

# Package ‘String2AdjMatrix’

January 30, 2018

**Type** Package

**Title** Creates an Adjacency Matrix from a List of Strings

**Version** 0.1.0

**Author** Tom Drake

**Maintainer** Tom Drake <t.drake@ed.ac.uk>

**Description** Takes a list of character strings and forms an adjacency matrix for the times the specified characters appear together in the strings provided. For use in social network analysis and data wrangling. Simple package, comprised of three functions.

**License** GPL-3

**Encoding** UTF-8

**Depends** stringr

**LazyData** true

**RoxygenNote** @import stringr 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-01-30 10:24:44 UTC

## R topics documented:

|                               |          |
|-------------------------------|----------|
| generate_adj_matrix . . . . . | 2        |
| string_2_matrix . . . . .     | 3        |
| <b>Index</b>                  | <b>5</b> |

---

generate\_adj\_matrix    *generate\_adj\_matrix*

---

## Description

Generates a blank adjacency matrix from a specified string

## Usage

```
generate_adj_matrix(string_data, data_separator = ", ", remove_spaces = F)
```

## Arguments

`string_data`    The 'string\_data' argument is the string from which the unique values and matrix will be generated.

`data_separator`    The 'data\_separator' argument is the character separating specified substrings in the given string. Default is ','.

`remove_spaces`    The 'remove\_spaces' argument will remove spaces from the header values (thus disrupting the search unless all spaces are removed in the given string in next steps). This is useful for separating strings with an irregular number of spaces between the same substrings.

## Details

Generates an adjacency matrix from a given string. Detects unique values and generates a blank matrix with colnames and rownames of each unique value in supplied string. Data must be provided as a character string.

## Author(s)

Tom Drake

## Examples

```
##Example
library(String2AdjMatrix)

#Start with character string to generate an adjacency matrix from
string_in = c('apples, pears, bananas', 'apples, bananas', 'apples, pears')

#Generate a new blank matrix
blank_matrix = generate_adj_matrix(string_in)

#Now fill the matrix
string_2_matrix(blank_matrix, string_in)
```

---

|                 |                        |
|-----------------|------------------------|
| string_2_matrix | <i>string_2_matrix</i> |
|-----------------|------------------------|

---

**Description**

Creates an adjacency matrix

**Usage**

```
string_2_matrix(new_matrix, supplied_string, self = 0)
```

**Arguments**

|                              |  |
|------------------------------|--|
| <code>new_matrix</code>      | The 'new_matrix' element of the function should be either the matrix generated by 'generate_adj_matrix()' or an empty data matrix of equal number of rows and columns. These should have unique values specified as the row names and column names.                      |
| <code>supplied_string</code> | The 'supplied_string' element refers to the string in which the search is to be performed. i.e 'list = c('apples, pears, bananas', 'apples, bananas', 'apples, pears)'   |
| <code>self</code>            | The 'self' option specifies how to handle data when the specified object is found within a string. Default is 0. i.e. the adjacency matrix does not count it when the substring is found, only when the substring is found in combination with another unique substring. |

**Value**

An adjacency matrix

**Note**

Generating large matrices is computationally intensive and may take a while.

**Author(s)**

Tom Drake

**Examples**

```
##Example
library(String2AdjMatrix)

#Start with character string to generate an adjacency matrix from
string_in = c('apples, pears, bananas', 'apples, bananas', 'apples, pears')

#Generate a new blank matrix
blank_matrix = generate_adj_matrix(string_in)
```

```
#Now fill the matrix  
string_2_matrix(blank_matrix, string_in)
```

# Index

`generate_adj_matrix`, [2](#)

`string_2_matrix`, [3](#)

`string_2_matrix_x(string_2_matrix)`, [3](#)