Bootstrap for sxenv*

Description

Compute bootstrap standard error for the scaled predictor envelope estimator.

Usage

```
boot.sxenv(X, Y, u, R, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope in the predictor space. An integer between 0 and p.
`R`	The number of replications of the scales. A vector, the sum of all elements of R must be p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the scaled envelope model in the predictor space by bootstrapping the residuals.

Value

The output is an p by r matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

B <- 100
## Not run: bootse <- boot.sxenv(X, Y, 2, R, B)
## Not run: bootse
```

`boot.xenv` *Bootstrap for xenv*

Description

Compute bootstrap standard error for the predictor envelope estimator.

Usage

```
boot.xenv(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`u`	Dimension of the envelope. An integer between 0 and p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model in predictor space by bootstrapping the residuals.

Value

The output is a p by r matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

## Not run: B <- 100
## Not run: bootse <- boot.xenv(X, Y, 2, B)
## Not run: bootse
```

`concrete` *Concrete Slump Test Dataset*

Description

Concrete is a highly complex material. This dataset investigates the relationship between the slump, flow and 28-day Compressive Strength of concrete and concrete ingredients.

Usage

```
data("concrete")
```

Format

A data frame with 103 entries on the following 10 variables.

`Cement` Cement, input gradients, kg in one M^3 concrete.

`Slag` Slag, input gradients, kg in one M^3 concrete.

`Fly_ash` Fly ash, input gradients, kg in one M^3 concrete.

`Water` Water, input gradients, kg in one M^3 concrete.

`SP` Superplasticizer, input gradients, kg in one M^3 concrete.

`Coarse_Aggr` Coarse aggregate, input gradients, kg in one M^3 concrete.

`Fine_Aggr` Fine aggregate, input gradients, kg in one M^3 concrete.

`Slump` Slump, output variable, in centimeter.

`Flow` Flow, output variable, in centimeter.

`Strength` 28-day Compressive Strength, in Mpa.

Details

Concrete is the most important material in civil engineering. This dataset include 103 data points that measure three characteristics of the concrete and the amount of seven input ingradents.

Source

UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/concrete+slump+test

References

Yeh, I-Cheng (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. Cement and Concrete Composites, 29(6), 474-480.

`cv.env` *Cross validation for env*

Description

Compute the prediction error for the response envelope estimator using m-fold cross validation.

Usage

```
cv.env(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- 5
nperm <- 50
cvPE <- cv.env(X, Y, 1, m, nperm)
cvPE
```

`cv.env.apweights` *Cross validation for env.apweights*

Description

Compute the prediction error using m-fold cross validation for the response envelope estimator that accommodates nonconstant variance.

Usage

```
cv.env.apweights(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]
## Not run: u <- u.env.apweights(X, Y)
## Not run: u

m <- 5
nperm <- 50
## Not run: cvPE <- cv.env.apweights(X, Y, 2, m, nperm)
## Not run: cvPE
```

`cv.env.tcond` *Cross validation for env.tcond*

Description

Compute the prediction error using m-fold cross validation for the response envelope estimator where the errors follow a t-distribution.

Usage

```
cv.env.tcond(X, Y, u, df, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`df`	Degrees of freedom of the t-distribution. A positive number that is greater than 2.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]
## Not run: u <- u.env.tcond(X, Y, 6)
## Not run: u
```

```
m <- 5
nperm <- 50
## Not run: cvPE <- cv.env.tcond(X, Y, 2, 6, m, nperm)
## Not run: cvPE
```

`cv.eppls` *Cross validation for peplos*

Description

Compute the prediction error for the Envelope-based Partial Partial Least Squares estimator using m-fold cross validation.

Usage

```
cv.eppls(X1, X2, Y, u, m, nperm)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(amitriptyline)

Y <- amitriptyline[ , 1:2]
X1 <- amitriptyline[ , 4:7]
X2 <- amitriptyline[ , 3]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.eppls(X1, X2, Y, 2, m, nperm)
## Not run: cvPE
```

`cv.genv` *Cross validation for genv*

Description

Compute the prediction error for the groupwise envelope estimator using m-fold cross validation.

Usage

```
cv.genv(X, Y, Z, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Z`	A group indicator vector of length n, where n denotes the number of observations.
`u`	Dimension of the groupwise envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate `m`-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

## Not run: m <- 5
## Not run: nperm <- 50
## Not run: cvPE <- cv.genv(X, Y, Z, 2, m, nperm)
## Not run: cvPE
```

`cv.henv` *Cross validation for henv*

Description

Compute the prediction error for the heteroscedastic envelope estimator using m-fold cross validation.

Usage

```
cv.henv(X, Y, u, m, nperm)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the heteroscedastic envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate `m`-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

m <- 5
nperm <- 50

## Not run: cvPE <- cv.henv(X, Y, 2, m, nperm)
## Not run: cvPE
```

`cv.logit.env` *Cross validation for logit.env*

Description

Compute the prediction error for the envelope estimator in logistic regression using m-fold cross validation.

Usage

```
cv.logit.env(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- 5
nperm <- 50
## Not run: cvPE <- cv.logit.env(X, Y, 1, m, nperm)
## Not run: cvPE
```

`cv.penv` *Cross validation for penv*

Description

Compute the prediction error for the partial envelope estimator using m-fold cross validation.

Usage

```
cv.penv(X1, X2, Y, u, m, nperm)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.penv(X1, X2, Y, 1, m, nperm)
## Not run: cvPE
```

`cv.pois.env` *Cross validation for pois.env*

Description

Compute the prediction error for the envelope estimator in poisson regression using m-fold cross validation.

Usage

```
cv.pois.env(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be counts.
`u`	Dimension of the envelope. An integer between 0 and p.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.pois.env(X, Y, 1, m, nperm)
## Not run: cvPE
```

`cv.rrenv` *Cross validation for rrenv*

Description

Compute the prediction error using m-fold cross validation for the reduced rank envelope estimator.

Usage

```
cv.rrenv(X, Y, u, d, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.

`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

m <- 5
nperm <- 50
## Not run: cvPE <- cv.rrenv(X, Y, 4, 2, m, nperm)
## Not run: cvPE
```

`cv.rrenv.apweights` *Cross validation for rrenv.apweights*

Description

Compute the prediction error using m-fold cross validation for the reduced rank envelope estimator that accommodates nonconstant error covariance.

Usage

```
cv.rrenv.apweights(X, Y, u, d, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

m <- 5
nperm <- 50
## Not run: cvPE <- cv.rrenv.apweights(X, Y, 3, 2, m, nperm)
## Not run: cvPE
```

`cv.senv` *Cross validation for senv*

Description

Compute the prediction error for the scaled response envelope estimator using m-fold cross validation.

Usage

```
cv.senv(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.senv(X, Y, 2, m, nperm)
## Not run: cvPE
```

`cv.stenv` *Cross validation for stenv*

Description

Compute the prediction error for the simultaneous envelope estimator using m-fold cross validation.

Usage

```
cv.stenv(X, Y, q, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`q`	Dimension of the X-envelope. An integer between 0 and p.
`u`	Dimension of the Y-envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension (q, u), the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.stenv(X, Y, 2, 3, m, nperm)
## Not run: cvPE
```

`cv.sxenv` *Cross validation for sxenv*

Description

Compute the prediction error for the scaled predictor envelope estimator using m-fold cross validation.

Usage

```
cv.sxenv(X, Y, u, R, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`R`	The number of replications of the scales. A vector, the sum of all elements of R must be p.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

m <- 5
```

```
nperm <- 50
## Not run: cvPE <- cv.sxenv(X, Y, 2, R, m, nperm)
## Not run: cvPE
```

`cv.xenv` *Cross validation for xenv*

Description

Compute the prediction error for the predictor envelope estimator using m-fold cross validation.

Usage

```
cv.xenv(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`u`	Dimension of the envelope. An integer between 0 and p.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.xenv(X, Y, 2, m, nperm)
## Not run: cvPE
```

`d.select` *Select the rank of beta*

Description

This function outputs the rank selected by a chi-squared test developed by Bura and Cook (2003) with specified significance level for the beta.

Usage

```
d.select(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Details

This function estimate the rank of beta using a chi-squared test. The test statistic and degrees of freedom are described in Bura and Cook (2003).

Value

`rank.beta`	Rank of beta guided by the Bura-Cook estimator.

References

Bura, E. and Cook, R. D. (2003). Rank estimation in reduced-rank regression. Journal of Multivariate Analysis, 87, 159 - 176.

`env` *Fit the response envelope model*

Description

Fit the response envelope model in multivariate linear regression with dimension u.

Usage

```
env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace. An r by u matrix. The default is the one generated by function envMU.

Details

This function fits the envelope model to the responses and predictors,

$$Y = \mu + \Gamma\eta X + \varepsilon, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`Sigma`	The envelope estimator of the error covariance matrix.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- env(X, Y, 1)
m
m$beta
```

`env.apweights` *Fit the envelope model with nonconstant variance*

Description

For fixed envelope dimension u, fit the envelope model in multivariate linear regression with nonconstant error variance.

Usage

```
env.apweights(X, Y, u, asy = TRUE)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

Details

This function fits the envelope model to the responses and predictors,

$$Y_i = \mu + \Gamma \eta X_i + \varepsilon_i, \Sigma = c_i(\Gamma \Omega \Gamma' + \Gamma_0 \Omega_0 \Gamma_0'), i = 1, ..., n,$$

using the maximum likelihood estimation. It allows that the error covariance matrix to be nonconstant. When the dimension of the envelope is between 1 and r-1, the alternating algorithm in Forzani and Su (2021) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`Sigma`	The envelope estimator of the error covariance matrix.
`Gamma`	An orthogonal basis of the envelope subspace.
`Gamma0`	An orthogonal basis of the complement of the envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator (with consideration of nonconstant variance) over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.
`C1`	The estimated weights c_i.

References

Forzani, L. and Su, Z. (2021). Envelopes for elliptical multivariate linear regression. Statist. Sinica 31, 301-332.

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]
## Not run: u <- u.env.apweights(X, Y)
## Not run: u

m <- env.apweights(X, Y, 2)
m
m$beta
```

`env.tcond` *Fit the envelope model with t-distributed errors*

Description

For fixed envelope dimension u, fit the envelope model in multivariate linear regression, where the errors follow a multivariate t-distribution.

Usage

```
env.tcond(X, Y, u, df, asy = TRUE)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`df`	Degrees of freedom of the t-distribution. A positive number that is greater than 2.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

Details

This function fits the envelope model to the responses and predictors,

$$Y_i = \alpha + \Gamma\eta X_i + \varepsilon_i, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0', i = 1, ..., n,$$

using the maximum likelihood estimation. The errors ε_i follow a multivariate t-distribution with scale matrix Σ and degrees of freedom `df`. When the dimension of the envelope is between 1 and r-1, the alternating algorithm in Forzani and Su (2021) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`Sigma`	The envelope estimator of the error covariance matrix.
`Gamma`	An orthogonal basis of the envelope subspace.
`Gamma0`	An orthogonal basis of the complement of the envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator (with consideration of nonconstant variance) over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Forzani, L. and Su, Z. (2021). Envelopes for elliptical multivariate linear regression. Statist. Sinica 31, 301-332.

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]
## Not run: u <- u.env.tcond(X, Y, 6)
## Not run: u

m <- env.tcond(X, Y, 2, 6)
m
m$beta
```

`eppls` *Fit the Envelope-based Partial Partial Least Squares model*

Description

Fit the Envelope-based Partial Partial Least Squares model for multivariate linear regression with dimension u.

Usage

```
eppls(X1, X2, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X1`	An n by $p1$ matrix of continuous predictors, where $p1$ is the number of continuous predictors with $p1 < n$.
`X2`	An n by $p2$ matrix of categorical predictors, where $p2$ is the number of categorical predictors with $p2 < n$.
`Y`	An n by r matrix of multivariate responses, where r is the number of responses.
`u`	A given dimension of the Envelope-based Partial Partial Least Squares. It should be an interger between 0 and $p1$.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace. An r by u matrix. The default is the one generated by function envMU.

Details

This function the Envelope-based Partial Partial Least Squares model for multivariate linear regression with dimension u,

$$Y = \mu + \Gamma\eta X + \varepsilon, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and $p1$-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is $p1$, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`muY`	The estimator of mean of `Y`.
`mu1`	The estimator of mean of `X1`.

`mu2`	The estimator of mean of `X2`.
`beta1`	A $p1$ by r matrix for the estimator of regression coefficients for continuous predictors.
`beta2`	A $p2$ by r matrix for the estimator of regression coefficients for categorical predictors.
`Gamma`	An $p1$ by d matrix for the orthogonal basis of the Envelope-based Partial Partial Least Squares.
`Gamma0`	An $p1$ by $(p1 - d)$ matrix for the orthogonal basis of the complement of the Envelope-based Partial Partial Least Squares.
`gamma`	A $p2$ by $p1$ matrix for the estimator of regression coefficients based on the regression of `X1` on `X2`.
`eta`	A d by $p1$ matrix for the coordinates of `beta1` with respect to `Gamma`.
`Omega`	A d by d matrix for the coordinates of `SigmaX1` with respect to `Gamma`.
`Omega0`	A $(p1 - d)$ by $(p1 - d)$ matrix for the coordinates of `SigmaX1` with respect to `Gamma0`.
`SigmaX1`	The estimator of error covariance matrix `Sigma[1|2]`.
`SigmaYcX`	The estimator of error covariance matrix `Sigma[Y|X]`.
`loglik`	The maximized log likelihood function.
`n`	The number of observations in the data.
`covMatrix1`	The asymptotic covariance of vec(beta1). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`covMatrix2`	The asymptotic covariance of vec(beta2). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE1`	The asymptotic standard error matrix for elements in `beta1`. The multiplication by the reciprocal of square root of n returns actual standard errors.
`asySE2`	The asymptotic standard error matrix for elements in `beta2`. The multiplication by the reciprocal of square root of n returns actual standard errors.

References

Park, Y., Su, Z. and Chung, D. (2022+) Envelope-based Partial Partial Least Squares with Application to Cytokine-based Biomarker Analysis for COVID-19.

Examples

```
data(amitriptyline)

Y <- amitriptyline[ , 1:2]
X1 <- amitriptyline[ , 4:7]
X2 <- amitriptyline[ , 3]
u <- u.eppls(X1, X2, Y)

u

m <- eppls(X1, X2, Y, 2)
m
```

`felmdir` *Fit the functional envelope linear model*

Description

Fit the response and predictor envelope model in function-on-function linear regression with dimensions ux and uy, using the direct estimation.

Usage

```
felmdir(X, Y, ux, uy, t1, t2, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`ux`	Dimension of the predictor envelope. An integer between 0 and number of knots +2.
`uy`	Dimension of the response envelope. An integer between 0 and number of knots +2.
`t1`	The observed time points for the predictor functions.
`t2`	The observed time points for the response functions.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function fits the envelope model to the function-on-function linear regression,

$$Y = \alpha + BX + \epsilon$$

, where X and Y are random functions in Hilbert spaces H_X and H_Y, α is a fixed member in H_Y, ϵ is a random member of H_Y, and B: $H_X - > H_Y$ is a linear operator. We use cubic splines as the basis for both H_X and H_Y. The coefficients $[X]$ and $[Y]$ with respect to the basis are computed. The predictor and response envelope model is fitted on the linear regression model of $[Y]$ on $[X]$. In this method, we do not need to estimate the eigenfunctions of $Sigma_X$ and $Sigma_\epsilon$. Based on the estimation result, the fitted value of Y is calculated. The standard function-on-function regression model also works through the linear regression model of $[Y]$ on $[X]$. But instead of fitting an envelope model, it fits a standard linear regression model, based on which the fitted value of $[Y]$ is calculated. The details are elaborated in Section 5, direct estimation, in the reference of Su et al. (2022).

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`betafull`	The standard estimator, i.e., the OLS estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`alpha`	The envelope estimator of the intercept in the regression of $[Y]$ on $[X]$.
`alphafull`	The standard estimator of the intercept in the regression of $[Y]$ on $[X]$.
`fitted.env`	The fitted value of Y computed from the functional envelope linear model.
`fitted.full`	The fitted value of Y computed from the standard function-to-function linear model.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

m <- felmdir(X, Y, 3, 1, t1, t2)
head(m$fitted.env)
head(m$fitted.full)
```

`felmKL` *Fit the functional envelope linear model*

Description

Fit the response and predictor envelope model in function-on-function linear regression with dimensions ux and uy, using Karhunen-Loeve expansion based estimation.

Usage

```
felmKL(X, Y, ux, uy, t1, t2, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`ux`	Dimension of the predictor envelope. An integer between 0 and number of knots +2.
`uy`	Dimension of the response envelope. An integer between 0 and number of knots +2.
`t1`	The observed time points for the predictor functions.
`t2`	The observed time points for the response functions.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function fits the envelope model to the function-on-function linear regression,

$$Y = \alpha + BX + \epsilon,$$

where X and Y are random functions in Hilbert spaces H_X and H_Y, α is a fixed member in H_Y, ϵ is a random member of H_Y, and B: $H_X - > H_Y$ is a linear operator. We use cubic splines as the basis for both H_X and H_Y in the estimation of the eigenfunctions of $Sigma_X$ and $Sigma_\epsilon$. The coefficients $[X]$ and $[Y]$ with respect to the estimated eigenfunctions are computed. The predictor and response envelope model is fitted on the linear regression model of $[Y]$ on $[X]$. Based on its result, the fitted value of Y is calculated. The standard function-on-function regression model also works through the linear regression model of $[Y]$ on $[X]$. But instead of fitting an envelope model, it fits a standard linear regression model, based on which the fitted value of Y is calculated. The details are elaborated in Section 6, Karhunen-Lo‘eve expansion based estimation, in the reference of Su et al. (2022).

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`betafull`	The standard estimator, i.e., the OLS estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`alpha`	The envelope estimator of the intercept in the regression of $[Y]$ on $[X]$.
`alphafull`	The standard estimator of the intercept in the regression of $[Y]$ on $[X]$.
`phihat.cord`	The estimated coordinates of eigenfunctions of $Sigma_\epsilon$ with respect to the cubic splines.

`psihat.cord`	The estimated coordinates of eigenfunctions of $Sigma_X$ with respect to the cubic splines.
`fitted.env`	The fitted value of Y computed from the functional envelope linear model.
`fitted.full`	The fitted value of Y computed from the standard function-to-function linear model.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

m <- felmKL(X, Y, 4, 3, t1, t2)
head(m$fitted.env)
head(m$fitted.full)
```

`fiberpaper` *Pulp and Paper Data*

Description

Pulp and paper property

Usage

```
data("fiberpaper")
```

Format

A data frame with 62 observations on the following 8 variables.

`V1`	Breaking length.
`V2`	Elastic modulus.
`V3`	Stress at failure.
`V4`	Burst strength.
`V5`	Arithmetic fiber length.
`V6`	Long fiber fraction.
`V7`	Fine fiber fraction.
`V8`	Zero span tensile.

Details

This data set contains measurements of properties of pulp fibers and the paper made from them.

References

Johnson, R.A. and Wichern, D.W. (2007). Applied Multivariate Statistical Analysis, 6th edition.

`genv` *Fit the groupwise envelope model*

Description

Fit the groupwise envelope model in multivariate linear regression with dimension u. The groupwise envelope model is designed to accommodate both distinct regression coefficients and distinct error structures for different groups.

Usage

```
genv(X, Y, Z, u, asy = TRUE, fit = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Z`	A group indicator vector of length n, where n denotes the number of observations.
`u`	Dimension of the groupwise envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`fit`	Flag for computing the fitted response. The default is `TRUE`.
`init`	The user-specified value of Gamma for the groupwise envelope subspace. An r by u matrix. The default is the one generated by function genvMU.

Details

This function fits the groupwise envelope model to the responses and predictors,

$$Y_{(l)j} = \mu_{(l)} + \Gamma\eta_{(l)j}X_{(l)j} + \varepsilon_{(l)j}, \Sigma_{(l)} = \Gamma\Omega_{(l)}\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

for l = 1, ..., L, using the maximum likelihood estimation. When the dimension of the groupwise envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different. When L is 1, the groupwise envelope model degenerates to the envelope model in Cook et al. (2010).

Value

The output is a list that contains the following components:

`beta`	A list of r by p matrices for the estimator of regression coefficients. `beta`[[i]] indicates the estimator of regression coefficient for the ith group.
`Sigma`	A list of the estimator of error covariance matrix. `Sigma`[[i]] contains the estimated covariance matrix for the ith group.
`Gamma`	An orthonormal basis of the groupwise envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the groupwise envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimator of group mean. A r by L matrix whose ith column contains the mean for the group.
`loglik`	The maximized log likelihood function.
`covMatrix`	A list of the asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n. `covMatrix`[[i]] contains the asymptotic covariance matrix for the ith group.
`asySE`	A list of the asymptotic standard error for elements in beta under the groupwise envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n). `asySE`[[i]] contains the asymptotic standard error for elements in beta[[i]].
`ratio`	A list of the asymptotic standard error ratio of the standard multivariate linear regression estimator over the groupwise envelope estimator, for each element in beta. `ratio`[[i]] contains the asymptotic standard error ratio for the ith group.
`groupInd`	A matrix containing the unique values of group indicators. The matrix has p rows.
`n`	The number of observations in the data.
`ng`	The number of observations in each group.
`Yfit`	Fitted responses.

References

Park, Y., Su, Z. and Zhu, H. (2017) Groupwise envelope models for Imaging Genetic Analysis. Biometrics, to appear.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
## Not run: u <- u.genv(X, Y, Z)
## Not run: u
## Not run: m <- genv(X, Y, Z, 2)
```

`henv` *Fit the heteroscedastic envelope model*

Description

Fit the heteroscedastic envelope model derived to incorporate heteroscedastic error structure in the context of estimating multivariate means for different groups with dimension u.

Usage

```
henv(X, Y, u, asy = TRUE, fit = TRUE, init = NULL)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the heteroscedastic envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`fit`	Flag for computing the fitted response. The default is `TRUE`.
`init`	The user-specified value of Gamma for the heteroscedastic envelope subspace. An r by u matrix. The default is the one generated by function henvMU.

Details

This function fits the heteroscedastic envelope model to the responses,

$$Y_{(i)j} = \mu + \Gamma\eta_{(i)} + \varepsilon_{(i)j}, \Sigma_{(i)} = \Gamma\Omega_{(i)}\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

for i = 1, ..., p, using the maximum likelihood estimation. When the dimension of the heteroscedastic envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression for comparing group means. When the dimension is 0, it means there is no any group effect, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The heteroscedastic envelope estimator of the group main effect. An r by p matix, the ith column of the matrix contains the main effect for the ith group.
`Sigma`	A list of the heteroscedastic envelope estimator of the error covariance matrix. `Sigma`[[i]] contains the estimated covariance matrix for the ith group.
`Gamma`	An orthonormal basis of the heteroscedastic envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the heteroscedastic envelope subspace.
`eta`	A list of the coordinates of beta with respect to Gamma. `eta` [[i]] indicates the coordinates of the main effect of the ith group with respect to Gamma.
`Omega`	A list of the coordinates of Sigma with respect to Gamma. `Omega`[[i]] indicates the coordinates of the covariance matrix of the ith group with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The heteroscedastic envelope estimator of the grand mean. A r by 1 matrix.
`mug`	A list of the heteroscedastic envelope estimator of the group mean. An r by p matix, the ith column of the matrix contains the mean for the ith group.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of (mu, vec(beta)')'. An r(p + 1) by r(p + 1) matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the heteroscedastic envelope model. An r by p matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression for comparing group means over the heteroscedastic envelope estimator, for each element in beta. An r by p matrix.
`groupInd`	A matrix containing the unique values of group indicators. The matrix has p rows.
`n`	The number of observations in the data.
`ng`	The number of observations in each group.
`Yfit`	Fitted responses.

References

Su, Z. and Cook, R. D. (2013) Estimation of Multivariate Means with Heteroscedastic Error Using Envelope Models. Statistica Sinica, 23, 213-230.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: u <- u.henv(X, Y)
## Not run: u

## Not run: m <- henv(X, Y, 2)
```

`horseshoecrab` *Horseshoe Crab Data*

Description

The number of satellite male crabs near a female crab upon characteristic of the female horseshoe crabs.

Usage

```
data("horseshoecrab")
```

Format

A data frame with 173 observations on the following 5 variables.

- `V1` Color.
- `V2` Condition of spine.
- `V3` Width of shell.
- `V4` Satellite.
- `V5` Weight.

Details

This data set contains the number of satellite male crabs and characteristics of the female horseshoe crabs.

References

Agresti, A. (2007). An Introduction to Categorical Data Analysis, 2nd edition.

`logit.env` *Fit the envelope model in logistic regression*

Description

Fit the envelope model in logistic regression with dimension u.

Usage

```
logit.env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace in logistic regression. An p by u matrix. The default is the one generated by function logit.envMU.

Details

This function fits the envelope model in logistic regression,

$$Y = exp(\mu + \beta' X)/(1 + exp(\mu + \beta' X)), \Sigma_X = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. This model works the best when X is multivariate normal.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the canonical parameter.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The estimated beta of the canonical parameter with respect to Gamma.
`Omega`	The coordinates of SigmaX with respect to Gamma.

`Omega0`	The coordinates of SigmaX with respect to Gamma0.
`mu`	The estimated intercept of the canonical parameter.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

## Not run: u <- u.logit.env(X, Y)
## Not run: u

## Not run: m <- logit.env(X, Y, 1)
## Not run: m$beta
```

`NJdata` *New Jersey Open Covid-19 Dataset*

Description

Daily COVID-19 infections, weather information as well as mobility of people for all 21 counties in May and June, 2020.

Usage

```
data(NJdata)
```

Format

A data frame with 1281 entries on the following 9 variables.

`key`	The FIPS (Federal Information Processing System) county codes.
`date`	Date of measurements.
`mobility_retail_and_recreation`	Percentage change in visits to restaurants, cafes, shopping centers, theme parks, museums, libraries, and movie theaters compared to baseline. The baseline is the median value, for the corresponding day of the week, during the 5-week period Jan 3-Feb 6, 2020.
`mobility_grocery_and_pharmacy`	Percentage change in visits to places like grocery markets, food warehouses, farmers markets, specialty food shops, drug stores, and pharmacies compared to baseline. The baseline is the median value, for the corresponding day of the week, during the 5-week period Jan 3-Feb 6, 2020.
`mobility_workplaces`	Percentage change in visits to places of work compared to baseline. The baseline is the median value, for the corresponding day of the week, during the 5-week period Jan 3-Feb 6, 2020.
`new_confirmed`	Count of new cases confirmed after positive test on this date. Values can be negative, typically indicating a correction or an adjustment in the way they were measured. For example, a case might have been incorrectly flagged as recovered one date so it will be subtracted from the following date.
`new_deceased`	Count of new deaths from a positive COVID-19 case on this date. Values can be negative, typically indicating a correction or an adjustment in the way they were measured. For example, a case might have been incorrectly flagged as recovered one date so it will be subtracted from the following date.
`average_temperature`	Recorded hourly average temperature, in celsius.
`rainfall`	Rainfall during the entire day, in millimeters.

Details

This dataset contains COVID-19 new confirmed cases and deceased cases, average temperature and total rainfall, as well as the movement of people to different categories of places for each date-region pair.

Source

https://github.com/open-covid-19/data#open-covid-19-dataset

`penv` *Fit the partial envelope model*

Description

Fit the partial envelope model in multivariate linear regression with dimension u. The partial envelope model focuses on the coefficients of main interest.

Usage

```
penv(X1, X2, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the partial envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the partial envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the partial envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the partial envelope subspace. An r by u matrix. The default is the one generated by function envMU.

Details

This function fits the partial envelope model to the responses Y and predictors X1 and X2,

$$Y = \mu + \Gamma\eta X_1 + \beta_2 X_2 + \varepsilon, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and r - 1, we implemented the algorithm in Su and Cook (2011), but the partial envelope subspace is estimated using the blockwise coordinate descent algorithm in Cook et al. (2016). When the dimension is r, then the partial envelope model degenerates to the standard multivariate linear regression with Y as the responses and both X1 and X2 as predictors. When the dimension is 0, X1 and Y are uncorrelated, and the fitting is the standard multivariate linear regression with Y as the responses and X2 as the predictors.

Value

The output is a list that contains the following components:

`beta1`	The partial envelope estimator of beta1, which is the regression coefficients for X1.
`beta2`	The partial envelope estimator of beta2, which is the regression coefficients for X2.
`Sigma`	The partial envelope estimator of the error covariance matrix.
`Gamma`	An orthonormal basis of the partial envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the partial envelope subspace.
`eta`	The coordinates of beta1 with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept in the partial envelope model.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta), while beta = (beta1, beta2). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE1`	The asymptotic standard error for elements in beta1 under the partial envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`asySE2`	The asymptotic standard error for elements in beta2 under the partial envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the stanard multivariate linear regression estimator over the partial envelope estimator, for each element in beta1.
`n`	The number of observations in the data.

References

Su, Z. and Cook, R.D. (2011). Partial envelopes for efficient estimation in multivariate linear regression. Biometrika 98, 133 - 146.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
u <- u.penv(X1, X2, Y)
u

m <- penv(X1, X2, Y, 1)
m
m$beta1
```

`pois.env` *Fit the envelope model in poisson regression*

Description

Fit the envelope model in poisson regression with dimension u.

Usage

```
pois.env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be counts.
`u`	Dimension of the envelope. An integer between 0 and p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace in poisson regression. An p by u matrix. The default is the one generated by function pois.envMU.

Details

This function fits the envelope model in poisson regression,

$$Y = exp(\mu + \beta' X), \Sigma_X = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. This model works the best when X is multivariate normal.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the canonical parameter.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The estimated beta of the canonical parameter with respect to Gamma.
`Omega`	The coordinates of SigmaX with respect to Gamma.

`Omega0`	The coordinates of SigmaX with respect to Gamma0.
`mu`	The estimated intercept of the canonical parameter.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

## Not run: u <- u.pois.env(X, Y)
## Not run: u

m <- pois.env(X, Y, 1)
m$beta
```

`pred.env` *Estimation or prediction for env*

Description

Perform estimation or prediction under the response envelope model.

Usage

```
pred.env(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- env(X, Y, 1)
m

X <- as.matrix(X)
pred.res <- pred.env(m, X[2, ])
pred.res
```

`pred.env.apweights` *Estimation or prediction for env.apweights*

Description

Perform estimation or prediction under the response envelope model with accommodation to nonconstant variance.

Usage

```
pred.env.apweights(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.apweights.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model with nonconstant variance at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]
## Not run: u <- u.envapweights(X, Y)
## Not run: u

m <- env.apweights(X, Y, 2)
m

X <- as.matrix(X)
pred.res <- pred.env.apweights(m, X[2, ])
pred.res
```

`pred.env.tcond` *Estimation or prediction for env.tcond*

Description

Perform estimation or prediction under the response envelope model, where the errors follow a multivariate t-distribution.

Usage

```
pred.env.tcond(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.tcond.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model with t distributed errors at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]
## Not run: u <- u.env.tcond(X, Y, 6)
## Not run: u

m <- env.tcond(X, Y, 2, 6)
m

X <- as.matrix(X)
pred.res <- pred.env.tcond(m, X[2, ])
pred.res
```

`pred.eppls` *Estimation or prediction for eppls*

Description

Perform estimation or prediction under the Envelope-based Partial Partial Least Squares.

Usage

```
pred.eppls(m, X1new, X2new)
```

Arguments

`m`	A list containing estimators and other statistics inherited from eppls.
`X1new`	The value of X1 with which to estimate or predict Y. A p1 dimensional vector.
`X2new`	The value of X2 with which to estimate or predict Y. A p2 dimensional vector.

Details

This function evaluates the partial envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at X1new and X2new.
`covMatrix.estm`	The covariance matrix of the fitted value at X1new and X2new.
`SE.estm`	The standard error of the fitted value at X1new and X2new.
`covMatrix.pred`	The covariance matrix of the predicted value at X1new and X2new.
`SE.pred`	The standard error of the predicted value at X1new and X2new.

References

Park, Y., Su, Z. and Chung, D. (2022+) Envelope-based Partial Partial Least Squares with Application to Cytokine-based Biomarker Analysis for COVID-19.

Examples

```
data(amitriptyline)

Y <- amitriptyline[ , 1:2]
X1 <- amitriptyline[ , 4:7]
X2 <- amitriptyline[ , 3]
u <- u.eppls(X1, X2, Y)
```

```
u

m <- eppls(X1, X2, Y, 2)

pred.res <- pred.eppls(m, X1[1, ], X2[1])
pred.res
```

`pred.felmdir` *Estimation or prediction for felmdir*

Description

Perform estimation or prediction under the functional envelope linear model, using the direct estimation.

Usage

```
pred.felmdir(X, Y, ux, uy, t1, t2, Xnew, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`ux`	Dimension of the predictor envelope. An integer between 0 and number of knots +2.
`uy`	Dimension of the response envelope. An integer between 0 and number of knots +2.
`t1`	The observed time points for the predictor functions.
`t2`	The observed time points for the response functions.
`Xnew`	The value of X with which to estimate or predict Y. A T1 dimensional vector. The observed time points should be the same as those of X.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function evaluates the functional envelope linear model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided. The estimation method uses the direct estimation in Su et al. (2022) with cubic splines.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew. The fitted or predicted values are at the same observation points as Y.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

m <- pred.felmdir(X, Y, 3, 1, t1, t2, X[1,])
m$value
m$SE.estm
m$SE.pred
```

`pred.felmKL` *Estimation or prediction for felmKL*

Description

Perform estimation or prediction under the functional envelope linear model, using Karhunen-Loeve expansion based estimation.

Usage

```
pred.felmKL(X, Y, ux, uy, t1, t2, Xnew, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`ux`	Dimension of the predictor envelope. An integer between 0 and number of knots +2.
`uy`	Dimension of the response envelope. An integer between 0 and number of knots +2.
`t1`	The observed time points for the predictor functions.
`t2`	The observed time points for the response functions.
`Xnew`	The value of X with which to estimate or predict Y. A T1 dimensional vector. The observed time points should be the same as those of X.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function evaluates the functional envelope linear model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided. The estimation method uses the Karhunen-Loeve expansion based estimation in Su et al. (2022) with cubic splines.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew. The fitted or predicted values are at the same observation points as Y.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
```

```
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

m <- pred.felmKL(X, Y, 4, 3, t1, t2, X[1,])
m$value
m$SE.estm
m$SE.pred
```

`pred.genv` *Estimation or prediction for genv*

Description

Perform estimation or prediction under the groupwise envelope model.

Usage

```
pred.genv(m, Xnew, Znew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.
`Znew`	A group indicator of X.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew with a group indicator Z = Znew, or prediction: predict Y when X = Xnew with a group indicator Z = Znew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Znew.
`covMatrix.estm`	The covariance matrix of the fitted value at Znew.
`SE.estm`	The standard error of the fitted value at Znew.
`covMatrix.pred`	The covariance matrix of the predicted value at Znew.
`SE.pred`	The standard error of the predicted value at Znew.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
u <- u.genv(X, Y, Z)
u

m <- genv(X, Y, Z, 2)
m

X <- as.matrix(X)
pred.res <- pred.genv(m, X[2, ], Z[2])
pred.res
```

`pred.henv` *Estimation or prediction for henv*

Description

Perform estimation or prediction under the heteroscedastic envelope model.

Usage

```
pred.henv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from henv.
`Xnew`	The value of X with which to estimate or predict Y. An r by 1 vector.

Details

This function evaluates the heteroscedastic envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]


## Not run: m <- henv(X, Y, 2)

## Not run: pred.res <- pred.henv(m, X[2])
```

`pred.logit.env` *Estimation or prediction for logit.env*

Description

Perform estimation or prediction under the envelope model in logistic regression.

Usage

```
pred.logit.env(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew. The covariance matrix of estimation and the standard errors of estimation are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- logit.env(X, Y, 1)

pred.res <- pred.logit.env(m, X[1, ])
pred.res
```

`pred.penv` *Estimation or prediction for penv*

Description

Perform estimation or prediction under the partial envelope model.

Usage

```
pred.penv(m, X1new, X2new)
```

Arguments

`m`	A list containing estimators and other statistics inherited from penv.
`X1new`	The value of X1 with which to estimate or predict Y. A p1 dimensional vector.
`X2new`	The value of X2 with which to estimate or predict Y. A p2 dimensional vector.

Details

This function evaluates the partial envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at X1new and X2new.
`covMatrix.estm`	The covariance matrix of the fitted value at X1new and X2new.

`SE.estm`	The standard error of the fitted value at X1new and X2new.
`covMatrix.pred`	The covariance matrix of the predicted value at X1new and X2new.
`SE.pred`	The standard error of the predicted value at X1new and X2new.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- penv(X1, X2, Y, 1)

pred.res <- pred.penv(m, X1[1], X2[1, ])
pred.res
```

`pred.pois.env` *Estimation or prediction for pois.env*

Description

Perform estimation or prediction under the envelope model in poisson regression.

Usage

```
pred.pois.env(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope in poisson regression at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix of estimation and the standard errors of estimation are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- pois.env(X, Y, 1)

pred.res <- pred.pois.env(m, X[1, ])
pred.res
```

`pred.rrenv` *Estimation or prediction for rrenv*

Description

Perform estimation or prediction under the reduced rank envelope model.

Usage

```
pred.rrenv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from rrenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the reduced rank envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

m <- rrenv(X, Y, 4, 2)
m

X <- as.matrix(X)
pred.res <- pred.rrenv(m, X[2, ])
pred.res
```

`pred.rrenv.apweights` *Estimation or prediction for rrenv.apweights*

Description

Perform estimation or prediction under the reduced rank envelope model with nonconstant error covariance.

Usage

```
pred.rrenv.apweights(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from rrenv.apweights.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the reduced rank envelope model that accommodates nonconstant error covariance at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here

## Not run: m <- rrenv.apweights(X, Y, 3, 2)
## Not run: m

## Not run: X <- as.matrix(X)
## Not run: pred.res <- pred.rrenv.apweights(m, X[2, ])
## Not run: pred.res
```

`pred.senv` *Estimation or prediction for senv*

Description

Perform estimation or prediction under the scaled response envelope model.

Usage

```
pred.senv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from scale.env.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the scaled envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- senv(X, Y, 2)

pred.res <- pred.senv(m, X[2, ])
pred.res
```

`pred.stenv` *Estimation or prediction for stenv*

Description

Perform estimation or prediction under the simultaneous envelope model.

Usage

```
pred.stenv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from stenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the simultaneous envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , 5:7]
Y <- fiberpaper[ , 1:4]

m <- stenv(X, Y, 2, 3)
m

pred.res <- pred.stenv(m, X[1, ])
pred.res
```

`pred.sxenv` *Estimation or prediction for sxenv*

Description

Perform estimation or prediction under the scaled predictor envelope model.

Usage

```
pred.sxenv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from stenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the scaled envelope model in the predictor space at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

m <- sxenv(X, Y, 2, R)

pred.res <- pred.sxenv(m, X[1, ])
pred.res
```

`pred.xenv` *Estimation or prediction for xenv*

Description

Perform estimation or prediction under the predictor envelope model.

Usage

```
pred.xenv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

m <- xenv(X, Y, 2)
m

pred.res <- pred.xenv(m, X[1, ])
pred.res
```

`pred2.env` *Estimation or prediction for env*

Description

Perform estimation or prediction under the envelope model through partial envelope model.

Usage

```
pred2.env(X, Y, u, Xnew)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	The dimension of the constructed partial envelope model.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided. Compared to predict.env, this function performs prediction through partial envelope model, which can be more accurate if the partial envelope is of smaller dimension and contains less variant material information. The constructed partial envelope model is obtained by the following: Let A0 by a p by p-1 matrix, such that A = (Xnew, A0) has full rank. Let phi1 = beta * Xnew, phi2 = beta * A0, phi = (phi1, phi2) and X = inverse of A * X = (Z1, Z2')'. Then the model Y = alpha + beta * X + epsilon can be reparameterized as Y = alpha + phi1 * Z1 + phi2 * Z2 + epsilon. We then fit a partial envelope model with Z1 as the predictor of interest, and predict at (Z1, Z2')' = inverse of A * Xnew.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.pred2.env(X, Y, X[10, ])
pred.res <- pred2.env(X, Y, u$u.bic, X[10, ])
pred.res$SE.estm
pred.res$SE.pred
```

`rrenv` *Fit the reduced-rank envelope model*

Description

Fit the reduced-rank envelope model with rank d and fixed envelope dimension u.

Usage

```
rrenv(X, Y, u, d, asy = TRUE)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`asy`	Flag for computing the asymptotic variance of the reduced rank envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

Details

This function fits the reduced rank envelope model to the responses and predictors,

$$Y_i = \alpha + \Gamma\eta B X_i + \varepsilon_i, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0', i = 1, ..., n,$$

using the maximum likelihood estimation. The errors ε_i follow a normal distribution. When 0 < d < u < r, the estimation procedure in Cook et al. (2015) is implemented. When d < u = r, then the model is equivalent to a reduced rank regression model. When d = u, or d = p < r, then B can be taken as the identity matrix and the model reduces to a response envelope model. When the dimension is d = u = r, then the envelope model degenerates to the standard multivariate linear regression. When the u = 0, it means that X and Y are uncorrelated, and the fitting is different. If the error covariance matrix is nonconstant, see the function `rrenv.apweights`.

Value

The output is a list that contains the following components:

`Gamma`	An orthogonal basis of the envelope subspace.
`Gamma0`	An orthogonal basis of the complement of the envelope subspace.
`mu`	The estimated intercept.
`beta`	The envelope estimator of the regression coefficients.
`Sigma`	The envelope estimator of the error covariance matrix.
`eta`	The eta matrix in the coefficient matrix.
`B`	The B matrix in the coefficient matrix.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the reduced rank envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator (with consideration of nonconstant variance) over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Forzani, L. and Zhang, X. (2015). Envelopes and reduced-rank regression. Biometrika 102, 439-456.

Forzani, L. and Su, Z. (2021). Envelopes for elliptical multivariate linear regression. Statist. Sinica 31, 301-332.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
d <- d.select(X, Y, 0.01)
d

## Not run: u <- u.rrenv(X, Y, 2)
## Not run: u

m <- rrenv(X, Y, 4, 2)
m
m$beta
```

`rrenv.apweights` *Fit the reduced-rank envelope model with nonconstant variance*

Description

For rank d and fixed envelope dimension u, fit the reduced-rank envelope model with nonconstant error variance.

Usage

```
rrenv.apweights(X, Y, u, d, asy = TRUE)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`d`	The rank of the coefficient matrix. An integer between 0 and u.
`asy`	Flag for computing the asymptotic variance of the reduced rank envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

Details

This function fits the reduced rank envelope model to the responses and predictors,

$$Y_i = \alpha + \Gamma\eta B X_i + \varepsilon_i, \Sigma = c_i(\Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'), i = 1, ..., n,$$

using the maximum likelihood estimation. The errors ε_i follow a normal distribution. It allows that the error covariance matrix to be nonconstant. When 0 < d < u < r, the estimation procedure in Cook et al. (2015) is implemented. When d < u = r, then the model is equivalent to a reduced rank regression model. When d = u, or d = p < r, then B can be taken as the identity matrix and the model reduces to a response envelope model. When the dimension is d = u = r, then the envelope model degenerates to the standard multivariate linear regression. When the u = 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`Gamma`	An orthogonal basis of the envelope subspace.
`Gamma0`	An orthogonal basis of the complement of the envelope subspace.
`mu`	The estimated intercept.
`beta`	The envelope estimator of the regression coefficients.
`Sigma`	The envelope estimator of the error covariance matrix.
`eta`	The eta matrix in the coefficient matrix.
`B`	The B matrix in the coefficient matrix.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the reduced rank envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator (with consideration of nonconstant variance) over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.
`C1`	The estimated weights c_i.

References

Cook, R. D., Forzani, L. and Zhang, X. (2015). Envelopes and reduced-rank regression. Biometrika 102, 439-456.

Forzani, L. and Su, Z. (2021). Envelopes for elliptical multivariate linear regression. Statist. Sinica 31, 301-332.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
d <- d.select(X, Y, 0.01)
d

## Not run: u <- u.rrenv.apweights(X, Y, 2)
## Not run: u

## Not run: m <- rrenv.apweights(X, Y, 3, 2)
## Not run: m
## Not run: m$beta
```

`sales` *Sales staff Data*

Description

On the performance of a firm's sales staff

Usage

`data("sales")`

Format

A data frame with 50 observations on the following 7 variables.

`V1`	Index of sales growth.
`V2`	Index of sales profitability.
`V3`	Index of new account sales.
`V4`	Score on creativity.
`V5`	Score on mechanical reasoning test.
`V6`	Score on abstract reasoning test.
`V7`	Score on Mathematics test.

Details

This data set contains 3 measures of performance and 4 tests scores.

References

Johnson, R.A., Wichern, D.W. (2007). Applied Multivariage Statistical Analysis, 6th edition.

`senv` *Fit the scaled response envelope model*

Description

Fit the scaled response envelope model in multivariate linear regression with dimension u. The scaled response envelope model is a scale-invariant version of the response envelope model.

Usage

```
senv(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the scaled envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the scaled envelope subspace. An r by u matrix. The default is the one generated by function senvMU.

Details

This function fits the scaled envelope model to the responses and predictors,

$$Y = \mu + \Lambda\Gamma\eta X + \varepsilon, \Sigma = \Lambda\Gamma\Omega\Gamma'\Lambda + \Lambda\Gamma_0\Omega_0\Gamma_0'\Lambda$$

using the maximum likelihood estimation. When the dimension of the scaled envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the scaled envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The scaled envelope estimator of the regression coefficients.
`Sigma`	The scaled envelope estimator of the error covariance matrix.
`Lambda`	The matrix of estimated scale.
`Gamma`	An orthonormal basis of the scaled envelope subspace.

`Gamma0`	An orthonormal basis of the complement of the scaled envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Su, Z. (2013). Scaled Envelopes: scale Invariant and Efficient Estimation in Multivariate Linear Regression. Biometrika 100, 939 - 954.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u

m <- senv(X, Y, 2)
m$beta
```

`stenv` *Fit the simultaneous envelope model*

Description

Fit the simultaneous envelope model in multivariate linear regression with dimension (q, u).

Usage

```
stenv(X, Y, q, u, asy = TRUE, Pinit = NULL, Ginit = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`q`	Dimension of the X-envelope. An integer between 0 and p.
`u`	Dimension of the Y-envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`Pinit`	The user-specified value of Phi for the X-envelope subspace. An p by q matrix. The default is the one generated by function stenvMU.
`Ginit`	The user-specified value of Gamma for the Y-envelope subspace. An r by u matrix. The default is the one generated by function stenvMU.

Details

This function fits the envelope model to the responses and predictors simultaneously,

$$Y = \mu + \beta' X + \varepsilon, \beta = \Phi\eta\Gamma', \Sigma_{Y|X} = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0', \Sigma_X = \Phi\Delta\Phi' + \Phi_0\Delta_0\Phi_0'$$

using the maximum likelihood estimation. When the dimension of the Y-envelope is between 1 and r-1 and the dimension of the X-envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is (p, r), then the envelope model degenerates to the standard multivariate linear regression. When the dimension of the Y-envelope is r, then the envelope model degenerates to the standard envelope model. When the dimension of X-envelope is p, then the envelope model degenerates to the envelope model in the predictor space. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`SigmaYcX`	The envelope estimator of the error covariance matrix.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the Y-envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the Y-envelope subspace.
`eta`	The coordinates of beta with respect to Gamma and Phi.
`Omega`	The coordinates of SigmaYcX with respect to Gamma.
`Omega0`	The coordinates of SigmaYcX with respect to Gamma0.
`mu`	The estimated intercept.
`Phi`	An orthonormal basis of the X-envelope subspace.

`Phi0`	An orthonormal basis of the complement of the X-envelope subspace.
`Delta`	The coordinates of SigmaX with respect to Phi.
`Delta0`	The coordinates of SigmaX with respect to Phi0.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Simultaneous Envelopes for Multivariate Linear Regression. Technometrics 57, 11 - 25.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]
u <- u.stenv(X, Y)
u

m <- stenv(X, Y, 2, 3)
m
m$beta
```

`sxenv` *Fit the scaled predictor envelope model*

Description

Fit the scaled predictor envelope model in multivariate linear regression with dimension u. The scaled predictor envelope model is a scale-invariant version of the predictor envelope model.

Usage

```
sxenv(X, Y, u, R, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope in the predictor space. An integer between 0 and p.
`R`	The number of replications of the scales. A vector, the sum of all elements of R must be p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the scaled envelope subspace in the predictor space. An p by u matrix. The default is the one generated by function sxenvMU.

Details

This function fits the scaled envelope model in the predictor space to the responses and predictors,

$$Y = \mu_Y + \eta'\Gamma'\Lambda^{-1}(X - \mu_X) + \varepsilon, \Sigma_X = \Lambda\Gamma\Omega\Gamma'\Lambda + \Lambda\Gamma_0\Omega_0\Gamma_0'\Lambda$$

using the maximum likelihood estimation. When the dimension of the scaled envelope in the predictor space is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is p, then the scaled envelope model in the predictor space degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The scaled envelope estimator of the regression coefficients.
`Sigma`	The scaled envelope estimator of the error covariance matrix.
`Lambda`	The matrix of estimated scale.
`Gamma`	An orthonormal basis of the scaled envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the scaled envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`muY`	The mean of Y.
`muX`	The mean of X.
`loglik`	The maximized log likelihood function.

`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Su, Z. (2016). Scaled Predictor Envelopes and Partial Least Squares Regression. Technometrics 58, 155 - 165.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

m <- sxenv(X, Y, 2, R)
m$beta
```

`testcoef.env` *Hypothesis test of the coefficients of the response envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the response envelope model.

Usage

```
testcoef.env(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.

`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
m <- env(X, Y, 1)
m

L <- diag(6)
R <- as.matrix(1)
A <- matrix(0, 6, 1)

test.res <- testcoef.env(m, L, R, A)
test.res
```

`testcoef.env.apweights`

Hypothesis test of the coefficients of the response envelope model with nonconstant variance

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the response envelope model with nonconstant error variance.

Usage

```
testcoef.env.apweights(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.apweights.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model with nonconstant errors. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]
m <- env.apweights(X, Y, 2)
m

L <- diag(3)
R <- matrix(1, 7, 1)
A <- matrix(0, 3, 1)

test.res <- testcoef.env.apweights(m, L, R, A)
test.res
```

`testcoef.env.tcond` *Hypothesis test of the coefficients of the response envelope model with t-distributed errors*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the response envelope model with t-distributed errors.

Usage

```
testcoef.env.tcond(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.tcond.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model with t-distributed errors. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]
m <- env.tcond(X, Y, 2, 6)
m

L <- diag(3)
R <- matrix(1, 7, 1)
A <- matrix(0, 3, 1)

test.res <- testcoef.env.tcond(m, L, R, A)
test.res
```

`testcoef.genv` *Hypothesis test of the coefficients of the groupwise envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the groupwise envelope model.

Usage

```
testcoef.genv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from genv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta[[i]] R = A, versus Ha: L beta[[i]] R != A. The beta is estimated by the groupwise envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta[[i]] = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
u <- u.genv(X, Y, Z)
u

m <- genv(X, Y, Z, 2)
m

L <- diag(3)
R <- diag(2)
A <- matrix(0, 3, 2)

test.res <- testcoef.genv(m, L, R, A)
test.res
```

`testcoef.henv` *Hypothesis test of the coefficients of the heteroscedastic envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the heteroscedastic envelope model.

Usage

```
testcoef.henv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from genv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the heteroscedastic envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: m <- henv(X, Y, 2)
## Not run: m

L <- diag(4)
R <- matrix(c(1, -1, 0), 3, 1)
A <- matrix(0, 4, 1)

## Not run: test.res <- testcoef.henv(m, L, R, A)
## Not run: test.res
```

`testcoef.logit.env` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta = A versus the alternative hypothesis L * beta ~= A, where beta is estimated under the envelope model in logistic regression.

Usage

```
testcoef.logit.env(m, L, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from logit.env.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by 1 matrix.

Note that inputs `L` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta = A, versus Ha: L beta != A. The beta is estimated by the envelope model in predictor space. If L = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta - A) hatSigma^-1 vec(L beta - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta - A). The reference distribution is chi-squared distribution with degrees of freedom d1.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta).

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- logit.env(X, Y, 1)

L <- diag(7)
A <- matrix(0, 7, 1)

test.res <- testcoef.logit.env(m, L, A)
test.res
```

`testcoef.penv` *Hypothesis test of the coefficients of the partial envelope model*

Description

This function tests the null hypothesis L * beta1 * R = A versus the alternative hypothesis L * beta1 * R ~= A, where beta is estimated under the partial envelope model.

Usage

```
testcoef.penv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from penv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p1 by d2 matrix, while d2 is less than or equal to p1.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta1 R = A, versus Ha: L beta1 R != A. The beta is estimated by the partial envelope model. If L = Ir, R = Ip1 and A = 0, then the test is equivalent to the standard F test on if beta1 = 0. The test statistics used is vec(L beta1 R - A) hatSigma^-1 vec(L beta1 R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta1 R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta1 R).

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- penv(X1, X2, Y, 1)
m

L <- diag(4)
R <- as.matrix(1)
A <- matrix(0, 4, 1)

test.res <- testcoef.penv(m, L, R, A)
test.res
```

`testcoef.pois.env` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta = A versus the alternative hypothesis L * beta ~= A, where beta is estimated under the envelope model in poisson regression.

Usage

```
testcoef.pois.env(m, L, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from pois.env.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by 1 matrix.

Note that inputs `L` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta = A, versus Ha: L beta != A. The beta is estimated by the envelope model in predictor space. If L = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta - A) hatSigma^-1 vec(L beta - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta - A). The reference distribution is chi-squared distribution with degrees of freedom d1.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta).

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- pois.env(X, Y, 1)

L <- diag(7)
A <- matrix(0, 7, 1)

test.res <- testcoef.pois.env(m, L, A)
test.res
```

`testcoef.rrenv` *Hypothesis test of the coefficients of the reduced rank envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the reduced rank envelope model.

Usage

```
testcoef.rrenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from rrenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.

`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the reduced rank envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
m <- rrenv(X, Y, 4, 2)
m

L <- diag(4)
R <- matrix(1, 11, 1)
A <- matrix(0, 4, 1)

test.res <- testcoef.rrenv(m, L, R, A)
test.res
```

`testcoef.rrenv.apweights`

Hypothesis test of the coefficients of the reduced rank envelope model with nonconstant error variance

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the reduced rank envelope model with nonconstant error variance.

Usage

```
testcoef.rrenv.apweights(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from rrenv.apweights.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the reduced rank envelope model that accommodates nonconstant error variance. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
## Not run: m <- rrenv.apweights(X, Y, 3, 2)
## Not run: m

L <- diag(4)
```

```
R <- matrix(1, 11, 1)
A <- matrix(0, 4, 1)

## Not run: test.res <- testcoef.rrenv.apweights(m, L, R, A)
## Not run: test.res
```

`testcoef.senv` *Hypothesis test of the coefficients of the scaled response envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the scaled response envelope model.

Usage

```
testcoef.senv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from scale.env.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the scaled envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- senv(X, Y, 2)

L <- diag(4)
R <- as.matrix(c(1, 0, 0))
A <- matrix(0, 4, 1)

test.res <- testcoef.senv(m, L, R, A)
test.res
```

`testcoef.stenv` *Hypothesis test of the coefficients of the simultaneous envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the simultaneous envelope model.

Usage

```
testcoef.stenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from stenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`R`	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the simultaneous envelope model. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(fiberpaper)
X <- fiberpaper[ , 5:7]
Y <- fiberpaper[ , 1:4]
m <- stenv(X, Y, 2, 3)

L <- diag(3)
R <- as.matrix(c(1, 0, 0, 0), nrow = 4)
A <- matrix(0, 3, 1)

test.res <- testcoef.stenv(m, L, R, A)
test.res
```

`testcoef.sxenv` *Hypothesis test of the coefficients of the scaled predictor envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the scaled predictor envelope model.

Usage

```
testcoef.sxenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from scale.xenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`R`	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the scaled envelope model in the predictor space. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

m <- sxenv(X, Y, 2, R)

L <- diag(4)
R <- as.matrix(c(1, 0, 0))
A <- matrix(0, 4, 1)

test.res <- testcoef.sxenv(m, L, R, A)
test.res
```

`testcoef.xenv` *Hypothesis test of the coefficients of the predictor envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the predictor envelope model.

Usage

```
testcoef.xenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`R`	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model in predictor space. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
m <- xenv(X, Y, 2)
m

L <- diag(6)
R <- as.matrix(1)
A <- matrix(0, 6, 1)

test.res <- testcoef.xenv(m, L, R, A)
test.res
```

`u.env` *Select the dimension of env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the response envelope model.

Usage

```
u.env(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u
```

`u.env.apweights` *Select the dimension of env.apweights*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the response envelope model that accommodates nonconstant variance.

Usage

```
u.env.apweights(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(concrete)
X <- concrete[, 1:7]
Y <- concrete[, 8:10]
## Not run: u <- u.env.apweights(X, Y)
## Not run: u
```

`u.env.tcond` *Select the dimension of env.tcond*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the response envelope model with t-distributed errors.

Usage

```
u.env.tcond(X, Y, df, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`df`	Degrees of freedom of the t-distribution. A positive number that is greater than 2.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(concrete)
X <- concrete[1:78, 1:7]  # The first 78 observations are training data
Y <- concrete[1:78, 8:10]
## Not run: u <- u.env.tcond(X, Y, 6)
## Not run: u
```

`u.eppls` *Select the dimension of eppls*

Description

This function outputs dimensions selected by Bayesian information criterion (BIC) for the Envelope-based Partial Partial Least Squares.

Usage

```
u.eppls(X1, X2, Y)
```

Arguments

`X1`	An n by $p1$ matrix of continuous predictors, where $p1$ is the number of continuous predictors with $p1 < n$.
`X2`	An n by $p2$ matrix of categorical predictors, where $p2$ is the number of categorical predictors with $p2 < n$.
`Y`	An n by r matrix of multivariate responses, where r is the number of responses.

Value

`u.bic`	Dimension of the Envelope-based Partial Partial Least Squares.
`bic.seq`	BIC value for dimension from 0 to $p1$.

References

Park, Y., Su, Z. and Chung, D. (2022+) Envelope-based Partial Partial Least Squares with Application to Cytokine-based Biomarker Analysis for COVID-19.

Examples

```
data(amitriptyline)

Y <- amitriptyline[ , 1:2]
X1 <- amitriptyline[ , 4:7]
X2 <- amitriptyline[ , 3]
u <- u.eppls(X1, X2, Y)

u
```

`u.felmdir` *Find the envelope dimensions in the functional envelope linear model*

Description

Fit the dimensions of the response and predictor envelopes in function-on-function linear regression, under direct estimation.

Usage

```
u.felmdir(X, Y, t1, t2, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`t1`	The observed time points for the predictor functions.
`t2`	The observed time points for the response functions.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function finds the dimension of the predictor and response envelope model by Bayesian information criterion (BIC) performed on the direct estimation. To be more specific, consider the envelope model to the function-on-function linear regression,

$$Y = \alpha + BX + \epsilon,$$

where X and Y are random functions in Hilbert spaces H_X and H_Y, α is a fixed member in H_Y, ϵ is a random member of H_Y, and B: $H_X - > H_Y$ is a linear operator. We use cubic splines as the basis for both H_X and H_Y. The coefficients $[X]$ and $[Y]$ with respect to the basis are computed. The predictor and response envelope model is fitted on the linear regression model of $[Y]$ on $[X]$, and the dimensions of the predictor and response envelopes are calculated using BIC. The details are included in Section 7 of Su et al. (2022).

Value

The output is a list that contains the following components:

`ux`	The estimated dimension of the predictor envelope.
`uy`	The estimated dimension of the response envelope.

`beta`	The envelope estimator of the regression coefficients in the regression of $[Y]$ on $[X]$, when the dimensions of envelopes are taken at their estimated values.
`betafull`	The standard estimator, i.e., the OLS estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`alpha`	The envelope estimator of the intercept in the regression of $[Y]$ on $[X]$, when the dimensions of envelopes are taken at their estimated values.
`alphafull`	The standard estimator of the intercept in the regression of $[Y]$ on $[X]$.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

## Not run: m <- u.felmdir(X, Y, t1, t2)
## Not run: m$ux
## Not run: m$uy
```

`u.felmKL` *Find the envelope dimensions in the functional envelope linear model*

Description

Fit the dimensions of the response and predictor envelopes in function-on-function linear regression, under Karhunen-Loeve expansion based estimation.

Usage

```
u.felmKL(X, Y, t1, t2, knots = c(0, 0.25, 0.5, 0.75, 1))
```

Arguments

`X`	Predictor function. An n by T1 matrix, T1 is number of observed time points, which is the length of t1. Here we assume that each function is observed at the same time points.
`Y`	Response function. An n by T2 matrix, T2 is number of observed time points, which is the length of t2. Here we assume that each function is observed at the same time points.
`t1`	The observed time points for the predictor functions.

`t2`	The observed time points for the response functions.
`knots`	The location of knots of the cubic splines used for estimation. Locations should be positive. The default location of the knots are 0, 0.25, 0.5, 0.75, 1.

Details

This function finds the dimension of the predictor and response envelope model by Bayesian information criterion (BIC) performed on the Karhunen-Lo‘eve expansion based estimation. To be more specific, consider the envelope model to the function-on-function linear regression,

$$Y = \alpha + BX + \epsilon,$$

where X and Y are random functions in Hilbert spaces H_X and H_Y, α is a fixed member in H_Y, ϵ is a random member of H_Y, and B: $H_X - > H_Y$ is a linear operator. We use cubic splines as the basis for both H_X and H_Y in the estimation of the eigenfunctions of $Sigma_X$ and $Sigma_\epsilon$. The coefficients $[X]$ and $[Y]$ with respect to the estimated eigenfunctions are computed. The predictor and response envelope model is fitted on the linear regression model of $[Y]$ on $[X]$, and the dimensions of the predictor and response envelopes are calculated using BIC. The details are included in Section 7 of Su et al. (2022).

Value

The output is a list that contains the following components:

`ux`	The estimated dimension of the predictor envelope.
`uy`	The estimated dimension of the response envelope.
`beta`	The envelope estimator of the regression coefficients in the regression of $[Y]$ on $[X]$, when the dimensions of envelopes are taken at their estimated values.
`betafull`	The standard estimator, i.e., the OLS estimator of the regression coefficients in the regression of $[Y]$ on $[X]$.
`alpha`	The envelope estimator of the intercept in the regression of $[Y]$ on $[X]$, when the dimensions of envelopes are taken at their estimated values.
`alphafull`	The standard estimator of the intercept in the regression of $[Y]$ on $[X]$.

References

Su, Z., Li, B. and Cook, R. D. (2022+) Envelope model for function-on-function linear regression.

Examples

```
data(NJdata)
dataX <- matrix(NJdata[,6], nrow = 21)
X <- as.matrix(dataX[, 32:61])
dataY <- matrix(NJdata[,3], nrow = 21)
Y <- as.matrix(dataY[, 32:61])
t1 <- 0:29
t2 <- t1

## Not run: m <- u.felmKL(X, Y, t1, t2)
## Not run: m$ux
## Not run: m$uy
```

`u.genv` *Select the dimension of genv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the groupwise envelope model.

Usage

```
u.genv(X, Y, Z, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Z`	A group indicator vector of length n, where n denotes the number of observations.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the groupwise envelope subspace selected by AIC.
`u.bic`	Dimension of the groupwise envelope subspace selected by BIC.
`u.lrt`	Dimension of the groupwise envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

u <- u.genv(X, Y, Z)
u
```

`u.henv` *Select the dimension of henv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the heteroscedastic envelope model.

Usage

```
u.henv(X, Y, alpha = 0.01)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the heteroscedastic envelope subspace selected by AIC.
`u.bic`	Dimension of the heteroscedastic envelope subspace selected by BIC.
`u.lrt`	Dimension of the heteroscedastic envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: u <- u.henv(X, Y)
## Not run: u
```

`u.logit.env` *Select the dimension of logit.env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model in logistic regression.

Usage

```
u.logit.env(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to p.
`aic.seq`	AIC value for dimension from 0 to p.
`bic.seq`	BIC value for dimension from 0 to p.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

## Not run: u <- u.logit.env(X, Y)
## Not run: u
```

---

u.penv *Select the dimension of penv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the partial envelope model.

Usage

```
u.penv(X1, X2, Y, alpha = 0.01)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the partial envelope subspace selected by AIC.
`u.bic`	Dimension of the partial envelope subspace selected by BIC.
`u.lrt`	Dimension of the partial envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

u <- u.penv(X1, X2, Y)
u
```

`u.pois.env` *Select the dimension of pois.env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model in poisson regression.

Usage

```
u.pois.env(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be counts.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to p.
`aic.seq`	AIC value for dimension from 0 to p.
`bic.seq`	BIC value for dimension from 0 to p.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

## Not run: u <- u.pois.env(X, Y)
## Not run: u
```

`u.pred2.env` *Select the dimension of the constructed partial envelope for prediction based on envelope model*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the constructed partial envelope model.

Usage

```
u.pred2.env(X, Y, Xnew, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the constructed partial envelope subspace selected by AIC.
`u.bic`	Dimension of the constructed partial envelope subspace selected by BIC.
`u.lrt`	Dimension of the constructed partial envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.pred2.env(X, Y, X[10, ])
u
```

`u.rrenv` *Select the dimension of rrenv*

Description

This function outputs the envelope dimension selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the reduced rank envelope model.

Usage

```
u.rrenv(X, Y, d, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`d`	The rank of the coefficient matrix. An integer between 0 and r.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from d to r.
`aic.seq`	AIC value for dimension from d to r.
`bic.seq`	BIC value for dimension from d to r.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
d <- d.select(X, Y, 0.01)
d

## Not run: u <- u.rrenv(X, Y, 2)
## Not run: u
```

`u.rrenv.apweights` *Select the dimension of rrenv.apweights*

Description

This function outputs the envelope dimension selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the reduced rank envelope model with nonconstant error variance.

Usage

```
u.rrenv.apweights(X, Y, d, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`d`	The rank of the coefficient matrix. An integer between 0 and r.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from d to r.
`aic.seq`	AIC value for dimension from d to r.
`bic.seq`	BIC value for dimension from d to r.

Examples

```
data(vehicles)
X <- vehicles[, 1:11]
Y <- vehicles[, 12:15]
X <- scale(X)
Y <- scale(Y)  # The scales of Y are vastly different, so scaling is reasonable here
d <- d.select(X, Y, 0.01)
d

## Not run: u <- u.rrenv.apweights(X, Y, 2)
## Not run: u
```

`u.senv` *Select the dimension of senv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) for the scaled response envelope model.

Usage

```
u.senv(X, Y)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.

Value

`u.aic`	Dimension of the scaled envelope subspace selected by AIC.
`u.bic`	Dimension of the scaled envelope subspace selected by BIC.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u
```

`u.stenv` *Select the dimension of stenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the simultaneous envelope model.

Usage

```
u.stenv(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`d`	Rank of beta selected by the Bura-Cook estimator.
`u.aic`	Dimension of the simultaneous envelope subspace selected by AIC.
`u.bic`	Dimension of the simultaneous envelope subspace selected by BIC.
`u.lrt`	Dimension of the simultaneous envelope subspace selected by the likelihood ratio testing procedure.
`loglik.mat`	Log likelihood for dimension from (1, 1) to (p, r).
`aic.mat`	AIC value for dimension from (1, 1) to (p, r).
`bic.mat`	BIC value for dimension from (1, 1) to (p, r).

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.stenv(X, Y)
u
```

`u.sxenv` *Select the dimension of sxenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) for the scaled predictor envelope model.

Usage

```
u.sxenv(X, Y, R)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`R`	The number of replications of the scales. A vector, the sum of all elements of R must be p.

Value

`u.aic`	Dimension of the scaled envelope subspace in the predictor space selected by AIC.
`u.bic`	Dimension of the scaled envelope subspace in the predictor space selected by BIC.
`aic.seq`	AIC value for dimension from 0 to p.
`bic.seq`	BIC value for dimension from 0 to p.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u
```

`u.xenv` *Select the dimension of xenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the predictor envelope model.

Usage

```
u.xenv(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to p.
`aic.seq`	AIC value for dimension from 0 to p.
`bic.seq`	BIC value for dimension from 0 to p.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

u <- u.xenv(X, Y)
u
```

`vehicles` *Automobile Dataset*

Description

This dataset consists of 15 characteristics for 30 vehicles.

Usage

```
data("vehicles")
```

Format

A data frame with 30 entries on the following 15 variables.

`turbo`	Indicator for vehicles with turbo aspiration.
`two.doors`	Indicator for vehicles with two doors.
`hatchback`	Indicator for hatchback vehicles.
`wheel.base`	Wheelbase of the vehicle.
`length`	Length of the vehicle.
`width`	Width of the vehicle.
`height`	Height of the vehicle.
`curb.weight`	Curb weight of the vehicle.
`eng.size`	Engine size of the vehicle.
`horsepower`	Horsepower of the vehicle.

`peak.rpm` Peak revolutions per minute.

`price` Price in dollars.

`symbol` Insurance risk rating. A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.

`city.mpg` Fuel efficiency in city.

`highway.mpg` Fuel efficiency in highway.

Source

This dataset is a subset of the vehicles dataset in the R package plsdepot.

References

1) 1985 Model Import Car and Truck Specifications, 1985 Ward's Automotive Yearbook. 2) Personal Auto Manuals, Insurance Services Office, 160 Water Street, New York, NY 10038. 3) Insurance Collision Report, Insurance Institute for Highway Safety, Watergate 600, Washington, DC 20037.

Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Automobile

---

`waterstrider` *Water strider data*

Description

Measures of characteristics of the water striders

Usage

```
data("waterstrider")
```

Format

A data frame with 90 observations on the following 9 variables.

`V1` Index of water strider species.

`V2` Logarithm of length of the first antennal segment.

`V3` Logarithm of length of the second antennal segment.

`V4` Logarithm of length of the third antennal segment.

`V5` Logarithm of length of the fourth antennal segment.

`V6` Logarithm of length of fomora of middle leg.

`V7` Logarithm of length of tibiae of middle leg.

`V8` Logarithm of length of fomora of hind leg.

`V9` Logarithm of length of tibiae of hind leg.

Details

This data set contains 8 measures of water striders and an indicator of the species of water striders.

References

Klingenberg, C. R. and Spence, J. R. (1993). Heterochrony and Allometry Lessons from the Water Strider Genus Limnoporus. Evolution 47, 1834-1853

`weighted.env` *Weighted response envelope estimator*

Description

Compute the weighted response envelope estimator with weights computed from BIC.

Usage

```
weighted.env(X, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(Y)), boot.resi = "full")
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`bstrpNum`	Number of bootstrap samples. A positive integer.
`min.u`	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`max.u`	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`boot.resi`	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when `bstrpNum`>0.

Details

This function computes the weighted envelope estimator in a standard multivariate linear regression. And the weighted envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{r} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{r} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap u =(`min.u`, `max.u`), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

`beta`	The weighted envelope estimator of the regression coefficients.
`mu`	The weighted estimated intercept.
`Sigma`	The weighted envelope estimator of the error covariance matrix.
`w`	Weights computed based on BIC.
`loglik`	The log likelihood function computed with weighted envelope estimator.
`n`	The number of observations in the data.
`bootse`	The standard error for elements in beta computed by residual bootstrap. This output is available only when `bstrpNum`>0.
`ratios`	The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted envelope estimator for each element in beta. This output is available only when `bstrpNum`>0.
`bic_select`	A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when `bstrpNum`>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
m <- weighted.env(X, Y)
m$w
m$beta

## Not run: m2 <- weighted.env(X, Y, bstrpNum = 100, min.u = 1, max.u = 6, boot.resi = "full")
## Not run: m2$bic_select
## Not run: m2$bootse
```

`weighted.penv` *Weighted partial envelope estimator*

Description

Compute the weighted partial envelope estimator with weights computed from BIC.

Usage

```
weighted.penv(X1, X2, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(Y)), boot.resi = "full")
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`bstrpNum`	Number of bootstrap samples. A positive integer.
`min.u`	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`max.u`	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`boot.resi`	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when `bstrpNum`>0.

Details

This function computes the weighted partial envelope estimator in a standard multivariate linear regression. And the weighted partial envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{r} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the partial envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{r} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the partial envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted partial envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap u =(`min.u`, `max.u`), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

`beta`	The weighted partial envelope estimator of the regression coefficients.
`mu`	The weighted estimated intercept.
`Sigma`	The weighted partial envelope estimator of the error covariance matrix.
`w`	Weights computed based on BIC.
`loglik`	The log likelihood function computed with weighted partial envelope estimator.
`n`	The number of observations in the data.
`bootse`	The standard error for elements in beta1 computed by residual bootstrap. This output is available only when `bstrpNum`>0.
`ratios`	The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted partial envelope estimator for each element in beta1. This output is available only when `bstrpNum`>0.
`bic_select`	A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when `bstrpNum`>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- weighted.penv(X1, X2, Y)
m$w
m$beta1

m2 <- penv(X1, X2, Y, 2)
m2$beta1


## Not run: m3 <- weighted.penv(X1, X2, Y, bstrpNum = 100, boot.resi = "full")
## Not run: m3$w
## Not run: m3$bic_select
## Not run: m3$bootse
```

```
## Not run: boot.penv(X1, X2, Y, 2, 100)
```

`weighted.pred.env` *Estimation or prediction using weighted partial envelope*

Description

Perform estimation or prediction through weighted partial envelope model.

Usage

```
weighted.pred.env(X, Y, Xnew)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. But it does not provide the estimation or prediction error. This function performs prediction using the same procedure as in pred2.env, except that the partial envelope estimator with dimension u is replaced by a weighted partial envelope estimator. The weights are decided based on BIC values.

Value

`value`	The fitted value or the predicted value evaluated at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

## Not run: pred.res <- weighted.pred.env(X, Y, X[10, ])
```

`weighted.xenv` *Weighted predictor envelope estimator*

Description

Compute the weighted predictor envelope estimator with weights computed from BIC.

Usage

```
weighted.xenv(X, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(X)), boot.resi = "full")
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`bstrpNum`	Number of bootstrap samples. A positive integer.
`min.u`	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`max.u`	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`boot.resi`	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted predictor envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when `bstrpNum`>0.

Details

This function computes the weighted predictor envelope estimator in a standard multivariate linear regression. And the weighted predictor envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{p} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the predictor envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{p} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the predictor envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted predictor envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap u =(`min.u`, `max.u`), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

`beta`	The weighted predictor envelope estimator of the regression coefficients.
`mu`	The weighted estimated intercept.
`SigmaX`	The weighted predictor envelope estimator of the covariance matrix of X.
`SigmaYcX`	The weighted predictor envelope estimator of the error covariance matrix.
`w`	Weights computed based on BIC.
`loglik`	The log likelihood function computed with weighted predictor envelope estimator.
`n`	The number of observations in the data.
`bootse`	The standard error for elements in beta computed by residual bootstrap. This output is available only when `bstrpNum`>0.
`ratios`	The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted predictor envelope estimator for each element in beta. This output is available only when `bstrpNum`>0.
`bic_select`	A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when `bstrpNum`>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
m <- weighted.xenv(X, Y)
m$w
m$beta

## Not run: m2 <- weighted.xenv(X, Y, bstrpNum = 100, min.u = 2, max.u = 4, boot.resi = "full")
## Not run: m2$w
## Not run: m2$bootse
```

`wheatprotein` *Wheat Protein Data*

Description

The protein content of ground wheat samples.

Usage

```
data(wheatprotein)
```

Format

A data frame with 50 observations on the following 8 variables.

`V1`	Measurements of the reflectance of NIR radiation by the wheat samples at 1680nm. The measurements were made on the log(1/reflectance) scale.
`V2`	Measurements of the reflectance of NIR radiation by the wheat samples at 1806nm.
`V3`	Measurements of the reflectance of NIR radiation by the wheat samples at 1932nm.
`V4`	Measurements of the reflectance of NIR radiation by the wheat samples at 2058nm.
`V5`	Measurements of the reflectance of NIR radiation by the wheat samples at 2184nm.
`V6`	Measurements of the reflectance of NIR radiation by the wheat samples at 2310nm.
`V7`	The protein content of each sample (in percent).
`V8`	Binary indicator, 0 for high protein content and 1 for low protein content. The cut off point is if the protein content is smaller than 9.75.

Details

The data are the result of an experiment to calibrate a near infrared reflectance (NIR) instrument for measuring the protein content of ground wheat samples. The protein content of each sample (in percent) was measured by the standard Kjeldahl method. In Fearn (1983), the problem is to find a linear combination of the measurements that predicts protein content. The estimated coefficients can then be entered into the instrument allowing the protein content of future samples to be read directly. The first 24 cases were used for calibration and the last 26 samples were used for prediction.

References

Fearn, T. (1983). A misuse of ridge regression in the calibration of a near infrared reflectance instrument.

`xenv` *Fit the predictor envelope model*

Description

Fit the predictor envelope model in linear regression with dimension u.

Usage

```
xenv(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`u`	Dimension of the envelope. An integer between 0 and p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace in the predictor space. An p by u matrix. The default is the one generated by function envMU.

Details

This function fits the envelope model in the predictor space,

$$Y = \mu + \eta'\Omega^{-1}\Gamma'X + \varepsilon, \Sigma_X = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is p, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The estimated eta. According to the envelope parameterization, beta = Gamma * Omega^-1 * eta.
`Omega`	The coordinates of SigmaX with respect to Gamma.
`Omega0`	The coordinates of SigmaX with respect to Gamma0.
`mu`	The estimated intercept.
`SigmaYcX`	The estimated conditional covariance matrix of Y given X.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).

`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Helland, I. S. and Su, Z. (2013). Envelopes and Partial Least Squares Re- gression. Journal of the Royal Statistical Society: Series B 75, 851 - 877.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

See Also

`simpls.fit` for partial least squares (PLS).

Examples

```
## Fit the envelope in the predictor space
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
u <- u.xenv(X, Y)
u

m <- xenv(X, Y, 4)
m
m$beta

## Fit the partial least squares
## Not run: m1 <- pls::simpls.fit(X, Y, 4)
## Not run: m1$coefficients
```

Index

∗ **datasets**
  amitriptyline, 6
  Berkeley, 6
  concrete, 23
  fiberpaper, 51
  horseshoecrab, 56
  NJdata, 58
  sales, 87
  vehicles, 129
  waterstrider, 130
  wheatprotein, 137

amitriptyline, 6

Berkeley, 6
boot.env, 8
boot.env.apweights, 9
boot.env.tcond, 10
boot.eppls, 11
boot.genv, 12
boot.henv, 13
boot.logit.env, 14
boot.penv, 15
boot.pois.env, 16
boot.rrenv, 17
boot.rrenv.apweights, 18
boot.senv, 19
boot.stenv, 20
boot.sxenv, 21
boot.xenv, 22

concrete, 23
cv.env, 24
cv.env.apweights, 25
cv.env.tcond, 26
cv.eppls, 27
cv.genv, 28
cv.henv, 29
cv.logit.env, 30
cv.penv, 31
cv.pois.env, 32
cv.rrenv, 33
cv.rrenv.apweights, 34
cv.senv, 35
cv.stenv, 36
cv.sxenv, 38
cv.xenv, 39

d.select, 40

env, 40
env.apweights, 42
env.tcond, 44
eppls, 46

felmdir, 48
felmKL, 49
fiberpaper, 51

genv, 52

henv, 54
horseshoecrab, 56

logit.env, 57

NJdata, 58

penv, 60
pois.env, 62
pred.env, 63
pred.env.apweights, 65
pred.env.tcond, 66
pred.eppls, 67
pred.felmdir, 68
pred.felmKL, 69
pred.genv, 71
pred.henv, 72
pred.logit.env, 73
pred.penv, 74
pred.pois.env, 75

pred.rrenv, 76
pred.rrenv.apweights, 77
pred.senv, 78
pred.stenv, 79
pred.sxenv, 80
pred.xenv, 81
pred2.env, 82

Renvlp-package, 4
rrenv, 83
rrenv.apweights, 85

sales, 87
senv, 88
simpls.fit, *140*
stenv, 89
sxenv, 91

testcoef.env, 93
testcoef.env.apweights, 94
testcoef.env.tcond, 96
testcoef.genv, 97
testcoef.henv, 98
testcoef.logit.env, 99
testcoef.penv, 101
testcoef.pois.env, 102
testcoef.rrenv, 103
testcoef.rrenv.apweights, 104
testcoef.senv, 106
testcoef.stenv, 107
testcoef.sxenv, 108
testcoef.xenv, 109

u.env, 111
u.env.apweights, 112
u.env.tcond, 113
u.eppls, 114
u.felmdir, 115
u.felmKL, 116
u.genv, 118
u.henv, 119
u.logit.env, 120
u.penv, 121
u.pois.env, 122
u.pred2.env, 123
u.rrenv, 124
u.rrenv.apweights, 125
u.senv, 126
u.stenv, 126
u.sxenv, 127
u.xenv, 128

vehicles, 129

waterstrider, 130
weighted.env, 131
weighted.penv, 133
weighted.pred.env, 135
weighted.xenv, 136
wheatprotein, 137

xenv, 138