

Package ‘simrel’

May 6, 2020

Title Simulation of Multivariate Linear Model Data

Version 2.0.2

Description Simulate multivariate linear model data is useful in research and education weather for comparison or create data with specific properties. This package lets user to simulate linear model data of wide range of properties with few tuning parameters. The package also consist of function to create plots for the simulation objects and A shiny app as RStudio gadget. It can be a handy tool for model comparison, testing and many other purposes.

Depends R (>= 3.5.0)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0.9000

VignetteBuilder knitr

BugReports <https://github.com/simulatr/simrel/issues>

URL <https://simulatr.github.io/simrel/>

Repository CRAN

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

Suggests covr, knitr, pls

NeedsCompilation no

Author Raju Rimal [aut, cre],
Solve Sæbø [aut, ths] (Original Creator of the package)

Maintainer Raju Rimal <raju.rimal@nmbu.no>

Date/Publication 2020-05-06 12:40:06 UTC

R topics documented:

AppSimrel	2
bisimrel	3
cov_mat	5
cov_plot_data	5
cov_xy	6
cov_zw	7
cov_zy	7
expect_subset	8
ggsimrelplot	8
mbrd	9
mbrdsim	11
msim	12
multisimrel	14
parse_parm	16
plot_beta	17
plot_cov	18
plot_covariance	19
plot_simrel	20
prepare_design	21
simrel	22
simrelplot	23
tidy_beta	24
tidy_lambda	25
tidy_sigma	25
unisimrel	26
Index	30

 AppSimrel

Simulation of Multivariate Linear Model Data

Description

Simulation of Multivariate Linear Model Data

Usage

AppSimrel()

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 for 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 <i>bivariate</i>
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	<i>Extract various sigma matrices</i>
---------	---------------------------------------

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	<i>Prepare data for Plotting Covariance Matrix</i>
---------------	--

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	<i>Covariance between Z and W</i>
--------	-----------------------------------

Description

Helper Functions

Usage

```
cov_zw(obj)
```

Arguments

obj A simrel object

Value

A covariance matrix of Z and W

cov_zy	<i>Covariance between Z and Y</i>
--------	-----------------------------------

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	<i>Extra test functions</i>
---------------	-----------------------------

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.

Examples

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

ggsimrelplot	<i>Simulation Plot with ggplot: The true beta, relevant component and eigen structure</i>
--------------	---

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 = 100, 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))
```

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

<code>l2levels</code>	A vector indicating the number of log2-levels for each factor. E.g. <code>c(2, 3)</code> means 2 factors, the first with $2^2 = 4$ levels, the second with $2^3 = 8$ levels
<code>fraction</code>	Design fraction at bit-level. Full design: <code>fraction=0</code> , half-fraction: <code>fraction=1</code> , and so on...
<code>gen</code>	list of generators at bit-factor level. Same as generators in function <code>FrF2</code> .
<code>fnames1</code>	Factor names of original multi-level factors (optional).
<code>fnames2</code>	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 $L \in \{2, 4, 8, \dots\}$. 2) Replace any given multi-level factor by a set of $\ln(L)$ two-level "bit factors". The complete bit-factor design can then be 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 a 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

<code>BitDesign</code>	The design at bit-factor level (inherits from <code>FrF2</code>). Function <code>design.info()</code> can be used to get extra design info of the bit-design, and <code>plot</code> for plotting of the bit-level design.
<code>Design</code>	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$Design, pch=20, cex=2, col=2)
#Three variables with 8 levels each, a 1/16-fraction.
res <- mbrd(c(3,3,3),fraction=4)
#library(rgl)
#plot3d(res$Design, type="s", col=2)
```

mbrdsim	<i>A function to set up a design for a given set of factors with their specific levels using the MBR-design method.</i>
---------	---

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 <code>FrF2</code> function. See documentation of <code>FrF2</code> for details on how to set the generators.

Value

BitDesign	The design at bit-factor level. The object is of class <code>design</code> , as output from <code>FrF2</code> . Function <code>design.info()</code> can be used to get extra design info of the bit-design. The bit-factors are named. <code>numbered</code> 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 ----
design <- mbrdsim(sim_list, fraction = 3)[["Design"]]
design <- cbind(
  design,
  q = lapply(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})
## Not run:
# library(DoE.base)
# design.info(des$BitDesign)

## End(Not run)
```

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 for 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 <i>univariate</i> or <i>multivariate</i>
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 for 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 <i>univariate</i> or <i>multivariate</i>
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
RsqrW	Coefficient of determination corresponding to response components
RsqrY	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 semi-colon. 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	<i>Plotting Functions</i>
-----------	---------------------------

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	<i>Plot Covariance between predictor (components) and response (components)</i>
-----------------	---

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	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 arguments for respective simulations. See unisimrel , bisimrel and multisimrel

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 <i>univariate</i> or <i>multivariate</i>
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.

simrelplot	<i>Simulation Plot: The true beta, relevant component and eigen structure</i>
------------	---

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 confirmation 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	<i>Tidy Functions to make plotting easy</i>
-----------	---

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	<i>Extract Eigenvalues of predictors</i>
-------------	--

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	<i>Tidy covariance matrix</i>
------------	-------------------------------

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 $(y, z_1, z_2, z_3, \dots, z_p)^t$ where y is the response variable and $z = (z_1, \dots, z_p)^t$ 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 $x = (x_1, x_2, \dots, x_p)^t$ 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 $m \leq q < p$ 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&B](#), [Alm&Y](#) and [Helland \(2015\)](#).

Value

A simrel object with list of following items,

<code>call</code>	The call to simrel.
<code>X</code>	The (n x p) simulated predictor matrix.
<code>Y</code>	The n-vector of simulated response values.
<code>beta</code>	The vector of true regression coefficients.
<code>beta0</code>	The true intercept. This is zero if <code>muY=NULL</code> and <code>muX=NULL</code>
<code>muY</code>	The true mean of the response variable.
<code>muX</code>	The p-vector of true means of the predictor variables.
<code>relpred</code>	The index of the true relevant predictors, that is the x-variables with non-zero true regression coefficients.
<code>TESTX</code>	The (ntest x p) matrix of optional test samples.
<code>TESTY</code>	The ntest-vector of responses of the optional test samples.
<code>n</code>	The number of simulated samples.
<code>p</code>	The number of predictor variables.
<code>m</code>	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 $(y, z_1, z_2, \dots, z_p)^t$.
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 $(x_1, \dots, x_p)^t$.
type	The type of response generated, either "univariate" as returned from simrel, or "bivariate" as returned from simrel2.

Author(s)

Solve Sørensen and Kristian H. Liland

References

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

Sørensen, S., Almøy, 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)
```

```
## Not run:
#Estimating model parameters using partial least squares regression with
#cross-validation to determine the number of relevant components.
require(pls)
plsfit <- pls(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)

## End(Not run)
```

Index

- *Topic **Design**
 - mbrdsim, 11
- *Topic **MBRD**
 - mbrdsim, 11
- *Topic **Model**
 - unisimrel, 26
- *Topic **Simulation**
 - unisimrel, 26
- *Topic **datagen**
 - bisimrel, 3
 - msim, 12
 - multisimrel, 14
 - simrel, 22
- *Topic **design**
 - mbrd, 9
- *Topic **intgernal**
 - cov_xy, 6
 - cov_zw, 7
 - cov_zy, 7
- *Topic **multivariate**
 - multisimrel, 14

abs_sigma(tidy_beta), 24
AppSimrel, 2

bisimrel, 3, 22

cov_mat, 5
cov_plot_data, 5
cov_xy, 6
cov_zw, 7
cov_zy, 7

expect_subset, 8

ggsimrelplot, 8

mbrd, 9
mbrdsim, 11
msim, 12
multisimrel, 14, 22

parse_parm, 16
plot_beta, 17
plot_cov, 18
plot_covariance, 19
plot_simrel, 20
prepare_design, 21

simrel, 22
simrelplot, 23

tidy_beta, 24
tidy_lambda, 25
tidy_sigma, 25

unisimrel, 22, 26