Package: CHOIRBM (via r-universe)

August 21, 2024

Title Plots the CHOIR Body Map

Version 0.0.2.9000

Description Collection of utility functions for visualizing body map data collected with the Collaborative Health Outcomes Information Registry.

License MIT + file LICENSE

URL `https://github.com/emcramer/CHOIRBM`

BugReports `https://github.com/emcramer/CHOIRBM/issues`

Depends R (>= 3.5.0)

Imports broom, ggplot2, rlang, stringr

Suggests knitr, rmarkdown, testthat (>= 2.0.0)

VignetteBuilder knitr

Config/testthat/edition 2

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Repository https://emcramer.r-universe.dev

RemoteUrl https://github.com/emcramer/choirbm

RemoteRef HEAD

RemoteSha ba5a7d2eae3e79db6d2a1712666bbb6e603a6cec

Contents

`agg_choirbm_list` *Converts a list of CBMs to a single data frame*

Description

Takes a list of data frames where each data frame is the CBM of a patient and the values column is a binary endorsement of the CBM segment. Then it collapses the information to a single data frame for plotting by adding the 'value' columns of each data frame.

Usage

```
agg_choirbm_list(cbm_list)
```

Arguments

`cbm_list`	a list of CBMs to collapse

Value

map_df a single CBM data frame with the value column summed.

Examples

```
## Not run:
data(validation)
cbm_list <- lapply(validation[["bodymap_regions_csv"]], string_to_map)
agg_df <- agg_choirbm_list(cbm_list)

## End(Not run)
```

`comp_choirbm_chi` *Compare CBM segment endorsement across categorical variable using Chi-Square*

Description

Compare CBM segment endorsement across categorical variable using Chi-Square

Usage

```
comp_choirbm_chi(cbm_list, ...)
```

Arguments

`cbm_list`	a named list of CBMs
`...`	additional parameters passed to p.adjust()

Value

a data frame with the p-values, chi statistic, and degrees of freedom

Examples

```
## Not run:
data(validation)
# split male and female data
male_data <- validation[validation[['gender']] == "Male", ]
male_bodymap_list <- lapply(
 male_data[["bodymap_regions_csv"]]
 , string_to_map)
male_bodymap_df <- agg_choirbm_list(male_bodymap_list)
female_data <- validation[validation[['gender']] == "Female", ]
female_bodymap_list <- lapply(
 female_data[["bodymap_regions_csv"]]
 , string_to_map)
female_bodymap_df <- agg_choirbm_list(female_bodymap_list)
# compare with chi square test
chi_res <- comp_choirbm_chi(
 list("male" = male_bodymap_df
 , "female" = female_bodymap_df)
 , method = 'bonferroni'
)

## End(Not run)
```

`comp_choirbm_glm` *Examine the effect of a continuous variable on CBM location endorsement*

Description

Examine the effect of a continuous variable on CBM location endorsement

Usage

```
comp_choirbm_glm(in_df, comp_var, method = "bonferroni", ...)
```

Arguments

`in_df`	a data.frame with at least one column for the CBM as a delimited string, and another column as the continuous variable for modeling.
`comp_var`	the name of the variable to model as a string.
`...`	additional parameters passed to glm.

Value

a data.frame with the following columns: id, term, estimate, std.error, statistic, p.value. Each row is the result of one glm using the continuous variable to predict CBM location endorsement.

Examples

```
## Not run:
data(validation)
set.seed(123)
sampled_data <- validation[sample(1:nrow(validation), 100, replace = FALSE),]
model_ouput <- comp_choirbm_glm(sampled_data, "age")

## End(Not run)
```

`comp_choirbm_ztest` *Compare CBM segment endorsement across categorical variable using z-test(s)*

Description

Compare CBM segment endorsement across categorical variable using z-test(s)

Usage

```
comp_choirbm_ztest(cbm_list, tail = "two", p.method = "bonferroni")
```

Arguments

`cbm_list`	a named list of CBMs
`tail`	whether to do a single or two tailed z test
`...`	additional parameters passed to p.adjust()

Value

a data frame with the p-values and z statistic

Examples

```
library(CHOIRBM)
# isolate and process male data
male_data <- validation[validation[["gender"]] == "Male", ]
# isolate and process female data
female_data <- validation[validation[["gender"]] == "Female", ]
comp_choirbm_ztest(list( "male" = male_data, "female" = female_data), tail = "two")
```

`comp_cooccurrence` *Calculate the co-occurrence between locations on the CBM*

Description

Calculates the raw number of times two locations on the CBM are endorsed together in a data set.

Usage

```
comp_cooccurrence(df)
```

Arguments

`df`	a data.frame with the CBMs as delimited strings in a single column.

Value

a data.frame with every combination of CBM locations and the number of times those locations occur together (the "co-occurrence").

Examples

```
## Not run:
set.seed(123)
sampled_data <- validation[sample(1:nrow(validation), 100, replace = FALSE),]
colnames(sampled_data)[5] <- "bodymap"
con_mat <- comp_cooccurrence(sampled_data)

## End(Not run)
```

`convert_bodymap` *convert_bodymap Helper function to convert a single bodymap*

Description

convert_bodymap Helper function to convert a single bodymap

Usage

```
convert_bodymap(segments)
```

Arguments

`segments`	a character vector containing segment numbers as individual strings in the vector that need to be adjusted/standardized

Value

a character vector containing standardized segment numbers as individual strings in the vector

Examples

```
exampledata <- data.frame(
    GENDER = as.character(c("Male", "Female", "Female")),
    BODYMAP_CSV = as.character(c("112,125","112,113","128,117"))
 )
convert_bodymap(exampledata[2,2])
```

`convert_bodymaps` *convert_bodymaps Function to convert multiple bodymaps*

Description

convert_bodymaps Function to convert multiple bodymaps

Usage

```
convert_bodymaps(f_maps)
```

Arguments

`f_maps`	a character vector where each string is a CHOIR bodymap in csv form

Value

a character vector of bodymaps using the male CHOIR bodymap numberings as a standard. Each bodymap is in csv form

Examples

```
exampledata <- data.frame(
    GENDER = as.character(c("Male", "Female", "Female")),
    BODYMAP_CSV = as.character(c("112,125","112,113","128,117"))
 )
convert_bodymaps(
    as.character(
        exampledata$BODYMAP_CSV[exampledata$GENDER == 'Female']
     )
 )
```

`gen_example_data` *Generate Simple Example Data*

Description

Creates a data frame with CHOIR Body Map segment IDs and a randomly associated value. Also adds grouping information for facetting while plotting.

Usage

```
gen_example_data(seed = 123)
```

Arguments

`seed`	integer to seed the random number generator

Value

values data.frame

Examples

```
cbm_df <- gen_example_data()
head(cbm_df)
```

`num_areas` *Count the number of areas indicated in a CBM*

Description

Counts the number of areas a person endorses/indicates on their CHOIR Body Map.

Usage

```
num_areas(cbm_str, delim = ",")
```

Arguments

`cbm_str`	a delimited string of 3-digit codes indicating CBM areas.
`delim`	the delimiter character, defaults to a comma.

Value

nareas

Examples

```
cbm_str <- c("101,102,103,104")
num_areas(cbm_str, ",")
```

`plot_cooccurrence` *Plot a concurrence matrix*

Description

Generates a concurrence matrix as a heatmap to show which CBM locations are commonly endorsed together.

Usage

```
plot_cooccurrence(con_mat, ...)
```

Arguments

`con_mat`	a long form data frame or matrix produced by the plot_concurrence function, with every combination of locations and the number of times each combination occurs.
`...`	additional parameters for plotting

Value

a ggplot heatmap of the concurrence.

Examples

```
## Not run:
set.seed(123)
sampled_data <- validation[sample(1:nrow(validation), 100, replace = FALSE),]
con_mat <- comp_cooccurrence(sampled_data)
plot_cooccurrence(con_mat)

## End(Not run)
```

`plot_female_choirbm` *Plot the male CHOIR Body Map*

Description

Creates a new plot of the front and back of the female CHOIR body map.

Usage

```
plot_female_choirbm(df, value)
```

Arguments

`df`	data.frame
`value`	string

Value

ggrob

Examples

```
cbm_df <- gen_example_data()
plot_female_choirbm(cbm_df, "value")
```

`plot_male_choirbm` *Plot the male CHOIR Body Map*

Description

Creates a new plot of the male CHOIR body map.

Usage

```
plot_male_choirbm(df, value)
```

Arguments

`df`	data.frame
`value`	string

Value

ggrob

Examples

```
cbm_df <- gen_example_data()
plot_male_choirbm(cbm_df, "value")
```

`plot_nareas_histogram` *Plots a histogram of the number of CBM areas indicated*

Description

This is a wrapper for ggplot2's histogram function that incorporates calculating the number of CBM areas each individual indicates.

Usage

```
plot_nareas_histogram(cbms, ...)
```

Arguments

`cbms`	a list of delimited CBM strings
`...`	additional arguments passed to geom_histogram

Value

a histogram of the number of CBM areas endorsed by individuals in the dataset.

Examples

```
## Not run:
data(validation)
below20 <- validation[
 sapply(validation$bodymap_regions_csv, num_areas) < 20
 , ]
plot_nareas_histogram(
 below20$bodymap_regions_csv
 , binwidth = 1
 , fill = "grey"
 , color = "white")

## End(Not run)
```

`prep_bodymaps` *prep_bodymaps converts a single charcter vector of bodymaps into a list of character vectors, each a bodymap*

Description

prep_bodymaps converts a single charcter vector of bodymaps into a list of character vectors, each a bodymap

Usage

```
prep_bodymaps(maps)
```

Arguments

`maps`	a character vector containing the endorsed bodymap segments of patients in csv form

Value

a list of character vectors, where each vector contains the patient's endorsed segments

Examples

```
exampledata <- data.frame(
    GENDER = as.character(c("Male", "Female", "Female")),
    BODYMAP_CSV = as.character(c("112,125","112,113","128,117"))
 )
prep_bodymaps(as.character(exampledata$BODYMAP_CSV))
```

`string_to_map` *Converts a comma-separated string to a CHOIR BM*

Description

Takes a string of IDs that are separated by a comma and converts the information into a data frame with a binary indication of whether or not an ID appeared. Useful for plotting an individual's CHOIR BM or for isolating particular sections to highlight.

Usage

```
string_to_map(map_str = "", delim = ",")
```

Arguments

`map_str`	The delimited CBM string.
`delim`	The delimiter for the CBM string.

Value

ret_df data.frame with all of the CHOIR BM segment IDs with a 1 if the segment was present and 0 otherwise.

Examples

```
# from a choir database
cbm_str <- "101,102,103,104,201,202"
cbm_df <- string_to_map(cbm_str)
# plot in a male or female bodymap...
plot_male_choirbm(cbm_df, "value")

# from a REDCap project
cbm_str <- "b07,b18,b19,b23,b24,b28,b33,f01,f03,f08,f17,f27,f29"
cbm_df <- string_to_map(cbm_str)
# plot in a male or female bodymap...
plot_male_choirbm(cbm_df, "value")
```

`validation` *CHOIR Body Map data for approximately 7,000 patients*

Description

A non-identifiable, simulated data set generated by randomly permuting data from the CHOIR Body Map validation study.

Usage

```
data(validation)
```

Format

An object of class `"data.frame"`

id A randomly generated numeric code for each patient.

gender The patient's gender.

race The patient's race.

age The patient's age.

bodymap_regions_csv The patient's CHOIR Body Map in a comma separated string.

score A simulated pain score for demonstration purposes.

References

This data set was derived from the data collected during the study validating the CHOIR Body Map as an instrument for recording a patient's anatomical pain location. doi:10.1097/pr9.0000000000000880

Examples

```
data(validation)
head(validation)
```

Index