Package ‘randomizr’

July 22, 2016

Title Easy to Use Tools for Common Forms of Random Assignment

Version 0.4.1

Description Generates random assignments for common experimental designs: simple, complete, blocked, and clustered.

Depends R (>= 2.10.0)

License GPL-3

LazyData true

Suggests knitr, dplyr, blockTools, testthat, rmarkdown

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Alexander Coppock [aut, cre]

Maintainer Alexander Coppock <acoppock@gmail.com>

Repository CRAN

Date/Publication 2016-07-22 01:12:11

R topics documented:

`block_and_cluster_ra` *Blocked and Clustered Random Assignment.*

Description

Random assignment where units are assigned as clusters and clusters are nested within blocks.

Usage

```
block_and_cluster_ra(block_var, clust_var, prob = NULL, prob_each = NULL,
  block_m = NULL, block_m_each = NULL, block_prob_each = NULL,
  num_arms = NULL, condition_names = NULL, balance_load = FALSE)
```

Arguments

`block_var` A vector of length N that indicates which block each unit belongs to.

`clust_var` A vector of length N that indicates which cluster each unit belongs to.

`prob` Use for a two-arm design in which either floor(N_clusters_block*prob) or ceiling(N_clusters_block*prob) clusters are assigned to treatment within each block. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_clusters_block*prob) clusters will be assigned to treatment and with probability prob, ceiling(N_clusters_block*prob) clusters will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of clusters assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

`block_m` Use for a two-arm design in which block_m describes the number of units to assign to treatment within each block. Note that in previous versions of randomizr, block_m behaved like block_m_each.

`block_m_each` Use for a multi-arm design in which the values of block_m_each determine the number of clusters assigned to each condition. block_m_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the number of clusters to be assigned to each treatment arm within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). The columns should be in the order of condition_names, if specified.

`block_prob_each`

Use for a multi-arm design in which the values of block_prob_each determine the probabilties of assignment to each treatment condition. block_prob_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the probabilites of assignment to treatment within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). Use only if the probabilities of assignment should vary by block, otherwise use prob_each. Each row of block_prob_each must sum to 1.

`num_arms` The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names`
A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

`balance_load` logical, defaults to FALSE. This feature is experimental. If set to TRUE, the function will resolve rounding problems by randomly assigning "remainder" units to each possible treatment condition with equal probability, while ensuring that the total number of units assigned to each condition does not vary greatly from assignment to assignment. However, the true probabiltiies of assignment may be different from the nominal probabilities specified in prob_each or block_prob_each. Please use with caution and perform many tests before using in a real research scenario.

Value

A vector of length N that indicates the treatment condition of each unit.

Examples

```
clust_var <- rep(letters, times=1:26)

block_var <- rep(NA, length(clust_var))
block_var[clust_var %in% letters[1:5]] <- "block_1"
block_var[clust_var %in% letters[6:10]] <- "block_2"
block_var[clust_var %in% letters[11:15]] <- "block_3"
block_var[clust_var %in% letters[16:20]] <- "block_4"
block_var[clust_var %in% letters[21:26]] <- "block_5"


table(block_var, clust_var)

Z <- block_and_cluster_ra(block_var = block_var,
                          clust_var = clust_var)

table(Z, block_var)
table(Z, clust_var)

Z <- block_and_cluster_ra(block_var = block_var,
                          clust_var = clust_var,
                          num_arms = 3)
```

```
table(Z, block_var)
table(Z, clust_var)

Z <- block_and_cluster_ra(block_var = block_var,
                          clust_var = clust_var,
                          prob_each = c(.2, .5, .3))

block_m_each <- rbind(c(2, 3),
                      c(1, 4),
                      c(3, 2),
                      c(2, 3),
                      c(5, 1))

Z <- block_and_cluster_ra(block_var = block_var,
                          clust_var = clust_var,
                          block_m_each = block_m_each)

table(Z, block_var)
table(Z, clust_var)
```

block_and_cluster_ra_probabilities

Probabilties of assignment: Blocked and Clustered Random Assignment

Description

Probabilties of assignment: Blocked and Clustered Random Assignment

Usage

```
block_and_cluster_ra_probabilities(block_var, clust_var, prob = NULL,
  prob_each = NULL, block_m = NULL, block_m_each = NULL,
  block_prob_each = NULL, num_arms = NULL, condition_names = NULL,
  balance_load = FALSE)
```

Arguments

block_var	A vector of length N that indicates which block each unit belongs to.
clust_var	A vector of length N that indicates which cluster each unit belongs to.
prob	Use for a two-arm design in which either floor(N_clusters_block*prob) or ceiling(N_clusters_block*prob) clusters are assigned to treatment within each block. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_clusters_block*prob) clusters will be assigned to treatment and with probability prob, ceiling(N_clusters_block*prob) clusters will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of clusters assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

`block_m` Use for a two-arm design in which block_m describes the number of units to assign to treatment within each block. Note that in previous versions of randomizr, block_m behaved like block_m_each.

`block_m_each` Use for a multi-arm design in which the values of block_m_each determine the number of clusters assigned to each condition. block_m_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the number of clusters to be assigned to each treatment arm within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). The columns should be in the order of condition_names, if specified.

`block_prob_each`

Use for a multi-arm design in which the values of block_prob_each determine the probabilties of assignment to each treatment condition. block_prob_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the probabilites of assignment to treatment within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). Use only if the probabilities of assignment should vary by block, otherwise use prob_each. Each row of block_prob_each must sum to 1.

`num_arms` The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names`

A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

`balance_load` logical, defaults to FALSE. This feature is experimental. If set to TRUE, the function will resolve rounding problems by randomly assigning "remainder" units to each possible treatment condition with equal probability, while ensuring that the total number of units assigned to each condition does not vary greatly from assignment to assignment. However, the true probabiltiies of assignment may be different from the nominal probabilities specified in prob_each or block_prob_each. Please use with caution and perform many tests before using in a real research scenario.

Value

A matrix of probabilities of assignment

Examples

```
clust_var <- rep(letters, times=1:26)
block_var <- rep(NA, length(clust_var))
block_var[clust_var %in% letters[1:5]] <- "block_1"
block_var[clust_var %in% letters[6:10]] <- "block_2"
block_var[clust_var %in% letters[11:15]] <- "block_3"
block_var[clust_var %in% letters[16:20]] <- "block_4"
block_var[clust_var %in% letters[21:26]] <- "block_5"


prob_mat <- block_and_cluster_ra_probabilities(clust_var = clust_var,
                                               block_var = block_var)
head(prob_mat)

prob_mat <- block_and_cluster_ra_probabilities(clust_var = clust_var,
                                               block_var = block_var,
                                               num_arms = 3)
head(prob_mat)

prob_mat <- block_and_cluster_ra_probabilities(clust_var = clust_var,
                                               block_var = block_var,
                                               prob_each = c(.2, .5, .3))
head(prob_mat)

block_m_each <- rbind(c(2, 3),
                      c(1, 4),
                      c(3, 2),
                      c(2, 3),
                      c(5, 1))

prob_mat <- block_and_cluster_ra_probabilities(clust_var = clust_var,
                                               block_var = block_var,
                                               block_m_each = block_m_each)
head(prob_mat)
```

block_ra *Block Random Assignment*

Description

block_ra implements a random assignment in which units that are grouped into blocks defined by pre-treatment covariates are assiged using complete random assignment withing block. For example, imagine that 50 of 100 men are assigned to treatment and 75 of 200 women are assigned to treatment.

Usage

```
block_ra(block_var, prob = NULL, prob_each = NULL, block_m = NULL,
  block_m_each = NULL, block_prob_each = NULL, num_arms = NULL,
  condition_names = NULL, balance_load = FALSE)
```

Arguments

`block_var` A vector of length N that indicates which block each unit belongs to. Can be a character, factor, or numeric vector. (required)

`prob` Use for a two-arm design in which either floor(N_block*prob) or ceiling(N_block*prob) units are assigned to treatment within each block. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_block*prob) units will be assigned to treatment and with probability prob, ceiling(N_block*prob) units will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of units assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

`block_m` Use for a two-arm design in which block_m describes the number of units to assign to treatment within each block. Note that in previous versions of randomizr, block_m behaved like block_m_each.

`block_m_each` Use for a multi-arm design in which the values of block_m_each determine the number of units assigned to each condition. block_m_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the number of units to be assigned to each treatment arm within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). The columns should be in the order of condition_names, if specified.

`block_prob_each`
Use for a multi-arm design in which the values of block_prob_each determine the probabilties of assignment to each treatment condition. block_prob_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the probabilites of assignment to treatment within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). Use only if the probabilities of assignment should vary by block, otherwise use prob_each. Each row of block_prob_each must sum to 1.

`num_arms` The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names`
A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm

trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

`balance_load` logical, defaults to FALSE. This feature is experimental. If set to TRUE, the function will resolve rounding problems by randomly assigning "remainder" units to each possible treatment condition with equal probability, while ensuring that the total number of units assigned to each condition does not vary greatly from assignment to assignment. However, the true probabiltiies of assignment may be different from the nominal probabilities specified in prob_each or block_prob_each. Please use with caution and perform many tests before using in a real research scenario.

Value

A vector of length N that indicates the treatment condition of each unit. Is numeric in a two-arm trial and a factor variable (ordered by condition_names) in a multi-arm trial.

Examples

```
# Two-arm Designs

block_var <- rep(c("A", "B","C"), times = c(50, 100, 200))
Z <- block_ra(block_var = block_var)
table(block_var, Z)

Z <- block_ra(block_var = block_var, prob = .3)
table(block_var, Z)
Z <- block_ra(block_var = block_var, block_m = c(20, 30, 40))
table(block_var, Z)

block_m_each <- rbind(c(25, 25),
                 c(50, 50),
                 c(100, 100))

Z <- block_ra(block_var = block_var, block_m_each = block_m_each)
table(block_var, Z)

block_m_each <- rbind(c(10, 40),
                 c(30, 70),
                 c(50, 150))

Z <- block_ra(block_var = block_var, block_m_each = block_m_each,
              condition_names = c("control", "treatment"))
table(block_var, Z)

# Multi-arm Designs
Z <- block_ra(block_var = block_var, num_arms = 3)
table(block_var, Z)

block_m_each <- rbind(c(10, 20, 20),
                 c(30, 50, 20),
                 c(50, 75, 75))
Z <- block_ra(block_var = block_var, block_m_each = block_m_each)
table(block_var, Z)

Z <- block_ra(block_var = block_var, block_m_each = block_m_each,
              condition_names = c("control", "placebo", "treatment"))
table(block_var, Z)

Z <- block_ra(block_var = block_var, prob_each = c(.1, .1, .8))
table(block_var, Z)


# Experimental feature: load balancing
# This procedure constrains the total number of units in each arm.
# This will never exceed 5 treated units total.

block_var <- rep(c("A", "B","C"), times=c(3, 3, 3))
Z <- block_ra(block_var = block_var, balance_load = TRUE)
table(block_var, Z)

# compare to block_ra without load balancing
# Sometimes this procedure assigns 6 total units to treatment
Z <- block_ra(block_var = block_var, balance_load = FALSE)
table(block_var, Z)
```

block_ra_probabilities

Probabilties of assignment: Block Random Assignment

Description

Probabilties of assignment: Block Random Assignment

Usage

```
block_ra_probabilities(block_var, prob = NULL, prob_each = NULL,
  block_m = NULL, block_m_each = NULL, block_prob_each = NULL,
  num_arms = NULL, condition_names = NULL, balance_load = FALSE)
```

Arguments

`block_var` A vector of length N that indicates which block each unit belongs to. Can be a character, factor, or numeric vector. (required)

`prob` Use for a two-arm design in which either floor(N_block*prob) or ceiling(N_block*prob) units are assigned to treatment within each block. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_block*prob) units will be assigned to treatment and with probability prob, ceiling(N_block*prob) units will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of units assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

`block_m` Use for a two-arm design in which block_m describes the number of units to assign to treatment within each block. Note that in previous versions of randomizr, block_m behaved like block_m_each.

`block_m_each` Use for a multi-arm design in which the values of block_m_each determine the number of units assigned to each condition. block_m_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the number of units to be assigned to each treatment arm within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). The columns should be in the order of condition_names, if specified.

`block_prob_each` Use for a multi-arm design in which the values of block_prob_each determine the probabilties of assignment to each treatment condition. block_prob_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the probabilites of assignment to treatment within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). Use only if the probabilities of assignment should vary by block, otherwise use prob_each. Each row of block_prob_each must sum to 1.

`num_arms` The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names` A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

`balance_load` logical, defaults to FALSE. This feature is experimental. If set to TRUE, the function will resolve rounding problems by randomly assigning "remainder" units to each possible treatment condition with equal probability, while ensuring that the total number of units assigned to each condition does not vary greatly from assignment to assignment. However, the true probabiltiies of assignment may be different from the nominal probabilities specified in prob_each or block_prob_each. Please use with caution and perform many tests before using in a real research scenario.

Value

A matrix of probabilities of assignment

Examples

```
block_var <- rep(c("A", "B","C"), times = c(50, 100, 200))
prob_mat <- block_ra_probabilities(block_var = block_var)
head(prob_mat)

block_m_each <- rbind(c(25, 25),
                 c(50, 50),
                 c(100, 100))

prob_mat <- block_ra_probabilities(block_var = block_var, block_m_each = block_m_each)
head(prob_mat)

block_m_each <- rbind(c(10, 40),
                 c(30, 70),
                 c(50, 150))

prob_mat <- block_ra_probabilities(block_var = block_var,
                                   block_m_each = block_m_each,
                                   condition_names = c("control", "treatment"))
head(prob_mat)

prob_mat <- block_ra_probabilities(block_var = block_var, num_arms = 3)
head(prob_mat)

block_m_each <- rbind(c(10, 20, 20),
                 c(30, 50, 20),
                 c(50, 75, 75))
prob_mat <- block_ra_probabilities(block_var = block_var, block_m_each = block_m_each)
head(prob_mat)

prob_mat <- block_ra_probabilities(block_var=block_var, block_m_each=block_m_each,
                       condition_names=c("control", "placebo", "treatment"))
head(prob_mat)

prob_mat <- block_ra_probabilities(block_var=block_var, prob_each=c(.1, .1, .8))
head(prob_mat)
```

cluster_ra *Cluster Random Assignment*

Description

cluster_ra implements a random assignment procedure in which groups of units are assigned together (as a cluster) to treatment conditions. This function conducts complete random assignment at the cluster level, unless simple = TRUE, in which case simple_ra analogues are used.

Usage

```
cluster_ra(clust_var, m = NULL, m_each = NULL, prob = NULL,
  prob_each = NULL, num_arms = NULL, condition_names = NULL,
  simple = FALSE)
```

Arguments

clust_var	A vector of length N that indicates which cluster each unit belongs to.
m	Use for a two-arm design in which m clusters are assigned to treatment and N-m clusters are assigned to control. (optional)
m_each	Use for a multi-arm design in which the values of m_each determine the number of clusters assigned to each condition. m_each must be a numeric vector in which each entry is a nonnegative integer that describes how many clusters should be assigned to the 1st, 2nd, 3rd... treatment condition. m_each must sum to N. (optional)
prob	Use for a two-arm design in which either floor(N_clusters*prob) or ceiling(N_clusters*prob) clusters are assigned to treatment. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_clusters*prob) clusters will be assigned to treatment and with probability prob, ceiling(N_clusters*prob) clusters will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)
prob_each	Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of clusters assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)
num_arms	The total number of treatment arms. If unspecified, will be determined from the length of m_each or condition_names.
condition_names	A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named T1, T2, T3, etc.
simple	logical, defaults to FALSE. If TRUE, simple random assignment of clusters to conditions is used. When simple = TRUE, please do not specify m or m_each.

Value

A vector of length N that indicates the treatment condition of each unit.

Examples

```
# Two Group Designs
clust_var <- rep(letters, times=1:26)

Z <- cluster_ra(clust_var = clust_var)
table(Z, clust_var)
```

```
Z <- cluster_ra(clust_var = clust_var, m = 13)
table(Z, clust_var)

Z <- cluster_ra(clust_var = clust_var, m_each = c(10, 16),
                condition_names = c("control", "treatment"))
table(Z, clust_var)

# Multi-arm Designs
Z <- cluster_ra(clust_var = clust_var, num_arms = 3)
table(Z, clust_var)

Z <- cluster_ra(clust_var = clust_var, m_each = c(7, 7, 12))
table(Z, clust_var)

Z <- cluster_ra(clust_var = clust_var, m_each = c(7, 7, 12),
                condition_names = c("control", "placebo", "treatment"))
table(Z, clust_var)

Z <- cluster_ra(clust_var = clust_var,
                condition_names = c("control", "placebo", "treatment"))
table(Z, clust_var)
```

cluster_ra_probabilities

Probabilties of assignment: Cluster Random Assignment

Description

Probabilties of assignment: Cluster Random Assignment

Usage

```
cluster_ra_probabilities(clust_var, m = NULL, m_each = NULL, prob = NULL,
  prob_each = NULL, num_arms = NULL, condition_names = NULL,
  simple = FALSE)
```

Arguments

`clust_var` A vector of length N that indicates which cluster each unit belongs to.

`m` Use for a two-arm design in which m clusters are assigned to treatment and N-m clusters are assigned to control. (optional)

`m_each` Use for a multi-arm design in which the values of m_each determine the number of clusters assigned to each condition. m_each must be a numeric vector in which each entry is a nonnegative integer that describes how many clusters should be assigned to the 1st, 2nd, 3rd... treatment condition. m_each must sum to N. (optional) `prob` Use for a two-arm design in which either floor(N_clusters*prob) or ceiling(N_clusters*prob) clusters are assigned to treatment. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N_clusters*prob) clusters will be assigned to treatment and with probability prob, ceiling(N_clusters*prob) clusters will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of clusters assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

`num_arms` The total number of treatment arms. If unspecified, will be determined from the length of m_each or condition_names.

`condition_names`
A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named T1, T2, T3, etc.

`simple` logical, defaults to FALSE. If TRUE, simple random assignment of clusters to conditions is used. When simple = TRUE, please do not specify m or m_each.

Value

A matrix of probabilities of assignment

Examples

```
# Two Group Designs
clust_var <- rep(letters, times = 1:26)
prob_mat <- cluster_ra_probabilities(clust_var = clust_var)
head(prob_mat)

prob_mat <- cluster_ra_probabilities(clust_var = clust_var, m = 10)
head(prob_mat)

prob_mat <- cluster_ra_probabilities(clust_var = clust_var,
                                     m_each = c(9, 17),
                                     condition_names = c("control", "treatment"))

# Multi-arm Designs
prob_mat <- cluster_ra_probabilities(clust_var = clust_var, num_arms = 3)
head(prob_mat)

prob_mat <- cluster_ra_probabilities(clust_var = clust_var, m_each = c(7, 7, 12))
head(prob_mat)

prob_mat <- cluster_ra_probabilities(clust_var = clust_var, m_each = c(7, 7, 12),
                         condition_names=c("control", "placebo", "treatment"))
head(prob_mat)
```

```
prob_mat <- cluster_ra_probabilities(clust_var = clust_var,
                       condition_names=c("control", "placebo", "treatment"))
head(prob_mat)

prob_mat <- cluster_ra_probabilities(clust_var = clust_var,
                                  prob_each = c(.1, .2, .7))
head(prob_mat)
```

complete_ra *Complete Random Assignment*

Description

complete_ra implements a random assignment procedure in which fixed numbers of units are assigned to treatment conditions. The canonical example of complete random assignment is a procedure in which exactly m of N units are assigned to treatment and N-m units are assigned to control.

Users can set the exact number of units to assign to each condition with m or m_each. Alternatively, users can specify probabilities of assignment with prob or prob_each and complete_ra will infer the correct number of units to assign to each condition. In a two-arm design, complete_ra will either assign floor(N*prob) or ceiling(N*prob) units to treatment, choosing between these two values to ensure that the overall probability of assignment is exactly prob. In a multi-arm design, complete_ra will first assign floor(N*prob_each) units to their respective conditions, then will assign the remaining units using simple random assignment, choosing these second-stage probabilties so that the overall probabilities of assignment are exactly prob_each.

In most cases, users should specify N and not more than one of m, m_each, prob, prob_each, or num_arms.

If only N is specified, a two-arm trial in which N/2 units are assigned to treatment is assumed. If N is odd, either floor(N/2) units or ceiling(N/2) units will be assigned to treatment.

Usage

```
complete_ra(N, m = NULL, m_each = NULL, prob = NULL, prob_each = NULL,
  num_arms = NULL, condition_names = NULL)
```

Arguments

N The number of units. N must be a positive integer. (required)

m Use for a two-arm design in which m units are assigned to treatment and N-m units are assigned to control. (optional)

m_each
: Use for a multi-arm design in which the values of m_each determine the number of units assigned to each condition. m_each must be a numeric vector in which each entry is a nonnegative integer that describes how many units should be assigned to the 1st, 2nd, 3rd... treatment condition. m_each must sum to N. (optional)

prob
: Use for a two-arm design in which either floor(N*prob) or ceiling(N*prob) units are assigned to treatment. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N*prob) units will be assigned to treatment and with probability prob, ceiling(N*prob) units will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

prob_each
: Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of units assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

num_arms
: The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

condition_names
: A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

Value

A vector of length N that indicates the treatment condition of each unit. Is numeric in a two-arm trial and a factor variable (ordered by condition_names) in a multi-arm trial.

Examples

```
# Two-arm Designs
Z <- complete_ra(N = 100)
table(Z)

Z <- complete_ra(N = 100, m = 50)
table(Z)

Z <- complete_ra(N = 100, prob = .111)
table(Z)

Z <- complete_ra(N = 100, condition_names = c("control", "treatment"))
table(Z)


# Multi-arm Designs
Z <- complete_ra(N = 100, num_arms = 3)
```

```
table(Z)

Z <- complete_ra(N = 100, m_each = c(30, 30, 40))
table(Z)

Z <- complete_ra(N = 100, prob_each = c(.1, .2, .7))
table(Z)

Z <- complete_ra(N = 100, condition_names = c("control", "placebo", "treatment"))
table(Z)

# Special Cases
# Two-arm trial where the condition_names are by default "T1" and "T2"
Z <- complete_ra(N = 100, num_arms = 2)
table(Z)

# If N = m, assign with 100% probability...
complete_ra(N=2, m=2)

# except if N = m = 1, in which case assign with 50% probability
complete_ra(N=1, m=1)
```

complete_ra_probabilities

Probabilties of assignment: Complete Random Assignment

Description

Probabilties of assignment: Complete Random Assignment

Usage

```
complete_ra_probabilities(N, m = NULL, m_each = NULL, prob = NULL,
  prob_each = NULL, num_arms = NULL, condition_names = NULL)
```

Arguments

N The number of units. N must be a positive integer. (required)

m Use for a two-arm design in which m units are assigned to treatment and N-m units are assigned to control. (optional)

m_each Use for a multi-arm design in which the values of m_each determine the number of units assigned to each condition. m_each must be a numeric vector in which each entry is a nonnegative integer that describes how many units should be assigned to the 1st, 2nd, 3rd... treatment condition. m_each must sum to N. (optional)

prob
: Use for a two-arm design in which either floor(N*prob) or ceiling(N*prob) units are assigned to treatment. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N*prob) units will be assigned to treatment and with probability prob, ceiling(N*prob) units will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

prob_each
: Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of units assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

num_arms
: The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

condition_names
: A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

Value

A matrix of probabilities of assignment

Examples

```
# 2-arm designs
prob_mat <- complete_ra_probabilities(N=100)
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, m=50)
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, prob = .3)
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, m_each = c(30, 70),
                          condition_names = c("control", "treatment"))
head(prob_mat)

# Multi-arm Designs
prob_mat <- complete_ra_probabilities(N=100, num_arms=3)
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, m_each=c(30, 30, 40))
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, m_each=c(30, 30, 40),
                          condition_names=c("control", "placebo", "treatment"))
```

```
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, condition_names=c("control", "placebo", "treatment"))
head(prob_mat)

prob_mat <- complete_ra_probabilities(N=100, prob_each = c(.2, .7, .1))
head(prob_mat)
```

`conduct_ra` *Conduct a declared random assignment.*

Description

Conduct a declared random assignment.

Usage

```
conduct_ra(ra_declaration)
```

Arguments

`ra_declaration` A random assignment declaration, created by `declare_ra`.

Examples

```
declaration <- declare_ra(N=100, m_each=c(30, 30, 40))
Z <- conduct_ra(ra_declaration = declaration)
table(Z)
```

`declare_ra` *Declare a random assignment procedure.*

Description

Declare a random assignment procedure.

Usage

```
declare_ra(N = NULL, block_var = NULL, clust_var = NULL, m = NULL,
  m_each = NULL, prob = NULL, prob_each = NULL, block_m = NULL,
  block_m_each = NULL, block_prob_each = NULL, num_arms = NULL,
  condition_names = NULL, simple = FALSE, balance_load = FALSE)
```

Arguments

N
: The number of units. N must be a positive integer. (required)

block_var
: A vector of length N that indicates which block each unit belongs to.

clust_var
: A vector of length N that indicates which cluster each unit belongs to.

m
: Use for a two-arm design in which m units (or clusters) are assigned to treatment and N-m units (or clusters) are assigned to control. (optional)

m_each
: Use for a multi-arm design in which the values of m_each determine the number of units (or clusters) assigned to each condition. m_each must be a numeric vector in which each entry is a nonnegative integer that describes how many units (or clusters) should be assigned to the 1st, 2nd, 3rd... treatment condition. m_each must sum to N. (optional)

prob
: Use for a two-arm design in which either floor(N*prob) or ceiling(N*prob) units (or clusters) are assigned to treatment. The probability of assignment to treatment is exactly prob because with probability 1-prob, floor(N*prob) units (or clusters) will be assigned to treatment and with probability prob, ceiling(N*prob) units (or clusters) will be assigned to treatment. prob must be a real number between 0 and 1 inclusive. (optional)

prob_each
: Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. Because of integer issues, the exact number of units assigned to each condition may differ (slightly) from assignment to assignment, but the overall probability of assignment is exactly prob_each. (optional)

block_m
: Use for a two-arm design in which block_m describes the number of units to assign to treatment within each block. Note that in previous versions of randomizr, block_m behaved like block_m_each.

block_m_each
: Use for a multi-arm design in which the values of block_m_each determine the number of units (or clusters) assigned to each condition. block_m_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the number of units (or clusters) to be assigned to each treatment arm within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). The columns should be in the order of condition_names, if specified.

block_prob_each
: Use for a multi-arm design in which the values of block_prob_each determine the probabilties of assignment to each treatment condition. block_prob_each must be a matrix with the same number of rows as blocks and the same number of columns as treatment arms. Cell entries are the probabilites of assignment to treatment within each block. The rows should respect the ordering of the blocks as determined by sort(unique(block_var)). Use only if the probabilities of assignment should vary by block, otherwise use prob_each. Each row of block_prob_each must sum to 1.

num_arms
: The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names`
A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

`simple` logical, defaults to FALSE. If TRUE, simple random assignment is used. When simple = TRUE, please do not specify m, m_each, block_m, or block_m_each.

`balance_load` logical, defaults to FALSE. This feature is experimental. If set to TRUE, the function will resolve rounding problems by randomly assigning "remainder" units to each possible treatment condition with equal probability, while ensuring that the total number of units assigned to each condition does not vary greatly from assignment to assignment. However, the true probabiltiies of assignment may be different from the nominal probabilities specified in prob_each or block_prob_each. Please use with caution and perform many tests before using in a real research scenario.

Value

A list of class "ra_declaration". The list has five entries: $ra_function, a function that generates random assignments accroding to the declaration. $ra_type, a string indicating the type of random assignment used $probabilities_matrix, a matrix with N rows and num_arms columns, describing each unit's probabilities of assignment to conditions. $block_var, the blocking variable. $clust_var, the clustering variable.

Examples

```
# The declare_ra function is used in three ways:

# 1. To obtain some basic facts about a randomization:
declaration <- declare_ra(N=100, m_each=c(30, 30, 40))
declaration

# 2. To conduct a random assignment:

Z <- conduct_ra(declaration)
table(Z)

# 3. To obtain observed condition probabilities

probs <- obtain_condition_probabilities(declaration, Z)
table(probs, Z)

# Simple Random Assignment Declarations

declare_ra(N=100, simple = TRUE)
declare_ra(N=100, prob = .4, simple = TRUE)
declare_ra(N=100, prob_each=c(0.3, 0.3, 0.4),
           condition_names=c("control", "placebo", "treatment"), simple=TRUE)

# Complete Random Assignment Declarations
```

```
declare_ra(N=100)
declare_ra(N=100, m_each = c(30, 70),
           condition_names = c("control", "treatment"))
declare_ra(N=100, m_each=c(30, 30, 40))


# Block Random Assignment Declarations

block_var <- rep(c("A", "B","C"), times=c(50, 100, 200))

block_m_each <- rbind(c(10, 40),
                 c(30, 70),
                 c(50, 150))
declare_ra(block_var=block_var, block_m_each=block_m_each)


# Cluster Random Assignment Declarations

clust_var <- rep(letters, times=1:26)
declare_ra(clust_var=clust_var)
declare_ra(clust_var=clust_var, m_each=c(7, 7, 12))

# Blocked and Clustered Random Assignment Declarations

clust_var <- rep(letters, times=1:26)
block_var <- rep(NA, length(clust_var))
block_var[clust_var %in% letters[1:5]] <- "block_1"
block_var[clust_var %in% letters[6:10]] <- "block_2"
block_var[clust_var %in% letters[11:15]] <- "block_3"
block_var[clust_var %in% letters[16:20]] <- "block_4"
block_var[clust_var %in% letters[21:26]] <- "block_5"

table(block_var, clust_var)

declare_ra(clust_var = clust_var, block_var = block_var)
declare_ra(clust_var = clust_var, block_var = block_var, prob_each = c(.2, .5, .3))
```

obtain_condition_probabilities

Obtain the probabilities of units being in the conditions that they are in.

Description

This function is especially useful when units have different probabilties of assignment and the analyst plans to use inverse-probability weights.

Usage

```
obtain_condition_probabilities(ra_declaration, assignment)
```

Arguments

`ra_declaration`	A random assignment declaration, created by `declare_ra`.
`assignment`	A vector of random assignments, often created by `conduct_ra`.

Examples

```
# Conduct a block random assignment
block_var <- rep(c("A", "B","C"), times=c(50, 100, 200))
block_m_each <- rbind(c(10, 40),
                 c(30, 70),
                 c(50, 150))
declaration <- declare_ra(block_var = block_var, block_m_each = block_m_each)
Z <- conduct_ra(ra_declaration = declaration)
table(Z, block_var)

observed_probabilities <-
   obtain_condition_probabilities(ra_declaration = declaration, assignment = Z)

# Probabilities in the control group:
table(observed_probabilities[Z == 0], block_var[Z == 0])

# Probabilities in the treatment group:
table(observed_probabilities[Z == 1], block_var[Z == 1])
```

`randomizr` *randomizr*

Description

randomizr

`simple_ra` *Simple Random Assignment*

Description

simple_ra implements a random assignment procedure in which units are independently assigned to treatment conditions. Because units are assigned independently, the number of units that are assigned to each condition can vary from assignment to assignment. For most experimental applications in which the number of experimental units is known in advance, `complete_ra` is better because the number of units assigned to each condition is fixed across assignments.

In most cases, users should specify N and not more than one of prob, prob_each, or num_arms.

If only N is specified, a two-arm trial with prob = 0.5 is assumed.

Usage

```
simple_ra(N, prob = NULL, prob_each = NULL, num_arms = NULL,
  condition_names = NULL)
```

Arguments

`N` The number of units. N must be a positive integer. (required)

`prob` Use for a two-arm design. prob is the probability of assignment to treatment and must be a real number between 0 and 1 inclusive. (optional)

`prob_each` Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. (optional)

`num_arms` The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

`condition_names` A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

Value

A vector of length N that indicates the treatment condition of each unit. Is numeric in a two-arm trial and a factor variable (ordered by condition_names) in a multi-arm trial.

Examples

```
# Two Group Designs

Z <- simple_ra(N=100)
table(Z)

Z <- simple_ra(N=100, prob=0.5)
```

```
table(Z)

Z <- simple_ra(N=100, prob_each = c(0.3, 0.7),
               condition_names = c("control", "treatment"))
table(Z)

# Multi-arm Designs
Z <- simple_ra(N=100, num_arms=3)
table(Z)

Z <- simple_ra(N=100, prob_each=c(0.3, 0.3, 0.4))
table(Z)

Z <- simple_ra(N=100, prob_each=c(0.3, 0.3, 0.4),
               condition_names=c("control", "placebo", "treatment"))
table(Z)

Z <- simple_ra(N=100, condition_names=c("control", "placebo", "treatment"))
table(Z)
```

simple_ra_probabilities *Probabilties of assignment: Simple Random Assignment*

Description

Probabilties of assignment: Simple Random Assignment

Usage

```
simple_ra_probabilities(N, prob = NULL, prob_each = NULL, num_arms = NULL,
  condition_names = NULL)
```

Arguments

N	The number of units. N must be a positive integer. (required)
prob	Use for a two-arm design. prob is the probability of assignment to treatment and must be a real number between 0 and 1 inclusive. (optional)
prob_each	Use for a multi-arm design in which the values of prob_each determine the probabilties of assignment to each treatment condition. prob_each must be a numeric vector giving the probability of assignment to each condition. All entries must be nonnegative real numbers between 0 and 1 inclusive and the total must sum to 1. (optional)
num_arms	The number of treatment arms. If unspecified, num_arms will be determined from the other arguments. (optional)

condition_names
: A character vector giving the names of the treatment groups. If unspecified, the treatment groups will be named 0 (for control) and 1 (for treatment) in a two-arm trial and T1, T2, T3, in a multi-arm trial. An execption is a two-group design in which num_arms is set to 2, in which case the condition names are T1 and T2, as in a multi-arm trial with two arms. (optional)

Value

A matrix of probabilities of assignment

Examples

```
# Two Group Designs
prob_mat <- simple_ra_probabilities(N=100)
head(prob_mat)

prob_mat <- simple_ra_probabilities(N=100, prob=0.5)
head(prob_mat)

prob_mat <- simple_ra_probabilities(N=100, prob_each = c(0.3, 0.7),
                        condition_names = c("control", "treatment"))
head(prob_mat)

# Multi-arm Designs
prob_mat <- simple_ra_probabilities(N=100, num_arms=3)
head(prob_mat)

prob_mat <- simple_ra_probabilities(N=100, prob_each=c(0.3, 0.3, 0.4))
head(prob_mat)

prob_mat <- simple_ra_probabilities(N=100, prob_each=c(0.3, 0.3, 0.4),
                        condition_names=c("control", "placebo", "treatment"))
head(prob_mat)

prob_mat <- simple_ra_probabilities(N=100, condition_names=c("control", "placebo", "treatment"))
head(prob_mat)
```

Index