

Package ‘GenderInfer’

October 12, 2022

Type Package

Title This is a Collection of Functions to Analyse Gender Differences

Version 0.1.0

Maintainer Rita Giordano <giordanor@rsc.org>

Description Implementation of functions, which combines binomial calculation and data visualisation, to analyse the differences in publishing authorship by gender described in Day et al. (2020) <[doi:10.1039/C9SC04090K](https://doi.org/10.1039/C9SC04090K)>. It should only be used when self-reported gender is unavailable.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports ggplot2, binom

Depends R (>= 2.10)

Suggests dplyr, knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Rita Giordano [aut, cre],
Aileen Day [aut],
John Boyle [aut],
Colin Batchelor [ctb],
Royal Society of Chemistry [cph]

Repository CRAN

Date/Publication 2021-09-29 08:00:05 UTC

R topics documented:

assign_gender	2
authors	3
balloon_plot	3

bar_chart	4
baseline	4
bullet_chart	5
bullet_line_chart	6
calculate_binom_baseline	7
gender_names	7
percent_df	8
reshape_for_binomials	8
stacked_bar_chart	9
theme_gd	10
total_gender_df	10

Index	11
--------------	-----------

assign_gender	<i>Assign gender by first name</i>
---------------	------------------------------------

Description

This function use the data source based on combined US/UK censor data to assign gender based on first name.

Usage

```
assign_gender(data_df, first_name_col)
```

Arguments

data_df, input dataframe containing the first name
first_name_col, first name column's name to assign gender to

Value

The input data frame with the gender column:
gender - assigned gender (F/M/U)

Examples

```
gender <- assign_gender(authors, "first_name")
```

authors	<i>names dataset</i>
---------	----------------------

Description

This data sets contains all the name fro UK and US social security

Usage

```
authors
```

Format

a data frame with 1000 rows of four variables:

first_name first name

last_name last lame

country_code country

publication_years publication year

balloon_plot	<i>Function to create the balloon plot for gender first name</i>
--------------	--

Description

Function to create the balloon plot for gender first name

Usage

```
balloon_plot(data_df, gender_var, cutoff)
```

Arguments

`data_df`, data frame containing 'first name' and 'gender' columns from [assign_gender](#)
`gender_var`, gender possible values are F for female, M for male and U for unknown
`cutoff`, numerical value indicating where to cut the counting data

Value

The output is a gg object from ggplot2 which shows the most frequent names as a balloon plot.

Examples

```
gender <- assign_gender(authors, "first_name")  
bp <- balloon_plot(gender, "M", cutoff = 5)
```

bar_chart	<i>Function to create a bar chart of the total number by gender</i>
-----------	---

Description

Function to create a bar chart of the total number by gender

Usage

```
bar_chart(data_df, x_label, y_label)
```

Arguments

data_df,	dataframe from total_gender_df
x_label,	label for x axis.
y_label,	label for y axis.

Value

A bar chart as `ggplot2` object showing on the y axis the total number per gender and on the x axis the level previously defined in [total_gender_df](#).

baseline	<i>Calculate the female baseline</i>
----------	--------------------------------------

Description

baseline calculate the female baseline giving a dataframe containing the gender information.

Usage

```
baseline(data_df, gender_col)
```

Arguments

data_df,	dataframe containing the gender column.
gender_col,	the name of the column containing the gender information.

Value

The function returns a numeric vector containing the baseline values

Examples

```
## df is the dataframe in output from the function assign_gender
df <- data.frame(first_name = c("anna", "john", "ernest", "colin", "aileen"),
                 gender = c("F", "M", "M", "M", "F"),
                 stringsAsFactors = FALSE)
baseline <- baseline(df, gender_col = "gender")
```

bullet_chart	<i>Create a bullet chart with significance bars to compare different baselines in percentage for gender analysis</i>
--------------	--

Description

Create a bullet chart with significance bars to compare different baselines in percentage for gender analysis

Usage

```
bullet_chart(data_df, baseline_female, x_label, y_label, baseline_label)
```

Arguments

data_df, dataframe in output from [percent_df](#)
baseline_female, numeric vector containing the baseline for each level
x_label, label for x axis
y_label, label for y axis
baseline_label, label used to define the baseline name.

Value

This function create a bullet chart containing the percentage of submission with the corresponding baseline for the level defined in [percent_df](#).

bullet_line_chart	<i>Function to create a bullet chart with a line chart in the same graphical frame; to compare different baselines for gender analysis.</i>
-------------------	---

Description

Function to create a bullet chart with a line chart in the same graphical frame; to compare different baselines for gender analysis.

Usage

```
bullet_line_chart(  
  data_df,  
  baseline_female,  
  x_label,  
  y_bullet_chart_label,  
  baseline_label,  
  line_chart_df,  
  line_chart_scaling,  
  y_line_chart_label,  
  line_label  
)
```

Arguments

data_df,	dataframe in output from percent_df
baseline_female,	numeric vector containing the baseline for each level
x_label,	label for x axis for both charts
y_bullet_chart_label,	label for y axis of the bullet chart
baseline_label,	label used to define the baseline name.
line_chart_df,	data frame containing the total number of submissions
line_chart_scaling,	factor of conversion for second y-axis
y_line_chart_label,	label the y-axis of the line chart
line_label,	label used to define the line chart.

Value

The function create a bullet chart containing the percentage of male and female with the corresponding baseline for the level defined in [percent_df](#). The total number of submissions are displayed on the top of the bullet chart.

 calculate_binom_baseline

Calculate binomials and significance for multiple baselines.

Description

Function to calculate the lower CI, upper CI, percentages and counts, and significance of difference from one or multiple baseline percentages, given supplied confidence level using

Usage

```
calculate_binom_baseline(data_df, baseline_female, confidence_level = 0.95)
```

Arguments

data_df, dataframe in output from [reshape_for_binomials](#) containing the columns: female, male, which contain the integer counts of males and females respectively and must be a numeric vector greater than 0.

baseline_female, female baseline in percentage from [baseline](#).

confidence_level, confidence level to use for significance calculation, default is 0.95

Value

This function returns a dataframe with additional columns than the input one:

lower_CI = lower confidence level of confidence interval expressed as a percentage

upper_CI = upper confidence level of confidence interval expressed as a percentage

lower_CI_count = lower confidence level of confidence interval expressed as a count

upper_CI_count = upper confidence level of confidence interval expressed as a count

significance = flag indicating whether difference of female percentage with baseline percentage is significant for the row in consideration. It has values "significant" or "" if not.

 gender_names

Gender names dataset

Description

This data sets contains all the name fro UK and US social security

Usage

```
gender_names
```

Format

a data frame of two variables:

Name First name

UKUS_Gender Gender of the first name

percent_df	<i>Create a dataframe that will be the input to generate stacked bar chart and bullet chart that show percentage to compare proportions among gender.</i>
------------	---

Description

Create a dataframe that will be the input to generate stacked bar chart and bullet chart that show percentage to compare proportions among gender.

Usage

```
percent_df(data_df)
```

Arguments

data_df,	dataframe containing level, lower_CI, upper_CI, significance and female and male percentages from calculate_binom_baseline
----------	--

Value

The output dataframe contains the columns x_values, y_values, gender, labels

reshape_for_binomials	<i>Reshape the dataframe to make it easier to carry out binomial calculations.</i>
-----------------------	--

Description

reshape dataframe from long format to wide format.

Usage

```
reshape_for_binomials(data_df, gender_col, level)
```

Arguments

data_df,	dataframe containing the columns gender and counts
gender_col,	the name of the column containing the gender values.
level,	variable to compare for the baseline.

Value

The output is a dataframe containing more columns than the input one, such as:

level : the variable used to perform the binomials
 total_for_level: the total amount of each gender including unknowns
 total_female_male: the total amount of male and female
 female_percentage: the percentage of female in the total_female_male
 male_percentage: the percentage of male in the total_female_male

Examples

```
authors_df <- assign_gender(data_df = authors, first_name_col = "first_name")
female_count <- dplyr::count(authors_df, gender)

## create a new data frame to be used for the binomial calculation.
df_gender <- reshape_for_binomials(data = female_count, gender_col = "gender",
                                  level = 2020)
```

stacked_bar_chart	<i>Create a stacked bar chart with significance bars to compare with the female baseline for gender analysis.</i>
-------------------	---

Description

Create a stacked bar chart with significance bars to compare with the female baseline for gender analysis.

Usage

```
stacked_bar_chart(data_df, baseline_female, x_label, y_label, baseline_label)
```

Arguments

data_df, is the output dataframe from [percent_df](#)
 baseline_female, female baseline in percentage from [baseline](#)
 x_label, label for x axis
 y_label, label for y axis
 baseline_label, label used to define the baseline name.

Value

This function create a bar chart containing the percentage of submission with the corresponding baseline.

theme_gd	<i>This function create a gender diversity theme for chart based on ggplot2</i>
----------	---

Description

This function create a gender diversity theme for chart based on ggplot2

Usage

```
theme_gd()
```

Value

an object of the class theme defined in ggplot2 own class system.

Examples

```
require(ggplot2)
ggplot(authors, aes(x = publication_years)) + geom_bar() + theme_gd()
```

total_gender_df	<i>Create a dataframe that will be the input to generate the bar chart of the full amount of female and male</i>
-----------------	--

Description

Create a dataframe that will be the input to generate the bar chart of the full amount of female and male

Usage

```
total_gender_df(data_df, level)
```

Arguments

data_df,	dataframe from calculate_binom_baseline containing Level, LCI, UCI, Significance and Male and Female percentages
level,	name of level

Value

The output is a dataframe with the columns x_values, total_female_male, gender, y_values. This data frame is the input to create the bar chart for [bar_chart](#)

Index

* datasets

authors, [3](#)

gender_names, [7](#)

assign_gender, [2](#), [3](#)

authors, [3](#)

balloon_plot, [3](#)

bar_chart, [4](#), [10](#)

baseline, [4](#), [7](#), [9](#)

bullet_chart, [5](#)

bullet_line_chart, [6](#)

calculate_binom_baseline, [7](#), [8](#), [10](#)

gender_names, [7](#)

percent_df, [5](#), [6](#), [8](#), [9](#)

reshape_for_binomials, [7](#), [8](#)

stacked_bar_chart, [9](#)

theme_gd, [10](#)

total_gender_df, [4](#), [10](#)