Package: cipheR (via r-universe)

November 1, 2024

Title Encryption and Decryption with Text Ciphers

Version 1.0.0

Description Encrypts and decrypts using basic ciphers. None of these should be used in place of real encryption using state of the art tools. The ciphers included use methods described in the ciphers's Wikipedia and cryptography hobby websites.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://cihper.guslipkin.me/, https://github.com/guslipkin/cipheR

BugReports https://github.com/guslipkin/cipheR/issues

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

RemoteUrl https://github.com/guslipkin/cipher

RemoteRef HEAD

RemoteSha a9b983073ce787d0d4b771e4cc8759eab510489f

Contents

atbash *Encrypt or decrypt an Atbash Cipher*

Description

This can be used to create (encrypt) and solve (decrypt) an Atbash Cipher. An Atbash Cipher swaps letters' places in the alphabet. Thus, 'a' becomes 'z', 'b' becomes 'y', and so on. The function does not differentiate between the two.

The Atbash Cipher Wikipedia entry provides more information on the methods used: https://en.wikipedia.org/wiki/Atbash

Usage

```
atbash(x)
```

Arguments

x	A vector to be encoded or decoded.

Value

A character vector of length one that has been transformed

Examples

```
(e1 <- atbash("abcde"))
atbash(e1)

(e2 <- atbash("cipheR is a great R package!"))
atbash(e2)

(e3 <- atbash("Isn't this fun?"))
atbash(e3)
```

caesar *Encrypt or decrypt a Caesar Cipher*

Description

This can be used to create (encrypt) and solve (decrypt) a Caesar cipher. The function does not differentiate between the two.

The Caesar Cipher Wikipedia entry provides more information on the methods used: https://en.wikipedia.org/wiki/Caesar_cipher

Usage

```
caesar(x, n = 1, preserve_spaces = TRUE, dict = NULL, preset = NULL)
```

Arguments

x	A vector to be shifted
n	(Default: 1) The number of places to shift by. This can be either positive or negative. Zero returns x as it was given to the function.
preserve_spaces	(Default: TRUE) A boolean describing if spaces should be preserved. This is helpful when working with sentences.
dict	The dictionary used for shifting. This defaults to NULL in which case a dictionary is built from the sorted unique values of x.
preset	A pre-made dictionary using ASCII codes from https://www.ascii-code.com/. Note that delete is excluded as a character.

- NULL (the default)
- "alphanumeric": ASCII characters 48:57, 65:90, and 97:122. Numbers 0-9 and both uppercase and lowercase letters from the English alphabet.
- "keyboard": ASCII characters 32:126. The characters you'll find on a US English keyboard.
- "letters": ASCII characters 65:90 and 97:122. Both uppercase and lowercase letters from the English alphabet.
- "lowercase": ASCII characters 97:122. Lowercase letters from the English alphabet.
- "uppercase": ASCII characters 65:90. Uppercase letters from the English alphabet.

Value

A character vector of length one that has been shifted.

Examples

```
(e1 <- caesar("abcde", 1))
caesar(e1, -1)

(e2 <- caesar("cipheR is a great R package!", -5))
caesar(e2, 5)

(e3 <- caesar("Isn't this fun?", 2, preserve_spaces = FALSE))
caesar(e3, -2, preserve_spaces = FALSE)
```

railfence *Encrypt or decrypt a Railfence Cipher*

Description

This can be used to create (encrypt) and solve (decrypt) a Railfence Cipher. A Railfence Cipher maps each letter to a cosine wave of the specified height where each letter resides at an integer rail height.

The Railfence Cipher Wikipedia entry provides more information on the methods used: https://en.wikipedia.org/wiki/Rail_fence_cipher

Usage

```
railfence(x, n = 1, decrypt = FALSE)
```

Arguments

x	A vector to be encoded or decoded.
n	(Default: 1) The width of the rail to be used. A width of one will have no effect.
decrypt	(Default: FALSE) The default FALSE will encrypt while using TRUE will decrypt a given value of x.

Value

A character vector of length one that has been transformed

Examples

```
(e1 <- railfence("abcde", 2))
railfence(e1, 2, decrypt = TRUE)

(e2 <- railfence("cipheR is a great R package!", 4))
railfence(e2, 4, decrypt = TRUE)

(e3 <- railfence("Isn't this fun?", 3))
railfence(e3, 3, decrypt = TRUE)
```

running_key *Encrypt or decrypt a Running Key Vigenere Cipher*

Description

This can be used to create (encrypt) and solve (decrypt) a Running Key Vigenere Cipher. A Vigenere cipher uses a table of alphabetic Caesar shifts for one to twenty-six. The key is made to have an equal length to the text by adding the first letters of the text to the key. Each letter and corresponding key value determine the grid location to choose the obfuscated letter from.

The Running Key Cipher Wikipedia entry provides more information on the methods used: https://en.wikipedia.org/wiki/Running_key_cipher

Usage

```
running_key(x, key, decrypt = FALSE, keep_punctuation = FALSE)
```

Arguments

x	A vector to be encoded or decoded.
key	A character vector of length one to use as a key
decrypt	(Default: FALSE) The default FALSE will encrypt while using TRUE will decrypt a given value of x.
keep_punctuation	(Default: FALSE) The default FALSE will ignore case and punctuation and return a lowercase result. TRUE will match the input's case and punctuation.

Value

A character vector of length equal to x that has been transformed

Examples

```
key <- "thisismysupersecurekey"
(e1 <- running_key("abcde", key))
running_key(e1, key, decrypt = TRUE)

(e2 <- running_key("cipheR is a great R package!", key))
running_key(e2, key, decrypt = TRUE)

(e3 <- running_key("Isn't this fun?", key, keep_punctuation = TRUE))
running_key(e3, key, decrypt = TRUE, keep_punctuation = TRUE)
```

vigenere *Encrypt or decrypt a Vigenere Cipher*

Description

This can be used to create (encrypt) and solve (decrypt) a Vigenere Cipher. A Vigenere cipher uses a table of alphabetic Caesar shifts for one to twenty-six. Each letter and corresponding key value determine the grid location to choose the obfuscated letter from.

The Vigenere Cipher Wikipedia entry provides more information on the methods used: https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

Usage

```
vigenere(x, key, decrypt = FALSE, keep_punctuation = FALSE)
```

Arguments

x	A vector to be encoded or decoded.
key	A character vector of length one to use as a key
decrypt	(Default: FALSE) The default FALSE will encrypt while using TRUE will decrypt a given value of x.
keep_punctuation	(Default: FALSE) The default FALSE will ignore case and punctuation and return a lowercase result. TRUE will match the input's case and punctuation.

Value

A character vector of length equal to x that has been transformed

Examples

```
(e1 <- vigenere("abcde", "key"))
vigenere(e1, "key", decrypt = TRUE)

(e2 <- vigenere("cipheR is a great R package!", "key"))
vigenere(e2, "key", decrypt = TRUE)

(e3 <- vigenere("Isn't this fun?", "key", keep_punctuation = TRUE))
vigenere(e3, "key", decrypt = TRUE, keep_punctuation = TRUE)
```

Index