

Package: lfebd3 (via r-universe)

July 21, 2026

Type Package

Title Generation and Analysis of 3-Level, 4-Level and 5-Level
Factorial Block Designs

Version 0.3.0

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

Description Provides tools to generate and analyze 3-level, 4-level and 5-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 construction, defining-contrast identification, alias and confounding summaries, incidence matrix construction, and selected design-characteristic diagnostics. The methodological framework follows foundational work on factorial block designs, including Gupta (1983) <[doi:10.1111/j.2517-6161.1983.tb01253.x](https://doi.org/10.1111/j.2517-6161.1983.tb01253.x)>.

License GPL-3

Encoding UTF-8

Imports MASS, Matrix, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

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

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-21 12:20:07 UTC

RemoteUrl <https://github.com/cran/lfebd3>

RemoteRef HEAD

RemoteSha 7f7d9f05be5f82e6ad7bd4bede1d7adaf6bfe7b2

Contents

lfebd3-package	3
build_block_matrix	3
build_effect_matrix	4
convert_to_blocks	4
FactChar	5
FactChar_fast	6
generate_Tn_full	7
get_Tn_square	7
lfebd3	8
lfebd3.cf	8
lfebd3.cf.full	9
lfebd3.ff	10
lfebd3.fr	10
lfebd3_analyze	11
lfebd4	12
lfebd4.cf	13
lfebd4.cf.full	13
lfebd4.ff	14
lfebd4.fr	15
lfebd4_analyze	15
lfebd5	16
lfebd5.cf	17
lfebd5.cf_independent_confounding	18
lfebd5.fr	18
lfebd5_analyze	19
print.lfebd3_analysis	20
print.lfebd3_analyze_result	21
print.lfebd3_cf_design	22
print.lfebd3_cf_principal	22
print.lfebd3_fr	23
print.lfebd4_analyze_result	23
print.lfebd4_cf_design	24
print.lfebd4_cf_principal	24
print.lfebd4_fr_design	25
print.lfebd5_analyze_result	25
print.lfebd5_cf_design	26
print.lfebd5_fr_design	27

Index

28

lfebd3-package	<i>Generation and Analysis of 3-, 4-, and 5-Level Factorial Block Designs</i>
----------------	---

Description

Provides generators and analysis tools for complete, fractional, and confounded factorial block designs at three, four, and five levels, together with defining-relation, aliasing, incidence-matrix, and design-characteristic summaries.

Author(s)

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

Authors:

- Vankudoth Kumar
- Sukanta Dash
- Med Ram Verma

build_block_matrix	<i>Build a treatment-by-block incidence matrix</i>
--------------------	--

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\(\)](#)

Examples

```
N <- build_block_matrix(list(B1 = c("00", "11"), B2 = c("01", "10")))
stopifnot(ncol(N) == 2)
```

build_effect_matrix	<i>Build a model matrix for factorial effects</i>
---------------------	---

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\(\)](#)

Examples

```
X <- build_effect_matrix(NULL, c(3, 3))
stopifnot(nrow(X) == 9)
```

convert_to_blocks	<i>Convert a generated design object to a block list</i>
-------------------	--

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()`

Examples

```
blocks <- convert_to_blocks(lfebd3(2))
stopifnot(length(blocks) == 1)
```

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. For a symmetric s -level design, an effect with coefficient vector a is marked as confounded only when $\sum(a[j] * x[j])$ is congruent to zero modulo s for every treatment combination x in the principal block. A constant nonzero residue is not marked as confounded. The 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()`, `lfebd3_analyze()`, `lfebd3.cf.full()`

Examples

```
d <- lfebd3.cf.full(2, 1)
res <- FactChar(c(3, 3), convert_to_blocks(d))
stopifnot(inherits(res, "lfebd3_analysis"))
```

FactChar_fast

Fast factorial block-design screening diagnostics

Description

Computes a lightweight subset of the diagnostics returned by `FactChar()`. This function is intended for large designs, especially 5-level designs with $n > 3$, where the exact all-effects diagnostics require large generalized inverses and pairwise distance matrices. It preserves the same printed-object class as `FactChar()` so the result can still be displayed with `print.lfebd3_analysis()`.

Usage

```
FactChar_fast(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.

Value

An object of class "lfebd3_analysis" containing fast design properties and block-confounding summaries. Expensive exact diagnostics are set to NA or skipped and reported as such by `print()`.

Examples

```
d <- as.data.frame(lfebd5.fr(3, c = 1))
blk <- list(B1 = apply(d, 1, paste0, collapse = ""))
res <- FactChar_fast(rep(5, 3), blk)
stopifnot(inherits(res, "lfebd3_analysis"))
```

generate_Tn_full	<i>Generate the full ternary construction matrix</i>
------------------	--

Description

Compatibility wrapper for the construction function used in earlier versions of the package.

Usage

```
generate_Tn_full(n)
```

Arguments

n Non-negative integer. Number of recursive expansions.

Value

The transposed recursive construction matrix.

Examples

```
dim(generate_Tn_full(1))
```

get_Tn_square	<i>Extract the square ternary construction matrix</i>
---------------	---

Description

Compatibility wrapper retaining the earlier package interface.

Usage

```
get_Tn_square(n)
```

Arguments

n Positive integer. Number of factors.

Value

A matrix containing the first 3^n rows of the recursive construction generated at expansion level n .

Examples

```
T2 <- get_Tn_square(2)
stopifnot(nrow(T2) == 9)
```

lfebd3	<i>Generate a complete 3-level LFBD run table</i>
--------	---

Description

Backward-compatible complete-design interface. The conceptual construction is provided by [lfebd3.ff\(\)](#), while this function returns treatment labels in the run-table format used by earlier package versions.

Usage

```
lfebd3(n)
```

Arguments

n Positive integer. Number of factors.

Value

A data frame with Run and Treatment columns.

Examples

```
d <- lfebd3(2)
stopifnot(nrow(d) == 9)
```

lfebd3.cf	<i>Generate the principal block of a confounded 3-level design</i>
-----------	--

Description

Generates a principal block of size 3^p from a 3^n complete factorial design. The first reduction uses the confounded selection rule and subsequent reductions use the fractional rule.

Usage

```
lfebd3.cf(n, p, row_numbers = FALSE)
```

Arguments

n Integer greater than or equal to two giving the number of factors.

p Positive integer giving the target block-size exponent, with $1 \leq p \leq n - 1$. The principal block contains 3^p treatments.

row_numbers Logical. If TRUE, prepend retained full-design row information.

Value

An object of class "lfebd3_cf_principal" containing design_name, design, confound_factors, design_matrix, and selected_rows.

See Also

[lfebd3.cf.full\(\)](#), [lfebd3_analyze\(\)](#)

Examples

```
d <- lfebd3.cf(3, p = 2)
stopifnot(nrow(d$design) == 9)
```

lfebd3.cf.full	<i>Generate a full confounded 3-level factorial block design</i>
----------------	--

Description

Forms all additive translates of the principal block, producing $3^{(n-r)}$ blocks of size 3^r and covering all 3^n treatment combinations.

Usage

```
lfebd3.cf.full(n, r, max_blocks_display = 12, return_info = FALSE)
```

Arguments

n	Positive integer giving the number of factors.
r	Positive integer giving the block-size exponent, with $1 \leq r \leq n$.
max_blocks_display	Retained for backward compatibility.
return_info	Logical. If TRUE, return the design and construction details in a list.

Value

A block-wise data frame of class "lfebd3_cf_design", or a list containing construction details when return_info = TRUE.

See Also

[lfebd3.cf\(\)](#), [convert_to_blocks\(\)](#), [FactChar\(\)](#)

Examples

```
d <- lfebd3.cf.full(3, r = 2)
stopifnot(nrow(d) == 9, ncol(d) == 4)
```

lfebd3.ff	<i>Generate a complete 3-level factorial design</i>
-----------	---

Description

Selects columns $3^0, 3^1, \dots, 3^{(n-1)}$ from the recursive ternary matrix to obtain all 3^n treatment combinations.

Usage

```
lfebd3.ff(n)
```

Arguments

n	Positive integer giving the number of factors.
---	--

Value

A data frame with 3^n rows and n factor columns named F1, F2, and so on.

See Also

[lfebd3\(\)](#), [lfebd3.fr\(\)](#), [lfebd3.cf.full\(\)](#)

Examples

```
d <- lfebd3.ff(3)
stopifnot(nrow(d) == 27, ncol(d) == 3)
```

lfebd3.fr	<i>Generate a fractional 3-level factorial design</i>
-----------	---

Description

Generates a $1/3^p$ fraction of a 3^n complete factorial design using recursive row-selection rules. The resulting design contains $3^{(n-p)}$ runs.

Usage

```
lfebd3.fr(n, p, row_numbers = FALSE)
```

Arguments

n	Positive integer giving the number of factors.
p	Non-negative integer giving the fractionation exponent, with $0 \leq p \leq n - 1$.
row_numbers	Logical. If TRUE, prepend the selected full-design row number and its position modulo nine.

Value

An object of class "lfebd3_fr" containing design_name, design, defining_factors, design_matrix, selected_rows, and defining_contrasts.

See Also

[lfebd3.ff\(\)](#), [lfebd3.cf\(\)](#), [lfebd3_analyze\(\)](#)

Examples

```
d <- lfebd3.fr(4, p = 2)
stopifnot(nrow(d$design) == 9)
```

lfebd3_analyze	<i>Generate and analyze a 3-level LFD in one call</i>
----------------	---

Description

Generates complete, fractional or confounded 3-level designs using the revised FF, FrF and CF construction rules and optionally applies the shared design-characteristic diagnostics.

Usage

```
lfebd3_analyze(
  type = base::c("lfebd3", "lfebd3.ff", "lfebd3.fr", "lfebd3.cf", "lfebd3.cf.full"),
  n,
  c = NULL,
  p = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE,
  fast = NULL,
  exact_limit = Inf,
  print_limit = 50
)
```

Arguments

type	Generator name: "lfebd3", "lfebd3.ff", "lfebd3.fr", "lfebd3.cf", or "lfebd3.cf.full".
n	Positive integer. Number of factors.
c	Optional backward-compatible alias for the fractional exponent.
p	Optional fractional exponent for lfebd3.fr, or principal-block exponent for lfebd3.cf.
r	Optional block-size exponent for confounded designs.

<code>block_size</code>	Block size for complete and fractional run-ordered designs.
<code>show_design</code>	Logical. Whether the print method displays the design.
<code>run_analysis</code>	Logical. Whether to calculate design diagnostics.
<code>fast</code>	Logical or NULL. Selects <code>FactChar_fast()</code> or <code>FactChar()</code> .
<code>exact_limit</code>	Maximum number of treatments for automatic exact mode.
<code>print_limit</code>	Maximum rows displayed by the print method.

Value

An object of class "lfebd3_analyze_result".

Examples

```
res <- lfebd3_analyze(
  type = "lfebd3.fr",
  n = 4,
  p = 2,
  block_size = 9,
  show_design = FALSE,
  run_analysis = FALSE
)
stopifnot(inherits(res, "lfebd3_analyze_result"))
```

lfebd4

Generate a complete 4-level factorial design

Description

A concise alias for `lfebd4.ff()` that matches the naming convention of `lfebd3()` and `lfebd5()`.

Usage

```
lfebd4(n)
```

Arguments

`n` Positive integer. Number of factors.

Value

An integer matrix with 4^n rows and n factor columns.

Examples

```
d <- lfebd4(2)
stopifnot(nrow(d) == 16)
```

lfebd4.cf	<i>Generate the principal block of a confounded 4-level design</i>
-----------	--

Description

Reduces a complete 4^n design to a principal block. The first reduction uses the confounded selection pattern and later reductions use the fractional pattern.

Usage

```
lfebd4.cf(n, fr = n - 1, return.rows = FALSE)
```

Arguments

n	Positive integer giving the number of factors.
fr	Positive integer giving the target block-size exponent. The principal block contains 4^{fr} treatments.
return.rows	Logical. If TRUE, include selected full-design row numbers.

Value

A list of class "lfebd4_cf_principal" containing the principal-block design and independent confounded effects.

Examples

```
d <- lfebd4.cf(3, fr = 2)
stopifnot(nrow(d$design) == 16)
```

lfebd4.cf.full	<i>Generate a full confounded 4-level factorial block design</i>
----------------	--

Description

Forms all additive translates of the principal block, giving $4^{(n-r)}$ blocks of size 4^r .

Usage

```
lfebd4.cf.full(n, r = 1, return.info = FALSE)
```

Arguments

n	Positive integer giving the number of factors.
r	Positive integer giving the block-size exponent.
return.info	Logical. If TRUE, return the design and construction details in a list.

Value

A data frame of class "lfebd4_cf_design", or a construction list when `return.info = TRUE`.

Examples

```
d <- lfebd4.cf.full(3, r = 2)
stopifnot(nrow(d) == 16, ncol(d) == 5)
```

lfebd4.ff

Generate a complete 4-level factorial design

Description

Constructs the complete 4^n factorial design using the recursive modulo-4 construction supplied with the package update.

Usage

```
lfebd4.ff(n, return.full.matrix = FALSE, return.cols = FALSE)
```

Arguments

`n` Positive integer. Number of factors.

`return.full.matrix` Logical. If TRUE, also return the full recursive construction matrix.

`return.cols` Logical. If TRUE, also return the selected recursive column positions.

Value

By default, an integer matrix with 4^n rows and n columns. When either return option is TRUE, a list containing design and the requested supplementary component(s) is returned.

Examples

```
d <- lfebd4.ff(2)
stopifnot(nrow(d) == 16, ncol(d) == 2)
```

lfebd4.fr	<i>Generate a fractional 4-level factorial design</i>
-----------	---

Description

Generates a $4^{(n-fr)}$ fraction from the complete 4^n design and reports independent defining factors. The special construction for $n = 5$ and $fr = 3$ is retained.

Usage

```
lfebd4.fr(n, fr = 1, return.rows = FALSE)
```

Arguments

n	Positive integer giving the number of factors.
fr	Non-negative integer giving the number of fractionation stages.
return.rows	Logical. If TRUE, include selected full-design row numbers.

Value

A list of class "lfebd4_fr_design" containing design, defining.factor, and optionally selected.rows.

Examples

```
d <- lfebd4.fr(3, fr = 1)
stopifnot(nrow(d$design) == 16)
```

lfebd4_analyze	<i>Generate and analyze a 4-level LFBF</i>
----------------	--

Description

Generates complete, fractional, or confounded 4-level factorial designs and optionally applies the shared package analysis routines with `factor_levels = rep(4, n)`.

Usage

```
lfebd4_analyze(
  type = c("lfebd4", "lfebd4.ff", "lfebd4.fr", "lfebd4.cf", "lfebd4.cf.full"),
  n,
  fr = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE,
```

```
fast = NULL,  
exact_limit = 256,  
print_limit = 50  
)
```

Arguments

type	Character string identifying the 4-level generator.
n	Positive integer giving the number of factors.
fr	Fractionation count for lfebd4.fr, or target block exponent for lfebd4.cf.
r	Optional block-size exponent for confounded designs.
block_size	Block size for complete and fractional run-ordered designs.
show_design	Logical. Whether printing displays the generated design.
run_analysis	Logical. Whether to compute design diagnostics.
fast	Logical or NULL; selects fast or exact shared diagnostics.
exact_limit	Maximum number of treatments for automatic exact mode.
print_limit	Maximum number of rows displayed by the print method.

Value

An object of class "lfebd4_analyze_result".

Examples

```
res <- lfebd4_analyze(  
  type = "lfebd4.fr", n = 3, fr = 1,  
  block_size = 16, run_analysis = FALSE  
)  
stopifnot(inherits(res, "lfebd4_analyze_result"))
```

lfebd5	<i>Generate a complete 5-level factorial design</i>
--------	---

Description

Creates the complete 5-level factorial layout for n factors. Factor columns are named F1, F2, and so on, and levels are coded as integers 0 through 4.

Usage

```
lfebd5(n)
```

Arguments

n	Positive integer. Number of factors.
---	--------------------------------------

Value

A data frame with 5^n rows and n factor columns.

Examples

```
d <- lfebd5(2)
stopifnot(nrow(d) == 25, ncol(d) == 2)
```

lfebd5.cf

Generate a confounded 5-level factorial block design

Description

Generates a 5-level confounded factorial design with 5^n treatments arranged in blocks of size 5^r . The result is returned in block-wise form.

Usage

```
lfebd5.cf(n, r = 1, return_info = FALSE)
```

Arguments

<code>n</code>	Integer. Number of factors. Must be at least 3.
<code>r</code>	Positive integer. Block-size exponent; each block has 5^r plots.
<code>return_info</code>	Logical. If TRUE, return a list containing the block design, principal block, and selection diagnostics.

Value

A data frame of class "lfebd5_cf_design" unless `return_info = TRUE`.

Examples

```
d <- lfebd5.cf(3, r = 1)
stopifnot(nrow(d) == 5, ncol(d) == 26)
```

lfebd5.cf_independent_confounding

Summarize independent confounding for a 5-level block design

Description

Returns independent confounded effects and related counts for a 5-level confounded design.

Usage

```
lfebd5.cf_independent_confounding(n, r, p = 5)
```

Arguments

n	Integer. Number of factors.
r	Positive integer. Block-size exponent.
p	Integer modulus. Defaults to 5.

Value

A list with confounding summaries and generator matrix.

Examples

```
info <- lfebd5.cf_independent_confounding(3, r = 1)
stopifnot(info$block_size == 5)
```

lfebd5.fr

Generate a fractional 5-level factorial design

Description

Generates a 5-level fractional factorial design by repeated row selection and attaches defining-factor and aliasing summaries as attributes.

Usage

```
lfebd5.fr(n, c = 1, return_info = FALSE)
```

Arguments

n	Integer. Number of factors. Must be at least 3.
c	Non-negative integer. Degree of fractionation; the design has $5^{(n - c)}$ runs.
return_info	Logical. If TRUE, return a list containing the design and selection diagnostics instead of a classed design data frame.

Value

A data frame of class "lfebd5_fr_design" unless return_info = TRUE.

Examples

```
d <- lfebd5.fr(3, c = 1)
stopifnot(nrow(d) == 25)
attr(d, "defining_factors")
```

lfebd5_analyze	<i>Generate and analyze a 5-level LFD in one call</i>
----------------	---

Description

Generates a complete, fractional, or confounded 5-level factorial block design and, optionally, computes design-characteristic diagnostics. The factor-level vector is inferred automatically as rep(5, n) unless supplied.

Usage

```
lfebd5_analyze(
  type = base::c("lfebd5", "lfebd5.fr", "lfebd5.cf"),
  factor_levels = NULL,
  n,
  c = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE,
  fast = NULL,
  exact_limit = 125,
  print_limit = 50
)
```

Arguments

type	Character string specifying the generator to use. Use "lfebd5" for complete factorial designs, "lfebd5.fr" for fractional factorial designs, and "lfebd5.cf" for confounded factorial block designs.
factor_levels	Optional integer vector. Must equal rep(5, n) when supplied.
n	Positive integer. Number of factors.
c	Optional non-negative integer. Degree of fractionation for type = "lfebd5.fr".
r	Optional positive integer. Block-size exponent for type = "lfebd5.cf"; each block has 5^r treatments.

<code>block_size</code>	Optional positive integer used to split run-ordered complete or fractional designs into consecutive blocks. Required for <code>type = "lfebd5"</code> and <code>type = "lfebd5.fr"</code> ; ignored for confounded designs because they are generated in block-wise form.
<code>show_design</code>	Logical. Stored in the returned object and used by <code>print()</code> to decide whether to display the generated design.
<code>run_analysis</code>	Logical. If TRUE, compute design-characteristic diagnostics.
<code>fast</code>	Logical or NULL. If NULL, fast mode is used automatically when the number of treatments is greater than <code>exact_limit</code> . If TRUE, use <code>FactChar_fast()</code> and skip the slow exact C-matrix, all-effects, Hamming, and J2 diagnostics. If FALSE, use exact <code>FactChar()</code> .
<code>exact_limit</code>	Positive integer. Maximum number of treatments for which exact <code>FactChar()</code> is used automatically when <code>fast = NULL</code> .
<code>print_limit</code>	Positive integer. Maximum number of rows printed from large generated designs and alias tables by <code>print()</code> . Full objects are still returned invisibly.

Value

An object of class `"lfebd5_analyze_result"`, a list with components `type`, `factor_levels`, `generated_design`, `design_table`, `blocks`, `analysis`, `show_design`, `show_confounding`, `fast`, and `print_limit`. Use `print()` to display a formatted report.

See Also

`lfebd5()`, `lfebd5.fr()`, `lfebd5.cf()`, `FactChar()`

Examples

```
res <- lfebd5_analyze(
  type = "lfebd5.fr",
  n = 3,
  c = 1,
  block_size = 25,
  show_design = FALSE,
  run_analysis = FALSE
)
stopifnot(inherits(res, "lfebd5_analyze_result"))
```

`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, show_confounding = FALSE, ...)
```

Arguments

`x` An object of class "lfebd3_analysis".
`show_confounding` Logical. If TRUE, display the block-confounding summary.
`...` Further arguments passed to or from other methods.

Value

Invisibly returns `x`.

Examples

```
d <- lfebd3.cf.full(2, 1)
res <- FactChar(c(3, 3), convert_to_blocks(d))
print(res)
```

```
print.lfebd3_analyze_result
```

Print a 3-level LFBD analysis result

Description

Print a 3-level LFBD analysis result

Usage

```
## S3 method for class 'lfebd3_analyze_result'
print(x, ...)
```

Arguments

`x` Object returned by `lfebd3_analyze()`.
`...` Further arguments passed to methods.

Value

Invisibly returns `x`.

```
print.lfebd3_cf_design
```

Print a full confounded 3-level block design

Description

Displays the block design and independent confounded factors.

Usage

```
## S3 method for class 'lfebd3_cf_design'
print(x, ...)
```

Arguments

x	Object returned by lfebd3.cf.full().
...	Further arguments passed to print().

Value

Invisibly returns x.

```
print.lfebd3_cf_principal
```

Print a principal block for a confounded 3-level design

Description

Displays the principal block and independent confounded factors.

Usage

```
## S3 method for class 'lfebd3_cf_principal'
print(x, ...)
```

Arguments

x	Object returned by lfebd3.cf().
...	Further arguments passed to print().

Value

Invisibly returns x.

print.lfebd3_fr	<i>Print a fractional 3-level factorial design</i>
-----------------	--

Description

Print a fractional 3-level factorial design

Usage

```
## S3 method for class 'lfebd3_fr'  
print(x, ...)
```

Arguments

x	Object returned by <code>lfebd3.fr()</code> .
...	Further arguments passed to <code>print()</code> .

Value

Invisibly returns x.

print.lfebd4_analyze_result	<i>Print a 4-level LFBD analysis result</i>
-----------------------------	---

Description

Displays the generated 4-level design and available diagnostics.

Usage

```
## S3 method for class 'lfebd4_analyze_result'  
print(x, ...)
```

Arguments

x	Object returned by <code>lfebd4_analyze()</code> .
...	Further arguments passed to methods.

Value

Invisibly returns x.

```
print.lfebd4_cf_design
```

Print a full confounded 4-level design

Description

Displays the block design and independent confounded effects.

Usage

```
## S3 method for class 'lfebd4_cf_design'
print(x, ...)
```

Arguments

x	Object returned by lfebd4.cf.full().
...	Further arguments passed to print().

Value

Invisibly returns x.

```
print.lfebd4_cf_principal
```

Print a principal block for a confounded 4-level design

Description

Displays the principal block and independent confounded effects.

Usage

```
## S3 method for class 'lfebd4_cf_principal'
print(x, ...)
```

Arguments

x	Object returned by lfebd4.cf().
...	Further arguments passed to print().

Value

Invisibly returns x.

```
print.lfebd4_fr_design
```

Print a fractional 4-level design

Description

Displays the fractional design and its defining factors.

Usage

```
## S3 method for class 'lfebd4_fr_design'
print(x, ...)
```

Arguments

x	Object returned by <code>lfebd4.fr()</code> .
...	Further arguments passed to <code>print()</code> .

Value

Invisibly returns x.

```
print.lfebd5_analyze_result
```

Print a 5-level LFBD analysis result

Description

Displays the generated design and analysis diagnostics stored in an object returned by `lfebd5_analyze()`. Printing is intentionally handled by this S3 method so that the constructor itself can be used silently in scripts, examples, and tests.

Usage

```
## S3 method for class 'lfebd5_analyze_result'
print(x, ...)
```

Arguments

x	An object returned by <code>lfebd5_analyze()</code> .
...	Further arguments passed to methods.

Value

Invisibly returns x.

Examples

```
res <- lfebd5_analyze(  
  type = "lfebd5.fr",  
  n = 3,  
  c = 1,  
  block_size = 25,  
  show_design = FALSE,  
  run_analysis = FALSE  
)  
print(res)
```

```
print.lfebd5_cf_design
```

Print a confounded 5-level design

Description

Displays a confounded 5-level block design and its confounded-effect summaries.

Usage

```
## S3 method for class 'lfebd5_cf_design'  
print(x, ...)
```

Arguments

x	Object of class "lfebd5_cf_design".
...	Further arguments passed to <code>print()</code> .

Value

Invisibly returns x.

Examples

```
d <- lfebd5.cf(3, r = 1)  
print(d)
```

```
print.lfebd5_fr_design
```

Print a fractional 5-level design

Description

Displays a fractional 5-level design with defining factors and main-effect aliases.

Usage

```
## S3 method for class 'lfebd5_fr_design'  
print(x, ...)
```

Arguments

<code>x</code>	Object of class "lfebd5_fr_design".
<code>...</code>	Further arguments passed to <code>print()</code> .

Value

Invisibly returns `x`.

Examples

```
d <- lfebd5.fr(3, c = 1)  
print(d)
```

Index

- * **package**
 - lfebd3-package, 3
- build_block_matrix, 3
- build_block_matrix(), 5
- build_effect_matrix, 4
- convert_to_blocks, 4
- convert_to_blocks(), 3, 9
- FactChar, 5
- FactChar(), 3-6, 9, 12, 20
- FactChar_fast, 6
- FactChar_fast(), 12, 20
- generate_Tn_full, 7
- get_Tn_square, 7
- lfebd3, 8
- lfebd3(), 10, 12
- lfebd3-package, 3
- lfebd3.cf, 8
- lfebd3.cf(), 9, 11
- lfebd3.cf.full, 9
- lfebd3.cf.full(), 5, 9, 10
- lfebd3.ff, 10
- lfebd3.ff(), 8, 11
- lfebd3.fr, 10
- lfebd3.fr(), 10, 23
- lfebd3_analyze, 11
- lfebd3_analyze(), 5, 9, 11, 21
- lfebd4, 12
- lfebd4.cf, 13
- lfebd4.cf.full, 13
- lfebd4.ff, 14
- lfebd4.ff(), 12
- lfebd4.fr, 15
- lfebd4_analyze, 15
- lfebd5, 16
- lfebd5(), 12, 20
- lfebd5.cf, 17
- lfebd5.cf(), 20
- lfebd5.cf_independent_confounding, 18
- lfebd5.fr, 18
- lfebd5.fr(), 20
- lfebd5_analyze, 19
- lfebd5_analyze(), 25
- print(), 23, 26, 27
- print.lfebd3_analysis, 20
- print.lfebd3_analysis(), 6
- print.lfebd3_analyze_result, 21
- print.lfebd3_cf_design, 22
- print.lfebd3_cf_principal, 22
- print.lfebd3_fr, 23
- print.lfebd4_analyze_result, 23
- print.lfebd4_cf_design, 24
- print.lfebd4_cf_principal, 24
- print.lfebd4_fr_design, 25
- print.lfebd5_analyze_result, 25
- print.lfebd5_cf_design, 26
- print.lfebd5_fr_design, 27