

# Package ‘crossword.r’

January 20, 2019

**Date** 2019-01-20

**Type** Package

**Title** Generating Crosswords from Word Lists

**Version** 0.3.6

**Author** Peter Meissner

**Maintainer** Peter Meissner <retep.meissner@gmail.com>

**Description** Generate crosswords from a list of words.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** R6 (>= 2.2.0), dplyr (>= 0.5.0), stringr (>= 1.2.0), magrittr (>= 1.5), jsonlite (>= 1.5), r6extended (>= 0.1.1)

**RoxygenNote** 6.0.1

**Suggests** covr, testthat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-01-20 06:40:03 UTC

## R topics documented:

Crossword . . . . .	2
cw_greplv . . . . .	3
cw_matrix_to_df . . . . .	3
cw_normalize_words . . . . .	4
cw_to_json . . . . .	4
cw_wordlist_animal_en . . . . .	4
%>% . . . . .	5

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

---

Crossword

*Crossword*

---

### Description

Crossword

### Usage

Crossword

### Format

An [R6Class](#) generator object for generating crosswords from word lists

### Fields

`letters` a character matrix representing the grid of the crossword

`words` a data.frame like (tibble) storing words, their position on the grid (row, col), their length in character, their direction ("right", "down") the word and the clue

### Methods

`add_words(words, clues = NULL)` this method will try to add words to the crossword by placing it on the grid; clues is optional and should be the same length;

`density()` gives back statistics on fill state of grid

`to_json(pretty = FALSE)` thi exports grid and word list data to JSON for external usage; pretty parameter determines if this is done in a human readable or more machine efficient way

### Examples

```
library(crossword.r)
cw <- Crossword$new(rows = 4, columns = 4)
cw$add_words(c("back", "nasa", "kick", "nuk", "ic", "sic"))
cw
cw$letters
cw$words
cw$density()
```

---

`cw_greplv` *a vectorized version of grep*

---

**Description**

a vectorized version of `grep`

**Usage**

```
cw_greplv(pattern, x, ignore.case = FALSE, perl = FALSE, fixed = FALSE,
          useBytes = FALSE)
```

**Arguments**

<code>pattern</code>	character string containing a <a href="#">regular expression</a> (or character string for <code>fixed = TRUE</code> ) to be matched in the given character vector. Coerced by <code>as.character</code> to a character string if possible. If a character vector of length 2 or more is supplied, the first element is used with a warning. Missing values are allowed except for <code>regexpr</code> and <code>gregexpr</code> .
<code>x</code>	a character vector where matches are sought, or an object which can be coerced by <code>as.character</code> to a character vector. <a href="#">Long vectors</a> are supported.
<code>ignore.case</code>	if <code>FALSE</code> , the pattern matching is <i>case sensitive</i> and if <code>TRUE</code> , case is ignored during matching.
<code>perl</code>	logical. Should Perl-compatible regexps be used?
<code>fixed</code>	logical. If <code>TRUE</code> , <code>pattern</code> is a string to be matched as is. Overrides all conflicting arguments.
<code>useBytes</code>	logical. If <code>TRUE</code> the matching is done byte-by-byte rather than character-by-character. See 'Details'.

---

`cw_matrix_to_df` *function that turn matrix into a data.frame in long format*

---

**Description**

function that turn matrix into a `data.frame` in long format

**Usage**

```
cw_matrix_to_df(x)
```

**Arguments**

<code>x</code>	the <code>data.frame</code> to transform
----------------	------------------------------------------

---

`cw_normalize_words`      *normalize words to be added to grid*

---

**Description**

normalize words to be added to grid

**Usage**

```
cw_normalize_words(words)
```

**Arguments**

`words`                  character vector of words to normalize for crossword usage

---

`cw_to_json`              *function implementing to\_json method*

---

**Description**

function implementing to\_json method

**Usage**

```
cw_to_json(cw, pretty = FALSE)
```

**Arguments**

`cw`                        an object of class crossword  
`pretty`                    should json formatted to be mor human readable or not

---

`cw_wordlist_animal_en`    *en - animals*

---

**Description**

data frame of words and clues

**Usage**

```
cw_wordlist_animal_en
```

**Format**

An object of class `data.frame` with 68 rows and 2 columns.

---

%>% *re-export magrittr pipe operator*

---

**Description**

re-export magrittr pipe operator

# Index

## \*Topic **datasets**

Crossword, [2](#)  
cw\_wordlist\_animal\_en, [4](#)  
%>%, [5](#)

as.character, [3](#)

Crossword, [2](#)  
cw\_greplv, [3](#)  
cw\_matrix\_to\_df, [3](#)  
cw\_normalize\_words, [4](#)  
cw\_to\_json, [4](#)  
cw\_wordlist\_animal\_en, [4](#)

Long vectors, [3](#)

R6Class, [2](#)  
regular expression, [3](#)