

Package ‘normfluodbf’

October 14, 2023

Title Cleans and Normalizes 'FLUOstar' 'DBF' and 'DAT' Files

Version 1.4.3.9000

Description Converts a 'FLUOstar' 'DBF' File into a Normalized Data Frame, Ready for Analysis.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

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

Config/testthat/edition 3

Imports data.table,
foreign,
tidyr,
tibble,
dplyr,
ggplot2,
cranlogs,
ggthemes,

Depends R (>= 2.10)

LazyData true

URL <https://github.com/AlphaPrime7/normfluodbf>, <https://alphaprime7.github.io/normfluodbf/>

BugReports <https://github.com/AlphaPrime7/normfluodbf/issues>

VignetteBuilder knitr

R topics documented:

clean_evendat	2
clean_odddat	3
comma_cleaner	4
dat_1	4
dat_2	5

dat_col_names	5
decimal_scaling	6
generic_identifier	7
gg_plot_triplets	7
liposomes_214	8
liposomes_215	9
liposomes_216	9
liposomes_218	9
liposomes_221	10
liposomes_227	10
log_transformation	10
min_max_norm	11
min_max_norm_percent	12
nfd_tracker	13
normfluodat	13
normfluordbf	14
norm_tidy_dbf	15
norm_z	16
resample_dat	17
resample_datv2	18
resample_dat_scale	19
unique_identifier	20

Index 21

clean_evendat	<i>Title: A function to partially clean dat files with an identified pattern from FLUostar experiments</i>
---------------	--

Description

The function takes a dirty data frame and returns a clean data frame with the only exception being that most values in the returned data frame have commas at the end. The data frame that goes into the function comes from reading dat files from FLUOstar experiments. See the example code.

Usage

```
clean_evendat(df)
```

Arguments

df A data frame with n number of rows

Value

A new data frame with most impurities taken out. See the example.

Note

This function only works on a specific type of Fluostar dat file and really serves to be made available for anyone to find a universal utility for the function.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat1_df <- read.table(file=fpath)
partial_cleaned_dat <- clean_evendat(dat1_df)
```

clean_odddat

Title: A function to partially clean dat files from FLUOstar experiments

Description

The function takes a dirty data frame and returns a clean data frame with the only exception being that most values in the returned data frame have commas at the end. The data frame that goes into the function comes from reading dat files from FLUOstar experiments. See the example code.

Usage

```
clean_odddat(df)
```

Arguments

df A data frame with n number of rows

Value

A new data frame with most impurities taken out; See the example

Note

This function should work on all types of Fluostar dat files (unlike the clean_evendat() function).

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
partial_cleaned_dat <- clean_odddat(dat_df)
```

comma_cleaner	<i>Title: A function to remove unwanted commas in a partially cleaned data frame</i>
---------------	--

Description

The function takes a dirty data frame (comma_df) derived from either the clean_odddat or clean_evendat functions. The resulting data frame is a cleaned data frame with numeric values.

Usage

```
comma_cleaner(comma_df)
```

Arguments

comma_df	A dirty data frame (numeric values have commas at the end) with n number of rows
----------	--

Value

A new data frame with commas at the end of numbers taken out

Note

This function is a subordinate function and follows a sequence of actions. In this package, this function cannot be used as a standalone.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
comma_dat <- clean_odddat(dat_df)
nocomma_dat <- comma_cleaner(comma_dat)
```

dat_1	<i>dat_1.</i>
-------	---------------

Description

FLUOstar .dat files used for creation of the update and unusable for data immediate data analysis.

Usage

```
dat_1
```

Format

An object of class data.frame with 320 rows and 12 columns.

dat_2	<i>dat_2.</i>
-------	---------------

Description

FLUOstar .dat files used for creation of the update and unusable for data immediate data analysis.

Usage

```
dat_2
```

Format

An object of class `data.frame` with 320 rows and 12 columns.

dat_col_names	<i>Title: A function to obtain attribute names for experimental samples</i>
---------------	---

Description

The function takes a clean data frame, data on the experiment and returns the column names that match the FLUOstar plate reader.

Usage

```
dat_col_names(df, rows_used = NULL, cols_used = NULL)
```

Arguments

df	A clean data frame obtained from the large scale delineation of samples.
rows_used	A character vector representing the plate rows used; eg <code>ru <- c('A','B','C')</code> . can be used in sequence or out of sequence.
cols_used	A numeric vector representing the plate columns used; eg <code>cu <- c(1,2,3,4)</code> . keep as null if all the columns were used or columns are used in sequence.

Value

Returns column names that will be added to the normalized data frame that contains all samples

Note

This function is a subordinate function and follows a sequence of actions. In this package, this function cannot be used as a standalone. Also, some work is needed here on the part of the user because i have no access to their setup file. A function that takes the setup excel file from the user should be part of the next update to prevent the user from doing much work.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
comma_dat <- clean_odddat(dat_df)
nocomma_dat <- comma_cleaner(comma_dat)
nocomma_dat <- as.data.frame(nocomma_dat)
resampled_scaled <- resample_dat_scale(nocomma_dat, tnp=3, cycles=40)
n = c('A','B','C')
sample_col_names <- dat_col_names(resampled_scaled, n , cols_used = NULL) # i used all columns (in sequence) so
```

decimal_scaling	<i>Title: A function that performs decimal scaling (not normalization) of attributes</i>
-----------------	--

Description

Title: A function that performs decimal scaling (not normalization) of attributes

Usage

```
decimal_scaling(x)
```

Arguments

x Column or attribute values passed into the decimal scaling function

Value

Attribute(s) with values that have the decimal place moved to facilitate machine learning

Note

The lapply function is required to apply the function across several columns in a data set. This is NOT a normalization function because the data still exist on a sliding scale. This function is here for demonstration purposes and should be used for exercise as it is here for educational purposes for the inventor of the package.

Author(s)

Tingwei Adeck

References

<https://www.statology.org/how-to-normalize-data-in-r/>

Examples

```
test_df <- as.data.frame(c(seq(40)))
colnames(test_df) <- "test"
test_df_norm <- lapply(test_df[1:ncol(test_df)], decimal_scaling)
```

generic_identifier	<i>Title: A generic identifier similar to unique identifier but end users supply a column name.</i>
--------------------	---

Description

A function that creates a column 1:nrow(df) and steps by 1 but gives you the option to use any column name.

Usage

```
generic_identifier(numrows, col_name)
```

Arguments

numrows	The number of rows in a data frame of interest (nrows(df) can be used).
col_name	The desired column name for the column.

Value

A new single column data frame with the desired attribute added.

Author(s)

Tingwei Adeck

Examples

```
generic_identifier(40,col_name="Cycle_No")
```

gg_plot_triplets	<i>Title: The visualizer function</i>
------------------	---------------------------------------

Description

Visualizes liposome flux assay experiments for all three sample types

Usage

```
gg_plot_triplets(df, x, y_list, xlim, ylim)
```

Arguments

df	clean normalized dat data frame
x	Cycle_No or Time
y_list	sample types (tnp = test, negative, positive)
xlim	x-axis range (0: number of cycles)
ylim	y-axis range (normalization range); 0:1 in this case

Value

A normalized data frame with the x-variable (Cycle_No), ready for analysis

Note

This function is not very modular so use with caution. It should work just fine in most cases but here for demonstration.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
fpath_dbf <- system.file("extdata", "liposomes_214.dbf", package = "normfluodbf", mustWork = TRUE)
n <- c('A', 'B', 'C')
normalized_fluo_dat <- normfluodat(dat=fpath, tnp = 3, cycles = 40, n)
normalized_dbf_scalez <- norm_tidy_dbf(file=fpath_dbf, norm_scale = 'z-score')
yvars <- c("A1", "B1", "C1")
yvars_dbf <- c('A01', 'A02', 'A03')
xvar <- c("Cycle_Number")
color <- c("Test", "Negative Control", "Positive Control")
xl <- c(0, 40)
yl <- c(0, 1)
yl_dbf <- c(-3, 3)
gg_plot_triplets(normalized_fluo_dat, x=xvar, y_list=yvars, xlim=xl, ylim=yl)
gg_plot_triplets(normalized_dbf_scalez, x=xvar, y_list=yvars_dbf, xlim=xl, ylim=yl_dbf)
```

liposomes_214

liposomes_214.

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

```
liposomes_214
```

Format

An object of class data.frame with 11 rows and 52 columns.

liposomes_215	<i>liposomes_215.</i>
---------------	-----------------------

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

liposomes_215

Format

An object of class data.frame with 11 rows and 52 columns.

liposomes_216	<i>liposomes_216.</i>
---------------	-----------------------

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

liposomes_216

Format

An object of class data.frame with 8 rows and 52 columns.

liposomes_218	<i>liposomes_218.</i>
---------------	-----------------------

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

liposomes_218

Format

An object of class data.frame with 11 rows and 52 columns.

liposomes_221	<i>liposomes_221.</i>
---------------	-----------------------

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

```
liposomes_221
```

Format

An object of class `data.frame` with 38 rows and 52 columns.

liposomes_227	<i>liposomes_227.</i>
---------------	-----------------------

Description

FLUOstar .dbf file in wide format and unable to use for data analysis.

Usage

```
liposomes_227
```

Format

An object of class `data.frame` with 29 rows and 52 columns.

log_transformation	<i>Title: A function that performs log transformation of data</i>
--------------------	---

Description

Title: A function that performs log transformation of data

Usage

```
log_transformation(x)
```

Arguments

x	Column or attribute values passed into the Z-standardization function
---	---

Value

Log transformed attributes

Note

The lapply function is required to apply the function across several columns in a data set.

Author(s)

Tingwei Adeck

References

<https://www.statology.org/how-to-normalize-data-in-r/>

Examples

```
test_df <- as.data.frame(c(seq(40)))
colnames(test_df) <- "test"
test_df_norm <- lapply(test_df[1:ncol(test_df)], log_transformation)
```

min_max_norm	<i>Title: A function that performs Min-Max normalization(0-1 range) of attributes that require normalization</i>
--------------	--

Description

Title: A function that performs Min-Max normalization(0-1 range) of attributes that require normalization

Usage

```
min_max_norm(x)
```

Arguments

x Column or attribute values passed into the min-max normalization function

Value

A normalized value (between 0 and 1) when used as a standalone function, or a normalized attribute(s) when used with lapply.

Note

The lapply function is required to apply the function across several columns in a data set.

Author(s)

Tingwei Adeck (Adapted from Statology)

References

<https://www.statology.org/how-to-normalize-data-in-r/>

Examples

```
test_df <- as.data.frame(c(seq(40)))  
colnames(test_df) <- "test"  
test_df_norm <- lapply(test_df[1:ncol(test_df)], min_max_norm)
```

min_max_norm_percent	<i>Title: A function that performs Min-Max normalization (0-100 range) of attributes that require normalization</i>
----------------------	---

Description

Title: A function that performs Min-Max normalization (0-100 range) of attributes that require normalization

Usage

```
min_max_norm_percent(x)
```

Arguments

x	Column or attribute values passed into the min-max normalization function
---	---

Value

A normalized value (between 0 and 100) when used as a standalone function, or a normalized attribute(s) when used with lapply.

Note

The lapply function is required to apply the function across several columns in a data set.

Author(s)

Tingwei Adeck

References

<https://www.statology.org/how-to-normalize-data-in-r/>

Examples

```
test_df <- as.data.frame(c(seq(40)))  
colnames(test_df) <- "test"  
test_df_norm <- lapply(test_df[1:ncol(test_df)], min_max_norm_percent)
```

nfd_tracker	<i>Title: Normfluodbf tracker</i>
-------------	-----------------------------------

Description

A simple tracker for the package. A fun addition to the package.

Usage

```
nfd_tracker(package, period = NULL, plot = NULL)
```

Arguments

package	package name as string (example: package = 'normfluodbf')
period	takes 'last-month' or 'last-week'; default is NULL
plot	takes 'cum' or 'daily'; default is null

Value

A data frame of cumulative downloads or a data frame of daily downloads

Note

The tracker is not modular so only works for normfluodbf hence in the package

Author(s)

Tingwei Adeck

Examples

```
nfd_tracker(package = 'normfluodbf', period = 'last-month', plot = 'cum')
```

normfluodat	<i>Title: The root function that returns a normalized data frame with the Cycle No ready for analysis</i>
-------------	---

Description

Input a dat file (dat file directory) and required parameters and BOOM the researcher has a normalized data frame ripe and ready for clean analysis

Usage

```
normfluodat(dat, tnp, cycles, rows_used = NULL, cols_used = NULL)
```

Arguments

<code>dat</code>	directory to the users FLUOstar dat file
<code>tnp</code>	Stands for test,negative,positive (sample types); the number should match the number of sample types in the plate reader even if repeating a sample type
<code>cycles</code>	The number of cycles chosen by the researcher. In the case of this package 40 is the standard but ensure to have the right number of samples
<code>rows_used</code>	A character vector of the rows used, eg <code>n = c('A','B','C')</code>
<code>cols_used</code>	A numeric vector of the columns used, eg <code>m = c(2,4,6)</code>

Value

A normalized data frame with the x-variable (Cycle_No), ready for analysis

Note

This is the MAIN function and stands alone but is dependent on the subordinate functions. If the user understands what they are doing this is all they need.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluordbf", mustWork = TRUE)
n <- c('A','B','C')
normalized_fluo_dat <- normfluodat(dat=fpath, tnp = 3, cycles = 40, n)
```

normfluordbf

Title: Cleans and Normalizes ".dbf" files obtained from experiments using the FLUOstar microplate reader.

Description

Input the path to a ".dbf" file obtained from the FLUOstar microplate (usually a 96-well microplate) reader; this function will create a data frame, clean the data frame, normalize the data frame, append a "Cycle_Number" column and return a data frame that is ready for analysis. Most importantly, this function is a single_step function. Also, the function can be extended to other ".dbf" files if they follow the format for which this function was designed; this is totally at the users' discretion.

Usage

```
normfluordbf(file = NULL, norm_scale = NULL, transformed = NULL, fun = NA, ...)
```

Arguments

file	A string ("liposomes_xxx.dbf") if the file is found within the present working directory (pwd) OR a path pointing directly to a ".dbf" file, from FLUOstar experiments.
norm_scale	This parameter can taken in 'hundred', 'one', or 'z-score' which denotes the normalization type; Initialized as NULL.
transformed	This parameter can take in 'log' which denotes a logarithmic box-cox transformation; Initialized as NULL.
fun	A variable defined as NA, used for boolean expressions or manipulation.
...	A container object that can be used to capture extra variables if needed.

Value

A normalized data frame with an appended "Cycle_Number" attribute.

Note

Re-nomenclature of norm_tidy_dbf to a more appropriate name that facilitates function utilization. Users can continue with the old name ("norm_tidy_dbf") but this is a better name in my opinion.

Examples

```
fpath <- system.file("extdata", "liposomes_214.dbf", package = "normfluordbf", mustWork = TRUE)
normalized_dbf <- normfluordbf(file=fpath)
```

norm_tidy_dbf	<i>Title: Cleans and Normalizes ".dbf" files obtained from experiments using the FLUOstar microplate reader.</i>
---------------	--

Description

Input the path to a ".dbf" file obtained from the FLUOstar microplate (usually a 96-well microplate) reader; this function will create a data frame, clean the data frame, normalize the data frame, append a "Cycle_Number" column and return a data frame that is ready for analysis. Most importantly, this function is a single_step function. Also, the function can be extended to other ".dbf" files if they follow the format for which this function was designed; this is totally at the users' discretion.

Usage

```
norm_tidy_dbf(
  file = NULL,
  norm_scale = NULL,
  transformed = NULL,
  fun = NA,
  ...
)
```

Arguments

file	A string ("liposomes_xxx.dbf") if the file is found within the present working directory (pwd) OR a path pointing directly to a ".dbf" file, from FLUOstar experiments.
norm_scale	This parameter can taken in 'raw', 'hundred', 'one', 'z-score' or 'decimal', which denotes the normalization type; Initialized as NULL.
transformed	This parameter can take in 'log' which denotes a logarithmic box-cox transformation; Initialized as NULL.
fun	A variable defined as NA, used for boolean expressions or manipulation.
...	A container object that can be used to capture extra variables if needed.

Value

A normalized data frame with an appended "Cycle_Number" attribute.

Note

Re-nomenclature of norm_tidy_dbf to a more appropriate name that facilitates function utilization. Users can continue with the old name ("norm_tidy_dbf") but this is a better name in my opinion. The default dbf normalization technique outputs values in the 0-1 range. The user needs to specify different aesthetics if they are used to seeing their plots on a different range. The function accounts for nine conditions and there is an active quest to find a better way to write this. Please NOTE that decimal scaling is a sliding scale and so should yield unwanted results. Please Note that the user must specify the norm_scale if they also want to specify log transformation(transformed argument). Backward compatibility is maintained and so there should be no issues with using the package the way the user was used to. The last NOTE is that if the user has no real clue how to analyze their data and just inputs a link to the dbf file, the program returns a 0-1 scale normalized data frame.

Examples

```
fpath <- system.file("extdata", "liposomes_214.dbf", package = "normfluodbf", mustWork = TRUE)
normalized_dbf_default <- norm_tidy_dbf(file=fpath)
normalized_dbf_scale100 <- norm_tidy_dbf(file=fpath, norm_scale = 'hundred')
xvar <- c("Cycle_Number")
yvars_dbf <- c("A01", "A02", "A03")
color <- c("Test", "Negative Control", "Positive Control")
xl <- c(0, 40)
yl_dbf <- c(0, 100)
gg_plot_triplets(normalized_dbf_scale100, x=xvar, y_list=yvars_dbf, xlim=xl, ylim=yl_dbf)
```

norm_z

Title: A function that performs Z-score standardization (or normalization) of attributes

Description

Title: A function that performs Z-score standardization (or normalization) of attributes

Usage

```
norm_z(x)
```


Arguments

x Column or attribute values passed into the Z-standardization function

Value

Attribute(s) with values that have mean 0 and standard deviation 1

Note

The lapply function is required to apply the function across several columns in a data set.

Author(s)

Tingwei Adeck

References

<https://www.statology.org/how-to-normalize-data-in-r/>

Examples

```
test_df <- as.data.frame(c(seq(40)))
colnames(test_df) <- "test"
test_df_norm <- lapply(test_df[1:ncol(test_df)], norm_z)
```

resample_dat	<i>Title: A function to extract sample types from the 120 tuples data cleaned data frame.</i>
--------------	---

Description

Designed as a prototype function to take a single column from the cleaned dat data frame and return the n=3 sample types (n can be any number based on the number of sample types used) as separate attributes with n=40 tuples (n can vary based on the number of cycles ran) The function is well thought and accounts for almost any scenario.

Usage

```
resample_dat(df, tnp, cycles, samples_per_tnp = NULL)
```

Arguments

df	A clean data frame with n number of rows
tnp	Stands for test,negative,positive (sample types); the number should match the number of sample types in the plate reader even if repeating a sample type
cycles	The number of cycles chosen by the researcher. In the case of this package 40 is the standard but ensure to have the right number of samples
samples_per_tnp	An optional parameter thought to be useful but had no use based on the approach taken to solve this problem; Will be useful in future functions

Value

A new data frame with attributes matching the number of sample types and tuples matching the number of cycles. In short it returns delineated samples.

Note

This function is a subordinate function and prototype that turned out very useful.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
comma_dat <- clean_odddat(dat_df)
nocomma_dat <- comma_cleaner(comma_dat)
col_1 <- nocomma_dat[,1]
col_1 <- as.data.frame(col_1)
samples_delineated <- resample_dat(col_1, tnp=3, cycles=40)
```

resample_datv2

Title: A function to extract sample types from the long-form (120 tuples) data cleaned data frame.

Description

Designed as a prototype function to take a single column from the cleaned dat data frame and return the n=3 sample types (n can be any number based on the number of sample types used) as separate attributes with n=40 tuples (n can vary based on the number of cycles ran) The function is well thought and accounts for almost any scenario.

Usage

```
resample_datv2(df, tnp, cycles)
```

Arguments

df	A clean data frame with n number of rows
tnp	Stands for test,negative,positive (sample types); the number should match the number of sample types in the plate reader even if repeating a sample type
cycles	The number of cycles chosen by the researcher. In the case of this package 40 is the standard but ensure to have the right number of samples

Value

A new data frame with attributes matching the number of sample types and tuples matching the number of cycles. In short it returns delineated samples.

Note

This function is a subordinate function and prototype that turned out very useful. Syntax modifications present for educational and learning purposes on my part-note taking. One parameter has been taken out compared to `resample_dat()`.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
comma_dat <- clean_odddat(dat_df)
nocomma_dat <- comma_cleaner(comma_dat)
col1 <- nocomma_dat[,1]
col1_80 <- nocomma_dat[c(1:80),1] #just for fun assume 2 samples only
col1 <- as.data.frame(col1)
col1_80 <- as.data.frame(col1_80 )
samples_delineated <- resample_datv2(col1, tnp=3, cycles=40)
two_sample_test <- resample_datv2(col1_80, tnp = 2, cycles = 40)
```

resample_dat_scale	<i>Title: A scaling up of the resample_dat() function</i>
--------------------	---

Description

Performs the role of the prototype `resample_dat()` function but it is scaled up to work on all columns in longer-form data frame.

Usage

```
resample_dat_scale(df, tnp, cycles)
```

Arguments

df	A clean data frame with n number of rows
tnp	Stands for test,negative,positive (sample types); the number should match the number of sample types in the plate reader even if repeating a sample type
cycles	The number of cycles chosen by the researcher. In the case of this package 40 is the standard but ensure to have the right number of samples

Value

A new data frame with attributes matching the number of sample types and tuples matching the number of cycles. In short it returns delineated samples.

Note

This function is the pre-requisite to the parent or main function of the update. As a matter of fact, this function is modified to produce the parent or main function.

Author(s)

Tingwei Adeck

See Also[resample_dat\(\)](#), [resample_datv2\(\)](#) and [dat_col_names\(\)](#)**Examples**

```
fpath <- system.file("extdata", "dat_1.dat", package = "normfluodbf", mustWork = TRUE)
dat_df <- read.table(file=fpath)
comma_dat <- clean_odddat(dat_df)
nocomma_dat <- comma_cleaner(comma_dat)
resampled_scaled <- resample_dat_scale(nocomma_dat, tnp=3, cycles=40)
```

unique_identifier

Title: A modified unique identifier function for FLUOstar normalized data

Description

The function creates a column called Cycle_Number and adds to the cleaned dbf data frame

Usage

```
unique_identifier(df)
```

Arguments

df A data frame with n number of rows

Value

A new data frame with the Cycle_Number attribute added.

Note

The FLUOstar microplate reader runs in cycles with the number of cycles determined by the experimenter. This function essentially counts the number of cycles and is a subordinate function to the main function `norm_tidy_dbf()`. Also the function can be used as a standalone but the only limitation is the column name will be Cycle_Number. For a more generic version of this program, use the `generic_identifier` included.

Author(s)

Tingwei Adeck

Examples

```
test_df <- as.data.frame(c(seq(40)))
colnames(test_df) <- "test"
unique_identifier(test_df)
```

Index

- * **datasets**
 - dat_1, [4](#)
 - dat_2, [5](#)
 - liposomes_214, [8](#)
 - liposomes_215, [9](#)
 - liposomes_216, [9](#)
 - liposomes_218, [9](#)
 - liposomes_221, [10](#)
 - liposomes_227, [10](#)
- clean_evendat, [2](#)
- clean_odddat, [3](#)
- comma_cleaner, [4](#)
- dat_1, [4](#)
- dat_2, [5](#)
- dat_col_names, [5](#)
- dat_col_names(), [20](#)
- decimal_scaling, [6](#)
- generic_identifler, [7](#)
- gg_plot_triplets, [7](#)
- liposomes_214, [8](#)
- liposomes_215, [9](#)
- liposomes_216, [9](#)
- liposomes_218, [9](#)
- liposomes_221, [10](#)
- liposomes_227, [10](#)
- log_transformation, [10](#)
- min_max_norm, [11](#)
- min_max_norm_percent, [12](#)
- nfd_tracker, [13](#)
- norm_tidy_dbf, [15](#)
- norm_z, [16](#)
- normfluodat, [13](#)
- normfluordbf, [14](#)
- resample_dat, [17](#)
- resample_dat(), [20](#)
- resample_dat_scale, [19](#)
- resample_datv2, [18](#)
- resample_datv2(), [20](#)
- unique_identifler, [20](#)