

# Package ‘textanalyzer’

July 22, 2025

**Type** Package

**Title** 'textanalyzer', an R Package to Analyze Text

**Version** 0.2.0

**Description**

It analyzes text to create a count of top n-grams, including tokens (one-word), bigrams(two-word), and trigrams (three-word), while removing all stopwords. It also plots the n-grams and corresponding counts as a bar chart.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** tidytext, tidyr, dplyr, ggplot2, utils, stats

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Pushker Ravindra [aut, cre]

**Maintainer** Pushker Ravindra <pushker@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-01-29 17:20:02 UTC

## Contents

analyze_bigrams . . . . .	2
analyze_ngrams . . . . .	2
analyze_tokens . . . . .	3
analyze_trigrams . . . . .	4
plot_ngrams . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

analyze\_bigrams      *Analyze Bigrams*

---

**Description**

Analyze text with ngram=2 (bigrams).

**Usage**

```
analyze_bigrams(in_text, top_rows = 25)
```

**Arguments**

in\_text            a character vector. Text to be analyzed as a character vector.  
top\_rows           a numeric vector of length 1. Number of top rows to be returned.

**Details**

analyze\_bigrams

**Value**

A data.frame with two columns - bigram (character vector) and count (numeric vector).

**Author(s)**

Ravindra Pushker

**Examples**

```
analyze_bigrams(in_text=c("The quick brown fox jumps over the lazy dog."))
```

---

analyze\_ngrams      *Analyze NGrams*

---

**Description**

Analyze text with ngram among 1, 2 or 3.

**Usage**

```
analyze_ngrams(in_text, ngram = 1, top_rows = 25)
```

**Arguments**

in\_text a character vector. Text to be analyzed as a character vector.  
 ngram a numeric\_vector of length 1. Ngram = 1, 2 or 3.  
 top\_rows a numeric vector of length 1. Number of top rows to be returned.

**Details**

analyze\_ngrams

**Value**

A data.frame with two columns - word/bigram/trigram (character vector) and count (integer vector).

**Author(s)**

Ravindra Pushker

**Examples**

```
analyze_ngrams(in_text=c("The quick brown fox jumps over the lazy dog."))
```

---

analyze_tokens	<i>Analyze Tokens</i>
----------------	-----------------------

---

**Description**

Analyze text with ngram=1

**Usage**

```
analyze_tokens(in_text, top_rows = 25)
```

**Arguments**

in\_text a character vector. Text to be analyzed as a character vector.  
 top\_rows a numeric vector of length 1. Number of top rows to be returned.

**Details**

analyze\_tokens

**Value**

A data.frame with two columns - word (character vector) and count (numeric vector).

**Author(s)**

Ravindra Pushker

**Examples**

```
analyze_tokens(in_text=c("The quick brown fox jumps over the lazy dog."))
```

---

analyze_trigrams	<i>Analyze Trigrams</i>
------------------	-------------------------

---

**Description**

Analyze text with ngram=3 (trigrams).

**Usage**

```
analyze_trigrams(in_text, top_rows = 25)
```

**Arguments**

in_text	a character vector. Text to be analyzed as a character vector.
top_rows	a numeric vector of length 1. Number of top rows to be returned.

**Details**

```
analyze_trigrams
```

**Value**

A data.frame with two columns - trigram (character vector) and count (numeric vector).

**Author(s)**

Ravindra Pushker

**Examples**

```
analyze_trigrams(in_text=c("The quick brown fox jumps over the lazy dog."))
```

---

`plot_ngrams`*Plot Ngrams*

---

**Description**

Plot ngrams - Word(s) vs. Count.

**Usage**

```
plot_ngrams(ngrams_data, top_rows = 25, plot_nrows = 25)
```

**Arguments**

<code>ngrams_data</code>	a data.frame containing word and n columns.
<code>top_rows</code>	a numeric vector of length 1. Number of top rows to be returned.
<code>plot_nrows</code>	a numeric vector of length 1. Number of rows to be plotted.

**Details**

`plot_ngrams`

**Value**

A ggplot plot object of bar chart with words and their counts.

**Author(s)**

Ravindra Pushker

**Examples**

```
plot_ngrams(data.frame(word=c("test1", "test2"), n=c(25, 30)))
```

# Index

`analyze_bigrams`, [2](#)  
`analyze_ngrams`, [2](#)  
`analyze_tokens`, [3](#)  
`analyze_trigrams`, [4](#)  
  
`plot_ngrams`, [5](#)