Package: simrel (via r-universe)

August 20, 2024

Title Simulation of Multivariate Linear Model Data

Version 2.1.0

Description Researchers have been using simulated data from a multivariate linear model to compare and evaluate different methods, ideas and models. Additionally, teachers and educators have been using a simulation tool to demonstrate and teach various statistical and machine learning concepts. This package helps users to simulate linear model data with a wide range of properties by tuning few parameters such as relevant latent components. In addition, a shiny app as an 'RStudio' gadget gives users a simple interface for using the simulation function. See more on: Sæbø, S., Almøy, T., Helland, I.S. (2015) <doi:10.1016/j.chemolab.2015.05.012> and Rimal, R., Almøy, T., Sæbø, S. (2018) <doi:10.1016/j.chemolab.2018.02.009>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.1

VignetteBuilder knitr

BugReports `https://github.com/simulatr/simrel/issues`

URL `https://simulatr.github.io/simrel/`

Imports FrF2, ggplot2, gridExtra, jsonlite, magrittr, methods, miniUI, purrr, reshape2, rlang, rstudioapi, scales, sfsmisc, shiny, testthat, tibble, tidyr

Suggests DoE.base, covr, knitr, markdown, pls

Repository https://simulatr.r-universe.dev

RemoteUrl https://github.com/simulatr/simrel

RemoteRef HEAD

RemoteSha fecd13c7a48d31dfda1b69a215b35941fbcc1113

Contents

`AppSimrel` *Simulation of Multivariate Linear Model Data*

Description

Simulation of Multivariate Linear Model Data

Usage

```
AppSimrel()
```

Value

No return value, runs the shiny interface for simulation bisimrel *Simulation of Multivariate Linear Model data with response*

Description

Simulation of Multivariate Linear Model data with response

Usage

```
bisimrel(
  n = 50,
  p = 100,
  q = c(10, 10, 5),
  rho = c(0.8, 0.4),
  relpos = list(c(1, 2), c(2, 3)),
  gamma = 0.5,
  R2 = c(0.8, 0.8),
  ntest = NULL,
  muY = NULL,
  muX = NULL,
  sim = NULL
)
```

Arguments

`n`	Number of training samples
`p`	Number of x-variables
`q`	Vector of number of relevant predictor variables for first, second and common to both responses
`rho`	A 2-element vector, unconditional and conditional correlation between y_1 and y_2
`relpos`	A list of position of relevant component for predictor variables. The list contains vectors of position index, one vector or each response
`gamma`	A declining (decaying) factor of eigen value of predictors (X). Higher the value of `gamma`, the decrease of eigenvalues will be steeper
`R2`	Vector of coefficient of determination for each response
`ntest`	Number of test observation
`muY`	Vector of average (mean) for each response variable
`muX`	Vector of average (mean) for each predictor variable
`sim`	A simrel object for reusing parameters setting

Value

A simrel object with all the input arguments along with following additional items

`X`	Simulated predictors
`Y`	Simulated responses
`beta`	True regression coefficients
`beta0`	True regression intercept
`relpred`	Position of relevant predictors
`testX`	Test Predictors
`testY`	Test Response
`minerror`	Minimum model error
`Rotation`	Rotation matrix of predictor (R)
`type`	Type of simrel object, in this case *bivariate*
`lambda`	Eigenvalues of predictors
`Sigma`	Variance-Covariance matrix of response and predictors

References

Sæbø, S., Almøy, T., & Helland, I. S. (2015). simrel—A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems, 146, 128-135.

Almøy, T. (1996). A simulation study on comparison of prediction methods when only a few components are relevant. Computational statistics & data analysis, 21(1), 87-107.

Examples

```
sobj <- bisimrel(
   n = 100,
   p = 10,
   q = c(5, 5, 3),
   rho = c(0.8, 0.4),
   relpos = list(c(1, 2, 3), c(2, 3, 4)),
   gamma = 0.7,
   R2 = c(0.8, 0.8)
)
# Regression Coefficients from this simulation
sobj$beta
```

cov_mat *Extract various sigma matrices*

Description

Extract various sigma matrices

Usage

```
cov_mat(obj, which = c("xy", "zy", "zw"), use_population = TRUE)
```

Arguments

obj	A simrel object
which	A character string to specify which covariance matrix to extract, possible values are "xy", "zy" and "zw"
use_population	A boolean whether to use compute population values or to estimate from sample

Value

A matrix of covariances with column equals to the number of response and row equals to the number of predictors

Examples

```
set.seed(1983)
sobj <- multisimrel()
cov_mat(sobj, which = "xy", use_population = TRUE)
cov_mat(sobj, which = "xy", use_population = FALSE)
```

cov_plot_data *Prepare data for Plotting Covariance Matrix*

Description

Prepare data for Plotting Covariance Matrix

Usage

```
cov_plot_data(sobj, type = "relpos", ordering = TRUE, facetting = TRUE)
```

Arguments

`sobj`	A simrel object
`type`	Type of covariance matrix - can take two values `relpos` for relevant position of principal components and `relpred` for relevant position of predictor variables
`ordering`	TRUE for ordering the covariance for block diagonal display
`facetting`	TRUE for facetting the predictor and response space. FALSE will give a single facet plot

Value

A data frame with covariances and related values based on `type` argument that is ready to plot

Examples

```
sobj <- simrel(n = 100, p = 10, q = c(4, 5), relpos = list(c(1, 2, 3), c(4, 6, 7)), m = 3,
            R2 = c(0.8, 0.7), ypos = list(c(1, 3), 2), gamma = 0.7, type = "multivariate")
head(cov_plot_data(sobj))
```

`cov_xy` *Covariance between X and Y*

Description

Covariance between X and Y

Usage

```
cov_xy(obj, use_population = TRUE)
```

Arguments

`obj`	A simrel object
`use_population`	A boolean to specify wheather to use population or sample

Value

A covariance matrix of X and Y

cov_zw *Covariance between Z and W*

Description

Helper Functions

Usage

```
cov_zw(obj)
```

Arguments

obj	A simrel object

Value

A covariance matrix of Z and W

cov_zy *Covariance between Z and Y*

Description

Covariance between Z and Y

Usage

```
cov_zy(obj, use_population = TRUE)
```

Arguments

obj	A simrel object
use_population	A boolean to specify wheather to use population or sample

Value

A covariance matrix of Z and Y

expect_subset *Extra test functions*

Description

Extra test functions

Usage

```
expect_subset(
  object,
  expected,
  info = NULL,
  label = NULL,
  expected.label = NULL
)
```

Arguments

object	object to test
expected	Expected value
info	extra information to be included in the message (useful when writing tests in loops).
label	object label. When ‘NULL‘, computed from deparsed object.
expected.label	Equivalent of ‘label‘ for shortcut form.

Value

Returns the object itself if expected value is found in the object as a subset else return Error

Examples

```
expect_subset(c(1, 2, 3, 4, 5), c(2, 4, 5))
```

ggsimrelplot *Simulation Plot with ggplot: The true beta, relevant component and eigen structure*

Description

Simulation Plot with ggplot: The true beta, relevant component and eigen structure

Usage

```
ggsimrelplot(
  obj,
  ncomp = min(obj$p, obj$n, 20),
  which = 1L:3L,
  layout = NULL,
  print.cov = FALSE,
  use_population = TRUE
)
```

Arguments

`obj`	A simrel object
`ncomp`	Number of components to plot
`which`	A character indicating which plot you want as output, it can take `TrueBeta`, `RelComp` and `EstRelComp`
`layout`	A layout matrix of how to layout multiple plots
`print.cov`	Output estimated covariance structure
`use_population`	Logical, TRUE if population values should be used and FALSE if sample values should be used

Value

A list of plots

Examples

```
sim.obj <- simrel(n = 50, p = 16, q = c(3, 4, 5),
   relpos = list(c(1, 2), c(3, 4), c(5, 7)), m = 5,
   ypos = list(c(1, 4), 2, c(3, 5)), type = "multivariate",
   R2 = c(0.8, 0.7, 0.9), gamma = 0.8)

ggsimrelplot(sim.obj, layout = matrix(c(2, 1, 3, 1), 2))

ggsimrelplot(sim.obj, which = c(1, 2), use_population = TRUE)

ggsimrelplot(sim.obj, which = c(1, 2), use_population = FALSE)

ggsimrelplot(sim.obj, which = c(1, 3), layout = matrix(c(1, 2), 1))
```

---

mbrd *Function to create MBR-design.*

Description

Function to create multi-level binary replacement (MBR) design (Martens et al., 2010). The MBR approach was developed for constructing experimental designs for computer experiments. MBR makes it possible to set up fractional designs for multi-factor problems with potentially many levels for each factor. In this package it is mainly called by the `mbrdsim` function.

Usage

```
mbrd(
  l2levels = c(2, 2),
  fraction = 0,
  gen = NULL,
  fnames1 = NULL,
  fnames2 = NULL
)
```

Arguments

`l2levels`	A vector indicating the number of log2-levels for each factor. E.g. `c(2,3)` means 2 factors, the first with $2^2 = 4$ levels, the second with $2^3 = 8$ levels
`fraction`	Design fraction at bit-level. Full design: fraction=0, half-fraction: fraction=1, and so on...
`gen`	list of generators at bit-factor level. Same as generators in function FrF2.
`fnames1`	Factor names of original multi-level factors (optional).
`fnames2`	Factor names at bit-level (optional).

Details

The MBR design approach was developed for designing fractional designs in multi-level multi-factor experiments, typically computer experiments. The basic idea can be summarized in the following steps: 1) Choose the number of levels L for each multi-level factor as a multiple of 2, that is $L \in \{2, 4, 8, ...\}$. 2) Replace any given multi-level factor by a set of $ln(L)$ two-level "bit factors". The complete bit-factor design can then by expressed as a 2^K design where K is the total number of bit-factors across all original multi-level factors. 3) Choose a fraction level P defining av fractional design $2^{(K-P)}$ (see e.g. Montgomery, 2008) as for regular two-levels factorial designs. 4) Express the reduced design in terms of the original multi-level factors.

Value

`BitDesign`	The design at bit-factor level (inherits from FrF2). Function `design.info()` can be used to get extra design info of the bit-design, and `plot` for plotting of the bit-level design.
`Design`	The design at original factor levels, non-randomized.

References

Martens, H., Måge, I., Tøndel, K., Isaeva, J., Høy, M. and Sæbø¸, S., 2010, Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems, *J, Chemom*, **24**, 748–756.

Montgomery, D., *Design and analysis of experiments*, John Wiley & Sons, 2008.

Examples

```
#Two variables with 8 levels each (2^3=8), a half-fraction design.
res <- mbrd(c(3,3),fraction=1, gen=list(c(1,4)))
#plot(res$Design, pch=20, cex=2, col=2)
#Three variabler with 8 levels each, a 1/16-fraction.
res <- mbrd(c(3,3,3),fraction=4)
#library(rgl)
#plot3d(res$Design,type="s",col=2)
```

---

mbrdsim — *A function to set up a design for a given set of factors with their specific levels using the MBR-design method.*

Description

The multi-level binary replacement (MBR) design approach is used here in order to facilitate the investigation of the effects of the data properties on the performance of estimation/prediction methods. The mbrdsim function takes as input a list containing a set of factors with their levels. The output is an MBR-design with the combinations of the factor levels to be run.

Usage

```
mbrdsim(simlist, fraction, gen = NULL)
```

Arguments

`simlist`	A named list containing the levels of a set of (multi-level) factors.
`fraction`	Design fraction at bit-level. Full design: fraction=0, half-fraction: fraction=1, and so on.
`gen`	Generators for the fractioning at the bit level. Default is `NULL` for which the generators are chosen automatically by the `FrF2` function. See documentation of `FrF2` for details on how to set the generators.

Value

`BitDesign`	The design at bit-factor level. The object is of class design, as output from FrF2. Function design.info() can be used to get extra design info of the bit-design. The bit-factors are named.numbered if the input factor list is named.
`Design`	The design at original factor level, non-randomized. The factors are named if the input factor list is named.

Author(s)

Solve Sæbø

References

Martens, H., Måge, I., Tøndel, K., Isaeva, J., Høy, M. and Sæbø¸, S., 2010, Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems, *J, Chemom*, **24**, 748–756.

Examples

```
# Input: A list of factors with their levels (number of levels must be a multiple of 2).
## Simrel Parameters ----
sim_list <- list(
  p = c(20, 150),
  gamma = seq(0.2, 1.1, length.out = 4),
  relpos = list(list(c(1, 2, 3), c(4, 5, 6)), list(c(1, 5, 6), c(2, 3, 4))),
  R2 = list(c(0.4, 0.8), c(0.8, 0.8)),
  ypos = list(list(1, c(2, 3)), list(c(1, 3), 2))
)
## 1/8 fractional Design ----
dgn <- mbrdsim(sim_list, fraction = 3)
design <- cbind(
  dgn[["Design"]],
  q = lapply(dgn[["Design"]][, "p"], function(x) rep(x/2, 2)),
  type = "multivariate",
  n = 100,
  ntest = 200,
  m = 3,
  eta = 0.6
)
## Simulation ----
sobj <- apply(design, 1, function(x) do.call(simrel, x))
names(sobj) <- paste0("Design", seq.int(sobj))

# Info about the bit-design including bit-level aliasing (and resolution if \code{gen = NULL})
if (requireNamespace("DoE.base", quietly = TRUE)) {
  dgn <- mbrdsim(sim_list, fraction = 3)
  DoE.base::design.info(dgn$BitDesign)
}
```

---

msim *Simulation of Multivariate Linear Model Data*

Description

Simulation of Multivariate Linear Model Data

Usage

```
msim(
  p = 15,
  q = c(5, 4, 3),
  m = 5,
  relpos = list(c(1, 2), c(3, 4, 6), c(5, 7)),
  gamma = 0.6,
  R2 = c(0.8, 0.7, 0.8),
  eta = 0,
  muX = NULL,
  muY = NULL,
  ypos = list(c(1), c(3, 4), c(2, 5))
)
```

Arguments

`p`	Number of variables
`q`	Vector containing the number of relevant predictor variables for each relevant response components
`m`	Number of response variables
`relpos`	A list of position of relevant component for predictor variables. The list contains vectors of position index, one vector or each relevant response components
`gamma`	A declining (decaying) factor of eigen value of predictors (X). Higher the value of `gamma`, the decrease of eigenvalues will be steeper
`R2`	Vector of coefficient of determination (proportion of variation explained by predictor variable) for each relevant response components
`eta`	A declining (decaying) factor of eigenvalues of response (Y). Higher the value of `eta`, more will be the declining of eigenvalues of Y. `eta = 0` refers that all eigenvalues of responses (Y) are 1.
`muX`	Vector of average (mean) for each predictor variable
`muY`	Vector of average (mean) for each response variable
`ypos`	List of position of relevant response components that are combined to generate response variable during orthogonal rotation

Value

A simrel object with all the input arguments along with following additional items

`X`	Simulated predictors
`Y`	Simulated responses
`W`	Simulated predictor components
`Z`	Simulated response components
`beta`	True regression coefficients
`beta0`	True regression intercept

`relpred`	Position of relevant predictors
`testX`	Test Predictors
`testY`	Test Response
`testW`	Test predictor components
`testZ`	Test response components
`minerror`	Minimum model error
`Xrotation`	Rotation matrix of predictor (R)
`Yrotation`	Rotation matrix of response (Q)
`type`	Type of simrel object *univariate* or *multivariate*
`lambda`	Eigenvalues of predictors
`SigmaWZ`	Variance-Covariance matrix of components of response and predictors
`SigmaWX`	Covariance matrix of response components and predictors
`SigmaYZ`	Covariance matrix of response and predictor components
`Sigma`	Variance-Covariance matrix of response and predictors
`RsqW`	Coefficient of determination corresponding to response components
`RsqY`	Coefficient of determination corresponding to response variables

References

Sæbø, S., Almøy, T., & Helland, I. S. (2015). simrel—A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems, 146, 128-135.

Almøy, T. (1996). A simulation study on comparison of prediction methods when only a few components are relevant. Computational statistics & data analysis, 21(1), 87-107.

`multisimrel` *Simulation of Multivariate Linear Model Data*

Description

Simulation of Multivariate Linear Model Data

Usage

```
multisimrel(
  n = 100,
  p = 15,
  q = c(5, 4, 3),
  m = 5,
  relpos = list(c(1, 2), c(3, 4, 6), c(5, 7)),
  gamma = 0.6,
  R2 = c(0.8, 0.7, 0.8),
```

```
    eta = 0,
    ntest = NULL,
    muX = NULL,
    muY = NULL,
    ypos = list(c(1), c(3, 4), c(2, 5))
  )
```

Arguments

`n`	Number of observations
`p`	Number of variables
`q`	Vector containing the number of relevant predictor variables for each relevant response components
`m`	Number of response variables
`relpos`	A list of position of relevant component for predictor variables. The list contains vectors of position index, one vector or each relevant response components
`gamma`	A declining (decaying) factor of eigen value of predictors (X). Higher the value of `gamma`, the decrease of eigenvalues will be steeper
`R2`	Vector of coefficient of determination (proportion of variation explained by predictor variable) for each relevant response components
`eta`	A declining (decaying) factor of eigenvalues of response (Y). Higher the value of `eta`, more will be the declining of eigenvalues of Y. `eta = 0` refers that all eigenvalues of responses (Y) are 1.
`ntest`	Number of test observation
`muX`	Vector of average (mean) for each predictor variable
`muY`	Vector of average (mean) for each response variable
`ypos`	List of position of relevant response components that are combined to generate response variable during orthogonal rotation

Value

A simrel object with all the input arguments along with following additional items

`X`	Simulated predictors
`Y`	Simulated responses
`W`	Simulated predictor components
`Z`	Simulated response components
`beta`	True regression coefficients
`beta0`	True regression intercept
`relpred`	Position of relevant predictors
`testX`	Test Predictors
`testY`	Test Response
`testW`	Test predictor components

testZ	Test response components
minerror	Minimum model error
Xrotation	Rotation matrix of predictor (R)
Yrotation	Rotation matrix of response (Q)
type	Type of simrel object *univariate* or *multivariate*
lambda	Eigenvalues of predictors
SigmaWZ	Variance-Covariance matrix of components of response and predictors
SigmaWX	Covariance matrix of response components and predictors
SigmaYZ	Covariance matrix of response and predictor components
Sigma	Variance-Covariance matrix of response and predictors
RsqW	Coefficient of determination corresponding to response components
RsqY	Coefficient of determination corresponding to response variables

References

Sæbø, S., Almøy, T., & Helland, I. S. (2015). simrel—A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems, 146, 128-135.

Almøy, T. (1996). A simulation study on comparison of prediction methods when only a few components are relevant. Computational statistics & data analysis, 21(1), 87-107.

---

parse_parm *Some helper function for simulation*

Description

These function helps to parse a character string into a list object and also creates parameters for performing multiple simulations

Usage

```
parse_parm(character_string, in_list = FALSE)
```

Arguments

character_string	A character string for parameter where the items in a list is separated by semicolon. For example: 1, 2; 3, 4
in_list	TRUE if the result need to wrap in a list, default is FALSE

Value

A list or a vector

Examples

```
parse_parm("1, 2; 3, 4")
parse_parm("1, 2")
```

---

plot_beta *Plotting Functions*

Description

Plotting Functions

Usage

```
plot_beta(obj, base_theme = theme_grey, lab_list = NULL, theme_list = NULL)
```

Arguments

obj	A simrel object
base_theme	Base ggplot theme to apply
lab_list	List of labs arguments such as x, y, title, subtitle
theme_list	List of theme arguments to apply in the plot

Value

A plot of true regression coefficients for the simulated data

Examples

```
sobj <- multisimrel()
sobj %>%
    plot_beta(
        base_theme = ggplot2::theme_bw,
        lab_list = list(
            title = "Regression Coefficients",
            subtitle = "From Simulation",
            y = "True Regression Coefficients"
        ),
        theme_list = list(
            legend.position = "bottom"
        )
    )
```

plot_cov *Plotting Covariance Matrix*

Description

Plotting Covariance Matrix

Usage

```
plot_cov(sobj, type = "relpos", ordering = TRUE, facetting = TRUE)
```

Arguments

sobj	A simrel object
type	Type of covariance matrix - can take two values `relpos` for relevant position of principal components and `relpred` for relevant position of predictor variables
ordering	TRUE for ordering the covariance for block diagonal display
facetting	TRUE for facetting the predictor and response space. FALSE will give a single facet plot

Value

A covariance plot

References

Sæbø, S., Almøy, T., & Helland, I. S. (2015). simrel—A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems, 146, 128-135.

Almøy, T. (1996). A simulation study on comparison of prediction methods when only a few components are relevant. Computational statistics & data analysis, 21(1), 87-107.

Rimal, R., Almøy, T., & Sæbø, S. (2018). A tool for simulating multi-response linear model data. Chemometrics and Intelligent Laboratory Systems, 176, 1-10.

Examples

```
sobj <- simrel(n = 100, p = 10, q = c(4, 5), relpos = list(c(1, 2, 3), c(4, 6, 7)), m = 3,
            R2 = c(0.8, 0.7), ypos = list(c(1, 3), 2), gamma = 0.7, type = "multivariate")
p1 <- plot_cov(sobj, type = "relpos", facetting = FALSE)
p2 <- plot_cov(sobj, type = "rotation", facetting = FALSE)
p3 <- plot_cov(sobj, type = "relpred", facetting = FALSE)
gridExtra::grid.arrange(p1, p2, p3, ncol = 3)
```

plot_covariance *Plot Covariance between predictor (components) and response (components)*

Description

Plot Covariance between predictor (components) and response (components)

Usage

```
plot_covariance(
  sigma_df,
  lambda_df = NULL,
  base_theme = theme_grey,
  lab_list = NULL,
  theme_list = NULL
)
```

Arguments

sigma_df	A data.frame generated by tidy_sigma
lambda_df	A data.frame generated by tidy_lambda
base_theme	Base ggplot theme to apply
lab_list	List of labs arguments such as x, y, title, subtitle
theme_list	List of theme arguments to apply in the plot

Value

A plot of true regression coefficients for the simulated data

Examples

```
sobj <- bisimrel(p = 12)
sigma_df <- sobj %>%
    cov_mat(which = "zy") %>%
    tidy_sigma() %>%
    abs_sigma()
lambda_df <- sobj %>%
    tidy_lambda()
plot_covariance(
    sigma_df,
    lambda_df,
    base_theme = ggplot2::theme_bw,
    lab_list = list(
        title = "Covariance between Response and Predictor Components",
        subtitle = "The bar represents the eigenvalues predictor covariance",
        y = "Absolute covariance",
```

```
        x = "Predictor Component",
        color = "Response Component"
    ),
    theme_list = list(
        legend.position = "bottom"
    )
)
```

---

plot_simrel *A wrapper function for a simrel object*

Description

A wrapper function for a simrel object

Usage

```
plot_simrel(
  obj,
  ncomp = min(obj$p, obj$n, 20),
  which = c(1L:4L),
  layout = NULL,
  print.cov = FALSE,
  use_population = TRUE,
  palette = "Set1",
  base_theme = ggplot2::theme_grey,
  lab_list = NULL,
  theme_list = NULL
)
```

Arguments

obj	A simrel object
ncomp	Number of components to show in x-axis
which	An integer specifying which simrel plot to obtain
layout	A layout matrix for arranging the simrel plots
print.cov	A boolean where to print covariance matrices
use_population	A boolean specifying weather to get plot for population or sample
palette	Name of color paletter compaticable with RColorBrewer
base_theme	Base ggplot theme to apply
lab_list	List of labs arguments such as x, y, title, subtitle. A nested list if the argument which has length greater than 1.
theme_list	List of theme arguments to apply in the plot. A nested list if the argument which has length greater than 1.

Value

Simrel Plot(s)

Examples

```
sobj <- bisimrel(p = 12)
plot_simrel(sobj, layout = matrix(1:4, 2, 2))
```

`prepare_design` *Prepare design for experiment from a list of simulation parameter*

Description

Prepare design for experiment from a list of simulation parameter

Usage

```
prepare_design(option_list, tabular = TRUE)
```

Arguments

`option_list`	A list of options that is to be parsed
`tabular`	logical if output is needed in tabular form or list format

Value

A list of parsed parameters for simulatr

Examples

```
opts <- list(
  n = rep(100, 2),
  p = c(20, 40),
  q = c("5, 5, 4",
        "10, 5, 5"),
  m = c(5, 5),
  relpos = c("1; 2, 4; 3",
             "1, 2; 3, 4; 5"),
  gamma = c(0.2, 0.4),
  R2 = c("0.8, 0.9, 0.7",
         "0.6, 0.8, 0.7"),
  ypos = c("1, 4; 2, 5; 3",
           "1; 2, 4; 3, 5"),
  ntest = rep(1000, 2)
)
design <- prepare_design(opts)
design
```

simrel *Simulation of Multivariate Linear Model Data*

Description

Simulation of Multivariate Linear Model Data

Usage

```
simrel(n, p, q, relpos, gamma, R2, type = "univariate", ...)
```

Arguments

`n`	Number of observations.
`p`	Number of variables.
`q`	Number of predictors related to each relevant components An integer for univariate, a vector of 3 integers for bivariate and 3 or more for multivariate simulation (for details see Notes).
`relpos`	A list (vector in case of univariate simulation) of position of relevant component for predictor variables corresponding to each response.
`gamma`	A declining (decaying) factor of eigenvalues of predictors (X). Higher the value of `gamma`, the decrease of eigenvalues will be steeper.
`R2`	Vector of coefficient of determination (proportion of variation explained by predictor variable) for each relevant response components.
`type`	Type of simulation - `univariate`, `bivariate` and `multivariate`
`...`	Since this is a wrapper function to simulate univariate, bivariate or multivariate, it calls their respective function. This parameter should contain all the necessary arguements for respective simulations. See `unisimrel`, `bisimrel` and `multisimrel`

Value

A simrel object with all the input arguments along with following additional items. For more detail on the return values see the individual simulation functions `unisimrel`, `bisimrel` and `multisimrel`.

Common returns from univariate, bivariate and multivariate simulation:

`call`	the matched call
`X`	simulated predictors
`Y`	simulated responses
`beta`	true regression coefficients
`beta0`	true regression intercept
`relpred`	position of relevant predictors
`n`	number of observations

`p`	number of predictors (as supplied in the arguments)
`p`	number of responses (as supplied in the arguments)
`q`	number of relevant predictors (as supplied in the arguments)
`gamma`	declining factor of eigenvalues of predictors (as supplied in the arguments)
`lambda`	eigenvalues corresponding to the predictors
`R2`	theoretical R-squared value (as supplied in the arguments)
`relpos`	position of relevant components (as supplied in the arguments)
`minerror`	minimum model error
`Sigma`	variance-Covariance matrix of response and predictors
`testX`	simulated test predictor (in univarite simulation `TESTX`)
`testY`	simulated test response (in univarite simulation `TESTY`)
`Rotation`	Random rotation matrix used to rotate latent components. Is equivalent to the transpose of eigenvector-matrix. In multivariate simulation, `Xrotation` (R) and `Yrotation` (Q) refers to this matrix corresponding to the predictor and response.
`type`	type of simrel object `univariate`, `bivariate` or *multivariate*

Returns from multivariate simulation:

`eta`	a declining factor of eigenvalues of response (Y) (as supplied in the arguments)
`ntest`	number of simulated test observations
`W`	simulated response components
`Z`	simulated predictor components
`testW`	test predictor components
`testZ`	test response components
`SigmaWZ`	Variance-Covariance matrix of components of response and predictors
`SigmaWX`	Covariance matrix of response components and predictors
`SigmaYZ`	Covariance matrix of response and predictor components
`RsqW`	Coefficient of determination corresponding to response components
`RsqY`	Coefficient of determination corresponding to response variables

Note

The parameter `q` represetns the number of predictor variables that forms a basis for each of the relevant componetns. For example, for `q = 8` and relevant components 1, 2, and 3 specified by parameter `relpos` then the randomly selected 8 predictor variables forms basis for these three relevant componets and thus in the model these 8 predictors will be revant for the response (outcome).

References

Sæbø, S., Almøy, T., & Helland, I. S. (2015). simrel—A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems, 146, 128-135.

Almøy, T. (1996). A simulation study on comparison of prediction methods when only a few components are relevant. Computational statistics & data analysis, 21(1), 87-107.

simrelplot	*Simulation Plot: The true beta, relevant component and eigen structure*

Description

Simulation Plot: The true beta, relevant component and eigen structure

Usage

```
simrelplot(
  obj,
  ncomp = min(obj$p, obj$n, 20),
  ask = TRUE,
  print.cov = FALSE,
  which = 1L:3L
)
```

Arguments

`obj`	A simrel object
`ncomp`	Number of components to plot
`ask`	logical, TRUE: functions ask for comfirmation FALSE: function layout plot on predefined format
`print.cov`	Output estimated covariance structure
`which`	A character indicating which plot you want as output, it can take `TrueBeta`, `RelComp` and `EstRelComp`

Value

A list of plots

tidy_beta	*Tidy Functions to make plotting easy*

Description

Tidy Functions to make plotting easy

Absolute value of sigma scaled by the overall maximum absolute value

Usage

```
tidy_beta(obj)

abs_sigma(sigma_df)
```

Arguments

`obj`	A Simrel Object
`sigma_df`	A tidy covariance data frame generated by tidy_sigma function

Value

A tibble with three columns: Predictor, Response and BetaCoef

Another data.frame (tibble) of same dimension with absolute covarinace scaled by overall maximum absolute values

Examples

```
sobj <- multisimrel()
beta_df <- tidy_beta(sobj)
beta_df
sobj <- multisimrel()
sobj %>%
    cov_mat("zy") %>%
    tidy_sigma() %>%
    abs_sigma()
```

`tidy_lambda` *Extract Eigenvalues of predictors*

Description

Extract Eigenvalues of predictors

Usage

```
tidy_lambda(obj, use_population = TRUE)
```

Arguments

`obj`	A simrel Object
`use_population`	A boolean to specify where to use population value or calculate from sample

Value

A dataframe of eigenvalues for each predictors

Examples

```
sobj <- multisimrel()
sobj %>%
    tidy_lambda()
```

`tidy_sigma` *Tidy covariance matrix*

Description

Tidy covariance matrix

Usage

```
tidy_sigma(covs)
```

Arguments

`covs`	A sigma matrix obtained from cov_mat function

Value

A tibble with three columns: Predictor, Response and Covariance

Examples

```
sobj <- multisimrel()
```

`unisimrel` *Function for data simulation*

Description

Functions for data simulation from a random regression model with one response variable where the data properties can be controlled by a few input parameters. The data simulation is based on the concept of relevant latent components and relevant predictors, and was developed for the purpose of testing methods for variable selection for prediction.

Usage

```
unisimrel(
  n,
  p,
  q,
  relpos,
  gamma,
  R2,
  ntest = NULL,
  muY = NULL,
  muX = NULL,
  lambda.min = .Machine$double.eps,
  sim = NULL
)
```

Arguments

`n`	The number of (training) samples to generate.
`p`	The total number of predictor variables to generate.
`q`	The number of relevant predictor variables (as a subset of p).
`relpos`	A vector indicating the position (between 1 and p) of the m relevant components, e.g. $c(1, 2)$ means that the first two latent components should be relevant. The length of relpos must be equal to m.
`gamma`	A number defining the speed of decline in eigenvalues (variances) of the latent components. The eigenvalues are assumed to decline according to an exponential model. The first eigenvalue is set equal to 1.
`R2`	The theoretical R-squared according to the true linear model. A number between 0 and 1.
`ntest`	The number of test samples to be generated (optional).
`muY`	The true mean of the response variable (optional). Default is muY=NULL.
`muX`	The p-vector of true means of the predictor variables (optional). Default is muX=NULL.
`lambda.min`	Lower bound of the eigenvalues. Defaults to .Machine$double.eps.
`sim`	A fitted simrel object. If this is given, the same regression coefficients will be used to simulate a new data set of requested size. Default is NULL, for which new regression coefficients are sampled.

Details

The data are simulated according to a multivariate normal model for the vector $(y, z_1, z_2, z_3, ..., z_p)^t$ where y is the response variable and $z = (z_1, ..., z_p)^t$ is the vector of latent (principal) components. The ordered principal components are uncorrelated variables with declining variances (eigenvalues) defined for component j as $e^{-\gamma * j}/e^{-\gamma}$. Hence, the variance (eigenvalue) of the first principal component is equal to 1, and a large value of γ gives a rapid decline in the variances. The variance of the response variable is by default fixed equal to 1.

Some of the principal components (ordered by their decreasing variances) are assumed to be relevant for the prediction of the response. The indices of the positions of the relevant components are set by the `relpos` argument. The joint degree of relevance for the relevant components is determined by the population R-squared defined by `R2`.

In order to obtain predictor variables $x = (x_1, x_2, ..., x_p)^t$ for y, a random rotation of the principal components is performed. Hence, $x = R^t * z$ for some random rotation matrix R. For values of q satisfying $m <= q < p$ only a subspace of dimension q containing the m relevant component(s) is rotated. This facilitates the possibility to generate q relevant predictor variables (x's). The indices of the relevant predictors is randomly selected with the only restriction that the index set contains the indices in `relpos`. The final index set of the relevant predictors is saved in the output argument `relpred`. If `q=p` all p predictor variables are relevant for the prediction of y.

For further details on the simulation approach, please see S<e6>b<f8>, Alm<f8>y and Helland (2015).

Value

A simrel object with list of following items,

`call`	The call to simrel.
`X`	The (n x p) simulated predictor matrix.
`Y`	The n-vector of simulated response values.
`beta`	The vector of true regression coefficients.
`beta0`	The true intercept. This is zero if muY=NULL and muX=NULL
`muY`	The true mean of the response variable.
`muX`	The p-vector of true means of the predictor variables.
`relpred`	The index of the true relevant predictors, that is the x-variables with non-zero true regression coefficients.
`TESTX`	The (ntest x p) matrix of optional test samples.
`TESTY`	The ntest-vector of responses of the optional test samples.
`n`	The number of simulated samples.
`p`	The number of predictor variables.
`m`	The number of relevant components.
`q`	The number of relevant predictors.
`gamma`	The decline parameter in the exponential model for the true eigenvalues.
`lambda`	The true eigenvalues of the covariance matrix of the p predictor variables.
`R2`	The true R-squared value of the linear model.
`relpos`	The positions of the relevant components.
`minerror`	The minimum achievable prediction error. Also the variance of the noise term in the linear model.
`r`	The sampled correlations between the principal components and the response.
`Sigma`	The true covariance matrix of $(y, z_1, z_2, ..., z_p)^t$.
`Rotation`	The random rotation matrix which is used to achieve the predictor variables as rotations of the latent components. Equals the transposed of the eigenvector-matrix of the covariance matrix of $(x_1, ..., x_p)^t$.
`type`	The type of response generated, either "univariate" as returned from `simrel`, or "bivariate" as returned from `simrel2`.

Author(s)

Solve S<e6>b<f8> and Kristian H. Liland

References

Helland, I. S. and Alm<f8>y, T., 1994, Comparison of prediction methods when only a few components are relevant, *J. Amer. Statist. Ass.*, **89**(426), 583 – 591.

S<e6>b<f8>, S., Alm<f8>y, T. and Helland, I. S., 2015, simrel - A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors, *Chemometr. Intell. Lab.*(in press),doi:10.1016/j.chemolab.2015.05.012.

Examples

```
#Linear model data, large n, small p
mydata <- unisimrel(n = 250, p = 20, q = 5, relpos = c(2, 4), gamma = 0.25, R2 = 0.75)

#Estimating model parameters using ordinary least squares
lmfit <- lm(mydata$Y ~ mydata$X)
summary(lmfit)

#Comparing true with estimated regression coefficients
plot(mydata$beta, lmfit$coef[-1], xlab = "True regression coefficients",
  ylab = "Estimated regression coefficients")
abline(0,1)

#Linear model data, small n, large p
mydata <- unisimrel(n = 50, p = 200, q = 25, relpos = c(2, 4), gamma = 0.25, R2 = 0.8 )

#Simulating more samples with identical distribution as previous simulation
mydata2 <- unisimrel(n = 2500, sim = mydata)

#Estimating model parameters using partial least squares regression with
#cross-validation to determine the number of relevant components.
if (requireNamespace("pls", quietly = TRUE)) {
  require(pls)
  plsfit <- plsr(mydata$Y ~ mydata$X, 15, validation = "CV")

  #Validation plot and finding the number of relevant components.
  plot(0:15, c(plsfit$validation$PRESS0, plsfit$validation$PRESS),
    type = "b", xlab = "Components", ylab = "PRESS")
  mincomp <- which(plsfit$validation$PRESS == min(plsfit$validation$PRESS))

  #Comparing true with estimated regression coefficients
  plot(mydata$beta, plsfit$coef[, 1, mincomp], xlab = "True regression coefficients",
    ylab = "Estimated regression coefficients")
  abline(0, 1)
}
```

Index