

Package ‘DOEpro’

August 4, 2026

Title Analysis of Designed Agricultural Experiments

Version 2.0.1

Description A 'shiny' application and supporting functions for the analysis of designed agricultural experiments, following the procedures set out by Gomez and Gomez (1984, ISBN:9780471870920). Handles completely randomised, randomised complete block, Latin square, factorial (up to four factors), split-plot and strip-plot designs, and pooled (combined) analysis over environments including factorial treatments, after Yates and Cochran (1938) <doi:10.1017/S0021859600050978>; analyses several response variables simultaneously; recommends and applies variance-stabilising transformations using the profile likelihood of Box and Cox (1964) <doi:10.1111/j.2517-6161.1964.tb00553.x>; reports standard errors and critical differences for every legitimate comparison, including the four distinct comparisons of a split plot, the mixed ones using the approximation of Satterthwaite (1946) <doi:10.2307/3002019>; and produces publication-format tables of means, diagnostic plots and a written interpretation.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.0)

Imports shiny, DT, ggplot2, rlang, stats, utils, graphics, grDevices

Suggests pagedown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/Uzairkhan11w/DOEpro>, <https://doepro.pages.dev>

BugReports <https://github.com/Uzairkhan11w/DOEpro/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Immad A. Shah [aut] (ORCID: <<https://orcid.org/0000-0003-2761-5112>>),
Uzair Javid Khan [aut, cre] (ORCID:
<<https://orcid.org/0009-0001-6368-0977>>),
M. Iqbal Jeelani [aut] (ORCID: <<https://orcid.org/0000-0002-2974-2871>>)

Maintainer Uzair Javid Khan <uzairkhan11w@gmail.com>
Repository CRAN
Date/Publication 2026-08-04 14:10:39 UTC

Contents

analyze	2
build_report	3
demo_data	4
DESIGNS	5
run_all	5
run_DOEpro	6
TRANS	7
Index	8

analyze	<i>Analyse one response from a designed experiment</i>
---------	--

Description

Fits the analysis of variance for a single response variable and returns the means, standard errors and critical differences for every legitimate comparison the design allows.

Usage

```
analyze(d, design, map, alpha = 0.05)
```

Arguments

d	A data frame in long format: one row per plot, with columns for the design factors and the response.
design	The design code. One of the values of DESIGNS , for example "RCBD", "SPLIT" or "POOLFRCBD".
map	A named list mapping roles to column names of d. Always needs response. Then, by design: treat (CRD, RCBD, LSD); block (RCBD, factorial RCBD); row, col (LSD); factors, a character vector of two to four column names (factorial); rep, main, sub (split and strip plots); env together with treat or factors, and rep for an RCBD base (pooled designs).
alpha	The significance level for the critical differences and the grouping letters. Defaults to 0.05.

Details

The error term is chosen to match the design. A split plot is fitted with `Error(rep/main)` and a strip plot with `Error(rep/(A+B))`, so the main-plot and sub-plot comparisons are each tested against their own error; the two mixed comparisons use a Satterthwaite-weighted *t*. In a pooled (combined) analysis over environments, each treatment effect is tested against its own interaction with the environment, and each environment by treatment interaction against the pooled error.

Value

A list with, among others: `anova` (the analysis of variance table), `effects` (one entry per effect, each holding the table of means with `sem`, `sed`, `cd5`, `cd1` and the grouping letters), `mse` and `dfe` (the error mean square and its degrees of freedom), `cv`, `grand`, `resid` and `lm`.

See Also

[run_all](#) to analyse several responses at once, and [build_report](#) to render the result.

Examples

```
d <- demo_data("RCBD")
res <- analyze(d, "RCBD", list(response = "Yield", treat = "Variety",
                               block = "Block"))

res$anova
res$effects[["Variety"]]$means
```

build_report	<i>Render an analysis as an HTML report</i>
--------------	---

Description

Turns the result of [run_all](#) into a complete, self-contained HTML report: the analysis of variance, the tables of means with their standard errors and critical differences, the checks of the assumptions, and a plain-English interpretation. The report carries the DOEpro citation in its footer.

Usage

```
build_report(rr, letters_on = TRUE, detailed = TRUE, screen = FALSE)
```

Arguments

<code>rr</code>	The object returned by run_all .
<code>letters_on</code>	Show the grouping letters. Letters are suppressed automatically for any effect whose F-test is not significant, whatever this is set to.
<code>detailed</code>	If TRUE, include the full per-effect tables of means. If FALSE, give the compact summary tables only.
<code>screen</code>	If TRUE, return a fragment styled for display inside the application. If FALSE, return a complete standalone HTML document.

Value

A character string of HTML, of length one.

Examples

```
rr <- run_all(demo_data("CRD"), "CRD", list(treat = "Treatment"), "Yield")
html <- build_report(rr)
substr(html, 1, 60)

# writeLines(html, "report.html")
```

demo_data

Example datasets for each design

Description

Generates a small, balanced example dataset for a design, so you can try DOEpro without your own data. These are the datasets behind the **Load example** button in the application.

Usage

```
demo_data(which)
```

Arguments

which	The example to build: "CRD", "RCBD", "LSD", "FRCBD" (factorial), "SPLIT", "STRIP", "POOLRCBD" or "POOLCRD" (a single treatment factor over environments), "POOLFACT" or "POOLFACTC" (factorial treatments over environments, RCBD and CRD base respectively).
-------	---

Value

A data frame in long format, ready to pass to [analyze](#).

Examples

```
head(demo_data("RCBD"))
str(demo_data("SPLIT"))
```

DESIGNS	<i>The designs DOEpro supports</i>
---------	------------------------------------

Description

A named character vector: the names are the labels shown in the application and the values are the design codes passed to [analyze](#).

Usage

DESIGNS

Format

A named character vector of length 11.

Value

A named character vector of length 11. Each element is the design code passed to the design argument of [analyze](#) or [run_all](#); the corresponding name is the label shown in the application.

Examples

```
DESIGNS
DESIGNS[["Split Plot Design"]]
```

<code>run_all</code>	<i>Analyse several responses from a designed experiment</i>
----------------------	---

Description

Runs [analyze](#) for every response variable in turn, using the same design and mapping, and optionally applying a variance-stabilising transformation to each. This is what the application calls when more than one response is selected.

Usage

```
run_all(d, design, map, responses, alpha = 0.05, trans = NULL, dtype = "auto")
```

Arguments

d	A data frame in long format.
design	The design code; see DESIGNS .
map	A named list mapping roles to columns; see analyze . The response element is set for each response in turn and need not be supplied.
responses	A character vector of response column names.
alpha	The significance level. Defaults to 0.05.
trans	Either NULL (analyse every response untransformed) or a named list or character vector giving a transformation for each response. The names are the response columns and the values are keys of TRANS , for example <code>list(Incidence = "arcsine")</code> .
dtype	Passed to the transformation adviser; "auto" lets DOEpro judge the type of each response from its name and its mean-variance behaviour.

Value

A list with `fits` (one entry per response, each containing the fitted analysis in `final`), together with the design, the mapping and the significance level.

Examples

```
d <- demo_data("FRCBD")
rr <- run_all(d, "FRCBD",
             list(factors = c("Nitrogen", "Variety"), block = "Block"),
             responses = "Yield")
rr$fits[["Yield"]][final]$anova
```

run_DOEpro

Run the DOEpro application

Description

Launches the DOEpro Shiny application, which analyses designed agricultural experiments: paste or upload your data, choose the design, and press **Run analysis**.

Usage

```
run_DOEpro(...)
```

Arguments

... Further arguments passed to [shinyApp](#), for example `options = list(port = 8080)`.

Details

The application returns the analysis of variance, tables of means with standard errors and critical differences, checks of the assumptions with advice on transformations, post-hoc comparisons, diagnostic plots, a plain-English interpretation, and a downloadable report.

Value

An object of class `shiny.appobj`. Called for its side effect of starting the application.

Examples

```
if (interactive()) {
  run_DOEpro()
}
```

TRANS

The transformations DOEpro offers

Description

A named list of the variance-stabilising transformations. Each entry holds a label (`lab`), the transformation (`f`) and its inverse (`inv`), which is what lets DOEpro report means on the original scale of measurement alongside the transformed ones.

Usage

TRANS

Format

A named list of length 9. The names are the keys used in the `trans` argument of `run_all`: "none", "log", "log1", "sqrt", "sqrt0.5", "arcsine", "arcsine01", "reciprocal" and "boxcox".

Value

A named list of length 9. Each element is itself a list with `lab`, the label shown in the application; `f`, a function applying the transformation; and `b`, a function applying its inverse, which is what allows means to be reported on the original scale of measurement.

Examples

```
names(TRANS)
vapply(TRANS, `[`, character(1), "lab")
```

Index

analyze, [2](#), [4–6](#)

build_report, [3](#), [3](#)

demo_data, [4](#)

DESIGNS, [2](#), [5](#), [6](#)

run_all, [3](#), [5](#), [5](#), [7](#)

run_DOEpro, [6](#)

shinyApp, [6](#)

TRANS, [6](#), [7](#)