

# Package ‘ggDoE’

May 8, 2026

**Type** Package

**Title** Modern Graphs for Design of Experiments with 'ggplot2'

**Version** 0.8

**Maintainer** Jose Toledo Luna <toledo60@protonmail.com>

**Description** Generate commonly used plots in the field of design of experiments using 'ggplot2'. 'ggDoE' currently supports the following plots: alias matrix, box cox transformation, box-plots, lambda plot, regression diagnostic plots, half normal plots, main and interaction effect plots for factorial designs, contour plots for response surface methodology, Pareto plot, and two dimensional projections of a latin hypercube design.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** <https://ggdoe.netlify.app>

**BugReports** <https://github.com/toledo60/ggDoE/issues>

**Imports** ggplot2 (>= 3.4.0), insight (>= 0.19.5)

**Suggests** roxygen2, tibble, MASS, viridisLite, ggrepel, patchwork, unrepx, geomtextpath, rsm, lhs, DoE.base

**RoxygenNote** 7.3.1

**Depends** R (>= 3.7.0)

**Language** en-US

**NeedsCompilation** no

**Author** Jose Toledo Luna [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-8559-0027>>)

**Repository** CRAN

**Date/Publication** 2024-02-10 04:50:02 UTC

## Contents

adapted_epitaxial . . . . .	2
aliased_design . . . . .	3
alias_matrix . . . . .	3
boxcox_transform . . . . .	4
design_to_tibble . . . . .	5
gg_boxplots . . . . .	5
gg_lm . . . . .	6
gg_rsm . . . . .	8
girder_experiment . . . . .	9
half_normal . . . . .	10
interaction_effects . . . . .	12
lambda_plot . . . . .	13
main_effects . . . . .	14
original_epitaxial . . . . .	15
pair_plots . . . . .	16
pareto_plot . . . . .	17
pulp_experiment . . . . .	18
theme_bw_nogrid . . . . .	19
viridisPalette . . . . .	20
<b>Index</b>	<b>21</b>

---

adapted_epitaxial	<i>Adapted epitaxial layer experiment</i>
-------------------	---

---

### Description

Same factors and levels as original epitaxial layer experiment but different data

### Usage

```
adapted_epitaxial
```

### Format

A tibble with 16 rows, 4 factors (A,B,C,D), and three responses (ybar,s2,lns2)

### Source

Wu, CF Jeff, and Michael S. Hamada. Experiments: planning, analysis, and optimization. John Wiley & Sons, 2011

---

aliased_design	<i>D-efficient minimal aliasing design for five factors in 12 runs</i>
----------------	--

---

**Description**

D-efficient minimal aliasing design for five factors in 12 runs

**Usage**

```
aliased_design
```

**Format**

A tibble with 12 rows, 5 factors

---

alias_matrix	<i>Color Map on Correlations</i>
--------------	----------------------------------

---

**Description**

Color Map on Correlations

**Usage**

```
alias_matrix(  
  design,  
  midpoint = 0.5,  
  digits = 3,  
  color_palette = "viridis",  
  alpha = 1,  
  direction = 1,  
  showplot = TRUE  
)
```

**Arguments**

design	Design of experiment (Without response)
midpoint	A midpoint value between (0,1) to split the color scheme of three colors
digits	number of digits to round correlation values. Default is 3
color_palette	A character string indicating the color map option to use. Eight options are available: "viridis", "cividis", "magma", "inferno", "plasma", "rocket", "mako", "turbo"
alpha	The alpha transparency, a number in [0,1]
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed
showplot	logical indicating to show the correlation plot. If false, the correlation/alias matrix is returned. Default is TRUE

**Value**

correlation matrix between main effects and interaction effects from the model.matrix. Alias matrix is also returned

**Examples**

```
alias_matrix(design=aliased_design)
alias_matrix(design=aliased_design, color_palette = "plasma")
alias_matrix(design=aliased_design, color_palette = "magma", direction = -1)
```

---

boxcox_transform	<i>Box-Cox Transformations</i>
------------------	--------------------------------

---

**Description**

Box-Cox Transformations

**Usage**

```
boxcox_transform(
  model,
  lambda = seq(-2, 2, 1/10),
  showlambda = TRUE,
  lambdaSF = 3,
  showplot = TRUE
)
```

**Arguments**

model	Model used for Box-Cox transformation
lambda	sequence of lambda values to consider for plot. Default is seq(-2,2,1/10)
showlambda	Default is TRUE, show lambda values in plot
lambdaSF	Digits to round lambda values shown in plot
showplot	Default is TRUE, if false plot will not be shown and a tibble is returned with a 95% confidence interval for lambda and lambda value which maximizes log-likelihood

**Value**

Box-Cox transformation plot with 95% confidence interval of lambda values to consider

**Examples**

```
model <- lm(s2 ~ (A+B+C+D), data = adapted_epitaxial)
boxcox_transform(model, lambda = seq(-5, 5, 0.2))
boxcox_transform(model, lambda = seq(-5, 5, 0.2), showplot=FALSE)
```

---

design_to_tibble	<i>Convert an object of class 'design' to 'tibble'</i>
------------------	--

---

**Description**

Convert an object of class 'design' to 'tibble'

**Usage**

```
design_to_tibble(design)
```

**Arguments**

design            An object of class 'design'

**Value**

Converted design to tibble

**Examples**

```
dat <- DoE.base::fac.design(factor.names = list(temp = c(16,32),
time = c(4,12)),replications = 5, randomize = FALSE)
Thk <- c(116.1, 106.7, 116.5, 123.2, 116.9, 107.5, 115.5, 125.1, 112.6, 105.9,
119.2, 124.5, 118.7, 107.1, 114.7, 124, 114.9, 106.5, 118.3, 124.7)
design <- DoE.base::add.response(dat, Thk)
design
design_to_tibble(design)
```

---

gg_boxplots	<i>Boxplots using ggplot2</i>
-------------	-------------------------------

---

**Description**

Boxplots using ggplot2

**Usage**

```
gg_boxplots(
  data,
  x,
  y,
  group_var = NULL,
  jitter_points = FALSE,
  horizontal = FALSE,
  point_size = 1,
```

```

alpha = 1,
color_palette = NA,
direction = 1,
show_mean = FALSE
)

```

### Arguments

data	provided dataset
x	A character string indicating the factor of the data
y	A character string indicating the response of the data
group_var	A character string indicating the groups for facet_wrap
jitter_points	Overlay jittered points to boxplots. Default is FALSE.
horizontal	Determine whether to change the orientation of the plot. Default is FALSE
point_size	Change size of points (outliers) in boxplots
alpha	The alpha transparency, a number in [0,1]
color_palette	A character string indicating the color map option to use. Eight options are available: "viridis", "cividis", "magma", "inferno", "plasma", "rocket", "mako", "turbo"
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed
show_mean	Display the mean for each boxplot. Default is FALSE

### Value

Boxplots created with ggplot2

### Examples

```

data <- ToothGrowth
data$dose <- factor(data$dose, levels = c(0.5, 1, 2), labels = c("D0.5", "D1", "D2"))
gg_boxplots(data, y = "len", x = "dose", alpha = 0.6)
gg_boxplots(data, y = "len", x = "dose", group_var = "supp",
alpha = 0.6, color_palette = 'viridis', jitter_points = TRUE)

```

---

gg\_lm

*Regression Diagnostic Plots with ggplot2*

---

### Description

Regression Diagnostic Plots with ggplot2

**Usage**

```
gg_lm(
  model,
  which_plots = 1:4,
  cooksD_type = 1,
  standard_errors = FALSE,
  point_size = 1.5,
  theme_color = "#21908CFF",
  n_columns = 2
)
```

**Arguments**

model	Model of class "lm" or "glm"
which_plots	Choose which diagnostic plots to choose from. Options are 1 = 'residual vs fitted', 2 = 'Normal-QQ', 3 = 'Scale-location', 4 = 'Residual vs Leverage', 5 = "Cook's Distance". 6 = "Collinearity". Default is 1:4
cooksD_type	An integer between 1 and 4 indicating the threshold to be computed for Cook's Distance plot. Default is 1. See details for threshold computation
standard_errors	Display confidence interval around geom_smooth, FALSE by default
point_size	Change size of points in plots
theme_color	Change color of the geom_smooth line and text labels for the respective diagnostic plot
n_columns	number of columns for grid layout. Default is 2

**Details**

**Plot 5: "Cook's Distance":** A data point having a large Cook's distance indicates that the data point strongly influences the fitted values of the model. The default threshold used for detecting or classifying observations as outliers is  $4/n$  (i.e. `cooksD_type=1`) where  $n$  is the number of observations. The thresholds computed are as follows:

**cooksD\_type = 1:**  $4/n$

**cooksD\_type = 2:**  $4/(n-p-1)$

**cooksD\_type = 3:**  $1/(n-p-1)$

**cooksD\_type = 4:**  $3 * \text{mean}(\text{cook's distance values})$

where  $n$  is the number of observations and  $p$  is the number of predictors.

**Plot 6: "Collinearity":** Considers the variance inflation factor (VIF) for multicollinearity:

Tolerance =  $1 - R_j^2$ , VIF =  $(1/\text{Tolerance})$  where  $R_j^2$  is the coefficient of determination of a regression of predictor  $j$  on all the other predictors. A general rule of thumb is that VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 indicates a multicollinearity problem

**Value**

Regression diagnostic plots

**References**

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. New York: John Wiley & Sons.

Sheather, S. (2009). *A modern approach to regression with R*. Springer Science & Business Media.

**Examples**

```
model <- lm(mpg ~ wt + am + gear, data = mtcars)
gg_lm(model)
```

---

gg\_rsm

*Contour plot(s) of a fitted linear model in ggplot2*


---

**Description**

Contour plot(s) of a fitted linear model in ggplot2

**Usage**

```
gg_rsm(
  rsm_model,
  formula,
  decode = FALSE,
  n_columns = 2,
  text_size = 3,
  bins = 6,
  ...
)
```

**Arguments**

rsm_model	Model of class "rsm"
formula	A formula, or a list of formulas
decode	This has an effect only if x is an rsm object or other model object that supports coded.data. In such cases, if decode is TRUE, the coordinate axes are transformed to their decoded values.
n_columns	number of columns for grid layout. Default is 2
text_size	size of text for labelled contour lines. Default is 3
bins	Number of contour bins. Overridden by binwidth
...	Other arguments passed on to contour(). For help with more arguments see ?rsm::contour.lm

**Value**

A grid of contour plot(s) of a fitted linear model in 'ggplot2'

**Examples**

```
## Not run:
heli.rsm <- rsm::rsm(ave ~ S0(x1, x2, x3), data = rsm::heli)

gg_rsm(heli.rsm, formula = ~x1+x2+x3, at = rsm::xs(heli.rsm), n_columns=3)

## End(Not run)
```

---

girder_experiment	<i>Girder experiment</i>
-------------------	--------------------------

---

**Description**

An experiment (Narayanan and Adorasio, 1983) to compare four methods for predicting the shear strength for steel plate girders. Data for nine girders in the form of the ratio of predicted to observed load for these procedures are given. Each of the four methods was used to predict the strength of each of the nine girders.

**Usage**

```
girder_experiment
```

**Format**

A tibble with 36 rows and 3 variables

**girders** A factor denoting one of the nine girders

**method** A factor denoting one of the four methods for predicting the shear strength for steel plate girders: Aarau, Karlsruhe, Lehigh, Cardiff

**response** The shear strength for steel plate girders

**Source**

Wu, CF Jeff, and Michael S. Hamada. Experiments: planning, analysis, and optimization. John Wiley & Sons, 2011

**Examples**

```
lm(response ~ method + girders, data = girder_experiment) |>
anova()
```

half\_normal

*Half-Normal Effects Plots***Description**

Half-Normal Effects Plots

**Usage**

```
half_normal(
  model,
  method = "Lenth",
  alpha = 0.05,
  label_active = FALSE,
  ref_line = FALSE,
  margin_errors = FALSE,
  point_color = "#21908CFF",
  showplot = TRUE
)
```

**Arguments**

model	object of class "lm"
method	Character value. Method to calculate PSE. Default is Lenth. Options include: Zahn, WZahn, Lenth, RMS, Dong, JuanPena, Daniel. See Details.
alpha	specify the significance level to compute margin of errors. Numeric significance level, between 0 and 1. Default is alpha=0.05
label_active	If TRUE, active effects are labeled if the effects cross the computed margin of error (ME). See method argument for more details
ref_line	Default is TRUE, if FALSE the abline with slope (1/PSE) is not displayed. Reference line should follow along most points that are not considered outliers.
margin_errors	Default is FALSE, if TRUE the cutoffs for margin of errors (ME) and simultaneous margin of error (SME) are shown
point_color	Change color of points in plot
showplot	Default is TRUE, if FALSE plot will not be shown and a tibble is returned used to create the plot along with the calculated PSE,ME,SME

**Details**

The method argument is a simple wrapper for the function PSE() from the unrepx R package. For more details you can use ?unrepx::PSE(). The *method* argument implements methods of estimating the standard error of effects estimates from unreplicated designs. The methods include

**Daniel:** The 68.3rd quantile of the absolute effects. See Daniel (1959)

- Dong:** The RMS method, applied after excluding all effects that exceed  $2.5 * PSE(\text{effects}, "SMedian")$  in absolute value. See Dong (1993)
- JuanPena:** An iterated median method whereby we repeatedly calculate the median of the absolute effects that don't exceed 3.5 times the previous median, until it stabilizes. The estimate is the final median, divided by .6578. See Juan and Pena (1992).
- Lenth (Default):** The SMedian method, applied after excluding all effects that exceed  $2.5 * PSE(\text{effects}, "SMedian")$  in absolute value. See Lenth (1989)
- RMS:** Square root of the mean of the squared effects. This is not a good PSE in the presence of active effects, but it is provided for sake of comparisons
- SMedian:** 1.5 times the median of the absolute effects
- Zahn, WZahn:** The Zahn method is the slope of the least-squares line fitted to the first  $m$  points of `unrep::hnplot(effects, horiz = FALSE)`, where  $m = \text{floor}(.683 * \text{length}(\text{effects}))$ . (This line is fitted through the origin.) The WZahn method is an experimental version of Zahn's method, based on weighted least-squares with weights decreasing linearly from  $m - .5$  to  $.5$ , but bounded above by  $.65m$

### Value

A tibble with the absolute effects and half-normal quantiles. A ggplot2 version of halfnormal plot for factorial effects is returned

### References

- Daniel, C (1959) Use of Half-Normal Plots in Interpreting Factorial Two-Level Experiments. *Technometrics*, 1(4), 311-341
- Dong, F (1993) On the Identification of Active Contrasts in Unreplicated Fractional Factorials. *Statistica Sinica* 3, 209-217
- Hamada and Balakrishnan (1998) Analyzing Unreplicated Factorial Experiments: A Review With Some New Proposals. *Statistica Sinica* 8, 1-41
- Juan, J and Pena, D (1992) A Simple Method to Identify Significant Effects in Unreplicated Two-Level Factorial Designs. *Communications in Statistics: Theory and Methods* 21, 1383-1403
- Lenth, R (1989) Quick and Easy Analysis of Unreligated Factorials *Technometrics* 31(4), 469-473
- Zahn, D (1975) Modifications of and Revised Critical Values for the Half-Normal Plot. *Technometrics* 17(2), 189-200

### Examples

```
model <- lm(ybar ~ (A+B+C+D)^4, data=adapted_epitaxial)
half_normal(model)
half_normal(model, method='Zahn', alpha=0.1, ref_line=TRUE,
            label_active=TRUE, margin_errors=TRUE)
```

---

interaction\_effects    *Two-Factor interaction effects plot for a factorial design*

---

### Description

Two-Factor interaction effects plot for a factorial design

### Usage

```
interaction_effects(  
  design,  
  response,  
  exclude_vars = c(),  
  linetypes = c("solid", "dashed"),  
  colors = c("#4260c9", "#d6443c"),  
  n_columns = 2,  
  showplot = TRUE  
)
```

### Arguments

design	Design of experiment (Factorial Design)
response	A character string indicating the response of the data
exclude_vars	A vector containing variables to exclude
linetypes	Change linetypes. Default are ('solid','dashed')
colors	Change color of lines/points. Default are (" #4260c9" , "#d6443c")
n_columns	number of columns for facet grid. Default is 2
showplot	logical indicating to show the interaction effect plots. If false, a list of tibbles is returned used to obtain the interaction effects for each factor. Default is TRUE

### Value

interaction effects plot between two factors

### Examples

```
interaction_effects(adapted_epitaxial,response = 'ybar',exclude_vars = c('s2','lms2'))
```

---

lambda_plot	<i>Lambda Plot: Trace of t-statistics</i>
-------------	---

---

## Description

Lambda Plot: Trace of t-statistics

## Usage

```
lambda_plot(
  model,
  lambda = seq(-2, 2, by = 0.1),
  color_palette = "viridis",
  alpha = 1,
  direction = 1,
  showplot = TRUE
)
```

## Arguments

model	Model of class "lm" or "glm"
lambda	sequence of lambda values to consider for plot. Default is seq(-2,2,0.1)
color_palette	A character string indicating the color map option to use. Eight options are available: "viridis", "cividis", "magma", "inferno", "plasma", "rocket", "mako", "turbo". Default is 'viridis'
alpha	The alpha transparency, a number in [0,1]
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed
showplot	logical indicating to show the main effect plots. If false, a list of tibbles is returned used to obtain the main effects for each factor. Default is TRUE

## Value

Lambda plot for tracing t-statistics across different values of lambda (in ggplot2)

## Examples

```
mod <- lm(s2 ~ (A+B+C)^2, data=original_epitaxial)
lambda_plot(mod)
lambda_plot(mod, lambda = seq(0, 2, 0.1))
lambda_plot(mod, lambda = seq(0, 2, 0.1), showplot = FALSE)
```

---

 main\_effects
 

---



---

*Obtain main effect plots in a factorial design*


---

**Description**

Obtain main effect plots in a factorial design

**Usage**

```
main_effects(
  design,
  response,
  exclude_vars = c(),
  n_columns = 2,
  color_palette = NA,
  alpha = 1,
  direction = 1,
  showplot = TRUE
)
```

**Arguments**

design	Design of experiment (Factorial Design)
response	A character string indicating the response of the data
exclude_vars	A vector containing variables to exclude
n_columns	number of columns for facet grid. Default is 2
color_palette	A character string indicating the color map option to use. Eight options are available: "viridis", "cividis", "magma", "inferno", "plasma", "rocket", "mako", "turbo"
alpha	The alpha transparency, a number in [0,1]
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed
showplot	logical indicating to show the main effect plots. If false, a list of data.frames is returned used to obtain the main effects for each factor. Default is TRUE

**Value**

Main effects plots, or a list of tibble with calculated main effects for each factors if showplot=FALSE.

**Examples**

```
main_effects(original_epitaxial, response='s2', exclude_vars = c('ybar', 'lns2'))
main_effects(original_epitaxial, response='ybar', exclude_vars=c('A', 's2', 'lns2'), n_columns=3)
```

---

original\_epitaxial      *Original epitaxial layer experiment*

---

### Description

One of the initial steps in fabricating integrated circuit (IC) devices is to grow an epitaxial layer on polished silicon wafers. The wafers are mounted on a six-faceted cylinder (two wafers per facet), called a susceptor, which is spun inside a metal bell jar. The jar is injected with chemical vapors through nozzles at the top of the jar and heated. The process continues until the epitaxial layer grows to a desired thickness

### Usage

original\_epitaxial

### Format

A tibble with 16 observations, 4 factors (A,B,C,D), and three responses (ybar,s2,lns2)

**Factor A** Susceptor-rotation method. Low level is oscillating and high level is continuous

**Factor B** Nozzle position. Low level is 2 and high level is 6

**Factor C** Deposition temperature (Celsius). Low level is 1210 and high level is 1220

**Factor D** Deposition time. Low level is low and high level is high

**ybar** average thickness

**s2** variance of thickness

**lns2** log variance of thickness

### Details

In the epitaxial layer growth process, suppose that the four experimental factors, susceptor rotation method, nozzle position, deposition temperature, and deposition time (labeled A, B, C and D) are to be investigated at the two levels each.

The purpose of this experiment is to find process conditions, that is, combinations of factor levels for A, B, C, and D, under which the average thickness is close to the target 14.5 micrometre with variation as small as possible. The most basic experimental design or plan is the full factorial design, which studies all possible combinations of factors at two levels.

### Source

Wu, CF Jeff, and Michael S. Hamada. Experiments: planning, analysis, and optimization. John Wiley & Sons, 2011

---

pair\_plots

*Two Dimensional Projections of Latin Hypercube Designs*

---

### Description

Two Dimensional Projections of Latin Hypercube Designs

### Usage

```
pair_plots(  
  design,  
  point_color = "#21908CFF",  
  grid = FALSE,  
  point_size = 1.5,  
  n_columns = 2  
)
```

### Arguments

design	A Latin hypercube design. Can be matrix, or data.frame
point_color	Change color of points in plots
grid	Logical argument. Specify if a grid should be added to each projection. The grid is constructed using <code>seq(0,1,length=nrows(design)+1)</code> . Default is FALSE
point_size	Change size of points in plots
n_columns	number of columns for grid layout. Default is 2

### Value

A grid of scatter plots from all two dimensional projections of a Latin hypercube design.

### Examples

```
set.seed(10)  
X <- lhs::randomLHS(n=15,k=4)  
pair_plots(X,n_columns = 3)
```

---

pareto\_plot                      *Pareto Plot of Effects*

---

### Description

Pareto Plot of Effects

### Usage

```
pareto_plot(
  model,
  alpha = 0.05,
  method = "Lenth",
  effect_colors = c("#F8766D", "#00BFC4"),
  margin_errors = TRUE,
  showplot = TRUE
)
```

### Arguments

model	Model of class "lm"
alpha	specify the significance level to compute margin of errors. Numeric significance level, between 0 and 1. Default is alpha=0.05
method	Character value. Method to calculate PSE. Default is Lenth. Options include: Zahn, WZahn, Lenth, RMS, Dong, JuanPena, Daniel. See Details.
effect_colors	Change colors of effects. Default are ('#F8766D','#00BFC4') for negative and positive effects, respectively.
margin_errors	Default is TRUE, if false the cutoffs for margin of errors (ME) and simultaneous margin of error (SME) are not shown
showplot	Default is TRUE, if false plot will not be shown and a tibble is returned with data used to create the pareto plot

### Details

The method argument is a simple wrapper for the function PSE() from the unrep R package. For more details you can use ?unrep::PSE(). The *method* argument implements methods of estimating the standard error of effects estimates from unreplicated designs. The methods include

**Daniel:** The 68.3rd quantile of the absolute effects. See Daniel (1959)

**Dong:** The RMS method, applied after excluding all effects that exceed  $2.5 * PSE(\text{effects}, "SME\text{-}dian")$  in absolute value. See Dong (1993)

**JuanPena:** An iterated median method whereby we repeatedly calculate the median of the absolute effects that don't exceed 3.5 times the previous median, until it stabilizes. The estimate is the final median, divided by .6578. See Juan and Pena (1992).

**Lenth (Default):** The SMedian method, applied after excluding all effects that exceed  $2.5 * PSE(\text{effects}, "SMedian")$  in absolute value. See Lenth (1989)

**RMS:** Square root of the mean of the squared effects. This is not a good PSE in the presence of active effects, but it is provided for sake of comparisons

**SMedian:** 1.5 times the median of the absolute effects

**Zahn, WZahn:** The Zahn method is the slope of the least-squares line fitted to the first  $m$  points of `unrep::hnplot(effects, horiz = FALSE)`, where  $m = \text{floor}(.683 * \text{length}(\text{effects}))$ . (This line is fitted through the origin.) The WZahn method is an experimental version of Zahn's method, based on weighted least-squares with weights decreasing linearly from  $m - .5$  to  $.5$ , but bounded above by  $.65m$

### Value

A bar plot with ordered effects, margin of error (ME) and simultaneous margin of error (SME) cutoffs.

### References

Daniel, C (1959) Use of Half-Normal Plots in Interpreting Factorial Two-Level Experiments. *Technometrics*, 1(4), 311-341

Dong, F (1993) On the Identification of Active Contrasts in Unreplicated Fractional Factorials. *Statistica Sinica* 3, 209-217

Hamada and Balakrishnan (1998) Analyzing Unreplicated Factorial Experiments: A Review With Some New Proposals. *Statistica Sinica* 8, 1-41

Juan, J and Pena, D (1992) A Simple Method to Identify Significant Effects in Unreplicated Two-Level Factorial Designs. *Communications in Statistics: Theory and Methods* 21, 1383-1403

Lenth, R (1989) Quick and Easy Analysis of Unreplicated Factorials *Technometrics* 31(4), 469-473

Zahn, D (1975) Modifications of and Revised Critical Values for the Half-Normal Plot. *Technometrics* 17(2), 189-200

### Examples

```
m1 <- lm(lns2 ~ (A+B+C+D)^4, data=original_epitaxial)
pareto_plot(m1)
pareto_plot(m1, method='Zahn', alpha=0.1)
```

**Description**

Plant performance is based on pulp brightness as measured by a reflectance meter. Each of the four shift operators (denoted by A, B, C, and D) made five pulp handsheets from unbleached pulp. Reflectance was read for each of the handsheets using a brightness tester

**Usage**

```
pulp_experiment
```

**Format**

A tibble with 5 rows, and 4 variables (A,B,C,D)

**Source**

Wu, CF Jeff, and Michael S. Hamada. Experiments: planning, analysis, and optimization. John Wiley & Sons, 2011

---

theme_bw_nogrid	<i>Theme for plots used in 'ggDoE'</i>
-----------------	--

---

**Description**

Theme for plots used in 'ggDoE'

**Usage**

```
theme_bw_nogrid()
```

**Value**

A simple black and white theme without grid.major and grid.minor for ggplot objects.

**Examples**

```
library(ggplot2)
data <- ToothGrowth
data$dose <- factor(data$dose, levels = c(0.5, 1, 2),
                    labels = c("D0.5", "D1", "D2"))

ggplot(data, aes(x=dose, y=len)) +
  geom_boxplot()+
  theme_bw_nogrid()
```

---

viridisPalette      *Simple viridisLite wrapper*

---

**Description**

Simple viridisLite wrapper

**Usage**

```
viridisPalette(  
  total_colors,  
  color_palette = "viridis",  
  alpha = 1,  
  direction = 1  
)
```

**Arguments**

total_colors	number of colors desired
color_palette	A character string indicating the color map option to use. Eight options are available: "viridis", "cividis", "magma", "inferno", "plasma", "rocket", "mako", "turbo"
alpha	The alpha transparency, a number in [0,1]
direction	Sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed

**Value**

Specified color palette of length 'total\_colors'

**Examples**

```
viridisPalette(5)  
viridisPalette(5,color_palette='magma',alpha=0.5)  
viridisPalette(5,color_palette='plasma',alpha=0.6,direction=-1)
```

# Index

## \* datasets

- adapted\_epitaxial, 2
- aliased\_design, 3
- girder\_experiment, 9
- original\_epitaxial, 15
- pulp\_experiment, 18

- adapted\_epitaxial, 2
- alias\_matrix, 3
- aliased\_design, 3

- boxcox\_transform, 4

- design\_to\_tibble, 5

- gg\_boxplots, 5
- gg\_lm, 6
- gg\_rsm, 8
- girder\_experiment, 9

- half\_normal, 10

- interaction\_effects, 12

- lambda\_plot, 13

- main\_effects, 14

- original\_epitaxial, 15

- pair\_plots, 16
- pareto\_plot, 17
- pulp\_experiment, 18

- theme\_bw\_nogrid, 19

- viridisPalette, 20