Package 'FSM'

October 12, 2022

Title Finite Selection Model

Version 1.0.0

Description Randomized and balanced allocation of units to treatment groups using the Finite Selection Model (FSM). The FSM was originally proposed and developed at the RAND corporation by Carl Morris to enhance the experimental design for the now famous Health Insurance Experiment. See Morris (1979) <doi:10.1016/0304-4076(79)90053-8> for details on the original version of the FSM.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Ambarish Chattopadhyay [aut, cre],
Carl Morris [aut],
Jose Zubizarreta [aut]

Maintainer Ambarish Chattopadhyay <ambarish_chattopadhyay@g.harvard.edu>

Repository CRAN

Date/Publication 2021-03-10 19:10:05 UTC

R topics documented:

---

crd   *Completely Randomized Design (CRD)*

Description

Generates an assignment under completely randomized design (CRD).

Usage

```
crd(data_frame, n_treat, treat_sizes, control = FALSE)
```

Arguments

data_frame	A data frame corresponding to the full sample of units.
n_treat	Number of treatment groups.
treat_sizes	A vector of treatment group sizes. If control = TRUE, the first element of treat_sizes should be the control group size.
control	If TRUE, treatments are labeled as 0,1,...,g-1 (0 representing the control group). If FALSE, they are labeled as 1,2,...,g.

Value

The original data frame augmented with the column of the treatment indicator.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Examples

```
# Consider N = 12, n1 = n2 = n3 = 4.
df_sample = data.frame(index = 1:12, x = c(20,30,40,40,50,60,20,30,40,40,50,60))
# Draw a random assignment from CRD.
fc = crd(data_frame = df_sample, n_treat = 3, treat_sizes = c(4,4,4))
# Get vector of treatment assignments.
Z_crd = fc$Treat
```

---

ess_model	*Model-based Effective Sample Size (ESS)*

Description

Computes the model-based effective sample size (ESS) of a collection of assignments under a given set of potential outcomes.

Usage

```
ess_model(X_cov, assign_matrix, Y_mat, contrast = c(1, -1))
```

Arguments

X_cov	A matrix of covariates or transformations thereof that will be used as explanatory variables in the linear outcome models within each treatment group.
assign_matrix	A matrix containing a collection of treatment assignment vectors, each column containing a particular assignment vector.
Y_mat	A matrix of potential outcomes, where rows represent units and columns represent treatment levels (ordered).
contrast	A vector of the coefficients of the treatment contrast of interest. For example, for estimating the average treatment effect of treatment 1 versus treatment 2, contrast = c(1,-1).

Value

A vector of effective sample sizes for the given collection of assignments.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Examples

```
# Consider the Lalonde dataset.
# Get the full sample size.
N = nrow(Lalonde)
# Get the treatment group sizes.
n1 = floor(N/2)
n2 = N-n1
# Generate an SOM.
som_obs = som(n_treat = 2, treat_sizes = c(n1,n2),include_discard = FALSE,
method = 'SCOMARS', marginal_treat = rep((n2/N),N), control = FALSE)
```

```
# Generate a treatment assignment given som_obs.
f = fsm(data_frame = Lalonde, SOM = som_obs, s_function = 'Dopt', eps = 0.0001,
ties = 'random', intercept = TRUE, standardize = TRUE, units_print = FALSE)
# Get assignment vector under the FSM.
Z_fsm_obs = f$data_frame_allocated$Treat
# Draw a random CRD.
Z_crd_obs = crd(data_frame = Lalonde, n_treat = 2, treat_sizes = c(n1, n2),
control = FALSE)$Treat
Z_big = cbind(Z_crd_obs, Z_fsm_obs)
# Generate the potential outcomes.
Y_1 = 100 - Lalonde$Age + 6 * Lalonde$Education - 20 * Lalonde$Black +
20 * Lalonde$Hispanic + 0.003 * Lalonde$Re75 + rnorm(N,0,4)
Y_1 = round(Y_1,2)
# Set unit level causal effect = tau = 0.
tau = 0
Y_2 = Y_1 + tau
# Get the matrix of potential outcomes.
Y_appended = cbind(Y_1, Y_2)
# Get the matrix of covariates.
X_cov = Lalonde[,-1]
ess = ess_model(X_cov = X_cov, assign_matrix = Z_big, Y_mat = Y_appended, contrast = c(1,-1))
```

---

ess_rand   *Randomization-based Effective Sample Size (ESS)*

Description

Computes the randomization-based effective sample size (ESS) of a collection of assignments under a given set of potential outcomes.

Usage

```
ess_rand(assign_array, Y_mat, contrast = c(1, -1))
```

Arguments

assign_array	A three dimensional array containing a set of independent realizations of a collection the designs. The first coordinate of the array represents the iterations for each design. The second coordinate represents the units. The third coordinate represents the design.
Y_mat	A matrix of potential outcomes, where rows represent units and columns represent treatment levels (ordered).
contrast	A vector of the coefficients of the treatment contrast of interest. For example, for estimating the average treatment effect of treatment 1 versus treatment 2, contrast = c(1,-1).

Value

A vector of effective sample sizes for the given collection of assignments.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Examples

```
# Consider N = 12, n1 = n2 = 6.
df_sample = data.frame(index = 1:12, x = c(20,30,40,40,50,60,20,30,40,40,50,60))
# Generate the potential outcomes.
Y_1 = 100 + (df_sample$x - mean(df_sample$x)) + rnorm(12, 0, 4)
Y_2 = Y_1 + 50
# Create matrix of potential outcomes.
Y_appended = cbind(Y_1, Y_2)
# Generate 100 assignments under CRD and the FSM.
Z_crd_iter = matrix(rep(0, 100 * 12), nrow = 100)
Z_fsm_iter = matrix(rep(0, 100 * 12), nrow = 100)
for(i in 1:100)
{
# Generate an assignment vector under CRD.
fc = crd(data_frame = df_sample, n_treat = 2, treat_sizes = c(6,6), control = FALSE)
Z_crd_iter[i,] = fc$Treat
# Generate an assignment vector under the FSM.
som_iter = som(data_frame = NULL, n_treat = 2,
treat_sizes = c(6, 6),include_discard = FALSE,
method = 'SCOMARS', marginal_treat = rep((6/12), 12), control = FALSE)
f = fsm(data_frame = df_sample, SOM = som_iter, s_function = 'Dopt',eps = 0.0001,
ties = 'random', intercept = TRUE, standardize = TRUE, units_print = FALSE)
Z_fsm_iter[i,] = f$data_frame_allocated$Treat
}
# Create a 3-dim array of assignments.
Z_array = array(0, dim = c(100, 12, 2))
Z_array[,,1] = Z_crd_iter
Z_array[,,2] = Z_fsm_iter
# Calculate the ESS.
ess_rand(assign_array = Z_array, Y_mat = Y_appended, contrast = c(1,-1))
```

---

fsm   *Finite Selection Model (FSM)*

Description

Generates a randomized assignment of a group of units to multiple groups of pre-determined sizes using the Finite Selection Model (FSM).

Usage

```
fsm(
  data_frame,
  SOM,
  s_function = "Dopt",
  Q_initial = NULL,
  eps = 0.001,
  ties = "random",
  intercept = TRUE,
  standardize = TRUE,
  units_print = TRUE,
  index_col = TRUE,
  Pol_mat = NULL,
  w_pol = NULL
)
```

Arguments

data_frame	A data frame containing a column of unit indices (optional) and covariates (or transformations thereof).
SOM	A selection order matrix.
s_function	Specifies a selection function, a string among 'constant', 'Dopt', 'Aopt', 'max pc', 'min pc', 'Dopt pc', 'max average', 'min average', 'Dopt average'. 'constant' selection function puts a constant value on every unselected unit. 'Dopt' use the D-optimality criteria based on the full set of covariates to select units. 'Aopt' uses the A-optimality criteria. 'max pc' (respectively, 'min pc') selects that unit that has the maximum (respectively, minimum) value of the first principal component. 'Dopt pc' uses the D-optimality criteria on the first principal component, 'max average' (respectively, 'min average') selects that unit that has the maximum (respectively, minimum) value of the simple average of the covariates. 'Dopt average' uses the D-optimality criteria on the simple average of the covariates.
Q_initial	A (optional) non-singular matrix (called 'initial matrix') that is added the $(X^T X)$ matrix of the choosing treatment group at any stage, when the $(X^T X)$ matrix of that treatment group at that stage is non-invertible. If FALSE, the $(X^T X)$ matrix for the full set of observations is used as the non-singular matrix. Applicable if s_function = 'Dopt' or 'Aopt'.
eps	Proportionality constant for Q_initial, the default value is 0.001.
ties	Specifies how to deal with ties in the values of the selection function. If ties = 'random', a unit is selected randomly from the set of candidate units. If ties = 'smallest', the unit that appears earlier in the data frame, i.e. the unit with the smallest index gets selected.
intercept	if TRUE, the design matrix of each treatment group includes a column of intercepts.
standardize	if TRUE, the columns of the X matrix other than the column for the intercept (if any), are standardized.

units_print	if TRUE, the function automatically prints the candidate units at each step of selection.
index_col	if TRUE, data_frame contains a column of unit indices.
Pol_mat	Policy matrix. Applicable only when s_function = 'Aopt'.
w_pol	A vector of policy weights. Applicable only when s_function = 'Aopt'.

Value

A list containing the following items.

data_frame_allocated: The original data frame augmented with the column of the treatment indicator.

som_appended: The SOM with augmented columns for the indices and covariate values for units selected.

som_split: som_appended, split by the levels of the treatment.

crit_print: The value of the objective function, at each stage of build up process. At each stage, the unit that maximizes the objective function is selected.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Morris, C. (1979), "A finite selection model for experimental design of the health insurance study", Journal of Econometrics, 11(1), 43–61.

Morris, C., Hill, J. (2000), "The health insurance experiment: design using the finite selection model", Public policy and statistics: case studies from RAND, Springer Science & Business Media, 29–53.

Examples

```
# Load the data.
df_sample = data.frame(index = 1:12, x = c(20,30,40,40,50,60,20,30,40,40,50,60))
# Generate an SOM with N = 12, n1 = n2 = 6.
som_sample = som(n_treat = 2, treat_sizes = c(6,6), method = 'SCOMARS', control = TRUE,
marginal_treat = rep(6/12,12))
# Assign units given the SOM.
f = fsm(data_frame = df_sample, SOM = som_sample, s_function = 'Dopt',
eps = 0.001, ties = 'random', intercept = TRUE, standardize = TRUE, units_print = TRUE,
index_col = TRUE)
```

| fsm_batch | *Batched FSM for sequential experiments* |

Description

Extension of the FSM to cases where units arrive sequentially in batches.

Usage

```
fsm_batch(
  data_frame,
  data_frame_past,
  t_ind,
  SOM,
  s_function = "Dopt",
  Q_initial = NULL,
  eps = 0.001,
  ties = "random",
  intercept = TRUE,
  index_col_past = TRUE,
  standardize = TRUE,
  units_print = TRUE,
  index_col = TRUE,
  Pol_mat = NULL,
  w_pol = NULL
)
```

Arguments

data_frame	Data frame containing a column of unit indices (optional) and covariates (or transformations thereof).
data_frame_past	
	A data frame of units already allocated to treatment groups. Data frame contains a column of unit indices (optional), columns of covariates (or transformations thereof), and a column for treatment indicator.
t_ind	column name containing the treatment indicator in `data_frame_past`.
SOM	Selection Order Matrix.
s_function	Specifies a selection function, a string among `'constant'`, `'Dopt'`, `'Aopt'`, `'max pc'`, `'min pc'`, `'Dopt pc'`, `'max average'`, `'min average'`, `'Dopt average'`. `'constant'` selection function puts a constant value on every unselected unit. `'Dopt'` use the D-optimality criteria based on the full set of covariates to select units. `'Aopt'` uses the A-optimality criteria. `'max pc'` (respectively, `'min pc'`) selects that unit that has the maximum (respectively, minimum) value of the first principal component. `'Dopt pc'` uses the D-optimality criteria on the first principal component, `'max average'` (respectively, `'min average'`) selects that unit that has the maximum (respectively, minimum) value of the simple average

	of the covariates. `'Dopt average'` uses the D-optimality criteria on the simple average of the covariates.
Q_initial	A (optional) non-singular matrix (called 'initial matrix') that is added the $(X^T X)$ matrix of the choosing treatment group at any stage, when the $(X^T X)$ matrix of that treatment group at that stage is non-invertible. If `FALSE`, the $(X^T X)$ matrix for the full set of observations is used as the non-singular matrix. Applicable if `s_function = 'Dopt'` or `'Aopt'`.
eps	Proportionality constant for `Q_initial`, the default value is 0.001.
ties	Specifies how to deal with ties in the values of the selection function. If `ties = 'random'`, a unit is selected randomly from the set of candidate units. If `ties = 'smallest'`, the unit that appears earlier in the data frame, i.e. the unit with the smallest index gets selected.
intercept	if `TRUE`, the design matrix of each treatment group includes a column of intercepts.
index_col_past	`TRUE` if column of unit indices is present in `data_frame_past`.
standardize	if `TRUE`, the columns of the X matrix other than the column for the intercept (if any), are standardized.
units_print	if `TRUE`, the function automatically prints the candidate units at each step of selection.
index_col	if `TRUE`, data_frame contains a column of unit indices.
Pol_mat	Policy matrix. Applicable only when `s_function = 'Aopt'`.
w_pol	A vector of policy weights. Applicable only when `s_function = 'Aopt'`.

Value

A list containing the following items.

`data_frame_allocated`: The original data frame augmented with the column of the treatment indicator.

`som_appended`: The SOM with augmented columns for the indices and covariate values for units selected.

`som_split`: som_appended, split by the levels of the treatment.

`data_frame_allocated_augmented`: data frame combining `data_frame_allocated` and `data_frame_past`.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R'.

Examples

```
# Consider N=18, number of treatments = 2, n1 = n2 = 9, batch sizes = 6,6,6.
# Get data frame for the first batch.
df_sample_1 = data.frame(index = 1:6, age = c(20,30,40,40,50,60))
# Obtain SOM for all the 12 units.
som_gen = som(data_frame = NULL, n_treat = 2, treat_sizes = c(9,9),
include_discard = FALSE, method = 'SCOMARS', marginal_treat = rep((9/18),18), control = FALSE)
# Assign the first batch.
f1 = fsm(data_frame = df_sample_1, SOM = som_gen[1:6,], s_function = 'Dopt',
eps = 0.0001, ties = 'random', intercept = TRUE, standardize = TRUE, units_print = TRUE)
f1_app = f1$data_frame_allocated
# Get data frame for the second batch.
df_sample_2 = data.frame(index = 7:12, age = c(20,30,40,40,50,60))
# Assign the second batch.
f2 = fsm_batch(data_frame = df_sample_2, SOM = som_gen[7:12,], s_function = 'Dopt',
eps = 0.0001, ties = 'random', intercept = TRUE, standardize = TRUE, units_print = TRUE,
data_frame_past = f1_app, t_ind = 'Treat', index_col_past = TRUE)
f2_app = f2$data_frame_allocated_augmented
# Get data frame for the third batch.
df_sample_3 = data.frame(index = 13:18, age = c(20,30,40,40,50,60))
# Assign the third batch.
f3 = fsm_batch(data_frame = df_sample_3, SOM = som_gen[13:18,], s_function = 'Dopt',
eps = 0.0001, ties = 'random', intercept = TRUE, standardize = TRUE, units_print = TRUE,
data_frame_past = f2_app, t_ind = 'Treat', index_col_past = TRUE)
f3_app = f3$data_frame_allocated_augmented
```

Lalonde   *The Lalonde experimental dataset*

Description

The Nationally Supported Work (NSW) experimental data set by Lalonde (1986).

References

LaLonde, R. J. (1986), "Evaluating the econometric evaluations of training programs with experimental data". The American Economic Review, pp. 604–620.

Dehejia, R., and Wahba, S. (1999), "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs", Journal of the American Statistical Association, 94(443), 1053–1062.

https://users.nber.org/~rdehejia/nswdata.html

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

---

love_plot   *Love plot*

Description

Generates a Love plot of Absolute Standardized Mean Differences (ASMD) or Target Absolute Standardized Differences (TASMD) between two groups under one or two designs.

Usage

```
love_plot(
  data_frame,
  index_col = TRUE,
  alloc1,
  alloc2 = NULL,
  imbalance = "TASMD",
  treat_lab = 1,
  vline = "",
  xupper = 1,
  mean_tar = NULL,
  sd_tar = NULL,
  denom = "target",
  legend_text = "FSM",
  legend_position = "topright"
)
```

Arguments

data_frame	Data frame containing a column of unit indices (optional) and covariates (or transformations thereof).
index_col	if TRUE, data_frame contains a column of unit indices.
alloc1	A vector of treatment assignment.
alloc2	A (optional) vector of treatment assignment.
imbalance	Measure of imbalance used. If imbalance = 'TASMD', imbalance is computed using the Target Absolute Standardized Mean Differences (TASMD). If imbalance = 'ASMD', imbalance is computed using the Absolute Standardized Mean Differences (ASMD)
treat_lab	Label of the treatment group in which the TASMD is computed. Applicable only when imbalance = 'TASMD'.
vline	A (optional) x-coordinate at which a vertical line is drawn.
xupper	Upper limit of the x-axis.
mean_tar	A (optional) vector of target profile of the covariates under consideration, e.g., mean of the covariates in the target population. Applicable only when imbalance = 'TASMD'. If mean_tar = NULL, the full-sample average of the covariates is considered as the target profile.

sd_tar	A optional vector of the standard deviation of the covariates in the target population. Applicable only when `imbalance = 'TASMD'`.
denom	Specifies the denominator for the computation of TASMD. If `denom = 'target'`, the standard deviations of the covariates in the target population are used. If `denom = 'group'`, the standard deviations of the covariates in the treatment group given by `treat_lab` are used. Applicable only when `imbalance = 'TASMD'`.
legend_text	Legend of the two designs under consideration.
legend_position	
	= Position of the legend in the plot. The default is `'topright'`.

Value

Love plot of the ASMD/TASMD of the covariates.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Love, T. (2004), "Graphical display of covariate balance", Presentation, See http://chrp.org/love/JSM2004RoundTableHandou 1364.

Examples

```
# Consider the Lalonde dataset.
# Get the full sample size.
N = nrow(Lalonde)
# Get the treatment group sizes.
n1 = floor(N/2)
n2 = N-n1
# Generate an SOM.
som_obs = som(n_treat = 2, treat_sizes = c(n1,n2),include_discard = FALSE,
method = 'SCOMARS', marginal_treat = rep((n2/N),N), control = FALSE)
# Generate a treatment assignment given som_obs.
f = fsm(data_frame = Lalonde, SOM = som_obs, s_function = 'Dopt', eps = 0.0001,
ties = 'random', intercept = TRUE, standardize = TRUE, units_print = FALSE)
# Get assignment vector under the FSM.
Z_fsm_obs = f$data_frame_allocated$Treat
# Draw a random CRD.
Z_crd_obs = crd(data_frame = Lalonde, n_treat = 2, treat_sizes = c(n1, n2),
control = FALSE)$Treat
# Draw Love plot.
love_plot(data_frame = Lalonde, index_col = TRUE, alloc1 = Z_fsm_obs, alloc2 = Z_crd_obs,
imbalance = 'TASMD', treat_lab = 1, mean_tar = NULL, sd_tar = NULL, denom = 'target',
vline = "", legend_text = c("FSM","CRD"), xupper = 0.15, legend_position = 'topright')
```

---

make_sq_inter   *Squares and two-way interactions of variables*

Description

Generates squares and/or two-way interactions (pairwise products) of the columns of a data frame.

Usage

```
make_sq_inter(
  data_frame,
  is_square = TRUE,
  is_inter = TRUE,
  keep_marginal = TRUE
)
```

Arguments

data_frame	Data frame containing the variables whose squares and interactions are to be created.
is_square	If TRUE, square of each column of data_frame is created.
is_inter	If TRUE, product of every pair of columns of data_frame is created.
keep_marginal	If TRUE, the original columns of data_frame are retained in the resulting data frame.

Value

A data frame containing the squares and/or pairwise products of data_frame.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

Examples

```
# Consider a data frame with N = 12 units and 2 covariates.
data_frame_sample = data.frame(male = c(rep(1,6),rep(0,6)),
age = c(20,30,40,40,50,60,20,30,40,40,50,60))
# Get a data frame with all possible squares and first order interactions.
make_sq_inter(data_frame = data_frame_sample, is_square = TRUE,
is_inter = TRUE, keep_marginal = FALSE)
```

---

perm_test   *Fisher's randomization test for sharp null hypothesis.*

Description

Performs Fisher's randomization test for sharp null hypotheses of the form $H_0 : c_1 Y_i(1) + c_2 Y_i(2) - \tau = 0$, for a vector of contrasts (c_1, c_2).

Usage

```
perm_test(
  Y_obs,
  alloc_obs,
  alloc,
  contrast = c(1, -1),
  tau = 0,
  method = "marginal mean",
  alternative = "not equal"
)
```

Arguments

Y_obs	Vector of observed outcome.
alloc_obs	Vector of observed treatment assignment.
alloc	A matrix of treatment assignments over which the randomization distribution of the test statistic is computed. Each row of alloc should correspond to an assignment vector.
contrast	A vector of the coefficients of the treatment contrast of interest. For example, for estimating the average treatment effect of treatment 1 versus treatment 2, contrast = c(1,-1).
tau	The value of the treatment contrast specified by the sharp null hypothesis.
method	The method of computing the test statistic. If method = 'marginal mean', the test statistic is $c_1 \hat{Y}_i(1) + c_2 \hat{Y}_i(2)$, where $\hat{Y}(z)$ is the mean of the observed outcome in the group $Z = z$, for $z = 0, 1$. If method = 'marginal rank', the test statistic is $c_1 \hat{Y}_i(1) + c_2 \hat{Y}_i(2)$, where $\hat{Y}(z)$ is the mean of the rank of the observed outcome in the group $Z = z$, for $z = 0, 1$
alternative	The type of alternative hypothesis used. For right-sided test, alternative = 'greater'. For left-sided test, alternative = 'less'. For both-sided test, alternative = 'not equal'.

Value

A list containing the following items.

test_stat_obs: The observed value of the test statistic.

test_stat_iter: A vector of values of the test statistic across repeated randomizations.

p_value: p-value of the test.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Examples

```
# Consider N = 12, n1 = n2 = 6.
# We test the sharp null of no treatment effect under CRD.
df_sample = data.frame(index = 1:12, x = c(20,30,40,40,50,60,20,30,40,40,50,60))
# True potential outcomes.
Y_1_true = 100 + (df_sample$x - mean(df_sample$x)) + rnorm(12, 0, 4)
Y_2_true = Y_1_true + 50
# Generate the realized assignment under CRD.
fc = crd(data_frame = df_sample, n_treat = 2, treat_sizes = c(6,6), control = FALSE)
Z_crd_obs = fc$Treat
# Get the observed outcomes
Y_obs = Y_1_true
Y_obs[Z_crd_obs == 2] = Y_2_true[Z_crd_obs == 2]
# Generate 1000 assignments under CRD.
Z_crd_iter = matrix(rep(0, 1000 * 12), nrow = 1000)
for(i in 1:1000)
{
fc = crd(data_frame = df_sample, n_treat = 2, treat_sizes = c(6,6), control = FALSE)
Z_crd_iter[i,] = fc$Treat
}
# Test for the sharp null H0: Y_i(1) = Y_i(0) for all i.
# Alternative: not H0 (two-sided test).
perm = perm_test(Y_obs = Y_obs, alloc_obs = Z_crd_obs, alloc = Z_crd_iter,
contrast = c(1,-1), tau = 0, method = "marginal mean", alternative = 'not equal')
# Obtain the p-value.
perm$p_value
```

---

som | *Selection Order Matrix (SOM)*

Description

Generates a Selection Order Matrix (SOM) in a deterministic/random manner.

Usage

```
som(
  data_frame = NULL,
  n_treat,
  treat_sizes,
```

```
    include_discard = FALSE,
    method = "SCOMARS",
    control = FALSE,
    marginal_treat = NULL
)
```

Arguments

data_frame	A (optional) data frame corresponding to the full sample of units. Required if include_discard = TRUE.
n_treat	Number of treatment groups.
treat_sizes	A vector of treatment group sizes. If control = TRUE, the first element of treat_sizes should be the control group size.
include_discard	
	TRUE if a discard group is considered.
method	Specifies the selection strategy used among 'global percentage', 'randomized chunk', 'SCOMARS'. 'SCOMARS' is applicable only if n_treat = 2.
control	If TRUE, treatments are labeled as 0,1,...,g-1 (0 representing the control group). If FALSE, they are labeled as 1,2,...,g.
marginal_treat	A vector of marginal probabilities, the jth element being the probability that treatment group (or treatment group 2 in case control = FALSE) gets to choose at the jth stage given the total number of choices made by treatment group upto the (j-1)th stage. Only applicable when method = 'SCOMARS'.

Value

A data frame containing the selection order of treatments, i.e. the labels of treatment groups at each stage of selection. If method = 'SCOMARS', the data frame contains an additional column of the conditional selection probabilities.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allocation of Units into Treatment Groups Using the Finite Selection Model for R".

Morris, C. (1983), "Sequentially controlled Markovian random sampling (SCOMARS)", Institute of Mathematical Statistics Bulletin,12(5), 237.

Examples

```
# Generate an SOM with N = 12, n1 = n2 = 6.
som_sample = som(data_frame = NULL, n_treat = 2, treat_sizes = c(6,6), include_discard = FALSE,
method = 'SCOMARS', control = FALSE, marginal_treat = rep(6/12,12))
```

---

tasmd_rand *Target Absolute Standardized Mean Differences (TASMD)*

Description

Computes the mean and standard deviation of Target Absolute Standardized Mean Differences (TASMD) of multiple covariates (or transformations thereof) in a treatment group relative to a target population or a target individual for a set of assignments under one or two designs.

Usage

```
tasmd_rand(
  data_frame,
  index_col = FALSE,
  alloc1,
  alloc2,
  treat_lab = 1,
  legend = c("CRD", "FSM"),
  mean_tar = NULL,
  sd_tar = NULL,
  denom = "target",
  roundoff = 3
)
```

Arguments

data_frame	Data frame containing a column of unit indices (optional) and covariates (or transformations thereof).
index_col	if TRUE, data_frame contains a column of unit indices.
alloc1	A matrix or vector of treatment assignments. If alloc1 is a matrix, then each row should correspond to an assignment vector.
alloc2	A (optional) matrix or vector of treatment assignment. If alloc2 is a matrix, then each row should correspond to an assignment vector.
treat_lab	Label of the treatment group in which the TASMD is computed.
legend	Legend of the two designs under consideration.
mean_tar	A (optional) vector of target profile of the covariates under consideration, e.g., mean of the covariates in the target population. Applicable only when imbalance = 'TASMD'. If mean_tar = NULL, the full-sample average of the covariates is considered as the target profile.
sd_tar	A optional vector of the standard deviation of the covariates in the target population. Applicable only when imbalance = 'TASMD'.
denom	Specifies the denominator for the computation of TASMD. If denom = 'target', the standard deviations of the covariates in the target population are used. If denom = 'group', the standard deviations of the covariates in the treatment group given by treat_lab are used. Applicable only when imbalance = 'TASMD'.

roundoff   A number indicating the number of decimal places to be used for rounding off
                   the TASMDs.

Value

A list containing the following items (if `alloc1` and `alloc2` are matrices)

`tasmd_table`: A matrix containing the means (standard deviations in parenthesis) of the TASMDs
for the designs under consideration. If `alloc1` or `alloc2` is a vector, the TASMD of the corre-
sponding assignment is returned.

`tasmd_mean`: A matrix containing the means of the TASMDs for the designs under consideration.

`tasmd_sd`: A matrix containing the standard deviations of the TASMDs for the designs under con-
sideration.

If `alloc1` and `alloc2` are vectors, `tasmd_rand` produces a data frame of the corresponding TAS-
MDs.

Author(s)

Ambarish Chattopadhyay, Carl N. Morris and Jose R. Zubizarreta.

References

Chattopadhyay, A., Morris, C. N., and Zubizarreta, J. R. (2020), "Randomized and Balanced Allo-
cation of Units into Treatment Groups Using the Finite Selection Model for R".

Examples

```
# Consider the Lalonde dataset.
# Get the full sample size.
N = nrow(Lalonde)
# Get the treatment group sizes.
n1 = floor(N/2)
n2 = N-n1
# Generate an SOM.
som_obs = som(n_treat = 2, treat_sizes = c(n1,n2),include_discard = FALSE,
method = 'SCOMARS', marginal_treat = rep((n2/N),N), control = FALSE)
# Generate a treatment assignment given som_obs.
f = fsm(data_frame = Lalonde, SOM = som_obs, s_function = 'Dopt', eps = 0.0001,
ties = 'random', intercept = TRUE, standardize = TRUE, units_print = FALSE)
# Get assignment vector under the FSM.
Z_fsm_obs = f$data_frame_allocated$Treat
# Draw a random CRD.
Z_crd_obs = crd(data_frame = Lalonde, n_treat = 2, treat_sizes = c(n1, n2),
control = FALSE)$Treat
# Calculate the TASMD.
TASMD = tasmd_rand(data_frame = Lalonde, index_col = TRUE, alloc1 = Z_crd_obs,
alloc2 = Z_fsm_obs, treat_lab = 1, mean_tar = NULL, sd_tar = NULL,
denom = 'target', legend = c('CRD','FSM'), roundoff = 3)
```

Index