Package ‘wordgraph’

July 1, 2020

Type Package

Title Graph Functionality of Free Associated Words

Version 0.1.0

Author Epaminondas Diamantopoulos

Maintainer Epaminondas Diamantopoulos <epdiamantopoulos@yahoo.gr>

Description Functions that help less experienced R users to make graph analysis for free associated words, or more generally for repeated nominal data for which a undirected graph analysis is meaningful. By corresponding to each word its centrality, it is possible to apply standard quantitative analysis methods in order to associate word selection with other variables. The functions are implemented with the aid of the 'tibble', 'tidygraph', 'ggraph' and 'ggplot2' packages. Supported centrality functions are centrality_alpha(), centrality_authority(), centrality_betweenness(), centrality_closeness(), centrality_pagerank(), centrality_eigen(). A data set is included.

Imports tidygraph, tibble, plyr, ggraph, purrr, stats, dplyr, ggplot2

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-01 08:40:03 UTC

R topics documented:

add.centrality.variables.to.data.frame

Main function of the package. Creates the graphs and adds the centrality variables to the data frame.

Description

Creates the graph of all words contained in the wordvars, computes the centrality of each word and for each wordvars variable, a column containing the corresponding centrality score is added to the dataframe.

Supported centrality functions are "centrality_alpha", "centrality_authority", "centrality_betweenness", "centrality_closeness", "centrality_pagerank", "centrality_eigen".

After applying add.centrality.variables.to.data.frame, subsequent quantitative analysis can follow where each word is replaced by its correspondent centrality score.

Usage

```
add.centrality.variables.to.data.frame(
  wordvars,
  centralityfunctionstr,
  data.df,
  iscircled = FALSE,
  verbose = FALSE)
```

Arguments

`wordvars` The vector containing the names of the variables containing the free associated words (from 1_st to n_th)

`centralityfunctionstr` The type of the centrality function to use (single string or vector of strings)

`data.df` The data frame where wordvars belong.

`iscircled` Set to TRUE if the graph is circled (that is, last word is related to the first). Default is FALSE.

`verbose` Set to TRUE if warning messages are desired. Default is FALSE.

Value

The initial data frame with additional columns, one for each variable in wordvars vector, having the correspondiding centrality of the contained words.

Author(s)

Epaminondas Diamantopoulos

Examples

```
data(freeassociationdata)

# It is a time consuming function...


wordvars = c("diet1stword", "diet2ndword", "diet3rdword")
df.with.centralities1 = add.centrality.variables.to.data.frame(wordvars,
"centrality_eigen", freeassociationdata)

# Variables with the centralities of the words in wordvars was added
# in the end of df.with.centralities1.

# Histogram of eigen centrality index
hist(df.with.centralities1$diet1stword_centrality_eigen)

# One may ask for two centralities...
df.with.centralities2 = add.centrality.variables.to.data.frame(wordvars,
c("centrality_authority", "centrality_betweenness"), freeassociationdata)

# Pearson correlation between authority and betweeness centrality indexes for the 1st word
# concerning diet.

cor(df.with.centralities2[c("diet1stword_centrality_authority",
"diet1stword_centrality_betweenness")], use="complete.obs", method = "pearson")

# ...or even all available centralities.
# Warning: depending on the number of variables and the size of the data frame, it may be a
# time consuming procedure...

df.with.centralities2 = add.centrality.variables.to.data.frame(wordvars,
"all", freeassociationdata)
```

---

create.centrality.variable

Return a vector with the centrality scores

Description

The function return a vector with the centrality scores of the items (i.e. words) of the variable columnwithoriginalwords. The centrality scores are retreived from the wordreport, an object that has been created previously by the function get.all.graphs.

Usage

```
create.centrality.variable(
  centralityfunctionstr,
  columnwithoriginalwords,
  awordreport,
  verbose = FALSE
)
```

Arguments

`centralityfunctionstr`	The type of the centrality function to use
`columnwithoriginalwords`	The column to look for the words
`awordreport`	The awordreport to look for the centrality of each word
`verbose`	Set to true to print every correspondence

Value

A vector containing the centralities of the words contained in columnwithoriginalwords

Author(s)

Epaminondas Diamantopoulos

Examples

```
# It is a time consuming function...


 allgraphs = get.all.graphs(c("diet1stword", "diet2ndword", "diet3rdword"),
 freeassociationdata)

 centralities_vector1 = create.centrality.variable("centrality_betweenness",
 freeassociationdata$exercise1stword, allgraphs$wordreport)

 centralities_vector2 = create.centrality.variable("centrality_alpha",
 freeassociationdata$exercise1stword, allgraphs$wordreport)

 centralities_vector3 = create.centrality.variable("centrality_authority",
 freeassociationdata$exercise1stword, allgraphs$wordreport)

 centralities_vector4 = create.centrality.variable("centrality_closeness",
 freeassociationdata$exercise1stword, allgraphs$wordreport)
```

```
centralities_vector5 = create.centrality.variable("centrality_pagerank",
freeassociationdata$exercise1stword, allgraphs$wordreport)

centralities_vector6 = create.centrality.variable("centrality_eigen",
freeassociationdata$exercise1stword, allgraphs$wordreport)
```

create_a_new_unique_var_name

Provides variable name that is not already exists in the data frame data

Description

Provides a column name like the likewhatvariable that is not already exists in the data frame data.

Usage

```
create_a_new_unique_var_name(likewhatvariable, data.df)
```

Arguments

`likewhatvariable`	The initial name (string)
`data.df`	The data frame to be added

Value

A string with the suggested name.

Author(s)

Epaminondas Diamantopoulos

Examples

```
newvariablename = create_a_new_unique_var_name(
          "diet1stword",
          freeassociationdata)

print(newvariablename)
```

find.word.code.in.wordreport

Find the index of the word

Description

Search at wordreporttolookat$name and find the index of the word with name awordtolookfor The index is the code of the word in the column.

Equivalent to which(grepl(awordtolookfor, allcentralitiesandgraphs$wordreport$name))

Usage

```
find.word.code.in.wordreport(
  awordtolookfor,
  wordreporttolookat,
  verbose = FALSE
)
```

Arguments

awordtolookfor	The word to look for
wordreporttolookat	The wordreport list to look for the centrality of the word (put the wordreport part of allcentralitiesandgraphs object)
verbose	Set to TRUE if quiet use is not desired. Default is set to FALSE (do not show warning messages)

Value

An integer with the index of the word awordtolookfor

Author(s)

Epaminondas Diamantopoulos

See Also

Equivalent to which(grepl(awordtolookfor, allcentralitiesandgraphs$wordreport$name))

Examples

```
# It is a time consuming function...


allcentralitiesandgraphs = get.all.graphs(c("diet1stword", "diet2ndword", "diet3rdword"),
  freeassociationdata_part)

awordtolookfor = "Diet"
```

```
get.word.centrality(awordtolookfor, "centrality_authority", allcentralitiesandgraphs$wordreport)
```

`freeassociationdata` *Free association data to demonstrate wordgraph package functionality.*

Description

Data that were collected in a research study on social representations. The participants were given a stimulus word regarding eight different life experiences (Diet, Exercise, Smoking, Alcoholic beverage, Nightlife, Lifestyle, Disease and Health) and asked to express three consecutively words associated with the stimulus word. Age, gender, marital status (married) and educational level (secondary) was also recorded. The nine Symptom Check List (SCL - 90R) psychopathology subscales are also administered and included in the data. Permission to make data publicly available was granted from Professor Mary Gouva.

Usage

```
freeassociationdata
```

Format

A data frame with 294 observations on the following 41 variables.

`aa` Index

`age` Age

`gender` Gender

`married` Marital status

`educationallevel` Secondary education

`diet1stword` First word (translated in English) that were associated to the stimulus word "Diet"

`diet2ndword` Second word (translated in English) that were associated to the stimulus word "Diet"

`diet3rdword` Third word (translated in English) that were associated to the stimulus word "Diet"

`exercise1stword` First word (translated in English) that were associated to the stimulus word "Exercise"

`exercise2ndword` Second word (translated in English) that were associated to the stimulus word "Exercise"

`exercise3rdword` Third word (translated in English) that were associated to the stimulus word "Exercise"

`smoking1stword` First word (translated in English) that were associated to the stimulus word "Smoking"

`smoking2ndword` Second word (translated in English) that were associated to the stimulus word "Smoking"

smoking3rdword Third word (translated in English) that were associated to the stimulus word "Smoking"

alcohol1stword First word (translated in English) that were associated to the stimulus word "Alcohol consumption"

alcohol2ndword Second word (translated in English) that were associated to the stimulus word "Alcohol consumption"

alcohol3rdword Third word (translated in English) that were associated to the stimulus word "Alcohol consumption"

nightlife1stword First word (translated in English) that were associated to the stimulus word "Nightlife"

nightlife2ndword Second word (translated in English) that were associated to the stimulus word "Nightlife"

nightlife3rdword Third word (translated in English) that were associated to the stimulus word "Nightlife"

lifestyle1stword First word (translated in English) that were associated to the stimulus word "Lifestyle"

lifestyle2ndword Second word (translated in English) that were associated to the stimulus word "Lifestyle"

lifestyle3rdword Third word (translated in English) that were associated to the stimulus word "Lifestyle"

disease1stword First word (translated in English) that were associated to the stimulus word "Disease"

disease2ndword Second word (translated in English) that were associated to the stimulus word "Disease"

disease3rdword Third word (translated in English) that were associated to the stimulus word "Disease"

health1stword First word (translated in English) that were associated to the stimulus word "Health"

health2ndword Second word (translated in English) that were associated to the stimulus word "Health"

health3rdword Third word (translated in English) that were associated to the stimulus word "Health"

diet1stword_el First word (original Greek) that were associated to the stimulus word "Diet"

diet2ndword_el Second word (original Greek) that were associated to the stimulus word "Diet"

diet3rdword_el Third word (original Greek) that were associated to the stimulus word "Diet"

exercise1stword_el First word (original Greek) that were associated to the stimulus word "Exercise"

exercise2ndword_el Second word (original Greek) that were associated to the stimulus word "Exercise"

exercise3rdword_el Third word (original Greek) that were associated to the stimulus word "Exercise"

smoking1stword_el First word (original Greek) that were associated to the stimulus word "Smoking"

`smoking2ndword_el` Second word (original Greek) that were associated to the stimulus word "Smoking"

`smoking3rdword_el` Third word (original Greek) that were associated to the stimulus word "Smoking"

`alcohol1stword_el` First word (original Greek) that were associated to the stimulus word "Alcohol consumption"

`alcohol2ndword_el` Second word (original Greek) that were associated to the stimulus word "Alcohol consumption"

`alcohol3rdword_el` Third word (original Greek) that were associated to the stimulus word "Alcohol consumption"

`nightlife1stword_el` First word (original Greek) that were associated to the stimulus word "Nightlife"

`nightlife2ndword_el` Second word (original Greek) that were associated to the stimulus word "Nightlife"

`nightlife3rdword_el` Third word (original Greek) that were associated to the stimulus word "Nightlife"

`lifestyle1stword_el` First word (original Greek) that were associated to the stimulus word "Lifestyle"

`lifestyle2ndword_el` Second word (original Greek) that were associated to the stimulus word "Lifestyle"

`lifestyle3rdword_el` Third word (original Greek) that were associated to the stimulus word "Lifestyle"

`disease1stword_el` First word (original Greek) that were associated to the stimulus word "Disease"

`disease2ndword_el` Second word (original Greek) that were associated to the stimulus word "Disease"

`disease3rdword_el` Third word (original Greek) that were associated to the stimulus word "Disease"

`health1stword_el` First word (original Greek) that were associated to the stimulus word "Health"

`health2ndword_el` Second word (original Greek) that were associated to the stimulus word "Health"

`health3rdword_el` Third word (original Greek) that were associated to the stimulus word "Health"

`SOM` Somatization

`OC` Obsessive–Compulsive

`IS` Interpersonal Sensitivity

`DEP` Depression

`ANX` Anxiety

`HOS` Hostility

`PHB` Phobia

`PAR` Paranoid Ideation

`PSY` Psychoticism

`GSI` Global Severity Index

`OAS` Other As Shamer Scale score

`ESS` Experiences of Shame Scale score

Source

The Social Representations of Health as a common Symbolic Place. E. Briseniou, N Skenteris, C. Hatzoglou, E. Diamantopoulos & M. Gouva. (Article 1).

Psychopathology and shame effect on words selected to describe eight common social representations. E. Briseniou, N Skenteris, C. Hatzoglou, E. Diamantopoulos & M. Gouva. (Article 2).

`freq.of.all.combinations.df` *Count all combinations between the values of column1 and column2.*

Description

Count all combinations between the values of column1 and column2 and return a data frame with the combination of the items and the respecting frequencies.

Usage

```
freq.of.all.combinations.df(column1, column2, data.df, order = 0, removeNA = TRUE)
```

Arguments

`column1`	The name of the first column
`column2`	The name of the second column
`data.df`	The data frame containing the columns column1, column2
`order`	0 (default): no order, 1: descenting frequancy order, 2: ascenting wordcode order, 3: ascenting word order.
`removeNA`	Should remove any NA values or count them in their combination? Default is TRUE (remove NA).

Value

A data frame containing the frequencies of all pairs of values between column1 and column2. It counts all pair occurences independent of position, thus the pairs of (column1, column2): (a, b), (a, b), (b, a) will result to the entry in freq.of.all.combinations.df of (word1 = a, word2 = b, freq = 3).

Author(s)

Epaminondas Diamantopoulos

Examples

```
# It is a time consuming function...
# Thus, for illustrative purposes, an example is provided in a small
# random subset of the original data...

freeassociationdata_part = dplyr::sample_n(freeassociationdata, 10)


df.with.frequencies = freq.of.all.combinations.df("diet1stword", "diet2ndword",
freeassociationdata_part)

head(df.with.frequencies)
```

---

freq.of.all.combinations.df.numeric

Special case of freq.of.all.combinations.df

Description

Count all combinations between the values of column1 and column2 and return a data frame with the combination of the items and the respecting frequencies. Special case of freq.of.all.combinations.df for numeric column1 and column2, thus no special reason to call.

Usage

```
freq.of.all.combinations.df.numeric(column1, column2, data.df, order = 0, removeNA = TRUE)
```

Arguments

`column1` The name of the first column

`column2` The name of the second column

`data.df` The data frame containing the columns var1, var2

`order` 0 (default): no order, 1: descenting frequancy order, 2: ascenting wordcode order, 3: ascenting word order.

`removeNA` Should remove any NA values or count them in their combination? Default is TRUE (remove NA).

Value

A data frame containing the frequencies of all pairs of values between column1 and column2. It counts all pair occurences independent of position, thus the pairs of (column1, column2): (a, b), (a, b), (b, a) will result to the entry in freq.of.all.combinations.df of (word1 = a, word2 = b, freq = 3).

Author(s)

Epaminondas Diamantopoulos

See Also

freq.of.all.combinations.df

`get.all.graphs` *Creates one ggraph object for each available centrality functions.*

Description

Creates a ggraph object for each available centrality function. A data frame with the words and all centralities indexes is also included. The combinations of all word pairs are computed for all concecutive pair of variables and the centrality of each word is calculated according to all available centrality functions. The wordreport list gives the opportunity to the researcher to compare centralities indexes for the words.

Usage

```
get.all.graphs(
          wordvars,
          data.df,
          iscircled = FALSE,
          verbose = FALSE)
```

Arguments

`wordvars`	The vector containing the names of the variables
`data.df`	The data frame where the variables belong.
`iscircled`	Should take also the combination between last and first variable (var_n - var1)? Default is FALSE.
`verbose`	Should show also warning messages? The default is FALSE.

Value

A list of objects "graphs" and "wordreport" which are themselves also lists.

`graphs`	The graphs list contains the graph data for each one centrality index.
`wordreport`	The wordreport list contains all centrality indexes for every different word that participates in a combination among wordvars variable vector.

Author(s)

Epaminondas Diamantopoulos

References

http://www.sthda.com/english/articles/33-social-network-analysis/136-network-analysis-and-manipulation-using-r/

See Also

get.graph.1.n.group.centrality.with.function

Examples

```
# It is a time consuming function...


allgraphs = get.all.graphs(c("diet1stword", "diet2ndword", "diet3rdword"),
freeassociationdata)

# Get insight in the relation of the centralities for your word set...
cor(allgraphs$wordreport[2:7])

# Plot a particular graph by giving the command
graph_plot(allgraphs$graphs$centrality_closeness)
```

`get.all.word.pairs.frequency`

Count all combinations between each concecutive pair of variables.

Description

Use multiple times the function freq.of.all.combinations.df to count all combinations between each concecutive pair of variables of wordvars (var1 - var2, var2 - var3, ..., var_n-1 - var_n). Then combines all pair frequencies into one data frame with the columns "word1code" "word2code" "word1" "word2" "cumsums"

Usage

```
get.all.word.pairs.frequency(wordvars, data.df, iscircled = FALSE)
```

Arguments

`wordvars`	The vector containing the names of the variables
`data.df`	The data frame where the variables belong.
`iscircled`	Should take also the combination between last and first variable (var_n - var1)? Default is FALSE.

Value

A data frame contains the columns "word1code" "word2code" "word1" "word2" "cumsums"

Author(s)

Epaminondas Diamantopoulos

Examples

```
# It is a time consuming function...
# Thus, for illustrative purposes, an example is provided in a small
# random subset of the original data...

freeassociationdata_part = dplyr::sample_n(freeassociationdata, 10)


df.all.pairs.frequency = get.all.word.pairs.frequency(c("diet1stword",
"diet2ndword", "diet3rdword"), freeassociationdata_part)

head(df.all.pairs.frequency)
```

---

get.graph.1.n.group.centrality.with.function

Creates a ggraph object.

Description

Creates the ggraph object, i.e. the graph depicting the words containing the variables vector wordvars The combinations of all word pairs are computed for all concecutive pair of variables and the centrality of each word is calculated according to centralityfunction.

Usage

```
get.graph.1.n.group.centrality.with.function(
        wordvars,
        centralityfunction,
        data.df,
        iscircled = FALSE)
```

Arguments

`wordvars` The vector containing the names of the variables

`centralityfunction` The centrality function to apply. Acceptable entries one of: tidygraph::centrality_authority(), tidygraph::centrality_betweenness(), tidygraph::centrality_closeness(), tidygraph::centrality_pagerank(), tidygraph::centrality_eigen(), tidygraph::centrality_alpha()

`data.df` The data frame where the variables belong.

`iscircled` Should take also the combination between last and first variable (var_n - var1)? Default is FALSE.

Value

A list contains the graph data for the selected centrality index.

Author(s)

Epaminondas Diamantopoulos

See Also

get.all.graphs

Examples

```
# It is a time consuming function...


agraph = get.graph.1.n.group.centrality.with.function(
    c("diet1stword", "diet2ndword", "diet3rdword"),
    tidygraph::centrality_authority(),
    freeassociationdata)

# To plot the graph simply call:
agraph

# The centrality scores are available at:agraph$data$centrality
hist(agraph$data$centrality)

# Other available centrality functions...

 agraph = get.graph.1.n.group.centrality.with.function(
             c("diet1stword", "diet2ndword", "diet3rdword"),
            tidygraph::centrality_betweenness(), freeassociationdata)

# Note: closeness centrality is not well-defined for disconnected graphs.
 agraph = get.graph.1.n.group.centrality.with.function(
             c("diet1stword", "diet2ndword", "diet3rdword"),
             tidygraph::centrality_closeness(), freeassociationdata)

 agraph = get.graph.1.n.group.centrality.with.function(
            c("diet1stword", "diet2ndword", "diet3rdword"),
            tidygraph::centrality_pagerank(), freeassociationdata)

 agraph = get.graph.1.n.group.centrality.with.function(
             c("diet1stword", "diet2ndword", "diet3rdword"),
             tidygraph::centrality_eigen(), freeassociationdata)

 agraph = get.graph.1.n.group.centrality.with.function(
            c("diet1stword", "diet2ndword", "diet3rdword"),
            tidygraph::centrality_alpha(), freeassociationdata)
```

get.word.centrality *Returns the centrality of the word.*

Description

Loop into the entries of the wordreporttolookat, find the centrality function described by centralityfunctionstr and returns the centrality of the word. If there is not a corresponding code then the function return NA

Usage

```
get.word.centrality(
  awordtolookfor,
  centralityfunctionstr,
  wordreporttolookat,
  verbose = FALSE)
```

Arguments

awordtolookfor	The word to look for
centralityfunctionstr	The centrality function description string
wordreporttolookat	The wordreport list to look for the centrality of the word
verbose	Set to TRUE if quiet use is not desired. Default is set to FALSE (do not show warning messages)

Value

A double with the centrality of the word awordtolookfor

Examples

```
# It is a time consuming function...

freeassociationdata_part = dplyr::sample_n(freeassociationdata, 10)

  allcentralitiesandgraphs = get.all.graphs(
                   c("diet1stword", "diet2ndword", "diet3rdword"),
                   freeassociationdata_part)

get.word.centrality("Enjoyment", "centrality_authority",
                       allcentralitiesandgraphs$wordreport)
```

get.word.combinations.as.tbl_graph *Calculate all word pairs frequencies.*

Description

Calculate all word pairs frequencies among concecutive variables in wordvars vector and return a tidygraph::as_tbl_graph object in order to be used to create the graph.

Usage

```
get.word.combinations.as.tbl_graph(
                        wordvars,
                        data.df,
                        iscircled = FALSE)
```

Arguments

`wordvars`	The vector containing the names of the variables
`data.df`	The data frame where the variables belong.
`iscircled`	Should take also the combination between last and first variable (var_n - var1)? Default is FALSE.

Value

A tidygraph::as_tbl_graph object containing the frequencies of all value pairs between concecutive variables in wordvars vector.

Author(s)

Epaminondas Diamantopoulos

Examples

```
# It is a time consuming function...
# Thus, for illustrative purposes, an example is provided in a
# small random subset of the original data...

freeassociationdata_part = dplyr::sample_n(freeassociationdata, 10)

df.word.combinations.as.tbl_graph = get.word.combinations.as.tbl_graph(
c("diet1stword", "diet2ndword", "diet3rdword"), freeassociationdata_part)

print(df.word.combinations.as.tbl_graph)
```

get_label_from_columnrow *Returns the label of a value.*

Description

Finds the label (string) of a variable specific value at the row. If there is not a label then the function return the value. If there is not value or the nrow is not in an acceptable range then it returns NA.

Usage

```
get_label_from_columnrow(
              arow,
              column.of.data.frame,
              verbose = FALSE)
```

Arguments

`arow`	The row to look at
`column.of.data.frame`	The column name to look for value labels
`verbose`	Set to TRUE if quiet use is not desired. Default is set to FALSE (do not show warning messages

Value

A string with the label of the value

Author(s)

Epaminondas Diamantopoulos

Examples

```
get_label_from_columnrow(21, freeassociationdata$gender)
```

graph_plot *Plots a graph.*

Description

Plots a graph created by get.all.graphs or get.grap h.1.n.group.centrality.with.function functions.

Usage

```
graph_plot(agraph_object)
```

Arguments

`agraph_object`	A graph object

Author(s)

Epaminondas Diamantopoulos

See Also

get.all.graphs

Examples

```
# The function graph_plot provides graph functionality for a graph
# object created by get.all.graphs function.


# First give...
 allgraphs = get.all.graphs(c("diet1stword", "diet2ndword",
     "diet3rdword"), freeassociationdata)

# Plot a particular graph by giving...
graph_plot(allgraphs$graphs$centrality_closeness)
```

`prepare.file` *Remove all the multiple blanks of the selected R output, in order to copied correctly to LibreOffice Calc.*

Description

Remove all the multiple blanks of the selected R output, in order to copied correctly to LibreOffice Calc.

Usage

```
prepare.file(afile)
```

Arguments

`afile`	A file

Value

The same file with reduced blank spaces.

Index