Package: optsize (via r-universe)

June 3, 2026

Title Optimal Plot Size Estimation for Field Experiments

Version 0.1.0

Description Provides methods for determining optimum plot size and shape in field experiments using Fairfield-Smith's variance law approach. It will evaluate field variability, determine optimum plot size and shape and study fertility trends across the field.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Y.A. Garde [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-0297-316X>), S.G. Patil [aut], M.S. Shitap [aut]

Maintainer Y.A. Garde <y.garde@yahoo.co.in>

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

Date/Publication 2025-10-06 08:30:02 UTC

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

RemoteRef HEAD

RemoteSha f6f9aa5ae875853b7b0b5eb1b1d773afc2a09442

Contents

`compute_moving_avg` *Compute 3x3 moving averages*

Description

Compute 3x3 moving averages

Usage

```
compute_moving_avg(mat)
```

Arguments

`mat`	A numeric matrix (at least 3 rows and 3 columns)

Value

A numeric matrix of 3x3 moving averages

`ferti_analysis` *Fertility Classes heatmap with 3 * 3 moving average values —*

Description

Fertility Classes heatmap with 3 * 3 moving average values —

Usage

```
ferti_analysis(mat)
```

Arguments

`mat`	A matrix to be converted into a horizontal vector

Value

Heatmap

`fit_variance_law` *Fit Fairfield-Smith's variance law to matrix data with ggplot2 plots*

Description

This function fits the Fairfield-Smith variance law, computes weighted R^2 on the log-log scale, identifies the optimum plot size and recommended shape, and produces two ggplot2 visualisations (original and log scale).

Usage

```
fit_variance_law(df_mat, plot_curve = TRUE)
```

Arguments

`df_mat`	numeric matrix of data
`plot_curve`	logical, if TRUE returns ggplot objects

Value

list with results:

- df_shapes: data.frame of plot shapes
- V1: variance at 1x1
- b_hat: estimated variance law coefficient
- R2_log: weighted R^2 on log-log scale
- x_opt: optimum plot size (units)
- Vx_opt: predicted variance at optimum
- best_shape: recommended shape for optimum plot size
- plots: list of ggplot objects if plot_curve = TRUE

`generate_plot_shapes` *Generate valid plot sizes and shapes*

Description

Generate valid plot sizes and shapes

Usage

```
generate_plot_shapes(df_mat)
```

Arguments

df_mat	numeric matrix of data

Value

data.frame of possible plot sizes and shapes

---

get_Tvals *Compute T values (sum of block totals) for a given h x w plot*

Description

Compute T values (sum of block totals) for a given h x w plot

Usage

```
get_Tvals(df_mat, h, w)
```

Arguments

df_mat	numeric matrix of data
h	rows in plot
w	cols in plot

Value

numeric vector of block totals

---

make_horizontal *Make a horizontal vector from a matrix*

Description

Creates a row-wise vector from a matrix. For every second row, the elements are reversed.

Usage

```
make_horizontal(mat)
```

Arguments

mat	A matrix to be converted into a horizontal vector

Value

A numeric vector

make_vertical *Make a vertical vector from a matrix*

Description

Creates a column-wise vector from a matrix. For every second column, the elements are reversed.

Usage

```
make_vertical(mat)
```

Arguments

mat	A matrix to be converted into a vertical vector

Value

A numeric vector

population_variance *Compute population variance for given h x w plot*

Description

Compute population variance for given h x w plot

Usage

```
population_variance(df_mat, h, w)
```

Arguments

df_mat	numeric matrix of data
h	rows in plot
w	cols in plot

Value

numeric variance

`serial_corr` *Compute first-order serial correlation of a vector*

Description

Computes the correlation between consecutive elements of a numeric vector.

Usage

```
serial_corr(vec)
```

Arguments

`vec`	A numeric vector

Value

Numeric value of the serial correlation

`serial_corrl` *computes the first-order serial correlation for both directions.*

Description

computes the first-order serial correlation for both directions.

Usage

```
serial_corrl(df_mat)
```

Arguments

`df_mat`	A numeric matrix

Value

A named list with two elements:

- vertical: first-order serial correlation along vertical snake
- horizontal: first-order serial correlation along horizontal snake

Index