Package ‘inti’

February 22, 2022

Type Package

Version 0.5.4

Title Tools and Statistical Procedures in Plant Science

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 technical writing.

Learn more about the ‘inkaverse’ project at <https://inkaverse.com/>.

Date 2022-02-20


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

Depends shiny, ggplot2, dplyr, tidyr, tibble, R (>= 2.10)

Imports lme4, agricolae, FactoMineR, emmeans, purrr, stringr, stringi, DT

Suggests gsheet, knitr, rmarkdown, bookdown

VignetteBuilder knitr

License GPL-3 | file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Flavio Lozano-Isla [aut, cre] (<https://orcid.org/0000-0002-0714-669X>), QuipoLab [ctb], Inkaverse [eph]

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

Repository CRAN

Date/Publication 2022-02-22 09:10:08 UTC
R topics documented:

colortext .......................................................... 2
figure2rmd .......................................................... 3
footnotes ............................................................ 4
gdoc2rmd ............................................................ 4
H2cal ................................................................. 5
include_figure ......................................................... 7
include_table ......................................................... 8
je_tombola ........................................................... 9
mean_comparison ....................................................... 10
met ................................................................. 11
metamorphosis ....................................................... 12
outliers_remove ....................................................... 13
plot_diag ............................................................. 14
plot_raw .............................................................. 15
plot_smr .............................................................. 17
potato ............................................................... 19
table2rmd ............................................................. 20
tarpuy ............................................................... 20
tarpuy_design ......................................................... 21
tarpuy_plex ........................................................... 22
tarpuy_plotdesign ..................................................... 24
tarpuy_varlist ......................................................... 25
web_table .............................................................. 26
yupana ............................................................... 27
yupana_analysis ...................................................... 28
yupana_export ......................................................... 29
yupana_import ......................................................... 31
yupana_mvr ............................................................ 32
yupana_reshape ......................................................... 33

Index ............................................................... 35

| colortext | Colourise text for display in the terminal |

Description

If R is not currently running in a system that supports terminal colours the text will be returned unchanged.

Usage

    colortext(text, fg = "red", bg = NULL)
**Arguments**

- **text**: character vector
- **fg**: foreground colour, defaults to white
- **bg**: background colour, defaults to transparent

**Details**

Allowed colours are: black, blue, brown, cyan, dark gray, green, light blue, light cyan, light gray, light green, light purple, light red, purple, red, white, yellow

**Author(s)**

testthat package

**Examples**

```r
print(colortext("Red", "red"))
cat(colortext("Red", "red"), "\n")
cat(colortext("White on red", "white", "red"), "\n")
```

---

**figure2rmd**  
*Figure to Rmarkdown*

**Description**

Use Articul8 Add-ons from Google docs to build Rarticles

**Usage**

```r
figure2rmd(text, path = ".", opts = NA, prefix = "Figure")
```

**Arguments**

- **text**: String with the table information
- **path**: Path of the image for the figure
- **opts**: chunk options in brackets.
- **prefix**: Prefix for the name of the figure

**Value**

Mutated string
footnotes

Footnotes in tables

Description

Include tables footnotes and symbols for kables in pandoc format

Usage

```r
code

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

gdoc2rmd

Google docs to Rmarkdown

Description

Use Articul8 Add-ons from Google docs to build Rarticles

Usage

```r
code

Arguments

doc

file Zip file path from Articul8 exported in md format
export path to export the files. Default file directory
prefix_fig Prefix for the name of the figure
prefix_tab Prefix for the name of the table
```
**H2cal**

**Value**

folder

---

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

**Description**

Heritability in plant breeding on a genotype difference basis

**Usage**

```r
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 for fixed and random effects (default = FALSE). Options: "base", "ggplot".
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.

For more information review the references.

Value

list

Author(s)

Maria Belen Kistner
Flavio Lozano Isla

References

Examples

```r
library(inti)

dt <- potato

hr <- H2cal(data = dt,
  , trait = "stemdw"
  , gen.name = "geno"
  , rep.n = 5
  , fixed.model = "0 + (1|bloque) + geno"
  , random.model = "1 + (1|bloque) + (1|geno)"
  , emmeans = TRUE
  , plot.diag = FALSE
  , outliers.rm = TRUE
)

hr$tabsmr
hr$blues
hr$blups
hr$outliers
```

Description

Include figures with title and notes using a data base

Usage

```
include_figure(figure, caption = NA, notes = NA, label = NA)
```
Arguments

- **figure**: Path or URL of the figure.
- **caption**: Figure caption (default = NA).
- **notes**: Figure notes (default = NA).
- **label**: Label for the notes (default = NA).

Value

Figure with caption and notes

Examples

```r
library(inti)

figure <- "https://inkaverse.com/reference/figures/logo.png"

figure %>% include_figure(caption = "Title test.", notes = "Note test.")
```

---

include_table

Table with footnotes

Description

Include tables with title and footnotes for word and html documents

Usage

```r
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
jc_tombola

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:")

jc_tombola

Description

Function for arrange journal club schedule

Usage

jc_tombola(
data,
members,
papers = 1,
group,
gr_lvl,
status,
st_lvl,
frq,
date,
seed = NULL
)

Arguments

data Data frame with members and their information.
members Columns with the members names.
papers Number of paper by meeting
group Column for arrange the group.
**mean_comparison**

| **gr_lvl** | Levels in the groups for the arrange. See details. |
| **status** | Column with the status of the members. |
| **st_lvl** | Level to confirm the assistance in the JC. See details. |
| **frq** | Number of the day for each session. |
| **date** | Date when start the first session of JC. |
| **seed** | Number for replicate the results (default = date). |

**Details**

The function could consider n levels for `gr_lvl`. In the case of two level the third level will be both. The suggested levels for `st_lvl` are: active or spectator. Only the active members will enter in the schedule.

**Value**

- data frame with the schedule for the JC

**Description**

Function to compare treatment from lm or aov using data frames

**Usage**

```r
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

```r
## Not run:
library(inti)
library(gsheet)
url <- paste0("https://docs.google.com/spreadsheets/d/
  "15r7ZwcZZHbEgl7F6g5FvCTFA-CFzVBWwg3mF1RyKPs/
  "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**  environment (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.
Description

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

Usage

outliers_remove(data, trait, model)

Arguments

data Experimental design data frame with the factors and traits.
trait Name of the trait.
model The fixed or random effects in the model.

Details

Function to remove outliers in MET experiments

Value

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

References


Examples

library(inti)

rmout <- outliers_remove(
  data = potato,
  trait = "stemdw",
  model = "0 + (1|bloque) + geno"
)

rmout$outliers
plot_diag  Diagnostic plots

Description

Function to plot the diagnostic of models

Usage

plot_diag(model, title = NA)

Arguments

model  Statistical model

title  Plot title

Value

plots

Examples

## Not run:

dt <- potato

lm <- aov(stemdw ~ bloque + geno*treat, dt)

plot(lm, which = 1)
plot_diag(lm)[3]

plot(lm, which = 2)
plot_diag(lm)[2]

plot(lm, which = 3)
plot_diag(lm)[4]

plot(lm, which = 4)
plot_diag(lm)[1]

## End(Not run)
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)
plot_raw

xtext : Text labels in x axis using a vector

ytext : 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 = NULL,
           ylab = NULL,
           xlab = NULL,
           glab = ""
  )

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

## End(Not run)
**plot_smr**

*Plot summary data*

---

**Description**

Graph summary data into bar o line plot

**Usage**

```r
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
plot_smr

```
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 output of mean_comparison(graph_opts = TRUE) function. It contain all the parameter for the plot.
You could add additional layer to the plot using "+" with ggplot2 options

Value

plot

Examples

```R
## 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 = "line",
           x = "geno",
           y = "hi",
           xlab = "",
           group = "treat",
           glab = "Tratamientos",
           ylimits = "",
           color = c("brown", "blue"),
           gtext = c("Irrigado", "Dry Down ")
)
```

## End(Not run)
**Water use efficiency in 15 potato genotypes**

**Description**

Experiment to evaluate the physiological response from 15 potato 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 experiment 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 efficiency (g/l)
- **twue**  tuber water use efficiency (g/l)
- **lfa**  leaf area (cm2)
### table2rmd

**Description**
Use Articul8 Add-ons from Google docs to build Rtkicles

**Usage**
```
table2rmd(text, opts = NA, prefix = "Table")
```

**Arguments**
- `text` String with the table information
- `opts` chunk options in brackets.
- `prefix` Prefix for the name of the table

**Value**
Mutated string

### tarpuy

**Description**
Invoke RStudio addin to create fieldbook designs

**Usage**
```
tarpuy(dependencies = FALSE)
```

**Arguments**
- `dependencies` Install package dependencies for run the app

**Details**
Tarpuy allow to create experimental designs under an interactive app.

**Value**
Shiny app
Examples

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

---

**tarpuy_design**

*Fieldbook experimental designs*

**Description**

Function to deploy experimental designs

**Usage**

```r
tarpuy_design(
  data,
  nfactors = 1,
  type = "crd",
  rep = 2,
  serie = 2,
  seed = 0,
  barcode = NA
)
```

**Arguments**

- `data`: Experimental design data frame with the factors and level. See examples.
- `nfactors`: Number of factors in the experiment (default = 1). See details.
- `type`: Type of experimental arrange (default = "crd"). See details.
- `rep`: Number of replications in the experiment (default = 3).
- `serie`: Digits in the plot id (default = 2).
- `seed`: Replicability of draw results (default = 0) always random. See details.
- `barcode`: Barcode prefix for data collection.

**Details**

The function allows to include the arguments in the sheet that have the information of the design. You should include 2 columns in the sheet: {arguments} and {values}. See examples. The information will be extracted automatically and deploy the design. `nfactors = 1`: crd, rcbd, lsd, lattice. `nfactors = 2` (factorial): split-crd, split-rcbd split-lsd `nfactors >= 2` (factorial): crd, rcbd, lsd.
Value

A list with the fieldbook design

Examples

```r
## Not run:

library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/",
               "183upHd4wriZz2UnInoo5Ate5YFd6k7c1hE0sMQ2x5iw/edit#gid=532773890")
# browseURL(url)

fb <- gsheet2tbl(url)

tarpuy_design(data = fb)

## End(Not run)
```

tarpuy_plex  Fieldbook plan information

Description

Information for build a plan for an experiment (PLEX)

Usage

```r
tarpuy_plex(
  data = NULL,
  idea = NULL,
  goal = NULL,
  hypothesis = NULL,
  rationale = NULL,
  objectives = NULL,
  plan = NULL,
  institutions = NULL,
  researchers = NULL,
  manager = NULL,
  location = NULL,
  altitude = NULL,
  georeferencing = NULL,
  environment = NULL,
  start = NA,
  end = NA,
)```
about = NULL,
fieldbook = NULL,
album = NULL,
github = NULL,
nfactor = 2,
design = "rcbd",
rep = 3,
serie = 2,
seed = 0
)

Arguments

data Data with the fieldbook information.
idea How the idea was born.
goal The main goal of the project.
hypothesis What are the expected results.
rationale Based in which evidence is planned the experiment.
objectives The objectives of the project.
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.
alitude 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.
about Short description of the project.
fieldbook Name or ID for the fieldbook/project.
album link with the photos of the project.
github link with the github repository.
nfactor Number of factors for the design.
design Type of design.
rep Number of replication.
serie Number of digits in the plots.
seed Seed for the randomization.

Details

Provide the information available.
tarpuy_plotdesign

Value
data frame or list of arguments:
   1. info
   2. variables
   3. design
   4. logbook
   5. timetable
   6. budget

Description
Plot fieldbook sketch designs based in experimental design

Usage
tarpuy_plotdesign(
data,
factor,
dim = NULL,
fill = "plots",
xlab = NULL,
ylab = NULL,
glab = NULL
)

Arguments
data Experimental design data frame with the factors and level. See examples.
factor Vector with the name of the columns with the factors.
dim Dimension for reshape the design arrangement.
fill Value for fill the experimental units (default = "plots").
xlab Title for x axis.
ylab Title for y axis.
glab Title for group axis.

Details
The function allows to plot the experimental design according the field experiment design.

Value
plot
**tarpuy_varlist**  

*Fieldbook variable list*

---

**Description**

Function to include the variables to evaluate in the fieldbook design.

**Usage**

```r
tarpuy_varlist(fieldbook, varlist = NULL)
```

**Arguments**

- `fieldbook`  
  Data frame with the fieldbook.

- `varlist`  
  Data frame with the variables information. See examples.

**Details**

The function allows to include the arguments in the sheet that have the information of the variables. You should include 3 columns in the sheet: `{abbreviation}`, `{evaluation}` and `{sampling}`. See examples. The information will be extracted automatically and deploy the list of variable for the fieldbook design.

**Value**

data frame

**Examples**

```r
## Not run:
library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/", "183upHd4wriZz2UnInoo5Ate5YFdtk7cZlhE0sMQ2x5iw/edit#gid=532773890")
# browseURL(url)

info <- gsheet2tbl(url)
fieldbook <- tarpuy_design(data = info)

url_var <- paste0("https://docs.google.com/spreadsheets/d/", "183upHd4wriZz2UnInoo5Ate5YFdtk7cZlhE0sMQ2x5iw/edit#gid=1335288687")

varlist <- gsheet2tbl(url_var)

tarpuy_varlist(fieldbook = fieldbook, varlist = varlist)
```
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",
  scrollly = NULL
)

Arguments

data Dataset.
caption Title for the table.
digits Digits number in the table exported.
rnames Row names.
bbuttons Buttons: "excel", "copy" or "none". Default c("excel", "copy")
file_name Excel file name
scrollly Windows height to show the table. Default "60vh"

Value

table in markdown format for html documents

Examples

## Not run:

library(inti)

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

```r
if(interactive()){
    inti::yupana()
}
```
yupana_analysis  
Fieldbook analysis report

---

**Description**

Function to create a complete report of the fieldbook.

**Usage**

```r
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`: Comparison test c("SNK", "TUKEY", "DUNCAN")
- `sig_level`: Significical 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

```r
yupana_export(
    data, 
    type = NA, 
    xlab = NA, 
    ylab = NA, 
    glab = NA, 
    ylims = 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/", "15r7ZwcZZHbEgl1F6g5FvCTFA-5FzVBWwg3mFLP1Kps/edit?gid=172957346")
# browseURL(url)

fb <- gsheet2tbl(url)

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

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

#> import
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/"
 , "15r7ZwcZZHbEglt1F6gSFvCkTFACFzVBWwlg3mf1RyKPs/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_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.
summary_by Variables for group the analysis.
groups Groups for color in PCA.
variables Variables to be use in the analysis.

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

Value
result and plots
Examples

```r
## Not run:
library(inti)
library(gsheet)

url <- paste0("https://docs.google.com/spreadsheets/d/", "15r7ZwcZZHbEgl1F6g5vCTFA-CFvBWy3mFlRyKPs/edit#gid=172957346")
# browseURL(url)

fb <- gsheet2tbl(url)

mv <- yupana_mvr(data = fb,
                 last_factor = "bloque",
                 summary_by = c("geno", "treat"),
                 groups = NULL)

FactoMineR::plot.PCA(mv$pca, choix = "ind", habillage = mv$param$groups)

## End(Not run)
```

---

**yupana_reshape**

Fieldbook reshape

**Description**

Function to reshape fieldbook according a separation character

**Usage**

```r
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, 11
  potato, 19
colortext, 2
figure2rmd, 3
footnotes, 4
gdoc2rmd, 4
H2cal, 5
include_figure, 7
include_table, 8
jc_tombola, 9
mean_comparison, 10
met, 11
metamorphosis, 12
outliers_remove, 13
plot_diag, 14
plot_raw, 15
plot_smr, 17
potato, 19
table2rmd, 20
tarpuy, 20
tarpuy_design, 21
tarpuy_plex, 22
tarpuy_plotdesign, 24
tarpuy_varlist, 25
web_table, 26
yupana, 27
yupana_analysis, 28
yupana_export, 29
yupana_import, 31
yupana_mvr, 32
yupana_reshape, 33