# Package ‘textworks’

October 2, 2017

**Title** All-in-one package for working with text data

**Version** 0.0.0.9000

**Description** In a nutshell? Do cool things with text data.

**Depends** R (>= 3.3.3)

**License** What license is it under?

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** jsonlite,
XML,
tidyverse,
httr,
magrittr

## R topics documented:

## breakitdown *Contingency table by referencing data and variables*

### Description

Produce a contingency table with percentage signs.

### Usage

```
breakitdown(x, var_org, var_per)
```

### Arguments

| | |
|---|---|
| x | Data frame to be passed through |
| var_org | The organising variable, i.e. variable for grouping categories.<br>For instance, age could be the grouping variable if you want to compare frequencies across age-groups. |
| var_per | The variable for which you want the percentages / frequencies. |

### Examples

```
breakitdown(mtcars,am,gear)
# A tibble: 4 x 4
# Groups:   am [2]
     am  gear     n rel.freq
* <dbl> <dbl> <int>    <chr>
1     0     3    15      79%
2     0     4     4      21%
3     1     4     8      62%
4     1     5     5      38%
```

## ctab *Pass a variable and return a contingency table by column percentages*

### Description

Use vector/variable as argument, and a contingency table by column percentages is returned as output. Function still works even if variable is not of factor class (unless it cannot be converted).

### Usage

```
ctab(vec)
```

### Arguments

| | |
|---|---|
| vec | Vector to be passed through. |

### Examples

```
Q12 <- sample(c(”Sandwich”,”Porridge”,”Muesli/cereal”,”Other”,”No breakfast”),2000,replace=TRUE,prob=c(0.2,0
Q12 <- factor(Q12)
ctab(Q12)
              Q   Prop
1 Muesli/cereal  0.201
2  No breakfast  0.249
3         Other 0.1485
4      Porridge  0.201
5      Sandwich  0.249
```

## `ctab.mc` *Contingency table for a multi-code question*

### Description

Use a data frame as argument, and a contingency table by column percentages is returned as output. First column will display the codes for the question Each subsequent column of the table will show the column percentages for each question code.

### Usage

```
ctab.mc(x)
```

### Arguments

`vec` Data frame to be passed through. Use select() to remove any unnecessary columns (including ID).

### Examples

```
Coming soon
```

## `fa.load` *Function to create a loadings file from the factanal() output*

### Description

Function to create a loadings file from the factanal() output

### Usage

```
fa.load(x, file_name = NULL, writeCSV = FALSE)
```

### Arguments

`x` factanal() model

fa.score *Function to create a score file from the factanal() output*

## Description

Function to create a score file from the factanal() output

## Usage

```
fa.score(x, file_name = NULL, writeCSV = FALSE)
```

## Arguments

| | |
|---|---|
| x | factanal() model |

hclustfunc *Run hierarchical clustering with arguments to specify methods.*

## Description

Run hierarchical clustering with arguments to specify methods.

## Usage

```
hclustfunc(x, method = "complete", dmeth = "euclidean")
```

likert.convert *Convert a Likert scale from one scale to another*

## Description

This is used for converting data, for instance, from a 6-point scale to a 5-point scale. The scale "dimensions" are specified in the function argument, and the function converts the numeric vector that is passed through.

## Usage

```
likert.convert(x, top.x, bot.x, top.y, bot.y)
```

## Arguments

| | |
|---|---|
| x | Numeric vector to be passed through. |
| top.x | Top value of the original scale. This would be 6 on a 0-6 scale. |
| bot.x | Bottom value of the original scale. This would be 0 on a 0-6 scale. |
| top.y | Top value of the new/target scale. This would be 5 on a 0-5 scale. |
| bot.y | Bottom value of the new/target scale. This would be 0 on a 0-5 scale. |

## Examples

```
data <-c(5, 4, 3, 2, 1)
likert.scaler(data,5,0,10,0) #5-point scale to 10-point scale
[1] 10  8  6  4  2
```

---

## `likert.reverse` *Reverse a Likert scale*

---

### Description

Reverse a Likert scale such that on a 0-10 scale, 10 becomes 0 and 0 becomes 10.

### Usage

```
likert.reverse(x, top, bottom)
```

### Arguments

| | |
|---|---|
| `x` | Numeric vector |
| `top` | Top value of the scale for the variable, e.g. 10 for a 0-10 scale. |
| `top` | Bottom value of the scale for the variable, e.g. 0 for a 0-10 scale. |

### Examples

```
data <-c(5, 4, 3, 2, 6)
likert.reverse(data, 6, 0)
[1] 1 2 3 4 0
```

---

## `maxmin` *Max-Min Scaling Function*

---

### Description

This function allows you to scale vectors or an entire data frame using the max-min scaling method, always returning a data frame.

### Usage

```
maxmin(x)
```

### Arguments

| | |
|---|---|
| `x` | Pass a vector or the required columns of a data frame through this argument. |

### Examples

```
rand.data <-cbind(sample(1000,234:697),sample(1000,234:697)) %>% as.data.frame()
maxmin(rand.data)

rand.data <-sample(1000,234:677)
maxmin(rand.data)
```

## plain_num *Function to convert all relevant columns to numeric Create a plain data frame without GUID and category*

### Description

Function to convert all relevant columns to numeric Create a plain data frame without GUID and category

### Usage

```
plain_num(x)
```

### Arguments

| | |
|---|---|
| x | data frame to pass through |

## split.tt *Split the data into a simple training and testing set*

### Description

Split the data into a simple training and testing set

### Usage

```
## S3 method for class 'tt'
split(x, part)
```

### Arguments

| | |
|---|---|
| x | Pass your data frame or matrix here. |
| part | A numeric value between 0 and 1 to represent the proportion of the whole data you want to use as the training set. |

### Examples

```
x <-as.data.frame(matrix(1:5000,250,20))
x.train <- split.tt(x,.7)$train
x.test <- split.tt(x,.7)$test

dim(x.train)
dim(x.test)
```

## trans.m *Translate function using the Microsoft Translator API*

### Description

This function allows you to translate character strings through the Microsoft Translator API. To use this function, you will need to first acquire an access token and pass it through the key argument.

### Usage

```
trans.m(what, from = NULL, to = "en", key = MyKey)
```

### Arguments

| | |
|---|---|
| what | What text string to translate |
| from | What language to translate from. Defaults to NULL |
| to | What language to translate to. Defaults to "en" |

### Examples

```
trans.m(what="Blanc",from="fr",to="en",key="F023kljadfoilkjlkj")
```

## trans.t *Translate function using transltr.org API*

### Description

This function allows you to translate character strings through the transltr.org API.

### Usage

```
trans.t(what = NULL, from = NULL, to = "en")
```

### Arguments

| | |
|---|---|
| what | What text string to translate |
| from | What language to translate from. Defaults to NULL |
| to | What language to translate to. Defaults to "en" |

### Examples

```
trans.t(what="Blanc",from="fr",to="en")
```

## trans.transltr *Translate function using transltr.org API*

### Description

This function allows you to translate character strings through the transltr.org API.

### Usage

```
trans.transltr(what, from = NULL, to)
```

### Arguments

| | |
|---|---|
| `what` | What tex string to translate |
| `from` | What language to translate from. Defaults to NULL |
| `to` | What language to translate to? Mandatory |

### Examples

```
trans.transltr()
```

# Index