Package ‘wordcloud’

February 20, 2015

Type Package
Title Word Clouds
Version 2.5
Date 2013-04-11
Author Ian Fellows
Maintainer Ian Fellows <ian@fellstat.com>
Description Pretty word clouds.
License LGPL-2.1
LazyLoad yes
Depends methods, RColorBrewer
Imports slam, Rcpp (>= 0.9.4)
Suggests tm (>= 0.6)
http://research.cens.ucla.edu/
LinkingTo Rcpp
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-06-13 00:56:24

R topics documented:

commonality.cloud ......................................................... 2
comparison.cloud .......................................................... 3
SOTU ........................................................................ 4
textplot ....................................................................... 4
wordcloud ..................................................................... 5
wordlayout ..................................................................... 7

Index 9
commonality.cloud  Plot a commonality cloud

Description
Plot a cloud of words shared across documents

Usage
commonality.cloud(term.matrix, comonality.measure=min, max.words=300,...)

Arguments
- term.matrix: A term frequency matrix whose rows represent words and whose columns represent documents.
- comonality.measure: A function taking a vector of frequencies for a single term, and returning a common frequency.
- max.words: Maximum number of words to be plotted. Least frequent terms dropped.
- ...: Additional parameters to be passed to wordcloud.

Value
nothing

Examples
if(require(tm)){
  data(SOTU)
  corp <- SOTU
  corp <- tm_map(corp, removePunctuation)
  corp <- tm_map(corp, content_transformer(tolower))
  corp <- tm_map(corp, removeNumbers)
  corp <- tm_map(corp, function(x)removeWords(x, stopwords()))

  term.matrix <- TermDocumentMatrix(corp)
  term.matrix <- as.matrix(term.matrix)
  colnames(term.matrix) <- c("SOTU 2010","SOTU 2011")
  comparison.cloud(term.matrix,max.words=40,random.order=FALSE)
  commonality.cloud(term.matrix,max.words=40,random.order=FALSE)
}
comparison.cloud

Plot a comparison cloud

Description

Plot a cloud comparing the frequencies of words across documents.

Usage

```r
comparison.cloud(term.matrix, scale=c(4,.5), max.words=300, 
random.order=FALSE, rot.per=.1, 
colors=brewer.pal(ncol(term.matrix),"Dark2"),
use.r.layout=FALSE,title.size=3,...)
```

Arguments

- `term.matrix`: A term frequency matrix whose rows represent words and whose columns represent documents.
- `scale`: A vector of length 2 indicating the range of the size of the words.
- `max.words`: Maximum number of words to be plotted. least frequent terms dropped.
- `random.order`: plot words in random order. If false, they will be plotted in decreasing frequency.
- `rot.per`: proportion words with 90 degree rotation.
- `colors`: color words from least to most frequent.
- `use.r.layout`: if false, then c++ code is used for collision detection, otherwise R is used.
- `title.size`: Size of document titles.
- `...`: Additional parameters to be passed to text (and strheight,strwidth).

Details

Let $p_{i,j}$ be the rate at which word i occurs in document j, and $p_j$ be the average across documents($\sum_i p_{i,j} / ndocs$). The size of each word is mapped to its maximum deviation ($max_j[p_{i,j} - p_j]$), and its angular position is determined by the document where that maximum occurs.

Value

nothing

Examples

```r
if(require(tm)){
data(SOTU)
corp <- SOTU
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, content_transformer(tolower))
corp <- tm_map(corp, removeNumbers)
```
corp <- tm_map(corp, function(x) removeWords(x, stopwords()))

term.matrix <- TermDocumentMatrix(corp)
term.matrix <- as.matrix(term.matrix)
colnames(term.matrix) <- c("SOTU 2010", "SOTU 2011")
comparison.cloud(term.matrix, max.words=40, random.order=FALSE)
commonality.cloud(term.matrix, max.words=40, random.order=FALSE)

---

SOTU  
*United States State of the Union Addresses (2010 and 2011)*

**Description**

Transcripts of the state of the union speeches. saved as a tm Corpus.

**Usage**

data(SOTU)

**Author(s)**

Barack Obama

---

**textplot**  
*Text Plot*

**Description**

An x y plot of non-overlapping text

**Usage**

textplot(x, y, words, cex=1, new=TRUE, show.lines=TRUE, ...)

**Arguments**

- **x**: x coordinates
- **y**: y coordinates
- **words**: the text to plot
- **cex**: font size
- **new**: should a new plot be created
- **show.lines**: if true, then lines are plotted between x,y and the word, for those words not covering their x,y coordinates
- **...**: Additional parameters to be passed to wordlayout and text.
wordcloud

Value

nothing

Examples

# calculate standardized MDS coordinates
dat <- sweep(USArrests, 2, colMeans(USArrests))
dat <- sweep(dat, 2, sqrt(diag(var(dat))))
loc <- cmdscale(dist(dat))

# plot with no overlap
textplot(loc[, 1], loc[, 2], rownames(loc))

# scale by urban population size
textplot(loc[, 1], loc[, 2], rownames(loc), cex = USArrests$UrbanPop/max(USArrests$UrbanPop))

# x limits sets x bounds of plot, and forces all words to be in bounds
textplot(loc[, 1], loc[, 2], rownames(loc), xlim = c(-3.5, 3.5))

# compare to text (many states unreadable)
plot(loc[, 1], loc[, 2], type = "n")
text(loc[, 1], loc[, 2], rownames(loc))

wordcloud  
Plot a word cloud

Description

Plot a word cloud

Usage

wordcloud(words, freq, scale = c(4, .5), min.freq = 3, max.words = Inf,
random.order = TRUE, random.color = FALSE, rot.per = .1,
colors = "black", ordered.colors = FALSE, use.r.layout = FALSE,
fixed.asp = TRUE, ...)

Arguments

words  the words
freq  their frequencies
scale  A vector of length 2 indicating the range of the size of the words.
min.freq  words with frequency below min.freq will not be plotted
max.words  Maximum number of words to be plotted. least frequent terms dropped
random.order  plot words in random order. If false, they will be plotted in decreasing frequency
random.color  choose colors randomly from the colors. If false, the color is chosen based on the frequency
rot.per  proportion words with 90 degree rotation
colors  color words from least to most frequent
ordered.colors  if true, then colors are assigned to words in order
use.r.layout  if false, then c++ code is used for collision detection, otherwise R is used
fixed.asp  if TRUE, the aspect ratio is fixed. Variable aspect ratio only supported if rot.per==0
...  Additional parameters to be passed to text (and strheight,strwidth).

Details
If freq is missing, then words can either be a character vector, or Corpus. If it is a vector and freq is missing, standard stop words will be removed prior to plotting.

Value
nothing

See Also
text

Examples

```r
wordcloud(c(letters, LETTERS, 0:9), seq(1, 1000, len = 62))
if(require(tm)){

    "Many years ago the great British explorer George Mallory, who was to die on Mount Everest, was asked why did he want to climb it. He said, \"Because it is there.\"

    Well, space is there, and we're going to climb it, and the moon and the planets are there, and new hopes for knowledge and peace are there. And, therefore, as we set sail we ask God's blessing on the most hazardous and dangerous and greatest adventure on which man has ever embarked.\",
    ,random.order=FALSE)

    data(crude)
    crude <- tm_map(crude, removePunctuation)
    crude <- tm_map(crude, function(x)removeWords(x, stopwords()))

    wordcloud(crude)
```
from frequency counts

tdm <- TermDocumentMatrix(crude)
m <- as.matrix(tdm)
v <- sort(rowSums(m), decreasing=TRUE)
d <- data.frame(word = names(v), freq=v)

wordcloud(d$word, d$freq)

# A bigger cloud with a minimum frequency of 2
wordcloud(d$word, d$freq,c(8,3),2)

# Now lets try it with frequent words plotted first
wordcloud(d$word, d$freq,c(8,.5),2,.FALSE,.1)

### with colors ###
if(require(RColorBrewer)){

pal <- brewer.pal(9,"BuGn")
pal <- pal[-(1:4)]
wordcloud(d$word, d$freq,c(8,3),2,.FALSE,.15,pal)

pal <- brewer.pal(6,"Dark2")
pal <- pal[-(1)]
wordcloud(d$word, d$freq,c(8,3),2,.TRUE,.15,pal)

# random colors
wordcloud(d$word, d$freq,c(8,3),2,.TRUE,.15,pal)
}

### with font ###
wordcloud(d$word, d$freq,c(8,3),2,.TRUE,.15,pal, vfont=c("gothic english","plain"))
wordcloud(d$word, d$freq,c(8,3),2,100,.TRUE,.15,pal,vfont=c("script","plain"))
wordcloud(d$word, d$freq,c(8,3),2,100,.TRUE,.15,pal,vfont=c("serif","plain"))

# End(Not run)
}

---

**wordlayout**

### Description ###

finds text plot layout coordinates such that no text overlaps
Usage

```r
wordlayout(x, y, words, cex=1, rotate90 = FALSE,
            xlim=c(-Inf,Inf), ylim=c(-Inf,Inf), tstep=.1, rstep=.1, ...)
```

Arguments

- `x`: x coordinates
- `y`: y coordinates
- `words`: the text to plot
- `cex`: font size
- `rotate90`: a value or vector indicating whether words should be rotated 90 degrees
- `xlim`: x axis bounds for text
- `ylim`: y axis bounds for text
- `tstep`: the angle (theta) step size as the algorithm spirals out
- `rstep`: the radius step size (in standard deviations) as the algorithm spirals out
- `...`: Additional parameters to be passed to strwidth and strheight.

Value

A matrix with columns representing x, y width and height.

Examples

```r
# calculate standardized MDS coordinates
dat <- sweep(USArrests, 2, colMeans(USArrests))
dat <- sweep(dat, 2, sqrt(diag(var(dat))), "/")
loc <- cmdscale(dist(dat))
x <- loc[, 1]
y <- loc[, 2]
w <- rownames(loc)

# plot with no overlap and all words visible
plot(x,y,type="n",xlim=c(-3,3),ylim=c(-3,2))
lay <- wordlayout(x,y,w,xlim=c(-3,3),ylim=c(-3,2))
text(lay[,1]+.5*lay[,3],lay[,2]+.5*lay[,4],w)

# notice north dakota is only partially visible
textplot(x,y,w)
```
Index

commonality.cloud, 2
comparison.cloud, 3

SOTU, 4

text, 6
textplot, 4

wordcloud, 5
wordlayout, 7