Package 'lfebd3'

April 24, 2026

Type Package

Title Generation and Analysis of Confounded and Fractional Factorial Block Designs

Version 0.1.0

Maintainer Sukanta Dash <sukanta.iasri@gmail.com>

Description Provides tools to generate and analyze 3-level linear factorial block designs, including complete factorial layouts, fractional factorial layouts, confounded factorial layouts, and design-characteristic summaries. The package includes utilities for recursive ternary construction, defining-contrast identification, alias/confounding summaries, incidence matrix construction, and design optimality diagnostics.The methodological framework follows foundational work on Gupta (1983) <doi:10.1111/j.2517-6161.1983.tb01253.x>. These methods assist in selecting, comparing, and studying factorial and fractional factorial block designs for large experimental situations.

License GPL-3

Encoding UTF-8

Imports MASS, Matrix, stats, utils

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Vankudoth Kumar [aut], Sukanta Dash [aut, cre], Med Ram Verma [aut]

Repository CRAN

Date/Publication 2026-04-24 20:30:06 UTC

Contents

`build_block_matrix` *Build a treatment-by-block incidence matrix*

Description

Creates the incidence matrix for a block design from a list of treatment labels grouped by block.

Usage

```
build_block_matrix(blocks)
```

Arguments

`blocks`	Named or unnamed list of blocks, where each block is a vector of treatment labels.

Value

A binary matrix with treatments in rows and blocks in columns.

See Also

[convert_to_blocks()], [FactChar()]

`build_effect_matrix` *Build a model matrix for factorial effects*

Description

Generates the treatment-effect model matrix for a factorial design using sum-to-zero contrasts.

Usage

```
build_effect_matrix(trts, factor_levels)
```

Arguments

`trts`	Character vector of treatment labels. Included for interface compatibility with the original script.
`factor_levels`	Integer vector giving the number of levels for each factor.

Value

A model matrix with one row per treatment combination.

See Also

[FactChar()]

`convert_to_blocks` *Convert a generated design object to a block list*

Description

Converts either a treatment-run data frame or a block-wise design data frame into a named list of blocks.

Usage

```
convert_to_blocks(design_df)
```

Arguments

`design_df`	Data frame returned by a design-generation function.

Value

A named list where each element is a character vector of treatment labels belonging to one block.

See Also

[build_block_matrix()], [FactChar()]

FactChar *Analyze factorial block-design characteristics*

Description

Computes incidence-based, estimability, balance, confounding, discrepancy, and optimality summaries for a factorial block design.

Usage

```
FactChar(factor_levels, blocks)
```

Arguments

factor_levels	Integer vector giving the number of levels for each factor.
blocks	List of blocks, where each block is a character vector of treatment labels.

Details

This function contains several local helper functions for pseudo-inverse calculation, contrast construction, Das-style diagnostics, discrepancy measures, and confounding checks. Those helper functions are scoped locally and are not intended to be documented as separate package-level functions.

Value

An object of class ‘"lfebd3_analysis"‘ containing the incidence matrix, C-matrix, design-property flags, confounding summary, discrepancy criteria, optimality diagnostics, and Das-style summaries. Use ‘print()‘ to display the formatted report.

See Also

[build_block_matrix()], [lfbd_analyze()], [lfbd.cf.full()]

---

fractional_confounding_summary
Summarize aliasing and confounding in a fractional LFBD

Description

Produces a low-order summary for main effects and two-factor interactions under the defining subgroup of a fractional 3-level factorial design.

Usage

```
fractional_confounding_summary(n, dc_basis)
```

Arguments

n	Integer. Number of factors.
dc_basis	Matrix whose columns form the basis of defining contrasts.

Value

A data frame describing each listed effect, its order, low-order aliases, and simple estimability/confounding indicators.

See Also

[find_defining_contrasts()], [lfbd.fr()]

---

generate_Tn_full *Generate the full ternary construction matrix*

Description

Builds the full recursive ternary matrix used as the foundation for generating complete, fractional, and confounded 3-level factorial layouts in this script.

Usage

```
generate_Tn_full(n)
```

Arguments

n	Integer. Number of recursion levels (or factors).

Details

The construction starts from a single-column seed and applies the 3-by-3 recursive expansion rule ‘n‘ times. The returned matrix is transposed so that downstream functions can extract the square design submatrix directly.

Value

A matrix with entries in `0`, `1`, and `2` representing the full ternary construction after `n` recursive expansions.

References

Elsawah, A.M. Multiple doubling: a simple effective construction technique for optimal two-level experimental designs. *Statistical Papers* **62**, 2923–2967 (2021). DOI: 10.1007/s00362-020-01221-0

Gupta, S. C. (1983). Some new methods for constructing block designs having orthogonal factorial structure. Journal of the Royal Statistical Society, Series B (Methodological), 45, 297-307.

See Also

[get_Tn_square()], [lfbd()], [lfbd.fr()], [lfbd.cf()]

Examples

```
generate_Tn_full(1)
dim(generate_Tn_full(2))
get_Tn_square(1)
dim(get_Tn_square(2))
d <- lfbd(2)
d
nrow(d)
d <- lfbd.fr(2, 1)
d
attr(d, "defining_contrasts")
lfbd.cf(2, 1)
dim(lfbd.cf(3, 2))
d <- lfbd.cf.full(2, 1)
d
colnames(d)
d <- lfbd(2)
blk <- convert_to_blocks(d)
blk
d <- lfbd.cf.full(2, 1)
convert_to_blocks(d)
d <- lfbd.cf.full(2, 1)
blk <- convert_to_blocks(d)
build_block_matrix(blk)
X <- build_effect_matrix(NULL, c(3, 3))
dim(X)
head(X)
d <- lfbd.cf.full(2, 1)
blk <- convert_to_blocks(d)
res <- FactChar(c(3, 3), blk)
print(res)
res <- lfbd_analyze(
  type = "lfbd.cf.full",
  factor_levels = c(3, 3),
  n = 2,
  r = 1,
  show_design = FALSE,
  run_analysis = FALSE
)
names(res)
```

`get_Tn_square` *Extract the square design matrix for a 3-level factorial construction*

Description

Computes the square submatrix used as the base design matrix for the LFBD generators.

Usage

```
get_Tn_square(n)
```

Arguments

n	Integer. Number of factors.

Value

A ‘3^n x n¿ style square matrix derived from the full ternary construction, with the first ‘3^n‘ rows retained after transposition.

See Also

[generate_Tn_full()], [lfbd()], [lfbd.fr()], [lfbd.cf()]

`lfbd` *Generate a complete factorial LFBD*

Description

Creates the complete 3-level factorial layout from the ternary design construction and returns treatment labels in run order.

Usage

```
lfbd(n)
```

Arguments

n	Integer. Number of factors.

Value

A data frame with columns:

Run Run labels such as ‘"run1"‘, ‘"run2"‘, ...

Treatment Treatment combination labels formed by concatenating factor levels coded as ‘0‘, ‘1‘, and ‘2‘.

See Also

[lfbd.fr()], [lfbd.cf.full()], [lfbd_analyze()]

lfbd.cf *Generate the principal block for a confounded LFBD*

Description

Extracts the principal block of size ‘3^r‘ from a complete 3-level factorial design using the recursive confounded selection rule.

Usage

```
lfbd.cf(n, r)
```

Arguments

n	Integer. Number of factors.
r	Integer. Exponent determining the principal block size, so the principal block contains ‘3^r‘ treatment combinations.

Value

A matrix representing the principal block.

See Also

[lfbd.cf.full()], [reduce_repeated()]

lfbd.cf.full *Generate the full confounded factorial design*

Description

Builds the complete set of blocks induced by the principal block of a confounded 3-level factorial design.

Usage

```
lfbd.cf.full(n, r, max_blocks_display = 12)
```

Arguments

n	Integer. Number of factors.
r	Integer. Exponent determining the principal block size.
max_blocks_display	Integer. Reserved display helper parameter from the original script.

Value

An invisible data frame whose first column identifies units and whose remaining columns correspond to blocks (‘blk1‘, ‘blk2‘, ...).

See Also

[lfbd.cf()], [convert_to_blocks()], [FactChar()]

`lfbd.fr` *Generate a fractional 3-level LFBD*

Description

Constructs a fractional layout from the complete ternary design by repeated fractional reduction and returns the design together with defining contrasts and a low-order confounding summary.

Usage

```
lfbd.fr(n, c)
```

Arguments

`n`	Integer. Number of factors.
`c`	Integer. Degree of fractionation, so the final number of runs is ‘3^(n - c)‘.

Value

An object of class ‘"lfbd_fr"‘ containing:

design A data frame with run labels and treatment labels.

design_matrix The reduced numeric design matrix.

defining_contrasts Character vector of defining contrast labels.

confounding_summary Data frame of main-effect and 2FI summaries.

See Also

[lfbd()], [lfbd.cf()], [lfbd_analyze()]

Examples

```
d <- lfbd.fr(2, 1)
d$design
d$defining_contrasts
```

`lfbd_analyze` *Generate and analyze an LFBD in one call*

Description

Wrapper that generates a complete, fractional, or confounded LFBD and then, optionally, analyzes its design characteristics.

Usage

```
lfbd_analyze(
  type = c("lfbd", "lfbd.fr", "lfbd.cf.full"),
  factor_levels,
  n,
  c = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE
)
```

Arguments

`type`	Character string specifying the generator to use. Must be one of ‘"lfbd"‘, ‘"lfbd.fr"‘, or ‘"lfbd.cf.full"‘.
`factor_levels`	Integer vector giving the number of levels for each factor.
`n`	Integer. Number of factors used by the generator.
`c`	Optional integer degree of fractionation for ‘type = "lfbd.fr"‘.
`r`	Optional integer controlling principal block size for ‘type = "lfbd.cf.full"‘.
`block_size`	Optional block size used when converting run-ordered designs into blocks.
`show_design`	Logical; if ‘TRUE‘, print the generated design.
`run_analysis`	Logical; if ‘TRUE‘, call [FactChar()] on the derived blocks.

Details

This wrapper uses internal helper functions to convert generated designs into block lists and to print the generated design before analysis.

Value

Invisibly returns a list with the selected design type, generated design object, derived block list, and analysis output.

See Also

[lfbd()], [lfbd.fr()], [lfbd.cf.full()], [FactChar()]

`print.lfebd3_analysis` *Print an lfebd3 analysis object*

Description

Displays the formatted report for an object returned by [FactChar()].

Usage

```
## S3 method for class 'lfebd3_analysis'
print(x, ...)
```

Arguments

`x`	An object of class ‘"lfebd3_analysis"‘.
`...`	Further arguments passed to or from other methods.

Value

Invisibly returns ‘x‘.

Index