

# Package ‘WeatherSentiment’

May 7, 2026

**Type** Package

**Version** 1.0

**Title** Comprehensive Analysis of Tweet Sentiments and Weather Data

**Description** A comprehensive suite of functions for processing, analyzing, and visualizing textual data from tweets is offered. Users can clean tweets, analyze their sentiments, visualize data, and examine the correlation between sentiments and environmental data such as weather conditions. Main features include text processing, sentiment analysis, data visualization, correlation analysis, and synthetic data generation. Text processing involves cleaning and preparing tweets by removing textual noise and irrelevant words. Sentiment analysis extracts and accurately analyzes sentiments from tweet texts using advanced algorithms. Data visualization creates various charts like word clouds and sentiment polarity graphs for visual representation of data. Correlation analysis examines and calculates the correlation between tweet sentiments and environmental variables such as weather conditions. Additionally, random tweets can be generated for testing and evaluating the performance of analyses, empowering users to effectively analyze and interpret 'Twitter' data for research and commercial purposes.

**License** GPL-3

**Depends** R (>= 4.1.0), tidyverse, wordcloud, sentimentr

**Imports** tidytext, ggplot2, stringr, data.table, RColorBrewer, tidyr

**Suggests** dplyr, syuzhet

**Maintainer** Leila Marvian Mashhad <Leila.marveian@gmail.com>

**NeedsCompilation** no

**Author** Andriette Bekker [aut],  
Mohammad Arashi [aut],  
Leila Marvian Mashhad [aut, cre],  
Priyanka Nagar [aut]

**Repository** CRAN

**Date/Publication** 2024-08-19 08:20:10 UTC

## Contents

corr_analys . . . . .	2
generate_tweets . . . . .	3
process_tweet . . . . .	3
sentiment_analys . . . . .	4
sentiment_polarity . . . . .	5
word_cloud_tweet . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

corr_analys	<i>Calculate Correlation between Sentiment and Weather Variable</i>
-------------	---

---

### Description

This function calculates the Pearson correlation coefficient between sentiment scores extracted from tweets and a weather variable (e.g., temperature) in a merged dataset.

### Usage

```
corr_analys(t, w, com_var = "Date", var1 = "T1", var2 = "T2")
```

### Arguments

t	A data.frame containing tweets with a 'text' column
w	A data.frame containing weather data with a column matching the 'com_var'
com_var	The name of the common variable for merging the tweet and weather data. Defaults to "Date".
var1	The name of the column in 't' containing the tweet text. Defaults to "T1".
var2	The name of the column in 'w' containing the weather variable. Defaults to "T2".

### Value

The Pearson correlation coefficient between sentiment scores and the weather variable.

### Author(s)

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

### Examples

```
Date1 <- c('2024-01-01', '2024-01-02')
T1 <- c('I love sunny days', 'Rainy days are the worst')
tweet <- data.frame(Date = Date1, T1 = T1)
weather <- data.frame(Date = Date1, T2 = c(25, 15))
cor1 <- corr_analys(tweet, weather, com_var = "Date", var1 = "T1", var2 = "T2")
print(cor1)
```

---

generate_tweets	<i>Generate Random Tweets</i>
-----------------	-------------------------------

---

**Description**

This function generates n random tweets about the weather. Each tweet consists of a randomly selected positive or negative phrase about a randomly selected weather condition.

**Usage**

```
generate_tweets(n)
```

**Arguments**

n                      The number of tweets to generate

**Value**

A data.frame containing two columns: Date: The date of the tweet T1: The text of the tweet

**Author(s)**

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

**Examples**

```
tweets <- generate_tweets(10)
print(tweets)
```

---

process_tweet	<i>Preprocess Tweets for Sentiment Analysis</i>
---------------	---

---

**Description**

This function takes a list of tweets as input and performs various preprocessing steps to prepare the data for sentiment analysis.

**Usage**

```
process_tweet(tweet)
```

**Arguments**

tweet                      A vector of tweets

**Value**

- A list including:
- A vector containing preprocessed tweets.
- A vector containing tokens of tweets.

**Author(s)**

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

**Examples**

```
tweets_data <- "I'm feeling really happy today! #goodvibes"  
  
preprocessed_tweets <- process_tweet(tweets_data)  
print(preprocessed_tweets)
```

---

sentiment\_analys      *Sentiment Analysis of a Tweet*

---

**Description**

This function analyzes the sentiment of a tweet and returns the sentiment score and the text of the tweet.

**Usage**

```
sentiment_analys(tweet)
```

**Arguments**

tweet                    A character string containing the text of the tweet

**Value**

A data.frame containing two columns: text: The text of the tweet ave\_sentiment: The sentiment score of the tweet In addition, it presents a plot to effectively visualize the spectrum of human emotions.

**Author(s)**

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

**Examples**

```
#Example 1
tweet_text <- "I love R!"
sentiment_result <- sentiment_analys(tweet_text)
print(sentiment_result)

#Example2
tweets <- c("I hate R!", "R is a great language!", "R is difficult to learn!")
sentiment_results <- sapply(tweets, sentiment_analys)
print(sentiment_results)
```

---

sentiment_polarity	<i>Analyze Sentiment Polarity of a Tweet</i>
--------------------	--

---

**Description**

This function takes a tweet text as input and performs sentiment analysis to visualize its overall sentiment polarity.

**Usage**

```
sentiment_polarity(tweet)
```

**Arguments**

tweet            A character vector containing the tweet text.

**Value**

A ggplot object displaying a bar chart with sentiment polarity (positive/negative) on the x-axis and sentiment score on the y-axis.

**Author(s)**

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

**Examples**

```
e <- c("The rain is ruining my outdoor plans today.",
      "I love the sunny weather today!")

s1 <- sentiment_polarity(e)
print(s1)
```

---

word\_cloud\_tweet      *Generate Word Cloud from Tweet Text*

---

**Description**

This function generates a word cloud visualization of the most frequent words in a tweet after basic cleaning. Stop words and words less than 3 characters are removed.

**Usage**

```
word_cloud_tweet(tweet)
```

**Arguments**

tweet      A character string containing the text of the tweet.

**Value**

void (generates a word cloud image).

**Author(s)**

Leila Marvian Mashhad and Andriette Bekker and Mohammad Arashi and Priyanka Nagar.

**Examples**

```
# Generate word cloud from a single tweet
tweet_text <- "This is a sample tweet for word cloud generation!"
word_cloud_tweet(tweet_text)

## This will generate a word cloud image where the most frequent words
## in the tweet will be displayed larger.
```

# Index

corr\_analys, [2](#)

generate\_tweets, [3](#)

process\_tweet, [3](#)

sentiment\_analys, [4](#)

sentiment\_polarity, [5](#)

word\_cloud\_tweet, [6](#)