Package: geocacheR (via r-universe)

November 3, 2024

Type Package

Imports dplyr, stringr, magrittr, tibble, threewords, tidyr

Title Tools for Geocaching

Version 0.1.0.9001

Date 2020-02-02

Description Tools for solving common geocaching puzzle types, and other Geocaching-related tasks.

License GPL-3

Encoding UTF-8

LazyData true

Suggests testthat

RoxygenNote 7.0.2

Repository https://alunhewinson.r-universe.dev

RemoteUrl https://github.com/alunhewinson/geocacher

RemoteRef HEAD

RemoteSha b5599e8449475a33f62a66e803e1caa98506ba63

Contents

alphash *Create "hashes" of the letters in a character vector*

Description

The "hash" here is not a true hash, since the output has a variable number of characters, but is instead a string representation of the number of characters in the input string in hexadecimal format.

Usage

```
alphash(x, case_sensitive = FALSE, alphabet = standard_alphabet)
```

Arguments

`x` A character vector to be hashed

`case_sensitive` NOT IN USE. This parameter will probably be removed

`alphabet` NOT IN USE. This parameter will probably be removed

Value

A character vector equal in length to the input x

analyse_frequency *Make a frequency table of the characters in a string*

Description

Make a frequency table of the characters in a string

Usage

```
analyse_frequency(
  x,
  case_sensitive = FALSE,
  alphabet = standard_alphabet,
  collapse = FALSE
)
```

Arguments

`x` A character vector

`case_sensitive` Logical. Should lower and upper case letters be kept separate

`alphabet` A list containing lower and upper case alphabets

`collapse` A character value to be passed to paste() to collapse x into a single string. If collapse=FALSE (default), no collapse is implemented

Value

A tibble of the frequencies, with one row for each input string

---

`base64` *A helper table for base64 conversion and lookup*

Description

A helper table for base64 conversion and lookup

Usage

```
base64
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 64 rows and 3 columns.

`expressCoordinates` *Express Decimal Coordinates in Other (text) Formats*

Description

Designed to convert into Geocaching-style style coordinates, but future styles may be accommodated.

Usage

```
expressCoordinates(x, style = "GC")
```

Arguments

`x`	A numeric vector of length 2
`style`	placeholder for future development if requirements emerge

Value

A character of length 1 with an alternative expression of the coordinates

Examples

```
expressCoordinates(c(55.9327, -3.25103))
```

`parseCoordinates` *Parse Coordinates into Numeric Format*

Description

`parseCoordinates` takes a variety of string inputs for coordinates in the following formats: - N00 00.000 W000 00.000 - N00 00 00 W000 00 00 - N00.0000 W00.0000 and converts them into a numeric vector of length 2

Usage

```
parseCoordinates(x)
```

Arguments

`x`	A string for the coordinates to be converted

Value

A numeric vector holding the n(orth) and e(ast) coordinates

Examples

```
parseCoordinates(”N55 55.555 W003 14.159”)
parseCoordinates(”N 55 55.555   E003 14.159”)
parseCoordinates(”N55.92592 W3.23598”)
```

`qqmiaiii` *Encrypt a string using the Vigenere cipher*

Description

This is a wrapper for `vigenere` where decrypt is set to FALSE

Usage

```
qqmiaiii(x, key, alphabet = standard_alphabet)
```

Arguments

`x`	A string to encrypt or decrypt
`key`	The encryption or decryption key
`alphabet`	A list of letters in lower and upper case

See Also

`vigenere`

`rot` *Caesar-shift a string by a given number of letters.*

Description

Caesar-shift a string by a given number of letters.

Usage

```
rot(x, n = 13, alphabet = standard_alphabet, showWarn = TRUE)
```

Arguments

`x`	A string.
`n`	A number of letters to shift the string by.
`alphabet`	A list containing lower and upper case alphabets
`showWarn`	boolean. Do you want to see warnings about alphabets?

Value

A string

Examples

```
rot(”abc”)
rot(”abc”, n=2)
rot(”abc”, n=5, list(lw=letters[1:7], up=LETTERS[1:7]))
```

`rot_all` *Caesar-shift a string over all possible number n*

Description

Caesar-shift a string over all possible number n

Usage

```
rot_all(x, alphabet = standard_alphabet)
```

Arguments

`x`	A string.
`alphabet`	A list containing lower and upper case alphabets.

Value

a vector of strings

Examples

```
rot_all("abc")
rot_all("abc", list(lw=letters[1:7], up=LETTERS[1:7]))
```

Scrabble *Value and frequency of Scrabble letters*

Description

Value and frequency of Scrabble letters

Usage

```
Scrabble
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 27 rows and 3 columns.

Scrabble_score *Find the Scrabble value of words*

Description

Find the Scrabble value of words

Usage

```
Scrabble_score(x, language = "en")
```

Arguments

`x`	A vector of character strings
`language`	A character string for the linguistic Scrabble edition, conforming to ISO 639-1 Current supported languages: en

Value

An integer vector

Examples

```
Scrabble_score(c("kwyjibo", "jozxyqk"))
```

`standard_alphabet` *The standard alphabet for the locale, for use in Caesar-based encryption etc.*

Description

The standard alphabet for the locale, for use in Caesar-based encryption etc.

Usage

```
standard_alphabet
```

Format

An object of class `list` of length 2.

`vigenere` *Encrypt or decrypt a string using a key*

Description

Encrypt or decrypt a string using a key

Usage

```
vigenere(x, key, decrypt = TRUE, alphabet = standard_alphabet)
```

Arguments

`x`	A string to encrypt or decrypt
`key`	The encryption or decryption key
`decrypt`	Are you decrypting an encrypted string?
`alphabet`	A list of letters in lower and upper case

Value

A string

Examples

```
vigenere("MN vdopf wq brcep zwtcd.", "midway")
vigenere("My treasure is buried he... find it who may.", "La Bouche", decrypt = FALSE)
```

w3w *What 3 Words wrapper*

Description

This function requires you to have a valid what3words API key called `W3WAPIKey` stored as an environment variable

Usage

```
w3w(x)
```

Arguments

`x`	A vector, or list, of words. Strings with dots in them will be split. After splitting, there must be a multiple of three words. Either a vector of words, for a single latitude/longitude pair, or a list of vectors for vectorised operations. This wrapper also accepts a single string of three words separated by full stops.

Value

a numeric vector of length 2, consisting of lat(itude) and lon(gitude)

Examples

```
## Not run:
w3w("president.always.lying")
w3w("unseen.academicals.football") ## returns NAs
w3w(list("special.tools.required", "cliffs.falling.rocks",
         "available.during.winter", "ultraviolet.light.required"))
w3w(c("protests", "memo", "consoles"))

## End(Not run)
```

word_score *Find the value of words*

Description

Find the value of words

Usage

```
word_score(x)
```

Arguments

x	A vector of character strings

Value

An integer vector

Examples

```
word_score(c("infinite", "monkey", "cage"))
```

Index