Package 'propalloc'

May 28, 2020

Title Oliver Wyman RShiny Property Allocation App

Version 0.0.1

Description Property Allocation model in R Shiny that allocates an insured's prospective year property insurance costs.

License MIT + file LICENSE

URL https://github.com/jimbrig2011/propalloc

BugReports https://github.com/jimbrig2011/propalloc/issues

Depends R (¿= 2.10)

Imports attempt,
dplyr,
DT,
fs,
fst,
htmltools,
janitor,
lubridate,
magrittr,
matchmaker,
purrr,
readr,
rhandsontable,
rintrojs,
rlang (¿= 0.1.2),
shiny,
shinycustomloader,
shinydashboard,
shinyjs,
shinyWidgets,
stats,
stringr,
tibble,
tidyr,
tidyselect,
utils

Suggests attachment,
devtools,
knitr,

rmarkdown,
testthat (¿= 2.1.0),
writexl

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

R topics documented:

`add_external_resources` *Add external resources to UI head*

Description

Adds external resources to shiny application from `propalloc`'s www directory.

Adds external resources to shiny application from `propalloc`'s www directory.

Usage

```
add_external_resources()
```

```
add_external_resources()
```

Value

HTML head tag

HTML head tag

Adds the following:

1. `shinyjs::useShinyjs()`
2. `shinyWidgets::useSweetAlert()`
3. `rintrojs::introjsUI()`
4. Custom CSS
5. Custom JavaScript

1. `shinyjs::useShinyjs()`
2. `shinyWidgets::useSweetAlert()`
3. `rintrojs::introjsUI()`
4. Custom CSS
5. Custom JavaScript

apply_labels *Apply Labels*

Description

Apply Labels

Usage

```
apply_labels(
  data,
  dict,
  from = "value",
  to = "value_label",
  by = "variable",
  names_from = "variable",
  names_to = "variable_label",
  dataset_name = NULL,
  ...
)
```

Arguments

data	dataset to apply labeling on.
dict	dictionary to use for application of labeling.
from	a column name or position defining words or keys to be replaced
to	a column name or position defining replacement values
by	character or integer - which column in dict defines the columns in data corresponding to each section of the dict. Defaults to "variable".
names_from	column name or position defining where to match names from. Defaults to "variable".
names_to	column name or position defining replacements for names(data). Defaults to "variable_label".
dataset_name	(optional) name of dataset to filter dict by
...	Passed to match_df

Value

a labelled, display friendly tibble

Examples

```
dictionary <- data.frame(
  var = c(rep("x1", 2),
          rep("x2", 3),
          rep("x3", 5)),
  var_lab = c(rep("Column 1", 2),
              rep("Column 2", 3),
              rep("Column 3", 5)),
  val = c(c(TRUE, FALSE),
```

```
          c(1:3),
          letters[1:5]),
  val_lab = c(c("YES", "NO"),
              paste0("Group ", c(1:3)),
              paste0("Area ", LETTERS[1:5]))
)

dat <- data.frame(
  "x1" = rep(c(TRUE, FALSE), 15),
  "x2" = rep(c(1:3), 10),
  "x3" = rep(letters[1:5], 6)
)

apply_labels(dat,
             dictionary,
             from = "val",
             to = "val_lab",
             by = "var",
             names_from = "var",
             names_to = "var_lab")
```

`apply_relativity` *Apply Relativity*

Description

Apply Relativity

Usage

```
apply_relativity(
  relativity_data,
  coverage = c("aop", "cat_eq", "cat_wind", "cat_flood", "terrorism"),
  sov_linker = NA,
  sov = sov
)
```

Arguments

`relativity_data`	df
`coverage`	cov
`sov_linker`	link
`sov`	df

Value

df

apply_rels *Apply Relativity Adjustment Factors*

Description

Apply Relativity Adjustment Factors

Usage

```
apply_rels(
  bu_rels = NULL,
  sprinkler_tier_rels = NULL,
  combustible_rels = NULL,
  sov = NULL
)
```

Arguments

bu_rels	BU rels
sprinkler_tier_rels	Sprinkler Tier rels
combustible_rels	Combustible rels
sov	SOV

Value

tibble

apply_surcharges *Apply Claim Count Surcharges*

Description

Apply Claim Count Surcharges

Usage

```
apply_surcharges(preliminary_allocation_data, count_buckets)
```

Arguments

preliminary_allocation_data	prelim data
count_buckets	count buckets table

Value

data.frame

`apply_threshold` *Apply Threshold*

Description

Apply Threshold

Usage

```
apply_threshold(dat, total_pct_chg, threshold)
```

Arguments

`dat`	data frame with required columns: uncapped_allocated and prior_allocated
`total_pct_chg`	total percent change since prior on the overall rate (premium / TIV) excluding expenses.
`threshold`	threshold to apply

Value

a tibble

`app_body_ui` *App Body UI Function*

Description

App Body UI Function

App Body UI Function

Usage

```
app_body_ui()
```

```
app_body_ui()
```

Value

HTML for a dashboardBody

HTML for a dashboardBody

app_header_ui *App Header UI Function*

Description

Adds header buttons and contacts.

Adds header buttons and contacts.

Usage

```
app_header_ui()
```


```
app_header_ui()
```


Value

HTML for a dashboardHeader

HTML for a dashboardHeader

app_server *App Server*

Description

App Server

Usage

```
app_server(input, output, session)
```

Arguments

input	shiny server input
output	shiny server output
session	shiny server session

Value

shiny app server

app_sidebar_ui *App Sidebar UI Function*

Description

App Sidebar UI Function

App Sidebar UI Function

Usage

```
app_sidebar_ui()
```

```
app_sidebar_ui()
```

Value

HTML for a dashboardSidebar

HTML for a dashboardSidebar

app_ui *App UI*

Description

App UI

App UI

Usage

```
app_ui()
```

```
app_ui()
```

Value

HTML tagList containing `shinydashboard::dashboardPage()`

HTML tagList containing `shinydashboard::dashboardPage()`

arrange_by_entity *Arrange a data.frame by entity*

Description

Since entity_id is a character field, this function parsed the number and arranges the data in increasing order by the entity id's number.

Usage

```
arrange_by_entity(data)
```

Arguments

data	data containing a column for "entity_id".

Value

an arranged tibble.

bu_rels *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
bu_rels
```

Format

A data frame with 2 rows and 6 variables:

bu character. DESCRIPTION.

aop_bu_relativity double. DESCRIPTION.

cat_eq_bu_relativity double. DESCRIPTION.

cat_wind_bu_relativity double. DESCRIPTION.

cat_flood_bu_relativity double. DESCRIPTION.

terrorism_bu_relativity double. DESCRIPTION.

combustible_rels *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
combustible_rels
```

Format

A data frame with 2 rows and 2 variables:

`aop_combustible` character. DESCRIPTION.

`aop_combustible_relativity` double. DESCRIPTION.

contacts *Contacts*

Description

Contacts

Usage

```
contacts()
```

Value

vector of `contact_item()`'s for usage in header dropdown.

contact_item *Contact Item*

Description

Creates an item to be placed in a contact dropdownmenu.

Usage

```
contact_item(
  name = "First Name, Last Name",
  role = "Role",
  phone = "###-###-####",
  email = "first.last@oliverwyman.com"
)
```

Arguments

`name`	Name
`role`	Role
`phone`	Phone
`email`	Email

Value

contact menu item

`contact_menu` *Creates a dropdown menu specific for contacts*

Description

Creates a dropdown menu specific for contacts

Usage

```
contact_menu(...)
```

Arguments

`...`	contact items to put into dropdown

Value

menu

`count_buckets` *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
count_buckets
```

Format

A data frame with 8 rows and 6 variables:

`bucket` character. DESCRIPTION.
`name` character. DESCRIPTION.
`min` double. DESCRIPTION.
`max` double. DESCRIPTION.
`percent_surcharge` double. DESCRIPTION.
`dollar_surcharge` double. DESCRIPTION.

entity_loss_summary *Derive Loss Data by Entity*

Description

This function derives total incurred, total claim counts, and claim counts by "count bucket" for use in the allocation model.

Usage

```
entity_loss_summary(
  loss_run,
  count_buckets,
  experience_period = NA,
  min_date = NA,
  max_date = NA
)
```

Arguments

loss_run	loss run
count_buckets	count bucket table
experience_period	defined experience period
min_date	minimum date
max_date	maximum data

Value

tibble

extract_costs *Extract Costs for Allocation*

Description

This function takes as input the initial cost table (by default uses the internal dataset `initial_costs`) and returns a named list containing:

- terrorism - Terrorism Cost
- all_risk - All Risk Cost (CAT + AOP)
- risk_transfer - Risk Transfer Cost (CAT + AOP + Terrorism)
- expenses - Total expenses
- total_w_expense - Total Including Expenses

Usage

```
extract_costs(cost_table)
```

Arguments

cost_table	initial costs data.frame to extract from (default equals initial_costs)

Value

list of costs

---

flucol *Fluid Column - Shiny fluidRow + Column*

Description

Fluid Column - Shiny fluidRow + Column

Usage

```
flucol(..., width = 12, offset = 0)
```

Arguments

...	elements to include within the flucol
width	width
offset	offset

Value

A column wrapped in fluidRow

---

get_col_classes *Get column classes from a data.frame*

Description

Get column classes from a data.frame

Usage

```
get_col_classes(data)
```

Arguments

data	data.frame input

Value

if each column only has one associated class, a named character vector is returned with the names equal to the column names and values equal to the classes. If some columns have more than a single class, a named list with column names as names and classes as values is returned.

Examples

```
data <- data.frame(a = c(1:3), b = letters[1:3], c = c(TRUE, FALSE, TRUE))
get_col_classes(data)
```

---

header_buttons *Header Buttons Server Module*

Description

Header Buttons Server Module

Usage

```
header_buttons(
  input,
  output,
  session,
  parent_session = NULL,
  include_tour = TRUE,
  tour_steps = NULL,
  include_refresh = TRUE,
  include_help = TRUE,
  include_disclaimer = TRUE,
  help_path = fs::path_package("propalloc", "reports/RMD/help.Rmd"),
  disclaimer_path = fs::path_package("propalloc", "reports/RMD/disclaimer.Rmd")
)
```

Arguments

input	shiny input
output	shiny output
session	shiny session
parent_session	shiny session of the parent environment where this module is called from. Only used if include_tour is TRUE.
include_tour	logical - include a 'tour app' button?
tour_steps	list of 'steps' for tour. Passed to rintrojs::introjs().
include_refresh	logical - include a 'refresh' button?
include_help	logical - include a 'help' button?
include_disclaimer	logical - include a 'disclaimer' button?
help_path	character - if include_help is TRUE; specify the path to the 'help' R Markdown document to utilize when button is pressed.
disclaimer_path	character - if include_disclaimer is TRUE; specify the path to the 'disclaimer' R Markdown document to utilize when button is pressed.

Value

server

header_buttons_ui *Header Buttons UI Module*

Description

Header Buttons UI Module

Usage

```
header_buttons_ui(
  id,
  include_tour = TRUE,
  include_refresh = TRUE,
  include_help = TRUE,
  include_disclaimer = TRUE,
  include_contact = TRUE,
  include_logout = TRUE,
  contacts = NULL
)
```

Arguments

id	namespace ID
include_tour	logical - include a 'tour app' button?
include_refresh	logical - include a 'refresh' button?
include_help	logical - include a 'help' button?
include_disclaimer	logical - include a 'disclaimer' button?
include_contact	logical - include a 'contact' button?
include_logout	logical - include a 'logout' button? Note this will replace the default shiny logout.
contacts	contacts list from `contacts()`.

Value

`htmltools::tagList()`

icon_text *Icon Text*

Description

Creates an HTML div containing the icon and text.

Usage

```
icon_text(icon, text)
```

Arguments

icon	fontawesome icon
text	text

Value

HTML div

Examples

```
icon_text("table", "Table")
```

---

ingest_relativities *Ingest Relativity Adjustment Factors*

Description

This function consumes the user-defined relativity adjustment factors and applies them to the TIV by entity resulting in a tibble::tibble with columns for each coverages relativity adjusted TIV.

Usage

```
ingest_relativities(rels_list, sov = NULL)
```

Arguments

rels_list	A multi-layered, named list of lists. The first level of lists names are in the format `name = list(applies_to = "<coverage>",data = <data.frame>)` where the name refers to the name used to indentify the relativity (i.e. sprinkler tier, business unit, etc), the applies_to defines which coverage to applu the relativity adjustments to, and data is a 2-column data.frame where the first column specifies character labels and the second column specifies numeric factors.
sov	SOV

Value

tibble

inputs_tab *Inputs tab Server*

Description

Inputs tab Server

Usage

```
inputs_tab(input, output, session)
```

Arguments

input	shiny server input
output	shiny server output
session	shiny server session

Value

a named list of all input tables

inputs_tab_ui *Inputs Tab Module UI*

Description

Inputs Tab Module UI

Usage

```
inputs_tab_ui(id)
```

Arguments

id	namespace ID

Value

shinydashboard::tabItem()

input_table *Table Input Module Server*

Description

Table Input Module Server

Table Input Module Server

Usage

```
input_table(
  input,
  output,
  session,
  data,
  dictionary,
  dataset_name,
  add_total_row = FALSE,
  digits = 0,
  currency_cols = NULL,
  percent_cols = NULL,
  percent_digits = 0,
  center_cols = NULL,
  stretch = TRUE
)

input_table(
  input,
  output,
  session,
  data,
  dictionary,
  dataset_name,
  add_total_row = FALSE,
  digits = 0,
  currency_cols = NULL,
  percent_cols = NULL,
  percent_digits = 0,
  center_cols = NULL,
  stretch = TRUE
)
```

Arguments

input	shiny input
output	shiny output
session	shiny session
data	initial data to pass to rhandsontable::rhandsontable().
dictionary	dictionary passed to apply_labels().

dataset_name	(optional) name of dataset to filter dict by
add_total_row	logical - should your table add a total row? Created via janitor::adorn_totals().
digits	numeric passed to jsonlite::toJSON
currency_cols	character or numeric vector to apply currency formatting to in table output via rhandsontable::hot_cols(cols = currency_cols,format = "$0,0"). Should use unlabeled 'R Friendly' column names.
percent_cols	character or numeric vector specifying columns to apply percent formatting to. Should use unlabeled 'R Friendly' column names.
percent_digits	integer specifying number of digits to display for any percent_cols. Not used if percent_cols = NULL.
center_cols	character or numeric vector specifying columns to center in table output. Passed to rhandsontable::hot_cols(cols = center_cols,halign = "htCenter"). Should use unlabeled 'R Friendly' column names.
stretch	logical-should table fill container width?

Value

data.frame with R-friendly syntax for usage in upstream modules and server logic.

data.frame with R-friendly syntax for usage in upstream modules and server logic.

---

input_table_ui *Input Table Module UI*

Description

Input Table Module UI

Input Table Module UI

Usage

```
input_table_ui(id, width = "100%", height = "100%")

input_table_ui(id, width = "100%", height = "100%")
```

Arguments

id	namespace ID
width	must be a valid CSS unit in pixels or a number, which will be coerced to a string and have "px" appended.
height	must be a valid CSS unit in pixels or a number, which will be coerced to a string and have "px" appended.

Value

rhandsontable::rHandsontableOutput

rhandsontable::rHandsontableOutput

insert_logo *Insert Logo*

Description

Insert Logo

Usage

```
insert_logo(
  file,
  style = "background-color: #FFF; width: 100%; height: 100%;",
  width = NULL,
  ref = "#"
)
```

Arguments

file	file
style	style
width	width
ref	ref

Value

tag

install_app_dependencies *Install App Dependencies*

Description

Install App Dependencies

Install App Dependencies

Usage

```
install_app_dependencies()
```

```
install_app_dependencies()
```

Value

invisible

invisible

load_demo_data *Load Demo Data for Use in Property Allocation Shiny App and Examples*

Description

Load Demo Data for Use in Property Allocation Shiny App and Examples

Usage

```
load_demo_data(data = "all", assign = TRUE)
```

Arguments

data	Character: name of dataset(s) or "all".
assign	Logical: Should the resulting data.frames be unpacked and assigned to global environment? If FALSE returns as a names list.

Value

If assign is TRUE returns objects to global environment invisibly. If assign is FALSE returns a named list.

Examples

```
library(propalloc)

## Not run:
# unpack to global environment
load_demo_data()

## End(Not run)

# return as a list
demo_data_list <- load_demo_data(assign = FALSE)
```

loss_run *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
loss_run
```

Format

A data frame with 482 rows and 8 variables:

`claim_number` character. DESCRIPTION.

`location_dud` character. DESCRIPTION.

`entity_id` character. DESCRIPTION.

`date_of_loss` double. DESCRIPTION.

`total_incurred` double. DESCRIPTION.

`accident_description` character. DESCRIPTION.

`accident_location` character. DESCRIPTION.

`year` double. DESCRIPTION.

`merge_entity_data` *Derive Entity Data*

Description

Derive Entity Data

Usage

```
merge_entity_data(
  sov,
  relativity_adjusted_tivs,
  entity_loss_data,
  rates,
  priors
)
```

Arguments

`sov`	sov
`relativity_adjusted_tivs`	rel adj tivs
`entity_loss_data`	inc, buckets, and cnts by ent
`rates`	rate table
`priors`	prior premium table

Value

entity_data database

preliminary_allocation *Perform Preliminary Allocation*

Description

Perform Preliminary Allocation

Usage

```
preliminary_allocation(entity_data, costs)
```

Arguments

entity_data	entity data
costs	renewal costs list

Value

data.frame

prepare_driver_summary *Prepare a driver summary using the final allocation*

Description

Prepare a driver summary using the final allocation

Usage

```
prepare_driver_summary(allocated_data, filter_vector)
```

Arguments

allocated_data	the data that has passed through all allocation stages (apply_threshold() is the final stage). This data has had filters applied to fields bu:department
filter_vector	Named character vector. Entries represent the filter (from Rshiny) to be applied to allocated_data. Values of the vector either NA (meaning no filter) or character. Names of vector refer to bu:department

Value

data.frame

`prepare_rhandsontable` *Prepare rhandsontable*

Description

Prepare rhandsontable

Usage

```
prepare_rhandsontable(
  r_data,
  dictionary,
  dataset_name,
  add_total_row = FALSE,
  digits = 0,
  currency_cols = NULL,
  percent_cols = NULL,
  percent_digits = 0,
  center_cols = NULL,
  stretch = TRUE
)
```

Arguments

`r_data`	data
`dictionary`	dict
`dataset_name`	name
`add_total_row`	logical
`digits`	number
`currency_cols`	character vector
`percent_cols`	character vector
`percent_digits`	number
`center_cols`	character vector
`stretch`	logical

Value

rhandsontable::hot_table

priors *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
priors
```

Format

A data frame with 739 rows and 12 variables:

`entity_id` character. DESCRIPTION.
`prior_tiv` double. DESCRIPTION.
`prior_expenses` double. DESCRIPTION.
`prior_aop_premium` double. DESCRIPTION.
`prior_cat_eq_premium` double. DESCRIPTION.
`prior_cat_wind_premium` double. DESCRIPTION.
`prior_cat_flood_premium` double. DESCRIPTION.
`prior_terrorism_premium` double. DESCRIPTION.
`prior_total_cat_premium` double. DESCRIPTION.
`prior_all_risk_premium` double. DESCRIPTION.
`prior_risk_transfer_premium` double. DESCRIPTION.
`prior_premium_incl_expenses` double. DESCRIPTION.

pull_unique *Pull all unique values for a variable*

Description

Pull all unique values for a variable

Usage

```
pull_unique(data, var, sort = TRUE, decreasing = FALSE, names = TRUE)
```

Arguments

`data`	data.frame
`var`	variable name
`sort`	logical (default = TRUE)
`decreasing`	logical (default = FALSE)
`names`	logical (default = TRUE)

Value

vector

rates *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
rates
```

Format

A data frame with 38 rows and 5 variables:

`rate_type` character. DESCRIPTION.

`rate_id` character. DESCRIPTION.

`prior_rate` double. DESCRIPTION.

`market_rate` double. DESCRIPTION.

`model_rate` double. DESCRIPTION.

rebalance_premiums *Rebalance premiums to a total value*

Description

Preserves the portion of the total held by the column that is being rebalanced.

Usage

```
rebalance_premiums(
  the_data,
  column_name_to_rebalance,
  total_to_rebalance_to,
  do_keep_column_name = TRUE
)
```

Arguments

`the_data`	data.frame
`column_name_to_rebalance`	character reflecting the name of the (numeric) column that is being rebalanced
`total_to_rebalance_to`	numeric
`do_keep_column_name`	boolean If FALSE, then automatically renames to "column_name_rebalanced"

Value

data.frame

Examples

```
data.frame(premiums = 1:3) %>% rebalance_premiums("premiums", 12)
# Returns data.frame(premiums = c(2, 4, 6))
```

---

renewal_costs *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
renewal_costs
```

Format

A data frame with 5 rows and 4 variables:

`cost_type` character. DESCRIPTION.

`description` character. DESCRIPTION.

`prior` double. DESCRIPTION.

`current` double. DESCRIPTION.

---

renewal_costs_tab *Inputs Tab Module Server*

Description

Inputs Tab Module Server

Usage

```
renewal_costs_tab(input, output, session, initial_costs, dictionary)
```

Arguments

`input`	shiny server input
`output`	shiny server output
`session`	shiny server session
`initial_costs`	data.frame for initial table representing renewal costs.
`dictionary`	internal

Value

List of input data.frames

renewal_costs_tab_ui *Renewal Costs Module UI*

Description

Renewal Costs Module UI

Usage

```
renewal_costs_tab_ui(id)
```

Arguments

id	Namespace ID

Value

shinydashboard::tabBox HTML

reverse_labels *Reverse Labels*

Description

Reverses link[apply_labels()].

Usage

```
reverse_labels(
  data,
  dict,
  from = "value",
  to = "value_label",
  by = "variable",
  names_from = "variable",
  names_to = "variable_label",
  dataset_name = NULL,
  ...
)
```

Arguments

data	dataset to apply labeling on.
dict	dictionary to use for application of labeling.
from	a column name or position defining words or keys to be replaced
to	a column name or position defining replacement values
by	character or integer - which column in dict defines the columns in data corresponding to each section of the dict. Defaults to "variable".

names_from	column name or position defining where to match names from. Defaults to "variable".
names_to	column name or position defining replacements for names(data). Defaults to "variable_label".
dataset_name	(optional) name of dataset to filter dict by
...	Passed to match_df

Value

un-labeled, R-friendly tibble

---

run_app *Run the Shiny Application*

Description

Run the Shiny Application

Run the Shiny Application

Usage

```
run_app(install_dependencies = FALSE)
```

```
run_app(install_dependencies = FALSE)
```

Arguments

install_dependencies	logical-install package/app dependencies? Defaults to FALSE.

---

sov *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
sov
```

Format

A data frame with 739 rows and 24 variables:

`entity_id` character. DESCRIPTION.
`loss_run_id` character. DESCRIPTION.
`bu` character. DESCRIPTION.
`region` character. DESCRIPTION.
`country` character. DESCRIPTION.
`state` character. DESCRIPTION.
`division` character. DESCRIPTION.
`location` character. DESCRIPTION.
`department` character. DESCRIPTION.
`tiv` double. DESCRIPTION.
`aop_coverage` logical. DESCRIPTION.
`cat_eq_coverage` logical. DESCRIPTION.
`cat_wind_coverage` logical. DESCRIPTION.
`cat_flood_coverage` logical. DESCRIPTION.
`terrorism_coverage` logical. DESCRIPTION.
`aop_sprinkler_tier` character. DESCRIPTION.
`aop_combustible` character. DESCRIPTION.
`aop_tiv_size_bucket` character. DESCRIPTION.
`cat_wind_hurricane` character. DESCRIPTION.
`aop_id` character. DESCRIPTION.
`cat_eq_id` character. DESCRIPTION.
`cat_wind_id` character. DESCRIPTION.
`cat_flood_id` character. DESCRIPTION.
`terrorism_id` character. DESCRIPTION.

`sprinkler_tier_rels` *DATASET_TITLE*

Description

DATASET_DESCRIPTION

Usage

```
sprinkler_tier_rels
```

Format

A data frame with 4 rows and 2 variables:

`aop_sprinkler_tier` character. DESCRIPTION.
`aop_sprinkler_tier_relativity` double. DESCRIPTION.

toproper *To Proper*

Description

To Proper

Usage

```
toproper(
  string,
  replace_underscores = TRUE,
  underscore_replacement = " ",
  return_as = c("titlecase", "uppercase", "lowercase", "asis"),
  uppers = c("Id", "Aop", "Cat", "Eq", "Tiv")
)
```

Arguments

string	string to manipulate on
replace_underscores	Logical: if TRUE replaces all underscores with specified underscore_replacement argument's value.
underscore_replacement	Character: if argument replace_underscores equals TRUE, will replace all "_"'s with specified string.
return_as	How should the string be returned? Options are: • "titlecase": Applies stringr::str_to_title. • "uppercase": Applies toupper. • "lowercase": Applied tolower. • "asis": No manipulation. Returns as is.
uppers	Character vector of any strings that should be displayed in upper-case (i.e. TPA, WC, AL, ABC, etc.)

Value

"Proper" string

See Also

str_replace.

Examples

```
s <- "variable_a is awesome"
toproper(s)
```

Index