

# Package ‘clustRcompaR’

July 20, 2018

**Type** Package

**Title** Easy Interface for Clustering a Set of Documents and Exploring Group- Based Patterns

**Version** 0.2.0

**Date** 2018-01-23

**Author** Joshua Rosenberg, Alex Lishinski

**Maintainer** Alex Lishinski <alexlishinski@gmail.com>

**Description** Provides an interface to perform cluster analysis on a corpus of text. Interfaces to Quanteda to assemble text corpuses easily. Deviationlizes text vectors prior to clustering using technique described by Sherin (Sherin, B. [2013]. A computational study of commonsense science: An exploration in the automated analysis of clinical interview data. Journal of the Learning Sciences, 22(4), 600-638. Chicago. <doi:10.1080/10508406.2013.836654>). Uses cosine similarity as distance metric for two stage clustering process, involving Ward's algorithm hierarchical agglomerative clustering, and k-means clustering. Selects optimal number of clusters to maximize ``variance explained" by clusters, adjusted by the number of clusters. Provides plotted output of clustering results as well as printed output. Assesses ``model fit" of clustering solution to a set of preexisting groups in dataset.

**License** GPL-3

**Depends** R (>= 3.1.3),

**URL** <https://github.com/alishinski/clustRcompaR>

**Imports** quanteda, dplyr, ggplot2, ppls

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**LazyData** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-01-28 21:04:49

## R topics documented:

assemble_corpus . . . . .	2
clean_dfm . . . . .	3
cluster . . . . .	3
cluster_text . . . . .	4
compare . . . . .	5
compare_plot . . . . .	6
compare_test . . . . .	6
deviationalize . . . . .	7
extract_terms . . . . .	7
inaugural_addresses . . . . .	8
<b>Index</b>	<b>9</b>

---

assemble_corpus	<i>First corpus building function</i>
-----------------	---------------------------------------

---

### Description

First corpus building function

### Usage

```
assemble_corpus(data, stopwords, remove_twitter)
```

### Arguments

data	The data from which the corpus is drawn with documents in first column
stopwords	Words to exclude from the clustering
remove_twitter	Whether to remove text associated with Twitter content, useful for when analyzing data from this source (defaults to FALSE)
...	The metadata columns following the text column

### Details

Puts together the corpus and dfm from the data frame provided

---

clean_dfm	<i>Cleans the DFM based on specified term minimums</i>
-----------	--

---

**Description**

Cleans the DFM based on specified term minimums

**Usage**

```
clean_dfm(corp, minimum_term_frequency, min_terms)
```

**Arguments**

corp	A corpus object as created by <code>assemble_corpus</code> .
minimum_term_frequency	Minimum number of occurrences for a term to be used
min_terms	Minimum number of terms for document to be used

**Details**

Removes terms and documents that don't meet term and doc minimums

---

cluster	<i>Cluster wrapper function</i>
---------	---------------------------------

---

**Description**

Cluster wrapper function

**Usage**

```
cluster(data, ..., n_clusters, minimum_term_frequency = 3, min_terms = 3,
        num_terms = 10, stopwords = NULL, remove_twitter = FALSE)
```

**Arguments**

data	The data frame comparing the text vector as the first column
...	Additional columns of the data frame containing metadata cfor comparison
n_clusters	The number of clusters to be used for the clustering solution
minimum_term_frequency	The minimum number of occurrences for a term to be included
min_terms	The minimum number of terms for a document to be included
num_terms	Number of terms to display in clustering summary output
stopwords	Additional stopwords to exclude from clustering analysis
remove_twitter	Whether to remove text associated with Twitter content, useful for when analyzing data from this source (defaults to FALSE)

**Details**

Performs the clustering half of the process, including assembling and cleaning the corpus, deviatonalizing and clustering.

**Examples**

```
library(clustRcompaR)
library(dplyr)
library(quanteda)

d <- inaugural_addresses
d <- mutate(d, century = ifelse(Year < 1800, "17th",
                               ifelse(Year >= 1800 & Year < 1900, "18th",
                                       ifelse(Year >= 1900 & Year < 2000, "19th", "20th"))))

three_clusters <- cluster(d, century, n_clusters = 3)
extract_terms(three_clusters)

three_clusters_comparison <- compare(three_clusters, "century")
compare_plot(three_clusters_comparison)
```

---

cluster\_text

*Clusters the vectors using 2-stage clustering algorithm*


---

**Description**

Clusters the vectors using 2-stage clustering algorithm

**Usage**

```
cluster_text(mat, dev_mat, n_clusters, cleanDFM, num_terms)
```

**Arguments**

mat	The clean dfm as a matrix and transposed, from deviatonalize
dev_mat	The deviation matrix of the dfm, from deviatonalize
n_clusters	number of desired clusters
cleanDFM	DFM object from clean_dfm function
num_terms	Minimum number of terms per document

**Details**

Applies 2 stage clustering algorithm, using Ward's method for hierarchical agglomerative clustering to set the centers for the specified number of clusters. K-means algorithm uses these centers as a starting point and fits its model. @export

---

compare	<i>Compare wrapper function</i>
---------	---------------------------------

---

## Description

Compare wrapper function

## Usage

```
compare(clustering_solution, compare_which, which_clusters = NULL,  
        which_groups = NULL)
```

## Arguments

`clustering_solution` The output from the cluster function.

`compare_which` A factor variable of the groups of interest for comparison.

`which_clusters` Clusters to be included in the comparison. Default is all clusters.

`which_groups` Levels of the grouping factor to be included in the comparison. Default is all levels.

## Details

Function for comparing clustering solution between subgroups. Output is contingency table for the specified groups and clusters.

## Examples

```
library(clustRcompaR)  
library(dplyr)  
library(quanteda)  
  
d <- inaugural_addresses  
d <- mutate(d, century = ifelse(Year < 1800, "17th",  
                               ifelse(Year >= 1800 & Year < 1900, "18th",  
                                     ifelse(Year >= 1900 & Year < 2000, "19th", "20th"))))  
  
three_clusters <- cluster(d, century, n_clusters = 3)  
extract_terms(three_clusters)  
  
three_clusters_comparison <- compare(three_clusters, "century")  
compare_plot(three_clusters_comparison)
```

---

compare_plot	<i>Compare plot function</i>
--------------	------------------------------

---

**Description**

Compare plot function

**Usage**

```
compare_plot(comparison_table)
```

**Arguments**

comparison\_table  
The table output from the compare function

**Details**

Creates a plot visualizing group clustering differences across the groups and clusters specified in the compare function. Creates a ggplot object, so default parameters can be overridden by adding layers to this object.

---

compare_test	<i>Compare test function</i>
--------------	------------------------------

---

**Description**

Compare test function

**Usage**

```
compare_test(comparison_table)
```

**Arguments**

comparison\_table  
The table output from the compare function

**Details**

Performs a chi-squared test across the groups and clusters specified in the compare function. Output gives omnibus test results and a table indicating significant individual chi-squared differences.

---

deviationalize	<i>Deviationalizes term vectors using Sherin's (2013) technique</i>
----------------	---

---

**Description**

Deviationalizes term vectors using Sherin's (2013) technique

**Usage**

```
deviationalize(cleaned_dfm)
```

**Arguments**

cleaned\_dfm     A clean dfm object as created by clean\_dfm

**Details**

Turns term vectors into deviation vectors. This turns the magnitude of each vector into a representation of its distance from the centroid, rather than its absolute direction. @export

---

extract_terms	<i>Extracts terms and term frequencies</i>
---------------	--

---

**Description**

Extracts terms and term frequencies

**Usage**

```
extract_terms(object)
```

**Arguments**

object            output from the cluster() function

**Details**

Extracts the terms and term frequencies from the output of the cluster() function

---

inaugural\_addresses    *Data on the inaugural addresses by every United States President  
(from the quanteda package)*

---

**Description**

Data on the inaugural addresses by every United States President (from the quanteda package)

**Usage**

inaugural\_addresses

**Format**

Data frame with columns #'

**texts** text contents of the inaugural addresses

**Year** year of the address

**President** last name of the President

**FirstName** first name of the President

**Source**

<http://docs.quanteda.io/>

# Index

## \*Topic **datasets**

inaugural\_addresses, 8

assemble\_corpus, 2

clean\_dfm, 3

cluster, 3

cluster\_text, 4

compare, 5

compare\_plot, 6

compare\_test, 6

deviationalize, 7

extract\_terms, 7

inaugural\_addresses, 8