

Package ‘ExpDesignR’

September 1, 2026

Type Package

Title Experimental Design and Randomization Methods for Biomedical and Veterinary Research

Version 1.0.0

Description Provides reproducible methods for experimental design and treatment allocation in biomedical, veterinary, agricultural, and clinical research. Includes simple, fixed-block, variable-block, stratified, stratified-block, cluster, matched-pair, restricted, minimization, and covariate-adaptive randomization, together with completely randomized, randomized-block, factorial, split-plot, Latin square, and crossover designs. Also provides allocation summaries, balance diagnostics, schedule export, and visualization. The methods are based on established principles of randomization and experimental design; see Rosenberger and Lachin (2015, ISBN:9781118742242) and Jones and Kenward (2014, ISBN:9781439861424).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2.0)

Imports dplyr, ggplot2, rlang, stats, tibble, utils

Suggests covr, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

URL <https://github.com/vinodhpm/ExpDesignR>

BugReports <https://github.com/vinodhpm/ExpDesignR/issues>

Language en-US

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

NeedsCompilation no

Author Vinodhkumar Obli Rajendran [aut, cre],
Keerthi Aaradhana [aut]

Maintainer Vinodhkumar Obli Rajendran <vinodhkumar.rajendran@gmail.com>

Repository CRAN
Date/Publication 2026-09-01 12:40:08 UTC

Contents

allocation_summary	2
balance_check	3
block_randomization	4
cluster_randomization	4
completely_randomized_design	5
covariate_adaptive_randomization	6
crossover_design	7
export_schedule	7
factorial_design	8
latin_square	9
matched_pair_randomization	9
minimization_randomization	10
plot_randomization	11
randomization_diagnostics	12
randomized_block_design	13
restricted_randomization	13
simple_randomization	14
split_plot_design	15
stratified_block_randomization	15
stratified_randomization	16
variable_block_randomization	17
Index	18

allocation_summary	<i>Allocation Summary</i>
--------------------	---------------------------

Description

Summarizes treatment allocations from a randomization schedule.

Usage

```
allocation_summary(schedule, group_col = "Group")
```

Arguments

- | | |
|-----------|--|
| schedule | A data frame or tibble produced by an ExpDesignR randomization function. |
| group_col | Name of the treatment group column. |

Value

A tibble summarizing the number and percentage of subjects in each treatment group.

Examples

```
sch <- simple_randomization(  
  n = 20,  
  groups = c("Control", "Treatment"),  
  seed = 123  
)  
  
allocation_summary(sch)
```

balance_check

Check Treatment Allocation Balance

Description

Summarize treatment-count and percentage imbalance in an allocation schedule.

Usage

```
balance_check(schedule, group_col = "Group")
```

Arguments

schedule	A data frame or tibble containing treatment assignments.
group_col	Name of the treatment column.

Value

A tibble with treatment counts, percentages, and balance statistics.

Examples

```
x <- block_randomization(40, c("A", "B"), 4, seed = 1)  
balance_check(x)
```

block_randomization *Fixed Block Randomization*

Description

Generate a balanced fixed-block randomization schedule.

Usage

```
block_randomization(n, groups, block_size = 4, seed = NULL, ratio = NULL)
```

Arguments

n	Number of subjects. It must be divisible by 'block_size'.
groups	Character vector of treatment groups.
block_size	Size of each block.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble with subject, block, and treatment assignment.

Examples

```
block_randomization(
  24,
  c("Control", "Treatment"),
  4,
  seed = 123
)
```

cluster_randomization *Cluster Randomization*

Description

Randomly assigns intact clusters to treatment groups.

Usage

```
cluster_randomization(clusters, groups, seed = NULL, ratio = NULL)
```

Arguments

clusters	Character or numeric vector of unique cluster IDs.
groups	Treatment groups.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble containing cluster assignments.

Examples

```
cluster_randomization(paste0("Farm_", 1:20), c("Control", "Treatment"), seed = 123)
```

`completely_randomized_design`*Completely Randomized Design*

Description

Randomly assign experimental units to treatment groups.

Usage

```
completely_randomized_design(n, treatments, seed = NULL, ratio = NULL)
```

Arguments

n	Number of units.
treatments	Treatment labels.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble containing unit and treatment.

`covariate_adaptive_randomization`*Covariate-Adaptive Randomization*

Description

Allocate subjects sequentially while reducing imbalance across categorical covariates. A probabilistic element preserves allocation randomness.

Usage

```
covariate_adaptive_randomization(  
  data,  
  covariates,  
  groups = c("Control", "Treatment"),  
  p = 0.75,  
  seed = NULL  
)
```

Arguments

<code>data</code>	Study-subject data frame.
<code>covariates</code>	Covariate column names.
<code>groups</code>	Treatment groups.
<code>p</code>	Probability of selecting one of the best-scoring groups.
<code>seed</code>	Optional random seed.

Value

A tibble containing the original data and 'Treatment'.

Examples

```
dat <- data.frame(ID = 1:20, Sex = rep(c("M", "F"), 10), Site = rep(c("A", "B"), 10))  
covariate_adaptive_randomization(dat, c("Sex", "Site"), seed = 1)
```

crossover_design	<i>Crossover Design</i>
------------------	-------------------------

Description

Generates a crossover design for clinical, veterinary, pharmaceutical and agricultural experiments.

Usage

```
crossover_design(  
  treatments,  
  subjects,  
  periods = length(treatments),  
  seed = NULL  
)
```

Arguments

treatments	Character vector of treatment labels.
subjects	Number of subjects.
periods	Number of study periods.
seed	Optional random seed.

Value

A tibble containing the crossover schedule.

Examples

```
crossover_design(  
  treatments = c("A", "B"),  
  subjects = 8,  
  periods = 2,  
  seed = 123  
)
```

export_schedule	<i>Export Randomization Schedule</i>
-----------------	--------------------------------------

Description

Export a randomization schedule to a CSV file.

Usage

```
export_schedule(schedule, file, row.names = FALSE)
```

Arguments

schedule	A data frame or tibble generated by ExpDesignR.
file	Character. Output CSV filename or path. This argument must be supplied explicitly.
row.names	Logical. Should row names be written?

Value

Invisibly returns the input schedule unchanged. The function writes the schedule to the CSV file specified by file.

Examples

```
sch <- simple_randomization(
  n = 20,
  groups = c("Control", "Treatment"),
  seed = 123
)

tf <- tempfile(fileext = ".csv")

export_schedule(
  sch,
  file = tf
)

unlink(tf)
```

factorial_design	<i>Factorial Design</i>
------------------	-------------------------

Description

Generate a randomized full-factorial treatment combination design.

Usage

```
factorial_design(factors, replicates = 1L, seed = NULL)
```

Arguments

factors	Named list of factor levels.
replicates	Number of replicates per combination.
seed	Optional random seed.

Value

A tibble containing randomized factorial combinations.

latin_square	<i>Latin Square Design</i>
--------------	----------------------------

Description

Generates a Latin Square design for experimental studies.

Usage

```
latin_square(treatments, randomize = TRUE, seed = NULL)
```

Arguments

treatments	Character vector of treatment labels.
randomize	Logical. Should rows, columns and treatments be randomized? Default is TRUE.
seed	Optional random seed.

Value

A matrix representing a Latin square.

Examples

```
latin_square(  
  treatments = LETTERS[1:4],  
  seed = 123  
)
```

matched_pair_randomization	<i>Matched-Pair Randomization</i>
----------------------------	-----------------------------------

Description

Randomly assigns one member of each matched pair to each of two treatments.

Usage

```
matched_pair_randomization(  
  data,  
  pair,  
  groups = c("Control", "Treatment"),  
  seed = NULL  
)
```

Arguments

data	Study-subject data frame.
pair	Column containing matched-pair IDs.
groups	Exactly two treatment labels.
seed	Optional random seed.

Value

A tibble containing the original data and treatment assignment.

Examples

```
dat <- data.frame(  
  ID = 1:10,  
  Pair = rep(1:5, each = 2)  
)  
  
matched_pair_randomization(  
  dat,  
  "Pair",  
  c("Control", "Treatment"),  
  seed = 1  
)
```

minimization_randomization

Minimization Randomization

Description

Perform covariate-adaptive minimization by selecting the treatment that gives the smallest resulting marginal imbalance, with optional randomness.

Usage

```
minimization_randomization(
  data,
  covariates,
  groups = c("Control", "Treatment"),
  probability = 0.8,
  seed = NULL
)
```

Arguments

data	Study-subject data frame.
covariates	Character vector of categorical covariate columns.
groups	Treatment groups.
probability	Probability of selecting a best-scoring group.
seed	Optional random seed.

Value

A tibble containing the original data and treatment allocation.

Examples

```
dat <- data.frame(ID = 1:30, Sex = rep(c("M", "F"), 15), Site = rep(LETTERS[1:3], 10))
minimization_randomization(dat, c("Sex", "Site"), seed = 123)
```

plot_randomization	<i>Plot Randomization Schedule</i>
--------------------	------------------------------------

Description

Creates a bar chart showing the number of subjects allocated to each treatment group.

Usage

```
plot_randomization(
  schedule,
  group_col = "Group",
  fill = "#2C7FB8",
  title = "Treatment Allocation"
)
```

Arguments

schedule	A data frame produced by ExpDesignR.
group_col	Character. Name of the treatment column.
fill	Character. Fill colour.
title	Character. Plot title.

Value

A ggplot object.

Examples

```
sch <- simple_randomization(  
  n = 40,  
  groups = c("Control", "Treatment"),  
  seed = 123  
)  
  
plot_randomization(sch)
```

randomization_diagnostics

Randomization Diagnostics

Description

Return compact diagnostics for a treatment allocation schedule.

Usage

```
randomization_diagnostics(schedule, group_col = "Group")
```

Arguments

schedule	A data frame or tibble containing treatment assignments.
group_col	Name of the treatment column.

Value

A named list of allocation diagnostics.

Examples

```
x <- simple_randomization(50, c("A", "B"), seed = 1)  
randomization_diagnostics(x)
```

`randomized_block_design`*Randomized Block Design*

Description

Randomize units within balanced blocks.

Usage

```
randomized_block_design(n, treatments, block_size = 4, seed = NULL)
```

Arguments

<code>n</code>	Number of units.
<code>treatments</code>	Treatment labels.
<code>block_size</code>	Block size.
<code>seed</code>	Optional random seed.

Value

A tibble with unit, block, and treatment.

`restricted_randomization`*Restricted Randomization*

Description

Generate simple random allocations subject to a maximum treatment-count imbalance.

Usage

```
restricted_randomization(  
  n,  
  groups,  
  max_imbalance = 1,  
  seed = NULL,  
  ratio = NULL  
)
```

Arguments

n	Number of subjects.
groups	Treatment groups.
max_imbalance	Maximum allowed difference between the largest and smallest group counts.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble containing the restricted allocation.

Examples

```
restricted_randomization(30, c("A", "B"), max_imbalance = 2, seed = 1)
```

simple_randomization *Simple Randomization*

Description

Generate a simple random allocation schedule.

Usage

```
simple_randomization(n, groups, seed = NULL, ratio = NULL)
```

Arguments

n	Number of subjects.
groups	Character vector of treatment groups.
seed	Optional random seed.
ratio	Optional positive allocation weights for the groups.

Value

A tibble containing subject IDs and assigned groups.

Examples

```
simple_randomization(20, c("Control", "Treatment"), seed = 123)
```

split_plot_design	<i>Split-Plot Design</i>
-------------------	--------------------------

Description

Generate a randomized split-plot treatment schedule.

Usage

```
split_plot_design(whole, sub, n_whole = length(whole), seed = NULL)
```

Arguments

whole	Whole-plot treatment labels.
sub	Sub-plot treatment labels.
n_whole	Number of whole plots.
seed	Optional random seed.

Value

A tibble with whole plots and sub-plots.

stratified_block_randomization
<i>Stratified Block Randomization</i>

Description

Perform blocked randomization independently within each stratum.

Usage

```
stratified_block_randomization(  
  data,  
  strata,  
  groups,  
  block_size = 4,  
  seed = NULL,  
  ratio = NULL  
)
```

Arguments

data	Study-subject data frame.
strata	Character vector of stratification columns.
groups	Treatment groups.
block_size	Block size.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble containing the original data, stratum, block, and treatment.

Examples

```
dat <- data.frame(ID = 1:40, Sex = rep(c("M", "F"), each = 20))
stratified_block_randomization(dat, "Sex", c("A", "B"), 4, seed = 1)
```

stratified_randomization

Stratified Randomization

Description

Randomize independently within one or more strata.

Usage

```
stratified_randomization(data, strata, groups, seed = NULL, ratio = NULL)
```

Arguments

data	Study-subject data frame.
strata	Character vector of stratification columns.
groups	Treatment groups.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

The input data with a ‘Treatment’ column.

Examples

```
dat <- data.frame(ID = 1:20, Sex = rep(c("M", "F"), each = 10))
stratified_randomization(dat, "Sex", c("Control", "Treatment"), seed = 123)
```

variable_block_randomization
Variable Block Randomization

Description

Generate randomization using randomly selected permitted block sizes.

Usage

```
variable_block_randomization(  
  n,  
  groups,  
  block_sizes = c(4, 6, 8),  
  seed = NULL,  
  ratio = NULL  
)
```

Arguments

n	Number of subjects.
groups	Treatment groups.
block_sizes	Permitted block sizes. Each must support the requested allocation ratio.
seed	Optional random seed.
ratio	Optional allocation weights.

Value

A tibble with subject, block, block size, and group.

Examples

```
variable_block_randomization(  
  30,  
  c("Control", "Treatment"),  
  c(4, 6, 8),  
  seed = 123  
)
```

Index

allocation_summary, [2](#)

balance_check, [3](#)

block_randomization, [4](#)

cluster_randomization, [4](#)

completely_randomized_design, [5](#)

covariate_adaptive_randomization, [6](#)

crossover_design, [7](#)

export_schedule, [7](#)

factorial_design, [8](#)

latin_square, [9](#)

matched_pair_randomization, [9](#)

minimization_randomization, [10](#)

plot_randomization, [11](#)

randomization_diagnostics, [12](#)

randomized_block_design, [13](#)

restricted_randomization, [13](#)

simple_randomization, [14](#)

split_plot_design, [15](#)

stratified_block_randomization, [15](#)

stratified_randomization, [16](#)

variable_block_randomization, [17](#)