Package 'inti'

October 1, 2026

Title Tools and Statistical Procedures in Plant Science

Version 0.7.5

Description The 'inti' package is part of the 'inkaverse' project for developing different procedures and tools used in plant science and experimental designs. The mean aim of the package is to support researchers during the planning of experiments and data collection (tarpuy()), data analysis and graphics (yupana()) , and scientific writing.
Learn more about the 'inkaverse' project at <https://inkaverse.com/>.

License GPL-3 | file LICENSE

URL https://inkaverse.com/, https://github.com/Flavjack/inti

BugReports https://github.com/Flavjack/inti/issues

Depends shiny, ggplot2, dplyr, tidyr, tibble, R (>= 4.1.0)

Imports lme4, agricolae, FactoMineR, emmeans, purrr, stringr, googlesheets4, DT, cowplot, cli, systemfonts

Suggests gsheet, huito, knitr, rmarkdown, quarto

VignetteBuilder knitr, quarto

LazyData true

Encoding UTF-8

SystemRequirements pandoc

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Flavio Lozano-Isla [aut, cre] (ORCID: <https://orcid.org/0000-0002-0714-669X>), Yoel Diaz-Saucedo [aut] (ORCID: <https://orcid.org/0009-0003-9765-136X>), Victor-Hugo Baldera-Chapoñan [aut] (ORCID: <https://orcid.org/0009-0003-4321-7103>), María Belén Kistner [ctb] (ORCID: <https://orcid.org/0000-0002-0947-6414>), QuipoLab [ctb], Inkaverse [cph]

Maintainer Flavio Lozano-Isla <flozanoisla@gmail.com>

Repository CRAN

Date/Publication 2026-09-30 22:00:03 UTC

Contents

design_augmented . . . 3
design_noreps . . . 4
design_repblock . . . 5
design_split . . . 7
design_split_rcbd . . . 8
footnotes . . . 9
H2cal . . . 10
include_pdf . . . 12
include_table . . . 13
mean_comparison . . . 14
met . . . 15
metamorphosis . . . 16
plot_augmented_design . . . 17
plot_diagnostic . . . 18
plot_raw . . . 19
plot_smr . . . 21
plot_split_rcbd_design . . . 23
plot_standard_design . . . 25
potato . . . 27
remove_outliers . . . 28
rticle . . . 29
split_folder . . . 29
tarpuy . . . 30
tarpuy_design . . . 31
tarpuy_plex . . . 33
tarpuy_plotdesign . . . 35
tarpuy_traits . . . 37
web_table . . . 39
yupana . . . 40
yupana_analysis . . . 41
yupana_export . . . 42
yupana_import . . . 44
yupana_mvr . . . 45
yupana_reshape . . . 46

Index **48**

`design_augmented` *Experimental design: Augmented*

Description

Fieldbook generator for augmented experimental designs. Every check occurs once in each block and every test entry occurs once in the complete design.

Usage

```
design_augmented(
  checks,
  entries,
  blocks = NULL,
  eu_block = NULL,
  random = TRUE,
  zigzag = FALSE,
  dim = NA,
  serie = 1000,
  seed = NULL,
  project = "inkaverse",
  qrcode = "{project}{plots}{entry}",
  separate_checks = TRUE
)
```

Arguments

`checks`	Vector of check treatments.
`entries`	Vector of new or test entries.
`blocks`	Optional number of blocks. If `NULL`, it is calculated from `entries`, `checks` and `eu_block`.
`eu_block`	Number of experimental units per block. It must be greater than the number of checks.
`random`	Logical. Randomize test-entry allocation and positions inside each block.
`zigzag`	Logical. Arrange the physical field layout in zigzag order.
`dim`	Optional physical layout dimensions `c(nrows, ncols)`. The product must equal the total number of experimental units.
`serie`	Base number used to generate plot identifiers.
`seed`	Random seed. `0`, `NA` or `NULL` means that no fixed seed is set inside this function. TARPUY stores an effective seed in the design sheet before calling the design generator.
`project`	Barcode prefix.
`qrcode`	QR-code column template. The default is `"{project}{plots}{entry}"`.
`separate_checks`	Logical. When enough positions are available, place checks in non-adjacent positions inside each block. With `random = FALSE`, this placement is deterministic.

Value

A list with `fieldbook` and `parameters`.

`design_noreps` *Experimental design without replications*

Description

Function to deploy field-book experiment without replications

Usage

```
design_noreps(
  factors,
  type = "sorted",
  zigzag = FALSE,
  nrows = NA,
  serie = 1000,
  seed = NULL,
  project = "inkaverse",
  qrcode = "{project}{plots}"
)
```

Arguments

`factors`	Lists with names and factor vector [`list`].
`type`	Randomization in the list [`character: "sorted", "unsorted"`]
`zigzag`	Experiment layout in zigzag [`logic: FALSE`].
`nrows`	Experimental design dimension by rows [`numeric: value`]
`serie`	Number to start the plot id [`numeric: 1000`].
`seed`	Replicability from randomization [`numeric: NULL`].
`project`	Bar code prefix for data collection [`character: "inkaverse"`].
`qrcode`	Concatenate the QR code [`character: "{project}{plots}{factors}"`]

Value

A list with the field-book design and parameters

Examples

```
## Not run:

library(inti)

factores <- list("geno" = c(1:99))

fb <- design_noreps(factors = factores
                    , type = "sorted"
                    , zigzag = F
                    , nrows = 10
                    )

dsg <- fb$fieldbook

fb %>%
  tarpuy_plotdesign(fill = "plots")

fb$parameters


## End(Not run)
```

`design_repblock` *Experimental design in CRD and RCBD*

Description

Function to deploy field-book experiments for completely randomized designs (CRD/DCA) and randomized complete block designs (RCBD/DBCA).

Usage

```
design_repblock(
  nfactors = 1,
  factors,
  type = "crd",
  rep = 3,
  zigzag = FALSE,
  nrows = NA,
  serie = 1000,
  seed = NULL,
  project = "inkaverse",
  qrcode = "{project}{plots}"
)
```

Arguments

nfactors	Number of factors in the experiment [numeric: 1].
factors	Named list with the levels of each factor [list].
type	Type of experimental arrangement [character: "crd", "rcbd"]. The aliases "dca" and "dbca" are also accepted. The former "lsd" branch is intentionally disabled because it did not generate a valid Latin square design.
rep	Number of replications or blocks in the experiment [numeric: 3].
zigzag	Arrange the physical layout in zigzag order [logical: FALSE].
nrows	Number of rows in the physical field layout. When missing, the number of replications or blocks is used.
serie	Base number used to generate plot identifiers [numeric: 1000].
seed	Seed used for reproducible randomization [numeric: NULL].
project	Barcode prefix for data collection [character: "inkaverse"].
qrcode	Template used to concatenate QR-code fields [character: "{project}{plots}"]. The placeholder {factors} expands to all factor columns.

Value

A list with the field-book design and parameters.

Examples

```
## Not run:

library(inti)

factores <- list(
  "geno" = c("A", "B", "C", "D", "D", 1, NA, NULL, "NA"),
  "salt stress" = c(0, 50, 200, 200, "T0", NA, NULL, "NULL"),
  "time" = c(30, 60, 90)
)

fb <- design_repblock(
  nfactors = 3,
  factors = factores,
  type = "rcbd",
  rep = 5,
  zigzag = TRUE,
  seed = 123,
  nrows = 5,
  qrcode = "{project}{plots}"
)

dsg <- fb$fieldbook

fb %>%
  tarpuy_plotdesign(fill = "plots")
```

```
fb$parameters


## End(Not run)
```

`design_split` *Split-plot experimental designs*

Description

Dispatch split-plot experimental designs.

Usage

```
design_split(
  nfactors = 2,
  factors,
  type = "split-rcbd",
  rep = 3,
  zigzag = FALSE,
  nrows = NA,
  serie = 1000,
  seed = NULL,
  project = "inkaverse",
  qrcode = "{project}{plots}{factors}"
)
```

Arguments

`nfactors`	Number of factors in the experiment.
`factors`	List with factor levels.
`type`	Split-plot design type.
`rep`	Number of replications.
`zigzag`	Field layout in zigzag.
`nrows`	Experimental design dimension by rows.
`serie`	Number to start the plot id.
`seed`	Seed for randomization.
`project`	Barcode prefix.
`qrcode`	String to concatenate the QR code.

Value

A list with the fieldbook design and parameters.

`design_split_rcbd` *Split-plot RCBD experimental design*

Description

Generate a split-plot design under a randomized complete block design (RCBD) structure for TARPUY.

Usage

```
design_split_rcbd(
  nfactors = 2,
  factors,
  type = "split-rcbd",
  rep = 3,
  zigzag = FALSE,
  nrows = NA,
  serie = 1000,
  seed = NULL,
  project = "inkaverse",
  qrcode = "{project}{plots}{factors}"
)
```

Arguments

`nfactors`	Number of factors in the experiment. Splitplot-RCBD requires exactly two factors.
`factors`	Named list with the factor levels. The first factor is the whole-plot factor and the second factor is the subplot factor.
`type`	Design type. The canonical value is `"split-rcbd"`; accepted aliases are normalized by `normalize_tarpuy_design_type()`.
`rep`	Number of replications or blocks.
`zigzag`	Logical. If `TRUE`, plot numbering follows a continuous vertical serpentine path through the whole plots and blocks.
`nrows`	Number of rows in the complete physical layout. The valid Splitplot-RCBD geometry is `rep * number_of_subplot_levels`; when missing, it is calculated automatically.
`serie`	Base number used to generate plot identifiers. For example, `serie = 1000` generates plots 1001, 1002, ... in block 1 and 2001, 2002, ... in block 2.
`seed`	Seed used for reproducible randomization. `NA` or `NULL` leaves the current random-number state unchanged.
`project`	Barcode or QR-code prefix.
`qrcode`	Template used to concatenate QR-code fields. The placeholder `{factors}` expands to both experimental factors.

Details

The first factor is the whole-plot factor and the second factor is the subplot factor. Whole plots are randomized within each block and subplot levels are randomized independently within every whole plot.

Value

A list with `fieldbook` and `parameters`.

Examples

```
## Not run:

factors <- list(
  Soil = c("S1", "S2", "S3", "S4"),
  Fertilizer = c("N1", "N2", "N3", "N4", "N5", "N6")
)

design_split_rcbd(
  factors = factors,
  rep = 3,
  zigzag = TRUE,
  seed = 123
)$fieldbook

## End(Not run)
```

`footnotes` *Footnotes in tables*

Description

Include tables footnotes and symbols for kables in pandoc format

Usage

```
footnotes(table, notes = NULL, label = "Note:", notation = "alphabet")
```

Arguments

`table`	Kable output in pandoc format.
`notes`	Footnotes for the table.
`label`	Label for start the footnote.
`notation`	Notation for the footnotes (default = "alphabet"). See details.

Details

You should use the pandoc format `kable(format = "pipe")`. You can add the footnote symbol using `{hypen}` in your table. `notation` could be use: "alphabet", "number", "symbol", "none".

Value

Table with footnotes for word and html documents

`H2cal` *Broad-sense heritability in plant breeding*

Description

Heritability in plant breeding on a genotype difference basis

Usage

```
H2cal(
  data,
  trait,
  gen.name,
  rep.n,
  env.n = 1,
  year.n = 1,
  env.name = NULL,
  year.name = NULL,
  fixed.model,
  random.model,
  summary = FALSE,
  emmeans = FALSE,
  weights = NULL,
  plot_diag = FALSE,
  outliers.rm = FALSE,
  trial = NULL
)
```

Arguments

`data`	Experimental design data frame with the factors and traits.
`trait`	Name of the trait.
`gen.name`	Name of the genotypes.
`rep.n`	Number of replications in the experiment.
`env.n`	Number of environments (default = 1). See details.
`year.n`	Number of years (default = 1). See details.
`env.name`	Name of the environments (default = NULL). See details.
`year.name`	Name of the years (default = NULL). See details.
`fixed.model`	The fixed effects in the model (BLUEs). See examples.
`random.model`	The random effects in the model (BLUPs). See examples.

summary	Print summary from random model (default = FALSE).
emmeans	Use emmeans for calculate the BLUEs (default = FALSE).
weights	an optional vector of 'prior weights' to be used in the fitting process (default = NULL).
plot_diag	Show diagnostic plots using ggplot2 and cowplot (default = FALSE).
outliers.rm	Remove outliers (default = FALSE). See references.
trial	Column with the name of the trial in the results (default = NULL).

Details

The function allows to made the calculation for individual or multi-environmental trials (MET) using fixed and random model.

1. The variance components based in the random model and the population summary information based in the fixed model (BLUEs).
2. Heritability under three approaches: Standard (ANOVA), Cullis (BLUPs) and Piepho (BLUEs).
3. Best Linear Unbiased Estimators (BLUEs), fixed effect.
4. Best Linear Unbiased Predictors (BLUPs), random effect.
5. Table with the outliers removed for each model.

For individual experiments is necessary provide the `{trait}`, `{gen.name}`, `{rep.n}`.

For MET experiments you should `{env.n}` and `{env.name}` and/or `{year.n}` and `{year.name}` according your experiment.

The BLUEs calculation based in the pairwise comparison could be time consuming with the increase of the number of the genotypes. You can specify `{emmeans = FALSE}` and the calculate of the BLUEs will be faster.

If `{emmeans = FALSE}` you should change 1 by 0 in the fixed model for exclude the intersect in the analysis and get all the genotypes BLUEs.

Diagnostic plots are produced with ggplot2 and assembled with cowplot. If `{outliers.rm = FALSE}`, fixed and random model diagnostics are displayed in one combined panel. If `{outliers.rm = TRUE}`, fixed and random diagnostics are displayed in separate panels comparing before and after cleaning.

For more information review the references.

Value

list

Author(s)

Maria Belen Kistner

Flavio Lozano Isla

References

Bernal Vasquez, Angela Maria, et al. "Outlier Detection Methods for Generalized Lattices: A Case Study on the Transition from ANOVA to REML." Theoretical and Applied Genetics, vol. 129, no. 4, Apr. 2016.

Buntaran, H., Piepho, H., Schmidt, P., Ryden, J., Halling, M., and Forkman, J. (2020). Cross validation of stagewise mixed model analysis of Swedish variety trials with winter wheat and spring barley. Crop Science, 60(5).

Schmidt, P., J. Hartung, J. Bennewitz, and H.P. Piepho. 2019. Heritability in Plant Breeding on a Genotype Difference Basis. Genetics 212(4).

Schmidt, P., J. Hartung, J. Rath, and H.P. Piepho. 2019. Estimating Broad Sense Heritability with Unbalanced Data from Agricultural Cultivar Trials. Crop Science 59(2).

Tanaka, E., and Hui, F. K. C. (2019). Symbolic Formulae for Linear Mixed Models. In H. Nguyen (Ed.), Statistics and Data Science. Springer.

Zystro, J., Colley, M., and Dawson, J. (2018). Alternative Experimental Designs for Plant Breeding. In Plant Breeding Reviews. John Wiley and Sons, Ltd.

Examples

```
library(inti)

md <- met %>%
  H2cal(trait = "yield",
  gen.name = "cultivar",
  rep.n = 2,
  env.name = "env",
  env.n = 18,
  fixed.model = ~ 0 + env + (1 | env:rep:alpha) + cultivar,
  random.model = ~ 1 + env +
    (1 | env:rep) + (1 | env:rep:alpha) +
    (1 | cultivar:env) + (1 | cultivar),
  summary = TRUE,
  plot_diag = TRUE,
  outliers.rm = TRUE,
  emmeans = FALSE
)

 md$tabsmr
 md$blues
 md$blups
 md$outliers
```

`include_pdf` *Include PDF in markdown documents*

Description

Insert PDF files in markdown documents

Usage

```
include_pdf(file, width = "100%", height = "600")
```

Arguments

file	file path from pdf file.
width	width preview file.
height	height preview file.

Value

html code for markdown

---

include_table *Table with footnotes*

Description

Include tables with title and footnotes for word and html documents

Usage

```
include_table(table, caption = NA, notes = NA, label = NA, notation = "none")
```

Arguments

table	Data frame.
caption	Table caption (default = NULL). See details.
notes	Footnotes for the table (default = NA). See details.
label	Label for start the footnote (default = NA).
notation	Notation for the symbols and footnotes (default = "none") Others: "alphabet", "number", "symbol".

Value

Table with caption and footnotes

Examples

```
library(inti)

table <- data.frame(
x = rep_len(1, 5)
, y = rep_len(3, 5)
, z = rep_len("c", 5)
)

table %>% inti::include_table(
  caption = "Title caption b) line 0
  a) line 1
  b) line 2"
  , notes = "Footnote"
  , label = "Where:"
  )
```

`mean_comparison` *Mean comparison test*

Description

Function to compare treatment from lm or aov using data frames

Usage

```
mean_comparison(
  data,
  response,
  model_factors,
  comparison,
  test_comp = "SNK",
  sig_level = 0.05
)
```

Arguments

`data`	Fieldbook data.
`response`	Model used for the experimental design.
`model_factors`	Factor in the model.
`comparison`	Significance level for the analysis (default = 0.05).
`test_comp`	Comparison test (default = "SNK"). Others: "TUKEY", "DUNCAN".
`sig_level`	Significance level for the analysis (default = 0.05).

Value

list

Examples

```
## Not run:

library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/"
              , "15r7ZwcZZHbEgltlF6gSFvCTFA-CFzVBWwg3mFlRyKPs/"
              , "edit#gid=172957346")
# browseURL(url)

fb <- gsheet2tbl(url)

mc <- mean_comparison(data = fb
                      , response = "spad_29"
                      , model_factors = "bloque* geno*treat"
                      , comparison = c("geno", "treat")
                      , test_comp = "SNK"
                      )
mc$comparison
mc$stat


## End(Not run)
```

---

met *Swedish cultivar trial data*

Description

The datasets were obtained from official Swedish cultivar tests. Dry matter yield was analyzed. All trials were laid out as alpha-designs with two replicates. Within each replicate, there were five to seven incomplete blocks.

Usage

```
met
```

Format

A data frame with 1069 rows and 8 variables:

zone Sweden is divided into three different agricultural zones: South, Middle, and North

location Locations: 18 location in the Zones

rep Replications (4): number of replication in the experiment

alpha Incomplete blocks (8) in the alpha-designs

cultivar Cultivars (30): genotypes evaluated

yield Yield in kg/ha

year Year (1): 2016

env enviroment (18): combination zone + location + year

Source

doi:10.1002/csc2.20177

`metamorphosis` *Transform fieldbooks based in a dictionary*

Description

Transform entire fieldbook according to data a dictionary

Usage

```
metamorphosis(fieldbook, dictionary, from, to, index, colnames)
```

Arguments

`fieldbook`	Data frame with the original information.
`dictionary`	Data frame with new names and categories. See details.
`from`	Column of the dictionary with the original names.
`to`	Column of the dictionary with the new names.
`index`	Column of the dictionary with the type and level of the variables.
`colnames`	Character vector with the name of the columns.

Details

The function require at least three columns.

1. Original names (`from`).
2. New names (`to`).
3. Variable type (`index`).

Value

List with two objects. 1. New data frame. 2. Dictionary.

plot_augmented_design *Plot augmented fieldbook design*

Description

Plot fieldbook sketches for augmented experimental designs generated by `design_augmented()`.

Usage

```
plot_augmented_design(
  data,
  factor = NA,
  fill = "plots",
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  text_size = NULL,
  wrap_width = NULL,
  font_family = "Open Sans",
  font_face = "plain"
)
```

Arguments

`data`	Fieldbook data frame from an augmented design.
`factor`	Character scalar. Column used to color experimental units. If missing, `"type"` is used.
`fill`	Character vector. Column or columns used as labels inside each experimental unit. Default is `"plots"`. When `ntreat` is selected, it is displayed as `T1`, `T2`, etc.
`xlab`	Character scalar. Optional x axis title. If `NULL`, `"Columns"` is used.
`ylab`	Character scalar. Optional y axis title. If `NULL`, the title is `"Blocks"` when every physical row corresponds to exactly one statistical block; otherwise `"Rows"` is used.
`glab`	Character scalar. Optional legend title. If `NULL`, the selected color factor is used.
`text_size`	Optional positive numeric scalar indicating the plot-label font size in typographic points (`pt`). If `NULL` or `NA`, a suitable default is selected according to the number of label columns. The value is converted internally to the unit expected by `ggplot2::geom_text()`.
`wrap_width`	Optional positive integer indicating the approximate maximum number of characters per line. If `NULL` or `NA`, the function calculates it automatically from the field dimensions, font size and number of label columns. Underscores are displayed as spaces only in the sketch; the original fieldbook values are not modified.

font_family	Character scalar. Font family used in the sketch. Defaults to "Open Sans". If the font cannot be verified through the optional systemfonts package, "sans" is used as a fallback.
font_face	Character scalar. Font face used in labels, axes and legends. Defaults to "plain".

Details

The function always uses the physical rows and cols coordinates stored in the fieldbook. Therefore, custom dimensions and zigzag layouts are preserved. The block column never replaces rows as a plotting coordinate. When each row represents exactly one block, only the visible y-axis title changes from "Rows" to "Blocks". The statistical block remains available as a color factor and, when a block occupies a complete rectangular region, its external border is highlighted.

Empty experimental units are represented by the level "empty". When factor = "type", checks, test entries and empty plots receive stable colors. Other factor columns use the regular TARPUY color palette.

Automatic wrapping and formatting affect only the displayed text. They do not modify entry, plots, ntreat, QR codes or any other fieldbook value.

Value

A ggplot object.

Examples

```
## Not run:

plot_augmented_design(
  data = fieldbook,
  factor = "type",
  fill = c("plots", "entry"),
  text_size = 9,
  font_family = "Open Sans",
  font_face = "plain"
)


## End(Not run)
```

---

plot_diagnostic *Diagnostic plots*

Description

Function to plot the diagnostic of models

Usage

```
plot_diagnostic(data, formula, title = NA)
```

Arguments

data	Experimental design data frame with the factors and traits.
formula	Mixed model formula
title	Plot title

Value

plots

Examples

```
## Not run:

library(inti)

plot_diagnostic(data = potato
                , formula = stemdw ~ (1|bloque) + geno*treat)


## End(Not run)
```

---

plot_raw *Plot raw data*

Description

Function use the raw data for made a boxplot graphic

Usage

```
plot_raw(
  data,
  type = "boxplot",
  x,
  y,
  group = NULL,
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  ylimits = NULL,
  xlimits = NULL,
  xrotation = NULL,
  legend = "top",
  xtext = NULL,
  gtext = NULL,
  color = TRUE,
```

```
  linetype = 1,
  opt = NULL
)
```

Arguments

data	raw data
type	Type of graphic. "boxplot" or "scatterplot"
x	Axis x variable
y	Axis y variable
group	Group variable
xlab	Title for the axis x
ylab	Title for the axis y
glab	Title for the legend
ylimits	Limits and break of the y axis c(initial, end, brakes)
xlimits	For scatter plot. Limits and break of the x axis c(initial, end, brakes)
xrotation	Rotation in x axis c(angle, h, v)
legend	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
xtext	Text labels in x axis using a vector
gtext	Text labels in groups using a vector
color	Colored figure (TRUE), black & white (FALSE) or color vector
linetype	Line type for regression. Default = 0
opt	Add new layers to the plot

Details

You could add additional layer to the plot using "+" with ggplot2 options

Value

plot

Examples

```
## Not run:

library(inti)

fb <- potato

fb %>%
  plot_raw(type = "box"
           , x = "geno"
           , y = "twue"
```

```
            #, group = "treat"
            , ylab = NULL
            , xlab = NULL
            , glab = ""
            )

fb %>%
  plot_raw(type = "sca"
            , x = "hi"
            , y = "twue"
            , group = "geno"
            )


## End(Not run)
```

---

plot_smr *Plot summary data*

Description

Graph summary data into bar o line plot

Usage

```
plot_smr(
  data,
  type = NULL,
  x = NULL,
  y = NULL,
  group = NULL,
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  ylimits = NULL,
  xrotation = c(0, 0.5, 0.5),
  xtext = NULL,
  gtext = NULL,
  legend = "top",
  sig = NULL,
  sigsize = 3,
  error = NULL,
  color = TRUE,
  opt = NULL
)
```

Arguments

data	Output from summary data
type	Type of graphic. "bar" or "line"
x	Axis x variable
y	Axis y variable
group	Group variable
xlab	Title for the axis x
ylab	Title for the axis y
glab	Title for the legend
ylimits	limits of the y axis c(initial, end, brakes)
xrotation	Rotation in x axis c(angle, h, v)
xtext	Text labels in x axis using a vector
gtext	Text labels in group using a vector
legend	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
sig	Column with the significance
sigsize	Font size in significance letters
error	Show the error bar ("ste" or "std")
color	colored figure (TRUE), black & white (FALSE) or color vector
opt	Add news layer to the plot

Details

If the table is a out put of `mean_comparison(graph_opts = TRUE)` function. Its contain all the parameter for the plot.

You could add additional layer to the plot using "+" with ggplot2 options

Value

plot

Examples

```
## Not run:

library(inti)

fb <- potato

yrs <- yupana_analysis(data = fb
                       , response = "hi"
                       , model_factors = "geno*treat"
                       , comparison = c("geno", "treat")
                       )
```

```
yrs$meancomp %>%
  plot_smr(type = "bar"
           , x = "geno"
           , y = "hi"
           , xlab = ""
           , group = "treat"
           , glab = "Tratamientos"
           , error = "ste"
           , sig = "sig"
           #, ylimits = c(0, 1, 0.2)
           , color = T #c("red", "black")
           , gtext = c("Irrigado", "Sequia")
           )

yrs$meancomp %>%
  plot_smr(type = "bar"
           , x = "treat"
           , y = "hi"
           , group = "geno"
           , glab = "Genotipo"
           , error = "ste"
           , sig = "sig"
           )


## End(Not run)
```

`plot_split_rcbd_design` *Plot Splitplot-RCBD fieldbook design*

Description

Plot fieldbook sketches for Splitplot-RCBD experimental designs generated by `design_split_rcbd()`.

Usage

```
plot_split_rcbd_design(
  data,
  factor = NA,
  fill = "plots",
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  text_size = NULL,
  wrap_width = NULL,
  font_family = "Open Sans",
```

```
  font_face = "plain"
)
```

Arguments

data	Fieldbook data frame from design_split_rcbd().
factor	Character scalar. Column used to color experimental units. If missing, "wp_sp" is used when available; otherwise, "ntreat" is used.
fill	Character vector. Column or columns used as labels inside each experimental unit. Default is "plots".
xlab	Character scalar. Optional x-axis title. If NULL, "Whole plots" is used.
ylab	Character scalar. Optional y-axis title. If NULL, "Subplots" is used when rows can be interpreted consistently as subplot positions within blocks; otherwise, "Rows" is used.
glab	Character scalar. Optional legend title.
text_size	Optional positive numeric scalar indicating the plot-label font size in typographic points (pt). If NULL or NA, a suitable default is selected according to the number of label columns. The value is converted internally to the unit expected by ggplot2::geom_text().
wrap_width	Optional positive integer indicating the approximate maximum number of characters per line. If NULL or NA, the function calculates it automatically from the dimensions of each block, font size and number of label columns. Underscores are displayed as spaces only in the sketch; the fieldbook values are not modified.
font_family	Character scalar. Font family used in the sketch. Defaults to "Open Sans". If the font cannot be verified through the optional systemfonts package, "sans" is used as a fallback.
font_face	Character scalar. Font face used in labels, axes, facet titles and legends. Defaults to "plain".

Details

The function does not recalculate or rearrange the experimental design. It uses the existing rows and cols coordinates from the fieldbook, so the physical arrangement and zigzag order generated by Tarpuy are preserved.

Every experimental unit is positioned with cols on the x axis and rows on the y axis. The block column is used only to identify the block panels; it never replaces the physical row coordinate.

Within each block, cols represents whole-plot positions. The visible y-axis tick labels are derived from the rank of the actual rows values inside each block, so the panels can display subplot positions while the geometry continues to use the original fieldbook coordinates. A black external border identifies each whole plot.

Automatic wrapping affects only the displayed labels. It does not change plots, ntreat, treatment factors, QR codes, rows, columns or any other fieldbook value.

Value

A ggplot object.

Examples

```
## Not run:

plot_split_rcbd_design(
  data = fieldbook,
  factor = "wp_sp",
  fill = c("plots", "ntreat"),
  text_size = 9,
  font_family = "Open Sans",
  font_face = "plain"
)


## End(Not run)
```

---

plot_standard_design *Plot standard fieldbook experimental designs*

Description

Plot standard fieldbook sketches for regular experimental designs generated in Tarpuy, including completely randomized designs (CRD/DCA) and randomized complete block designs (RCBD/DBCA).

Usage

```
plot_standard_design(
  data,
  factor = NA,
  fill = "plots",
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  text_size = NULL,
  wrap_width = NULL,
  font_family = "Open Sans",
  font_face = "plain"
)
```

Arguments

data	A fieldbook data frame. It must contain at least rows and cols.
factor	Character scalar. Name of the column used to color the experimental units. If missing, "block" is used when available; otherwise, the third column of data is used.
fill	Character vector. Names of one or more columns used as labels inside each experimental unit. When ntreat is used, it is displayed as T1, T2, etc.
xlab	Character scalar. Title for the x axis. If NULL, "Columns" is used.

ylab	Character scalar. Title for the y axis. If NULL, "Blocks" is shown for RCBD/DBCA only when each physical row corresponds to exactly one block; otherwise "Rows" is used. The plotting coordinate always remains the physical rows column.
glab	Character scalar. Legend title. If NULL, factor is used.
text_size	Optional positive numeric scalar indicating the plot-label font size in typographic points (pt). If NULL or NA, a suitable default is selected according to the number of label columns. The value is converted internally to the unit expected by ggplot2::geom_text().
wrap_width	Optional positive integer indicating the approximate maximum number of characters per line. If NULL or NA, the function calculates it automatically from the field dimensions, font size and number of label columns. Underscores are shown as spaces only in the plotted label; the original fieldbook values are not modified.
font_family	Character scalar. Font family used by the sketch. Defaults to "Open Sans". If the font cannot be verified through the optional systemfonts package, "sans" is used as a fallback.
font_face	Character scalar. Font face used in labels, axes and legends. Defaults to "plain".

Details

The function does not calculate or rearrange an experimental design. It uses the existing rows and cols coordinates from the fieldbook, so the physical layout and the zigzag order generated by Tarpuy are preserved.

All standard designs are plotted with cols on the x axis and rows on the y axis. A DBCA/RCBD is therefore displayed according to its actual physical coordinates rather than replacing the row coordinate with the block number. When rows and blocks have a strict one-to-one relationship, only the visible y-axis title changes to "Blocks". The block can still be selected as the color factor.

Automatic wrapping affects only the sketch. It does not change entry, ntreat, QR codes, factor levels or any other fieldbook value.

Value

A ggplot object.

Examples

```
## Not run:

plot_standard_design(
  data = fieldbook,
  factor = "geno",
  fill = c("plots", "entry"),
  text_size = 9,
  font_family = "Open Sans",
  font_face = "plain"
)


## End(Not run)
```

potato *Water use efficiency in 15 potato genotypes*

Description

Experiment to evaluate the physiological response from 15 potatos genotypes under water deficit condition. The experiment had a randomized complete block design with five replications. The stress started at 30 day after planting.

Usage

```
potato
```

Format

A data frame with 150 rows and 17 variables:

treat Water deficit treatments: sequia, irrigado

geno 15 potato genotypes

bloque blocks for the experimentl design

spad_29 Relative chlorophyll content (SPAD) at 29 day after planting

spad_83 Relative chlorophyll content (SPAD) at 84 day after planting

rwc_84 Relative water content (percentage) at 84 day after planting

op_84 Osmotic potential (Mpa) at 84 day after planting

leafdw leaf dry weight (g)

stemdw stem dry weight (g)

rootdw root dry weight (g)

tubdw tuber dry weight (g)

biomdw total biomass dry weight (g)

hi harvest index

ttrans total transpiration (l)

wue water use effiency (g/l)

twue tuber water use effiency (g/l)

lfa leaf area (cm2)

`remove_outliers` *Remove outliers using mixed models*

Description

Use the method M4 in Bernal Vasquez (2016). Bonferroni Holm test to judge residuals standardized by the re scaled MAD (BH MADR).

Usage

```
remove_outliers(
  data,
  formula,
  drop_na = FALSE,
  plot_diag = FALSE,
  title = "Diagnostic Plot"
)
```

Arguments

`data`	Experimental design data frame with the factors and traits.
`formula`	mixed model formula.
`drop_na`	drop NA values from the data.frame
`plot_diag`	Diagnostic plot based in the raw and clean data
`title`	Plot title

Details

Function to remove outliers in MET experiments

Value

list. 1. Table with date without outliers. 2. The outliers in the dataset.

References

Bernal Vasquez, Angela Maria, et al. “Outlier Detection Methods for Generalized Lattices: A Case Study on the Transition from ANOVA to REML.” Theoretical and Applied Genetics, vol. 129, no. 4, Apr. 2016.

Examples

```
library(inti)

rmout <- potato %>%
  remove_outliers(data = .
  , formula = stemdw ~ 0 + (1|bloque) + treat*geno
```

```
  , plot_diag = TRUE
  , drop_na = FALSE
  , title = "Plot Diagnostic"
  )

rmout
```

`rticle` *Convert a Markdown manuscript into a Quarto document*

Description

Reads a Markdown file exported from Google Docs and converts it into a Quarto-compatible manuscript. The function detects and separates text, figures, and tables, removes formatting artifacts, restores equations embedded as image placeholders, and inserts format-specific section breaks. The resulting document can be rendered directly using Quarto for HTML, Word, or PDF outputs.

Usage

```
rticle(file = "render2rticle.md", export = "files", type = c("asis", "list"))
```

Arguments

`file`	Character string indicating the path to the Markdown file. Default is "draft.md".
`export`	Character string specifying the output directory where the generated .qmd file will be saved. If NULL, the directory name is derived from the input file name.
`type`	Character string indicating how the manuscript should be organized. "asis" preserves the original structure, whereas "list" rearranges the content into text, figures, and tables.

Value

A character vector containing the full path of the generated Quarto (.qmd) file.

`split_folder` *Split folder*

Description

Function to split folder by size or number of elements

Usage

```
split_folder(
  folder,
  export,
  units = "megas",
  size = 500,
  zip = TRUE,
  remove = FALSE
)
```

Arguments

folder	Path of folder to split (path).
export	Path to export the split folders (path).
units	Units to split folder (string: "megas", "number").
size	Folder size by the units selected (numeric).
zip	Zip split folders (logical).
remove	Remove the split folder after zip (logical).

Value

zip files

Examples

```
## Not run:

split_folder("pictures/QUINOA 2018-2019 SC SEEDS EDWIN - CAMACANI/"
   , "pictures/split_num", remove = T, size = 400, units = "number")


## End(Not run)
```

---

tarpuy *Interactive fieldbook designs*

Description

Invoke RStudio addin to create fieldbook designs

Usage

```
tarpuy(dependencies = FALSE)
```

Arguments

dependencies	Logical. If TRUE, install any missing TARPUY runtime dependencies using the local inst/tarpuy/setup.R script included with the installed inti package.

Details

TARPUY allows users to create experimental designs through an interactive Shiny application.

Value

A Shiny application launched with shiny::runApp().

Examples

```
if (interactive()) {
  inti::tarpuy()
}
```

---

tarpuy_design *Fieldbook experimental designs*

Description

Function to deploy the experimental designs currently supported by TARPUY.

Usage

```
tarpuy_design(
  data,
  nfactors = 1,
  type = "crd",
  rep = 2,
  zigzag = FALSE,
  nrows = NA,
  serie = 100,
  seed = NULL,
  project = NA,
  qrcode = "{project}{plots}"
)
```

Arguments

data	Experimental design data frame containing factor names and levels. A design sheet may also include the columns {arguments} and {values} to override the function arguments.
nfactors	Number of factors in the experiment [default = 1].

type	Type of experimental arrangement [default = "crd"]. Supported designs are "crd", "rcbd", "augmented", and "split-rcbd". The aliases "dca" and "dbca" are accepted.
rep	Number of replications or blocks in the experiment [default = 2].
zigzag	Arrange the physical layout in zigzag order [logical: FALSE].
nrows	Number of rows in the physical field layout. When missing, the corresponding design function calculates the layout.
serie	Base number used to generate plot identifiers [numeric: 100].
seed	Seed used for reproducible randomization. 0, NA, and NULL preserve the historical TARPUY behavior of using a random seed.
project	Barcode prefix for data collection.
qrcode	Template used to concatenate QR-code fields [character: "{project}{plots}"].

Details

The design sheet can include two optional columns named {arguments} and {values}. Values supplied in those columns override the corresponding function arguments. Factor columns are the remaining columns whose names are not enclosed in braces ({}) or square brackets ([]).

TARPUY currently dispatches only designs with an implemented and validated generator: CRD/DCA, RCBD/DBCA, augmented, and split-plot RCBD. Other design identifiers are rejected explicitly instead of being routed to incomplete generators.

Value

A data frame containing the generated fieldbook.

Examples

```
## Not run:

library(inti)
library(gsheet)

url <- paste0(
  "https://docs.google.com/spreadsheets/d/",
  "1510fOKj0g4CDEAFkrpFbr-zNMnle_Hou9O_wuf7Vdo4/edit"
)

fb <- gsheet2tbl(url)

dsg <- fb %>% tarpuy_design()

dsg %>% tarpuy_plotdesign()


## End(Not run)
```

tarpuy_plex *Fieldbook plan information*

Description

Information for build a plan for an experiment (PLEX)

Usage

```
tarpuy_plex(
  data = NULL,
  title = NULL,
  short_title = NULL,
  objective = NULL,
  references = NULL,
  plan = NULL,
  institutions = NULL,
  researchers = NULL,
  manager = NULL,
  location = NULL,
  altitude = NULL,
  georeferencing = NULL,
  environment = NULL,
  start = NA,
  end = NA,
  project = NULL,
  repository = NULL,
  manuscript = NULL,
  album = NULL,
  nfactor = 2,
  design = "rcbd",
  rep = 4,
  zigzag = FALSE,
  nrows = NA,
  serie = 1000,
  seed = 0,
  qrcode = "{project}{plots}",
  aug_blocks = NA,
  aug_eu_block = NA,
  aug_random = TRUE
)
```

Arguments

data	Data with the fieldbook information.
title	Project title.

short_title	Short description of the project.
objective	The objectives of the project.
references	References.
plan	General description of the project (M & M).
institutions	Institutions involved in the project.
researchers	Persons involved in the project.
manager	Persons responsible of the collection of the data.
location	Location of the project.
altitude	Altitude of the experiment (m.a.s.l).
georeferencing	Georeferencing information.
environment	Environment of the experiment (greenhouse, lab, etc).
start	The date of the start of the experiments.
end	The date of the end of the experiments.
project	Name or ID for the fieldbook/project.
repository	link to the repository.
manuscript	link for manuscript.
album	link with the photos of the project.
nfactor	Number of factors for the design.
design	Type of design.
rep	Number of replication.
zigzag	Experiment layout in zigzag `[logic: FALSE]`
nrows	Experimental design dimension by rows `[numeric: value]`
serie	Number of digits in the plots.
seed	Seed for the randomization. A value of `0`, `NA`, or `NULL` generates and stores an effective seed in the design sheet.
qrcode	QR code template used to concatenate fieldbook identifiers.
aug_blocks	Optional number of blocks for augmented design.
aug_eu_block	Number of plots per block for augmented design.
aug_random	Logical. Randomize entries allocation in augmented design.

Details

Provide the information available.

Value

data frame or list of arguments:

1. info
2. variables

3. design
4. logbook
5. timetable
6. budget

`tarpuy_plotdesign` *Fieldbook plot experimental designs*

Description

Plot fieldbook sketches according to the experimental design type.

Usage

```
tarpuy_plotdesign(
  data,
  factor = NA,
  fill = "plots",
  xlab = NULL,
  ylab = NULL,
  glab = NULL,
  text_size = NULL,
  wrap_width = NULL,
  font_family = "Open Sans",
  font_face = "plain"
)
```

Arguments

`data`	Fieldbook data frame or a design object containing a fieldbook.
`factor`	Character scalar. Column used to color experimental units. When omitted, `NA`, an empty string or `"auto"`, TARPUY selects a design- appropriate default: the first experimental factor for CRD, RCBD and Splitplot-RCBD, and `type` for augmented designs.
`fill`	Character vector. Column or columns used as labels inside the experimental units. Defaults to `"plots"`.
`xlab`	Character scalar. Optional x-axis title. When `NULL`, each plotter determines the title from the design geometry.
`ylab`	Character scalar. Optional y-axis title. When `NULL`, each plotter determines whether the physical `rows` represent rows, blocks or subplot positions.
`glab`	Character scalar. Optional legend title.
`text_size`	Optional positive numeric scalar indicating the plot-label font size in typographic points (`pt`). When `NULL` or `NA`, each plotter calculates a suitable size automatically.

wrap_width	Optional positive integer retained for backward compatibility. When NULL or NA, label wrapping is calculated automatically by the selected plotter. This argument does not need to be exposed as a control in the TARPUY interface.
font_family	Character scalar retained for programmatic use. Defaults to "Open Sans"; the plotters fall back to "sans" when necessary.
font_face	Character scalar retained for programmatic use. One of "plain", "bold", "italic" or "bold.italic".

Details

This function is the common plotting interface used by TARPUY. It reads the design stored in the fieldbook, chooses the corresponding plotter and sends the same plotting arguments to all supported design types.

Supported designs and plotters are:

- CRD and RCBD: plot_standard_design().
- Augmented: plot_augmented_design().
- Splitplot-RCBD: plot_split_rcbd_design().

The function never recalculates or rearranges the design. All plotters use cols as the x coordinate and rows as the y coordinate. Columns such as block are used only for labels, grouping or faceting.

wrap_width, font_family and font_face remain available to avoid breaking existing programmatic calls, but the TARPUY frontend should expose only the general text_size control. Automatic label wrapping is used when wrap_width = NULL.

Value

A ggplot object.

Examples

```
## Not run:

tarpuy_plotdesign(
  data = fieldbook,
  factor = "auto",
  fill = c("plots", "ntreat"),
  text_size = 9
)


## End(Not run)
```

tarpuy_traits *Field book traits*

Description

Function to export a field book and its trait definitions for use in the Field Book app.

Usage

```
tarpuy_traits(fieldbook = NULL, last_factor = NULL, traits = NULL)
```

Arguments

fieldbook	Experimental field book [data.frame].
last_factor	Optional name of the last structural column to include in the CSV exported to Field Book [character: colname].
traits	Traits information [data.frame or list].

Details

The trait sheet can contain the columns variable, {trait}, {when}, {samples}, {format}, units, details, and categories.

Spaces inside {trait}, {when}, and the textual part of {samples} are removed. The components are joined with underscores. For example, {trait} = "G", {when} = "Dia 1", and {samples} = "plant3" generate G_Dia1_plant1, G_Dia1_plant2, and G_Dia1_plant3.

Value

A list with four elements:

- fieldbook: the field book including empty trait columns;
- traits: the trait definition table used to export the .trt file;
- fb: the base field book used to export the Field Book CSV;
- metadata: internal mapping between a stable Trait ID and every generated fieldbook column. This element is used by TARPUY and is not exported to the fieldbook or to Field Book mobile files.

Examples

```
library(inti)

fieldbook <- inti::potato

traits <- list(
  list(
    variable = "altura de planta",
    trait = "altp",
```

```
    format = "numeric",
    when = "Dia 30, Dia 40, Dia 50",
    samples = "plant3",
    units = "cm",
    details = NA,
    minimum = 0,
    maximum = 100
  ),
  list(
    variable = "severidad",
    trait = "svr",
    format = "scategorical",
    when = "30, 40, 50",
    samples = 1,
    units = "scale",
    details = NA,
    categories = "1, 3, 5, 7, 9"
  ),
  list(
    variable = "foto",
    trait = "foto",
    format = "photo",
    when = "hrv, pshrv",
    samples = NA,
    units = "image",
    details = NA
  ),
  list(
    variable = "germinacion",
    trait = "G",
    format = "boolean",
    when = "0, 1, 2",
    samples = 1,
    units = "logical",
    details = NA
  )
)

fbapp <- tarpuy_traits(fieldbook, last_factor = "bloque", traits)

## Not run:

library(inti)
library(gsheet)

url_ds <- paste0(
  "https://docs.google.com/spreadsheets/d/",
  "1510fOKj0g4CDEAFkrpFbr-zNMnle_Hou9O_wuf7Vdo4/edit?gid=1278145622"
)

ds <- gsheet2tbl(url_ds)
fb <- ds |> tarpuy_design()
```

```
url_trt <- paste0(
  "https://docs.google.com/spreadsheets/d/",
  "1510fOKj0g4CDEAFkrpFbr-zNMnle_Hou9O_wuf7Vdo4/edit?gid=1665653985"
)

traits <- gsheet2tbl(url_trt)
fbapp <- tarpuy_traits(fb, last_factor = "cols", traits)

## End(Not run)
```

`web_table` *HTML tables for markdown documents*

Description

Export tables with download, pasta and copy buttons

Usage

```
web_table(
  data,
  caption = NULL,
  digits = 2,
  rnames = FALSE,
  buttons = NULL,
  file_name = "file",
  scrolly = "45vh",
  columnwidth = "200px",
  width = "100%"
)
```

Arguments

`data`	Dataset.
`caption`	Title for the table.
`digits`	Digits number in the table exported.
`rnames`	Row names.
`buttons`	Buttons: "excel", "copy" or "none". Default c("excel", "copy")
`file_name`	Excel file name
`scrolly`	Windows height to show the table. Default "45vh"
`columnwidth`	Column width. Default '200px'
`width`	Width in pixels or percentage (Defaults to automatic sizing)

Value

table in markdown format for html documents

Examples

```
## Not run:

library(inti)

met %>%
  web_table(caption = "Web table")


## End(Not run)
```

---

yupana *Interactive data analysis*

Description

Invoke RStudio addin to analyze and graph experimental design data

Usage

```
yupana(dependencies = FALSE)
```

Arguments

dependencies	Install package dependencies for run the app

Details

Yupana: data analysis and graphics for experimental designs.

Value

Shiny app

Examples

```
if(interactive()){

 inti::yupana()

}
```

`yupana_analysis` *Fieldbook analysis report*

Description

Function to create a complete report of the fieldbook

Usage

```
yupana_analysis(
  data,
  last_factor = NULL,
  response,
  model_factors,
  comparison,
  test_comp = "SNK",
  sig_level = 0.05,
  plot_dist = "boxplot",
  plot_diag = FALSE,
  digits = 2
)
```

Arguments

`data`	Field book data.
`last_factor`	The last factor in your fieldbook.
`response`	Response variable.
`model_factors`	Model used for the experimental design.
`comparison`	Factors to compare
`test_comp`	Comprasison test c("SNK", "TUKEY", "DUNCAN")
`sig_level`	Significal test (default: p = 0.005)
`plot_dist`	Plot data distribution (default = "boxplot")
`plot_diag`	Diagnostic plots for model (default = FALSE).
`digits`	Digits number in the table exported.

Value

list

Examples

```
## Not run:

library(inti)

fb <- potato

rsl <- yupana_analysis(data = fb
                       , last_factor = "bloque"
                       , response = "spad_83"
                       , model_factors = "geno * treat"
                       , comparison = c("geno", "treat")
                       )


## End(Not run)
```

`yupana_export` *Graph options to export*

Description

Function to export the graph options and model parameters

Usage

```
yupana_export(
  data,
  type = NA,
  xlab = NA,
  ylab = NA,
  glab = NA,
  ylimits = NA,
  xrotation = c(0, 0.5, 0.5),
  xtext = NA,
  gtext = NA,
  legend = "top",
  sig = NA,
  error = NA,
  color = TRUE,
  opt = NA,
  dimension = c(20, 10, 100)
)
```

Arguments

data	Result from yupana_analysis or yupana_import.
type	Plot type
xlab	Title for the axis x
ylab	Title for the axis y
glab	Title for the legend
ylimits	limits of the y axis
xrotation	Rotation in x axis c(angle, h, v)
xtext	Text labels in x axis
gtext	Text labels in group
legend	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
sig	Column with the significance
error	Show the error bar ("ste" or "std").
color	colored figure (TRUE), otherwise black & white (FALSE)
opt	Add news layer to the plot
dimension	Dimension of graphs

Value

data frame

Examples

```
## Not run:

library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/"
              , "15r7ZwcZZHbEgltlF6gSFvCTFA-CFzVBWwg3mFlRyKPs/edit#gid=172957346")
# browseURL(url)

fb <- gsheet2tbl(url)

smr <- yupana_analysis(data = fb
                       , last_factor = "bloque"
                       , response = "spad_83"
                       , model_factors = "block + geno*riego"
                       , comparison = c("geno", "riego")
                       )

gtab <- yupana_export(smr, type = "line", ylimits = c(0, 100, 2))

#> import
```

```
url <- paste0("https://docs.google.com/spreadsheets/d/"
              , "15r7ZwcZZHbEgltlF6gSFvCTFA-CFzVBWwg3mFlRyKPs/edit#gid=1202800640")
# browseURL(url)

fb <- gsheet2tbl(url)

info <- yupana_import(fb)

etab <- yupana_export(info)

info2 <- yupana_import(etab)

etab2 <- yupana_export(info2)


## End(Not run)
```

---

yupana_import *Import information from data summary*

Description

Graph summary data

Usage

```
yupana_import(data)
```

Arguments

data	Summary information with options

Value

list

Examples

```
## Not run:

library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/"
         , "15r7ZwcZZHbEgltlF6gSFvCTFA-CFzVBWwg3mFlRyKPs/edit?gid=2137596914#gid=2137596914")
# browseURL(url)

fb <- gsheet2tbl(url)
```

```
info <- yupana_import(fb)


## End(Not run)
```

---

yupana_mvr *Multivariate Analysis*

Description

Multivariate analysis for PCA and HCPC

Usage

```
yupana_mvr(
  data,
  last_factor = NULL,
  summary_by = NULL,
  groups = NULL,
  variables = NULL
)
```

Arguments

data	Field book data.
last_factor	The last factor in your fieldbook [string: NULL].
summary_by	Variables for group the analysis.
groups	Groups for color in PCA.
variables	Variables to be use in the analysis [string: NULL].

Details

Compute and plot information for multivariate analysis (PCA, HCPC and correlation).

Value

result and plots

Examples

```
## Not run:

library(inti)

fb <- inti::potato
```

```
mv <- yupana_mvr(data = fb
                 , last_factor = "geno"
                 , summary_by = c("geno", "treat")
                 , groups = "treat"
                 , variables = c("all")
                 #, variables = c("wue", "twue")
                 )

mv$plot[1]

mv$data



## End(Not run)
```

---

yupana_reshape *Fieldbook reshape*

Description

Function to reshape fieldbook according a separation character

Usage

```
yupana_reshape(
  data,
  last_factor,
  sep,
  new_colname,
  from_var = NULL,
  to_var = NULL,
  exc_factors = NULL
)
```

Arguments

data	Field book raw data.
last_factor	The last factor in your field book.
sep	Character that separates the last value.
new_colname	The new name for the column created.
from_var	The first variable in case you want to exclude several. variables.
to_var	The last variable in case you want to exclude several variables.
exc_factors	Factor to exclude during the reshape.

Details

If you variable name is `variable_evaluation_rep`. The reshape function will help to create the column `rep` and the new variable name will be `variable_evaluation`.

Value

data frame

Index

∗ **datasets**
met, 15
potato, 27

design_augmented, 3
design_noreps, 4
design_repblock, 5
design_split, 7
design_split_rcbd, 8

footnotes, 9

H2cal, 10

include_pdf, 12
include_table, 13

mean_comparison, 14
met, 15
metamorphosis, 16

plot_augmented_design, 17
plot_diagnostic, 18
plot_raw, 19
plot_smr, 21
plot_split_rcbd_design, 23
plot_standard_design, 25
potato, 27

remove_outliers, 28
rticle, 29

shiny::runApp(), *31*
split_folder, 29

tarpuy, 30
tarpuy_design, 31
tarpuy_plex, 33
tarpuy_plotdesign, 35
tarpuy_traits, 37

web_table, 39

yupana, 40
yupana_analysis, 41
yupana_export, 42
yupana_import, 44
yupana_mvr, 45
yupana_reshape, 46