Package 'codename'

February 6, 2023

Type Package

Title Generation of Code Names for Organizations, People, Projects,
and Whatever Else

Version 0.5.0

Depends R (>= 3.5.0)

Maintainer Steve Miller <steven.v.miller@gmail.com>

Description This creates code names that a user can consider for their organiza-
tions, their projects, themselves, people
in their organizations or projects, or whatever else. The user can also supply a nu-
meric seed (and even a character seed)
for maximum reproducibility. Use is simple and the code names produced come in vari-
ous types too, contingent on what the
user may be desiring as a code name or nickname.

License GPL-2

Encoding UTF-8

LazyData true

Suggests tibble

RoxygenNote 7.2.3

URL https://github.com/svmiller/codename

BugReports https://github.com/svmiller/codename/issues/

NeedsCompilation no

Author Steve Miller [aut, cre] (<https://orcid.org/0000-0003-4072-6263>)

Repository CRAN

Date/Publication 2023-02-06 11:42:32 UTC

R topics documented:

1

---

adjectives   *A Data Frame of Adjectives*

Description

This is a data frame of adjectives in the English language to use as an attribute in generating a code name.

Usage

```
adjectives
```

Format

a data frame with 1,347 observations and 1 column.

value a character vector

Details

Adjectives found on Github as a gist from username "@hugsy".

---

animals   *A Data Frame of Animals*

Description

This is a data frame of animals in the English language to use as an object in generating a code name.

Usage

```
animals
```

Format

a data frame with 400 observations and 1 column.

`value` a character vector

Details

Animals found on Github as a gist from username "@atduskgreg".

---

char2seed	*Convert a character vector to a numeric integer for setting a reproducible seed*

Description

`char2seed()` is a parlor trick for converting a character vector into an integer for the sake of setting a reproducible seed.

Usage

```
char2seed(x)

char2seed_v1(x)
```

Arguments

x   a character vector

Details

Interested users can see how this works. Namely, letters (and numbers) in the character vector are assigned corresponding numbers. In `char2seed()`, these numbers are concatenated together an divided over 1 minus the absolute maximum number that R can handle by default (2^32). The remainder of this division is what ultimately becomes the reproducible seed.

In `char2seed_v1()`, these numbers are added together and then exponentiated to create a really big number. The number is divided over 1 minus the absolute maximum number that R can handle by default (2^32). The remainder of this division is what ultimately becomes the reproducible seed.

The nature of this parlor trick means there's a possibility, however small, that two different character vectors can return the same reproducible seed. This happened before https://github.com/svmiller/codename/issues/1, which is why I changed the function for generating the seed and moved the old function to `char2seed_v1()`.

The function may warn of "loss of accuracy", but this just means you supplied it a really long character vector.

Value

char2seed() takes a character vector and returns a reproducible seed for you to use for whatever purpose. It's used internally in codename(). char2seed_v1() does the same, but using the older method from v. 0.1.0 and v. 0.2.0 of this release.

Author(s)

Steven V. Miller

Examples

```
char2seed("ABCDEF")
char2seed("Go Bucks!")
char2seed("My Project Title")
```

---

codename	*Generate a unique codename for yourself, your organization, other people, your projects, and whatever else*

Description

codename() is a tool for generating codenames for various things.

Usage

```
codename(type = "any", seed)
```

Arguments

type   a type of code the user wants. Defaults to "any", but "gods", "nicka", "ubuntu", and "wu-tang" are available.

seed   an optional reproducible seed, which can be specified as a character or number.

Details

When type is 'any' (the default), the function combines all adjective data frames together (i.e. those from the Wu-Tang set, the colors, and the generic adjectives) and combines all noun data frames together (i.e. the animals, the generic nouns, the gods, and the Wu-Tang nouns). It then samples from each and combines them together.

When type is 'gods', the function randomly samples a god from the gods data frame and pastes it to a random color or generic adjective.

When type is 'ubuntu', a codename is derived that is broadly consistent with how Ubuntu versions their releases. This particular way of doing it starts with sampling an animal from the animals data frame. After an animal is selected, the function finds the first letter of the animal and matches it with something from the adjectives or xkcd_colors data sets. However, anything on the adjective side

of the equation must be of length one. In other words, you can get "yellow yak" but you cannot have "yellow orange yak" because the "yellow orange" color is a two-word modifier.

When type is 'wu-tang', a name is derived by randomly sampling from the `wu_adjs` and `wu_nouns` data frames.

When type is 'nicka', the function randomly samples a U.S. agency from the `nicka_blocks` data frame. This agency has a series of two-character blocks from which they can draw a noun and an adjective. Thereafter, the function randomly selects something from a compilation of the adjectives and the nouns that start with something from the two-character block that was selected, though the sampling procedure omits any adjective or noun that is more than one word. You can cross-reference your code name to the `nicka_blocks` data frame to see what agency you're representing for something that I'm sure is nefarious and may require a congressional hearing.

Value

`codename()` takes a preferred type of code and an optional reproducible seed and returns a code-name for the user to consider for whatever it is they want.

Author(s)

Steven V. Miller

Examples

```
codename()
codename(type = "ubuntu")
codename(type = "gods")
codename(type = "wu-tang")
codename(type = "nicka")
codename(type = "ubuntu", seed = "A Character Seed")
codename(type = "gods", seed = "A Character Seed")
codename(type = "wu-tang", seed = "A Character Seed")
codename(type = "nicka", seed = "A Character Seed")
```

---

codename_message	*Display package version for* **codename**

Description

`codename_message()` produces a message about the package version and the version of R making use of this package.

Usage

```
codename_message()
```

Value

codename_message() produces a message about the installed version of **codename**. Successive updates may (understandably) break an expected output from a reproducible seed in the codename() function. This just adds an extra layer of transparency.

Author(s)

Steven V. Miller

Examples

```
codename_message()
```

---

gods   *A Data Frame of Gods*

Description

This is a data frame of gods in the English language to use as an object in generating a code name.

Usage

```
gods
```

Format

a data frame with 221 observations and 1 column.

value   a character vector

Details

Gods data cobbled from the website "Godchecker" after searching for the top 10 most popular deities by various regions/religions on their website.

---

nicka_blocks *A Data Frame of NICKA Blocks*

Description

This is a data frame of the NICKA blocks used by the U.S. government for devising code names for operations.

Usage

```
nicka_blocks
```

Format

a data frame with 626 observations and 4 column.

block a numeric block identifier

first_last a character vector for the range of first and last characters assigned to the particular U.S. agency

letlet a character vector, derived from the first_last column

assigned_to a character vector for the U.S. agency that had the block in question

Details

Data are taken from here: <http://www.designation-systems.net/usmilav/codenames.html>

---

nouns *A Data Frame of Nouns*

Description

This is a data frame of nouns in the English language to use as an object in generating a code name.

Usage

```
nouns
```

Format

a data frame with 6801 observations and 1 column.

value a character vector

Details

Nouns data come by way of "The Great Nouns List"

---

variety_pack	*Get a variety pack of unique code names for yourself, your organization, other people, your projects, and whatever else*

Description

variety_pack() is produces all types of code names available in **codename**.

Usage

```
variety_pack(seed)
```

Arguments

seed   an optional reproducible seed, which can be specified as a character or number.

Value

variety_pack() takes an optional reproducible seed and produces all kinds of code names available in **codename**. In order, the return is "any", "gods", "nicka", "ubuntu", and "wu-tang".

Author(s)

Steven V. Miller

Examples

```
variety_pack()
variety_pack(seed = 8675309)
```

---

wu_adjs	*A Data Frame of Adjectives from the "Wu-Tang Name Generator"*

Description

This is a data frame of adjectives from the "Wu-Tang Name Generator" to use as an attribute in generating a code name.

Usage

```
wu_adjs
```

Format

a data frame with 45 observations and 1 column.

`value` a character vector

Details

These data are classic and apparently come from around 2002.

---

wu_nouns	*A Data Frame of Nouns from the "Wu-Tang Name Generator"*

Description

This is a data frame of nouns from the "Wu-Tang Name Generator" to use as an object in generating a code name.

Usage

`wu_nouns`

Format

a data frame with 40 observations and 1 column.

`value` a character vector

Details

These data are classic and apparently come from around 2002.

---

xkcd_colors	*A Data Frame of Colors*

Description

This is a data frame of colors in the English language to use as an attribute in generating a code name.

Usage

`xkcd_colors`

Format

a data frame with 949 observations and 1 column.

`value` a character vector

Details

Colors found by way of the web comic *XKCD*.

Index