Package ‘agricolaeplotr’

January 30, 2025

Type Package

Title Visualization of Design of Experiments from the 'agricolae' Package

Version 0.6.1

Maintainer Jens Harbers <`jensharbers@gmail.com`>

Description
Visualization of Design of Experiments from the 'agricolae' package with 'ggplot2' framework
The user provides an experiment design from the 'agricolae' package, calls the corresponding function and will receive a
visualization with 'ggplot2' based functions that are specific for each design. As there are many different designs, each design is tested on its type.
The output can be modified with standard 'ggplot2' commands or with other packages with 'ggplot2' function extensions.

License GPL (>= 3)

Encoding UTF-8

Imports ggplot2, agricolae, raster, sp (>= 2.0.0), methods, FielDHub, utils, tibble, sf, dplyr, tidyr, stringr, stplanr, ggspatial

RoxygenNote 7.2.3

Language en-US

Suggests testthat (>= 3.0.0), knitr, rmarkdown, leaflet

Config/testthat/edition 3

BugReports `https://github.com/jensharbers/agricolaeplotr/issues`

URL `https://github.com/jensharbers/agricolaeplotr`

Depends R (>= 4.0)

VignetteBuilder knitr

Note 'None'

NeedsCompilation no

Author Jens Harbers [aut, cre] (<`https://orcid.org/0000-0001-6634-623X`>)

Repository CRAN

Date/Publication 2025-01-30 16:50:02 UTC

Contents

`citations` *Citation*

Description

Generates citations of all loaded packages

Usage

```
citations(includeURL = TRUE, bibtex = TRUE)
```

Arguments

`includeURL`	boolean, Should the URL be returned?
`bibtex`	boolean, Should the citations be returned as bibtex?

Value

printed output to console

Examples

```
library(ggplot2)
library(agricolaeplotr)
library(agricolae)
library(raster)
citations()
```

`DOE_obj` *Measures of a Field Design*

Description

Returns a list with several useful information about the experiment

Usage

```
DOE_obj(p)
```

Arguments

`p`	`ggplot` object containing the data of the plot

Value

a list with several useful information about the experiment and the field

Examples

```
library(agricolae)
library(agricolaeplotr)
trt = c(2,3,4,5,6)
outdesign1 <-design.crd(trt,r=5,serie=2,2543,'Mersenne-Twister')
p <- plot_design_crd(outdesign1,
             ncols = 7,
             nrows = 4,
             width = 10,
             height = 10,
             reverse_y = TRUE)
stats <- DOE_obj(p)
stats
```

`full_control_positions` *full_control_positions*

Description

This function provides full control about the plotting. The user also may shift the coordinates as liked.

Usage

```
full_control_positions(
  design,
  x = "col",
  y = "row",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE,
  way_x = 0,
  way_y = 0,
  shift_columns = 0,
  shift_rows = 0,
  shift_x = 0,
  shift_y = 0,
  dist_x = 1,
  dist_y = 1,
  n_shift_columns = 0,
  n_shift_rows = 0,
  start_origin = FALSE
)
```

Arguments

`design`	data.frame containing the row and columns of an experiment
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	string Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE
`way_x`	numeric vector indicates the shift of the nth-plot in x-axis.
`way_y`	numeric vector indicates the shift of the nth-plot in y-axis.
`shift_columns`	numeric indicates the shift of the given plots of a specific row by n units in x-axis.
`shift_rows`	numeric indicates the shift of the given plots of a specific column by n units in y-axis.
`shift_x`	numeric indicates the shift in units in x-axis.
`shift_y`	numeric indicates the shift in units for the y-axis.
`dist_x`	numeric indicates the shift in plots in x-axis.
`dist_y`	numeric indicates the shift in plots for the y-axis.
`n_shift_columns`	numeric vector indicating the number of plots of shift_columns. negative number indicate shift to left, otherwise right
`n_shift_rows`	numeric vector indicating the number of plots of shift_rows. negative number indicate shift to left, otherwise right
`start_origin`	boolean. Should the design start at the origin (0|0)?

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
library(ggplot2)
varieties<-c('perricholi','yungay','maria bonita','tomasa')
outdesign <-design.youden(varieties,r=2,serie=2,seed=23)
design <- outdesign$book
design
p <- full_control_positions(design,"col","row","varieties","plots",
                       width=3,height=4.5,
                       space_width=0.5,space_height=0.5,
                      shift_x=(-0.5*3) + (-0.5*3*0.5),shift_y=-0.5*4.5 + (-0.5*4.5*0.5))
p
p <- full_control_positions(design,"col","row","varieties","plots",
                       width=3,height=4.5,
                       space_width=0.13,space_height=0.445,
               shift_x=(-0.5*3) + (-0.5*3*(1-0.13)),shift_y=-0.5*4.5 + (-0.5*4.5*(1-0.445)))
                       p

varieties<-LETTERS[1:12]
outdesign <-design.youden(varieties,r=12,serie=2,seed=23)
design <- outdesign$book
p <- full_control_positions(design,"col","row","varieties","plots",
                       width=3,height=4.5,
                       space_width=1,space_height=1,
                       shift_x=-0.5*3,shift_y=-0.5*4.5)
p

p <- full_control_positions(design,"col","row","varieties","plots",
                       width=3,height=4.5,
                       space_width=0.93,space_height=0.945,
                       start_origin = TRUE)
                       p

p <- full_control_positions(design,"col","row","varieties","plots",
width=3,height=4.5,
space_width=0.93,space_height=0.945,way_x = c(2,6,8,10,12),way_y=c(3,8),dist_x=2,dist_y=4,
start_origin = TRUE, reverse_y = FALSE,  reverse_x = FALSE);p

p <- full_control_positions(design,"col","row","varieties","plots",
                                    width=3,height=4.5,
                                    space_width=0.93,space_height=0.945,
                                    way_x = c(2,4,6,8,10,12),way_y=c(3,8),
                                    start_origin = FALSE, reverse_y = FALSE,
                                    reverse_x = FALSE);p
p <- full_control_positions(design,"col","row","varieties","plots",
                                    width=3,height=4.5,shift_columns=c(4,8),
                                    shift_rows=c(3,5,9),
                                    n_shift_columns=c(1,5),
                                    n_shift_rows=c(1,-2,6),
                                    space_width=0.93,space_height=0.945,
                                    way_x = c(2,4,6,8,10,12),way_y=c(3,8),
```

```
                                    start_origin = TRUE, reverse_y = FALSE,
                                    reverse_x = FALSE);p
```

---

make_polygons *make_polygons*

Description

This function coerces all rectangles from a 'ggplot' object to 'SpatialPolygonDataFrame'.

Usage

```
make_polygons(
  ggplot_object,
  north = 3454206.89,
  east = 5939183.21,
  projection_input = "+init=epsg:31467",
  projection_output = "+init=epsg:4326"
)
```

Arguments

`ggplot_object`	saved ggplot object, containing the coordinates of the rectangles of a 'ggplot' object of the first two layers
`north`	float added to the rows to have a northing ordinate
`east`	float added to the rows to have a easting ordinate
`projection_input`	string defines in which EPSG projection the ggplot object should be converted to a raster object? a projection with a metric unit is highly recommended
`projection_output`	string defines in which EPSG projection the SpatialPolygonDataFrame should be exported.

Value

a SpatialPolygonDataFrame object

Examples

```
library(agricolaeplotr)
library(agricolae)
trt = c(2,3,4)
outdesign1 <-design.crd(trt,r=5,serie=2,2543,'Mersenne-Twister')
plt <- plot_design_crd(outdesign1,ncols = 13,nrows = 3)
spat_df <- make_polygons(plt)
spat_df
```

`plot_alpha` *Plot Alpha design Experiments*

Description

Plot a design of an experiment with an alpha design from `agricolae` design.alpha

Usage

```
plot_alpha(
  design,
  x = "cols",
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-1:30
t <- length(trt)
# size block k
k<-3
# Blocks s
s<-t/k
# replications r
r <- 2
outdesign<- design.alpha(trt,k,r,serie=2)
plot_alpha(outdesign)
```

`plot_bib` *Plot Randomized Balanced Incomplete Block Designs*

Description

Plot a design of an experiment with an Randomized Balanced Incomplete Block Designs (BIB) from design.bib

Usage

```
plot_bib(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment

`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c('A','B','C','D')
k<-3
outdesign<-design.bib(trt,k,serie=2,seed =41,kinds ='Super-Duper') # seed = 41
plot_bib(outdesign)
#now let us change position of the columns
plot_bib(outdesign,reverse_x = TRUE)
```

`plot_cyclic` *Plot Cyclic Design*

Description

Plot a design of an experiment with an cyclic design from `agricolae` design.cyclic

Usage

```
plot_cyclic(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
k <- 2
r <- 6
trt <-c('CIP-101','CIP-201','CIP-301','CIP-401','CIP-501',LETTERS[1:2])
outdesign<- design.cyclic(trt,k=k, r=r, serie=3, rowcol=TRUE)
plot_cyclic(outdesign, factor_name = 'trt')
```

`plot_dau` *Plot Design of Augmented Blocks (dau)*

Description

Plot a design of an experiment with an augmented block design from `agricolae` design.dau

Usage

```
plot_dau(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('A','B','C','D','E','F')
T2<-letters[19:26]
outdesign <-design.dau(T1,T2, r=5,serie=2)
plot_dau(outdesign)
plot_dau(outdesign,reverse_y = TRUE)
```

`plot_design.factorial_crd` *Plot Factorial Complete Randomized Designs (crd)*

Description

Plot a design of a factorial experiment with completely randomized design (crd) from design.ab

Usage

```
plot_design.factorial_crd(
  design,
  ncols,
  nrows,
  y = "row",
  factor_name = "A",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`ncols`	integer value, choose the number of columns to which the experiment should be plotted
`nrows`	integer value, choose the number of rows to which the experiment should be plotted
`y`	Describes the y coordinates of a experiment design, default is row
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	string indicates the column of which the labels should be displayed
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c(3,2) # factorial 3x2
outdesign <- design.ab(trt, r=3, serie=2,design = 'crd')
plot_design.factorial_crd(outdesign,ncols = 8,nrows = 6)
plot_design.factorial_crd(outdesign,reverse_y = TRUE,ncols = 8,nrows = 6)
plot_design.factorial_crd(outdesign,reverse_y = TRUE,reverse_x = TRUE,ncols = 8,nrows = 6)
```

`plot_design.factorial_lsd`
Plot Factorial Latin Square Designs (lsd)

Description

Plot a design of a factorial experiment with latin square design (lsd) design from `agricolae` design.ab

Usage

```
plot_design.factorial_lsd(
  design,
  x = "col",
  y = "row",
  factor_name = "A",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment

`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c(3,2) # factorial 3x2
outdesign <-design.ab(trt, r=3, serie=2,design = 'lsd')
plot_design.factorial_lsd(outdesign,factor_name = 'B',reverse_x = TRUE)
```

`plot_design.factorial_rcbd`

Plot Factorial Designs with rcbd Design

Description

Plot a design of a factorial experiment with randomized complete block design (rcbd) from design.ab

Usage

```
plot_design.factorial_rcbd(
  design,
  y = "row",
  factor_name = "A",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_x = FALSE,
  reverse_y = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c(2,4)
k=6
outdesign<-design.ab(trt, r=k, serie=3,design='rcbd')
plot_design.factorial_rcbd(design=outdesign,factor_name = 'B')
plot_design.factorial_rcbd(outdesign,reverse_y = TRUE,reverse_x = TRUE)
```

`plot_design_crd` *Plot Complete Randomized Design*

Description

Plot a design of a factorial experiment with randomized complete block design from `agricolae` design.ab

Usage

```
plot_design_crd(
  design,
  ncols,
  nrows,
  y = "row",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`ncols`	integer value, choose the number of columns to which the experiment should be plotted
`nrows`	integer value, choose the number of rows to which the experiment should be plotted
`y`	Describes the y coordinates of a experiment design, default is row
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt = c(2,3,4,5,6)
outdesign1 <-design.crd(trt,r=5,serie=2,2543,'Mersenne-Twister')
plot_design_crd(outdesign1,ncols = 13,nrows = 3)
```

`plot_fieldhub` *Plot FielDHub Design*

Description

Plots designs from `FielDHub` package

Usage

```
plot_fieldhub(
  design,
  x = "COLUMN",
  y = "ROW",
  labels = "PLOT",
  factor_name = "TREATMENT",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE,
  shift_x = 0,
  shift_y = 0
)
```

Arguments

`design`	outdesign from `FielDHub` package with on of the following IDs: c(9,13,14,15,16)
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`labels`	string Describes the column from that the plots are taken to display them
`factor_name`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width

`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE
`shift_x`	numeric indicates the shift in units in x-axis.
`shift_y`	numeric indicates the shift in units for the y-axis.

Value

`ggplot` graphic that can be modified, if wished

Examples

```
## Not run:
library(agricolaeplotr)
library(FielDHub)
H <- paste("H", 1:4, sep = "")
V <- paste("V", 1:5, sep = "")

strip1 <- FielDHub::strip_plot(Hplots = H,
                              Vplots = V,
                              b = 1,
                              l = 1,
                              plotNumber = 101,
                              planter = "serpentine",
                              locationNames = "A",
                              seed = 333)


strip1$fieldBook$ROW <- as.numeric(ordered(strip1$fieldBook$VSTRIP,
                       levels = unique(strip1$fieldBook$VSTRIP)))
strip1$fieldBook$COLUMN <- as.numeric(ordered(strip1$fieldBook$HSTRIP,
                       levels = unique(strip1$fieldBook$HSTRIP)))

plot_fieldhub(strip1,
              x = "ROW",
              y = "COLUMN",
              labels = "HSTRIP",
              factor_name = "HSTRIP",
              width = 12,
              height = 10,
              reverse_y = FALSE,
              reverse_x = FALSE)


## End(Not run)
```

`plot_graeco` *Plot Graeco Latin Square Design*

Description

Plot a design of an experiment with an Graeco - latin square design from `agricolae` design.graeco

Usage

```
plot_graeco(
  design,
  x = "col",
  y = "row",
  factor_name = "T1",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d')
T2<-c('v','w','x','y','z','zz')
outdesign <- design.graeco(trt1=T1, trt2=T2, serie = 2,
 seed = 0, kinds = 'Super-Duper',randomization=TRUE)
plot_graeco(outdesign, factor_name = 'T2',reverse_y = TRUE)
plot_graeco(outdesign, factor_name = 'T2',reverse_x = TRUE)
```

`plot_latin_square` *Plot Latin Square Design*

Description

Plot a design of a factorial experiment with a latin square design from `agricolae` design.lsd

Usage

```
plot_latin_square(
  design,
  x = "col",
  y = "row",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width

`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-LETTERS[1:9]
outdesign<- design.lsd(trt,serie=2)
plot_latin_square(outdesign, reverse_y = TRUE)
```

`plot_lattice_simple` *Plot Simple Lattice Design*

Description

Plot a design of a factorial experiment with a lattice design from `agricolae` design.lattice with r=2

Usage

```
plot_lattice_simple(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book

`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-1:100
outdesign<-design.lattice(trt,r=2,serie=3) # simple lattice design, 10x10
plot_lattice_simple(outdesign,width = 2, height = 1)
```

`plot_lattice_triple` *Plot Triple Lattice Design*

Description

Plot a design of a factorial experiment with a latin square design from `agricolae` design.lattice with r=3

Usage

```
plot_lattice_triple(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-LETTERS[1:9]
outdesign<-design.lattice(trt,r=3,serie=2)
plot_lattice_triple(design=outdesign,reverse_x=TRUE)
```

`plot_longest_diagonal` *Plot the longest diagonal of a field*

Description

This function takes a field and plots the longest diagonal of the field. The field is divided into segments and points are sampled from these segments.

Usage

```
plot_longest_diagonal(
  field,
  n = 8,
  type = "random",
  n_segments = 2,
  distance_field_boundary = 3,
  width_diagonal_path = 2
)
```

Arguments

`field`	An object of class sf representing the field.
`n`	Integer, the number of sample points along the longest diagonal.
`type`	Type of sampling. Default is "random".
`n_segments`	Numeric, the number of segments to divide the longest diagonal (default is 2).
`distance_field_boundary`	Numeric, the distance to buffer the field for creating the boundary (default is 3.0).
`width_diagonal_path`	Numeric, the width to buffer the diagonal path (default is 2.0).

Value

- p: A ggplot object showing the field, the buffered field, the buffered line, and the sample points.
- buffered_line: A sf object representing the buffered line.
- my_line: A sf object representing the longest diagonal of the field.
- sample_points: A sf object representing the sampled points.
- length: A numeric value, representing the length of the longest line.

Examples

```
library(sf)
my_sf <- st_read(system.file("shape/gfn_schlaege.shp", package="agricolaeplotr"))
st_crs(my_sf) <- 25832
field <- my_sf[my_sf$SCHLAG_NR == 170,]
plot_longest_diagonal(field)
```

`plot_rcbd` *Plot randomized complete block designs*

Description

Plot a design of an experiment with randomized complete block design (rcbd) design from `agricolae` design.rcbd

Usage

```
plot_rcbd(
  design,
  y = "block",
  factor_name = "trt",
  labels = "plots",
  treatment_label = "trt",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  label_width = 10,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	Describes the y coordinates of a experiment design
`factor_name`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`treatment_label`	Describes the column(s) from the treatments the plots are taken to display them as a label text
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`label_width`	numeric value, describes the maximum width of a label
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
# 5 treatments and 6 blocks
trt<-c('A','B','C','D','E')
outdesign <-design.rcbd(trt,6,serie=2,986,'Wichmann-Hill') # seed = 986
plot_rcbd(outdesign)
plot_rcbd(outdesign,reverse_y = TRUE,reverse_x = TRUE)
```

`plot_split_crd` *Plot Split Plot Designs (crd)*

Description

Plot a design of a split plot experiment with a complete randomized design (crd) from design.split

Usage

```
plot_split_crd(
  design,
  nrows,
  ncols,
  factor_name_1 = "T1",
  factor_name_2 = "T2",
  labels = "plots",
  subplots = TRUE,
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`nrows`	Number of rows for the design
`ncols`	Number of columns for the design
`factor_name_1`	string Which factor should be used for plotting, needs to be a column in outdesign$book

`factor_name_2`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	string Describes the column from that the plots are taken to display them
`subplots`	should the plot function return the subplots (default) or main plots?
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d','e','f','g')
T2<-c('v','w','x','y','zzz')
r <- 4
outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,
serie = 2, seed = 0, kinds = 'Super-Duper',
randomization=TRUE,first=TRUE,design = 'crd')
plot_split_crd(outdesign2,ncols = 6,nrows=5)

outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,
serie = 2, seed = 0, kinds = 'Super-Duper',
randomization=FALSE,first=TRUE,design = 'crd')
plot_split_crd(outdesign2,ncols = 6,nrows=5)
```

`plot_split_lsd` *Plot Split Plot Design lsd*

Description

Plot a design of a split plot experiment with latin squared design (lsd) from design.split

Usage

```
plot_split_lsd(
  design,
  factor_name_1 = "T1",
  factor_name_2 = "T2",
  labels = "plots",
  subplots = TRUE,
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`factor_name_1`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`factor_name_2`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	string Describes the column from that the plots are taken to display them
`subplots`	should the plot function return the subplots (default) or main plots?
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d','e')
T2<-c('v','w','x','y')
```

```
outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,serie = 2,
                         seed = 0, kinds = 'Super-Duper',
                         randomization=TRUE,first=TRUE,design = 'lsd')
plot_split_lsd(outdesign2,width = 4,height = 4)
```

`plot_split_rcbd` *Plot Split Plot Designs with rcbd*

Description

Plot a design of a split plot experiment with randomized complete blocks design (rcbd) from design.split

Usage

```
plot_split_rcbd(
  design,
  y = "block",
  factor_name_1 = "T1",
  factor_name_2 = "T2",
  subplots = TRUE,
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`y`	string defines the block
`factor_name_1`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`factor_name_2`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`subplots`	should the plot function return the subplots (default) or main plots?
`labels`	string Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width

`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d','e')
T2<-c('v','w','x','y','z','zz')
r = 3
outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,serie = 2,
 seed = 0, kinds = 'Super-Duper',randomization=TRUE,
 first=TRUE,design = 'rcbd')
plot_split_rcbd(outdesign2,width = 1,height = 1)
plot_split_rcbd(outdesign2,width = 1,height = 1,reverse_y = TRUE)
plot_split_rcbd(outdesign2,width = 1,height = 1,reverse_x = TRUE,reverse_y = TRUE)
```

`plot_strip` *Plot Strip Design*

Description

Plot a design of an experiment with an Strip Plot design from `agricolae` design.strip

Usage

```
plot_strip(
  design,
  x = "col",
  y = "row",
  factor_name_1 = "T1",
  factor_name_2 = "T2",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name_1`	Which factor should be used for plotting, needs to be a column in outdesign$book
`factor_name_2`	Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	Describes the column from that the plots are taken to display them
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d')
T2<-c('v','w','x','y','z')
r = 3
outdesign <- design.strip(trt1=T1, trt2=T2, r=r,serie = 2,
 seed = 0, kinds = 'Super-Duper',randomization=TRUE)
plot_strip(outdesign,factor_name_1 = "T1",factor_name_2="T2")
plot_strip(outdesign,factor_name_1 = "T1",factor_name_2="T2",reverse_x = TRUE)
```

`plot_youden` *Plot Youden Design*

Description

Plot a Youden experiment design from `agricolae` design.youden

Usage

```
plot_youden(
  design,
  x = "col",
  y = "row",
  factor_name = "varieties",
  labels = "plots",
  width = 1,
  height = 1,
  space_width = 0.95,
  space_height = 0.85,
  reverse_y = FALSE,
  reverse_x = FALSE
)
```

Arguments

`design`	outdesign from `agricolae` package
`x`	Describes the x coordinates of a experiment design
`y`	Describes the y coordinates of a experiment design
`factor_name`	string Which factor should be used for plotting, needs to be a column in outdesign$book
`labels`	string Describes the column from that the plots are taken to display them.
`width`	numeric value, describes the width of a plot in an experiment
`height`	numeric value, describes the height of a plot in an experiment
`space_width`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of width
`space_height`	numeric value, describes the share of the space of the plots. 0=only space, 1=no space between plots in term of height
`reverse_y`	boolean, should the plots of the experiment be changed in reverse order in Row direction? Use reverse_y=TRUE to have same sketch as in agricolae. default:reverse_y=FALSE
`reverse_x`	boolean, should the plots of the experiment be changed in reverse order in column direction? default:reverse_x=FALSE

Value

`ggplot` graphic that can be modified, if wished

Examples

```
library(agricolaeplotr)
library(agricolae)
varieties<-c('perricholi','yungay','maria bonita','tomasa')
outdesign <-design.youden(varieties,r=2,serie=2,seed=23)
plot_youden(outdesign, labels = 'varieties')
```

`protective_layers` *Create Protective Layers for Design of Experiments (DOEs)*

Description

This function generates protective layers around the polygons of an experiment. These layers can be used to plot boundaries, for example, to protect agricultural on-farm experiments from accidental harvesting.

Usage

```
protective_layers(design, borders = c(0, 3, 5, 10))
```

Arguments

`design`	An `sf` object containing the polygons of the experiment. The coordinate reference system (crs) of the data needs to be in metric distance, not degrees.
`borders`	A numeric vector specifying the distances (in meters) for which protective layers should be created. The layers will be created with decreasing distances, starting from the largest.

Value

An `sf` object representing the protective layers around the experiment polygons.

Examples

```
library(agricolaeplotr)
library(sf)
library(ggplot2)
example("make_polygons")
polygo <- make_polygons(plt, north = 13454206.89, east = 7939183.21)
polygo <- st_transform(polygo, 25832)
pl <- protective_layers(polygo)
# plot experiment shape
ggplot(pl) + geom_sf(fill=c("black","orange","blue","red"))+ theme_minimal()
# write them to kml for Google Maps
# st_write(pl, "boundaries2.kml", append = FALSE)
```

`sample_locations` *Sample Locations*

Description

Returns locations to sample for each plot.

Usage

```
sample_locations(design, n, plot = TRUE, ...)
```

Arguments

`design`	Your experiment design of plot layouts.
`n`	Number of samples per plot (integer).
`plot`	Logical, indicating whether to visualize the sample locations as a ggplot2-based map.
`...`	further options for ‘st_sample‘ and ‘make_polygons‘

Details

This function takes an experiment design (plot layout) and returns random sample locations within each plot. The function uses the ‘sf‘ package to generate spatial polygons for the plots and then samples points within each polygon. Optionally, it can also display the sample locations as a ggplot2-based map.

Value

An ‘sf‘ object containing the sample locations within each plot.

Examples

```
library(agricolaeplotr)
library(agricolae)
library(ggplot2)
trt <- c('A', 'B', 'C', 'D')
k <- 3
outdesign <- design.bib(trt, k, serie = 2, seed = 41, kinds = 'Super-Duper')
plot_bib(outdesign)
p <- plot_bib(outdesign)
sample_locations(p, 3, TRUE, projection_output = 25832)
```

`serpentine` *Serpentine*

Description

This function produces a serpentine array of integers beginning by one

Usage

```
serpentine(n, times, m = 1)
```

Arguments

`n`	integer value indicating the upper cap of a numeric sequence
`times`	integer number of replications
`m`	integer value indicating the lower cap of a numeric sequence

Value

vector containing the serpentine sequence

Examples

```
serpentine(n=20,times = 15)
serpentine(n=20,times = 15,m=4)
```

`summary` *summary of a field Layout*

Description

print a summary of a FieldLayout object

Usage

```
summary(object, unit = "m", part = "net_plot", ...)
```

Arguments

`object`	an object, created by DOE_obj with a FieldLayout class
`unit`	a string that corresponds to measure unit (default is m)
`part`	which part of the summary are you interested? Choose one of the following: "net_plot","gross_plot","field","experiment","all"
`...`	further arguments passed to or from other methods

Examples

```
library(agricolaeplotr)
library(agricolae)
varieties<-c('perricholi','yungay','maria bonita','tomasa')
outdesign <-design.youden(varieties,r=2,serie=2,seed=23)
p <- plot_youden(outdesign, labels = 'varieties')
stats <- DOE_obj(p)
# print plot summary for net plot (plots without space)
summary(stats, part = "net_plot")
# print plot summary for gross plot (plots with space)
summary(stats, part = "gross_plot")
# print plot summary for entire field
summary(stats, part = "field")
# print plot summary for design summary
summary(stats, part = "experiment")
# print plot summary for all information shown above in one output
summary(stats, part = "all")
```

`test_input_extend` *Test if input for width and height is numeric*

Description

Test if input is numeric for field width and height

Usage

```
test_input_extend(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_input_extend(3)
```

`test_input_ncols` *checks matrix column input*

Description

checks if input is suitable for matrix column indication

Usage

```
test_input_ncols(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_input_ncols(9)
```

`test_input_nrows` *checks matrix rows input*

Description

checks if input is suitable for matrix row indication

Usage

```
test_input_nrows(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_input_nrows(10)
```

`test_input_reverse` *Test if input is a logical*

Description

Test if input is a logical

Usage

```
test_input_reverse(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_input_reverse(TRUE)
```

`test_input_shift` *Test if input for shift parameter is numeric*

Description

Test if input is numeric for shift parameter

Usage

```
test_input_shift(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_input_shift(0.5)
```

`test_names_design` *Test of experimental design*

Description

Test if the outdesign file contains book and parameter list

Usage

```
test_names_design(design)
```

Arguments

`design`	design from `agricolae` package

Value

error

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c(2,4)
k=6
outdesign<-design.ab(trt, r=k, serie=3,design='rcbd')
test_names_design(outdesign)
```

`test_name_in_column` *Test if input column names*

Description

Test if input is in column names of a table

Usage

```
test_name_in_column(x, design)
```

Arguments

`x`	string input
`design`	design from `agricolae` package

Value

error

Examples

```
library(agricolaeplotr)
library(agricolae)
trt<-c(2,4)
k=6
outdesign<-design.ab(trt, r=k, serie=3,design='rcbd')
test_name_in_column('B',outdesign)
```

`test_string` *Test if input is a string*

Description

Test if input is a string

Usage

```
test_string(x)
```

Arguments

`x`	input to be tested

Value

error

Examples

```
library(agricolaeplotr)
test_string('smallstring')
```

`theme_gi` *theme_gi*

Description

Creates a theme for ’ggplot’ based graphics to ensure to meet formal requirements for conferences of the Gesellschaft fuer Informatik

Usage

```
theme_gi()
```

Value

a ’ggplot’ graph with a modified theme

Examples

```
# example borrowed from ggplot2
library(ggplot2)
df <- data.frame(
gp = factor(rep(letters[1:3], each = 10)),
y = rnorm(30))

p <- ggplot() +
geom_point(data = df, aes(gp, y))
p <- p + theme_gi();p
```

`theme_poster` *ggplot2 theme for poster presentation*

Description

This theme is designed to increase font size to ensure readability on poster presentations

Usage

```
theme_poster()
```

Value

ggplot2 theme

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d','e','f','g')
T2<-c('v','w','x','y','z')
r <- 4
outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,
serie = 2, seed = 0, kinds = 'Super-Duper',
randomization=FALSE,first=TRUE,design = 'crd')
plot_split_crd(outdesign2,ncols = 6,nrows=5)+
theme_poster()
```

---

theme_pres *ggplot2 theme for outdoor presentation*

Description

This theme is designed to increase font size to ensure readability on outdoor used devices

Usage

```
theme_pres()
```

Value

ggplot2 theme

Examples

```
library(agricolaeplotr)
library(agricolae)
T1<-c('a','b','c','d','e','f','g')
T2<-c('v','w','x','y','z')
r <- 4
outdesign2 <- design.split(trt1=T1, trt2=T2, r=r,
serie = 2, seed = 0, kinds = 'Super-Duper',
randomization=FALSE,first=TRUE,design = 'crd')
plot_split_crd(outdesign2,ncols = 6,nrows=5)+
theme_pres()
```

---

to_table *to_table*

Description

Write field experiment information to a dataframe.

Usage

```
to_table(object, part = "net_plot", unit = "m", digits = 3, ...)
```

Arguments

`object`	an object, created by DOE_obj with a FieldLayout class
`part`	which part of the summary are you interested? Choose one of the following: "net_plot","gross_plot","field","experiment"
`unit`	a string that corresponds to measure unit (default is m)
`digits`	integer indicating the number of decimal places (round) or significant digits (signif) to be used. Negative values are allowed
`...`	further arguments passed to or from other methods

Value

dataframe with corresponding information about the experiment

Examples

```
library(agricolaeplotr)
library(agricolae)
varieties<-c('perricholi','yungay','maria bonita','tomasa')
outdesign <-design.youden(varieties,r=2,serie=2,seed=23)
p <- plot_youden(outdesign, labels = 'varieties', width=4, height=3)
stats <- DOE_obj(p)
r <- to_table(stats,part = "net_plot", digits = 2)
r
r <- to_table(stats,part = "gross_plot", digits = 2)
r
r <- to_table(stats,part = "field", digits = 2)
r
r <- to_table(stats,part = "experiment", digits = 2)
r
r <- to_table(stats,part = "all", digits = 2)
r
```

Index