Package: FactChar (via r-universe)

October 30, 2026

Type Package

Title Characterization and Diagnostic Tools for Factorial Block Designs

Version 1.0

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

Description Description: Provides comprehensive tools for analysing and characterizing mixed-level factorial designs arranged in blocks. Includes construction and validation of incidence structures, computation of C-matrices, evaluation of A-, D-, E-, and MV-efficiencies, checking of orthogonal factorial structure (OFS), diagnostics based on Hamming distance, discrepancy measures, B-criterion, Es^2 statistics, J2-distance and J2-efficiency, Phi-p optimality, and symmetry conditions for universal optimality. The methodological framework follows foundational work on factorial and mixed-level design assessment by Xu and Wu (2001) <doi:10.1214/aos/1013699993>, and Gupta (1983) <doi:10.1111/j.2517-6161.1983.tb01253.x>. These methods assist in selecting, comparing, and studying factorial block designs across a range of experimental situations.

License GPL-3

Encoding UTF-8

Imports MASS, Matrix

RoxygenNote 7.3.3

NeedsCompilation no

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

Repository https://sukantaiasri.r-universe.dev

Date/Publication 2025-12-12 21:40:02 UTC

RemoteUrl https://github.com/cran/FactChar

RemoteRef HEAD

RemoteSha 040eb7e41d90670dd3f27f0830921f9d1ec09ffd

Contents

FactChar 2

Index **4**

`FactChar` *Comprehensive Analysis of Factorial Block Designs*

Description

FactChar() performs an extensive diagnostic and characterization of mixed-level factorial designs arranged in blocks.

Usage

```
FactChar(factor_levels, blocks, verbose = TRUE)
```

Arguments

`factor_levels`	Integer vector giving the number of levels of each factor, in order. For example, factor_levels = c(3, 4) means factor 1 has 3 levels coded 0, 1, 2 and factor 2 has 4 levels coded 0, 1, 2, 3.
`blocks`	A named list of blocks. Each element is a character vector of treatment labels. A treatment label is a string of digits, one digit per factor, using 0-based coding that matches factor_levels. Example: `blocks <- list( B1 = c("00", "10", "20"), B2 = c("01", "11", "21"), B3 = c("02", "12", "22"), B4 = c("03", "13", "23") )`
`verbose`	Logical; if TRUE (default) a detailed summary is printed to the console. If FALSE, all informational output is suppressed and only the result object is returned (invisibly).

Details

The function computes, among other things: - incidence matrices - C-matrix and its eigenstructure - A-, D-, E-, and MV-efficiencies - balance checks for model effects - OFS (Orthogonal Factorial Structure) checks - Hamming distance based measures - discrepancy indices (DD) - B-criterion - Es^2 (Xu and Wu) for mixed-level designs - J2-distance and J2-optimization - Phi_p optimality values - symmetry checks for universal optimality

The function validates the user-supplied blocks so that all treatment combinations are consistent with the supplied factor_levels.

Internally, the function: - builds the treatment list implied by factor_levels - validates that all user-specified treatments in blocks are compatible with factor_levels - builds the incidence matrix N - computes the C-matrix and its eigenvalues and rank - evaluates estimability and balance of factorial effects - computes OFS (Orthogonal Factorial Structure) measures - computes various discrepancy and distance measures (Hamming distance, discrete discrepancy DD) - computes MMA (Moment Matrix Analysis: K-vector) - computes MA or GMA measures (A-vector) - computes Es^2 for

mixed-level designs - computes J2-distance, J2-based efficiencies, and a lower bound - computes Phi_p values, trace(C^2), and distance from equal eigenvalues - checks symmetry conditions that are sufficient for universal optimality

The implementation includes helper routines based on methods from Das, Dean, Stufken, Wu, Hamada, Xu, Wu, Cheng, and Mukerjee, among others.

Value

Invisibly returns a list containing the main diagnostic objects: incidence structure, C-matrix and eigenvalues, efficiency measures, OFS summary, discrepancy and distance measures, J2-criteria, Phi_p values, and symmetry diagnostics. If `verbose = TRUE`, a detailed summary is printed to the console.

Input validation

If factor_levels = c(s1, s2, ..., sm), then each treatment label inside the blocks list must: - be a character string of length m - use digits 0, 1, ..., s_i - 1 in position i - match one of the prod(s_i) possible treatment combinations

If the user enters any treatment combination that does not match factor_levels, the function stops with the message:

"treatment combination is not match with your factor_levels"

References

Xu, H. and Wu, C. F. J. (2001). Generalized minimum aberration for asymmetrical fractional factorial designs. Annals of Statistics, 29, 1066-1077.

Qin, H. and Ai, M. (2007). A note on the connection between uniformity and generalized minimum aberration. Statistical Papers, 48, 491-502.

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.

Examples

```
## A valid 3 x 4 factorial in 4 blocks
factor_levels <- c(3, 4)

blocks <- list(
  B1 = c("00", "10", "20"),
  B2 = c("01", "11", "21"),
  B3 = c("02", "12", "22"),
  B4 = c("03", "13", "23")
)

out <- FactChar(factor_levels, blocks, verbose = FALSE)
str(out)
```

Index

FactChar, 2