Package 'fun'

July 22, 2025

Type Package

Title Use R for Fun

Version 0.3

Maintainer Yihui Xie <xie@yihui.name>

Description This is a collection of R games and other funny stuff, such as the classic Mine sweeper and sliding puzzles.

License GPL

LazyData yes

URL https://github.com/yihui/fun

BugReports https://github.com/yihui/fun/issues

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Yihui Xie [aut, cre] (ORCID: <https://orcid.org/0000-0003-0645-5666>),
     Yixuan Qiu [aut],
     Taiyun Wei [aut]

Repository CRAN

Date/Publication 2020-10-23 18:00:02 UTC

Contents

Index   **14**

---

fun-package   *Use R for Fun*

Description

This is a collection of R games and other funny stuff, such as the classical Mine sweeper and sliding puzzles.

Details

New games are always welcome; if you know GIT and want to join the development, please go to https://github.com/yihui/fun; or simply contribute ideas at https://github.com/yihui/fun/issues.

Author(s)

Yihui Xie, Taiyun Wei, and Yixuan Qiu

Examples

```
## See the examples in each function, or check out the demos
demo(package = "fun")
```

---

alzheimer_test   *Test Alzheimer's disease by finding out the different character in a character rectangle*

Description

Please try hard to find the letter "N" in 300 "M"s, one "6" in 300 "9"s, etc.

Usage

```
alzheimer_test(
  char1 = c("9", "O", "M", "I", "F", "D"),
  char2 = c("6", "C", "N", "T", "E", "O"),
  nr = 10,
  nc = 30,
  seed = NULL,
  ...
)
```

Arguments

char1	the 'background' character
char2	the character to be found out
nr	number of rows of the character rectangle
nc	number of columns
seed	seed for random number generation
...	other arguments passed to set.seed

Value

If at least one test item has been passed, a data.frame will be returned telling the result of the test.

Note

Don't be too serious about this test. I'm no doctor, but I think this will be a good present to your friends.

Author(s)

Yihui Xie <https://yihui.org>

Examples

```
## Not run:
x = alzheimer_test()

## End(Not run)
```

---

gomoku	*The game of Gomoku, a.k.a Five in a row*

Description

There are two players in this game who play one after the other using black and white stones respectively. The winner is the first player to get an unbroken row of five stones horizontally, vertically, or diagonally.

Usage

```
gomoku(n = 19)
```

Arguments

n	the number of rows and columns in the board (the default 19 generates the standard board)

Value

NULL

Note

The players should judge the winner by themselves; this function does not do this job (patches are welcome, of course).

Author(s)

Yihui Xie <https://yihui.org>; modified from the code by pklin

References

https://d.cosx.org/d/104750, https://en.wikipedia.org/wiki/Gomoku

Examples

```
gomoku()
```

---

htmlspecialchars	*Replace HTML special characters with HTML entities*

Description

The characters c("&", '"', "'", "<", ">") will be replaced with c("&", """, "'", "<", ">"), respectively.

Usage

```
htmlspecialchars(string)
```

Arguments

string   the string with (or w/o) HTML special chars

Value

the string with special chars replaced.

Author(s)

Yihui Xie <https://yihui.org>

References

https://www.php.net/manual/en/function.htmlspecialchars.php

See Also

gsub

Examples

```
htmlspecialchars("<a href = 'https://yihui.org'>Yihui</a>")
# &lt;a href = &#039;https://yihui.org&#039;&gt;Yihui&lt;/a&gt;
```

---

lights_out   *Play the "Lights Out" game in R*

Description

By default, the white squares in the plot denote the lights that are on, and black ones for the off. When you click on a light, this light as well as the four neighbors will switch theirs status. Your mission is to close all the lights.

Usage

```
lights_out(
  width = 5,
  height = 5,
  steps = 3,
  cheat = FALSE,
  col.off = "black",
  col.on = "white",
  col.frame = "lightblue",
  seed = NULL
)
```

Arguments

width	number of lights in x axis
height	number of lights in y axis
steps	number of "seed" lights to initialize the puzzle. In general, the larger steps is, the more complex this puzzle may be
cheat	logical. If TRUE a data frame indicating the steps to solve this puzzle will be printed
col.off	color when lights off
col.on	color when lights on
col.frame	color of lights border
seed	seed for random number generator

Note

Linux/Mac users have to use X11(type = 'Xlib') or the Cairo graphics device Cairo() in the package **cairoDevice**.

Author(s)

Yixuan Qiu <yixuan.qiu@cos.name>

References

https://en.wikipedia.org/wiki/Lights_Out_(game)

Examples

```
## should use Xlib for the x11() device under *nix, e.g
if (interactive()) {
    if (.Platform$OS.type == "windows")
        x11() else x11(type = "Xlib")
    lights_out()
}
```

---

mine_sweeper   *Play the Mine Sweeper game in R*

Description

The controls should be familiar to you: Click the left mouse button to dig in an area, and right button to mark or unmark the area with flags.

Usage

```
mine_sweeper(width = 10, height = 10, mines = 20, cheat = FALSE)
```

Arguments

width	number of grids in horizontal axis
height	number of grids in vertical axis
mines	number of mines
cheat	logical. If TRUE a matrix indicating the mines will be printed

Note

Linux/Mac users have to use X11(type = 'Xlib') or the Cairo graphics device Cairo() in the package **cairoDevice**.

Author(s)

Yixuan Qiu <yixuan.qiu@cos.name>

References

<https://en.wikipedia.org/wiki/Minesweeper_(computer_game)>

Examples

```
## should use Xlib for the x11() device under *nix, e.g
if (interactive()) {
    if (.Platform$OS.type == "windows")
        x11() else x11(type = "Xlib")
    mine_sweeper()
}
```

---

random_password	*Generate a random password with a specified length*

Description

This function generates a random passord sampled from the ASCII table.

Usage

```
random_password(length = 12, replace = FALSE, extended = TRUE)
```

Arguments

length	length of the password
replace	sample from the ASCII table with (TRUE) or without (FALSE) replacement?
extended	if FALSE, use alphanumeric characters only; otherwise use all the ASCII characters

Value

a character string

Author(s)

Yihui Xie <<https://yihui.org>>

See Also

sample

Examples

```
random_password()
# set the seed to get fixed password every time; you may just remember the seed
# and forget the real password because it's reproducible
set.seed(123)
random_password()
# long password
random_password(20, TRUE)
```

---

shutdown   *Shut down the operating system with the command 'shutdown'*

Description

There is a command shutdown in both Windows and Linux, and this function uses it to shut down
a computer.

Usage

```
shutdown(wait = 0)
```

Arguments

wait   time to wait before shutting down (in seconds); passed to Sys.sleep

Details

After the time wait has passed, R will execute shutdown -s -t 0 (for Windows) or shutdown -h
now to shut down the computer.

Value

The status code of system.

Author(s)

Yihui Xie <https://yihui.org>

See Also

system, Sys.sleep

Examples

```
if (interactive()) {
    # when your code is extremely time-consuming, you may need this function; e.g.
    # you go to sleep, and R is running long computation... complex graphics... and
    # long long computation... at last,
    shutdown()
    # the next day you wake up, 'thank you, R' :)
}
```

---

sliding_puzzle   *Sliding puzzle in R*

Description

Use R to play sliding puzzle

Usage

```
sliding_puzzle(size = c(3, 3), bg = "lightblue", z = NULL)
```

Arguments

size   two dimensional vector, the size of sliding puzzle. Note: the element of size
            must be greater than 1.

bg   the background color of blocks.

z   the matrix of sliding puzzle, if z is specified, size will be omited.

Details

If size is specified and z is NULL, then the function will generate a solvable sliding puzzle.

Note

Linux/Mac users have to use X11(type = 'Xlib') or the Cairo graphics device Cairo() in the
package **cairoDevice**.

Author(s)

Taiyun Wei

References

About the sliding puzzle: https://en.wikipedia.org/wiki/Sliding_puzzle

Examples

```
## should use Xlib for the x11() device under *nix, e.g
if (interactive()) {
    if (.Platform$OS.type == "windows")
        x11() else x11(type = "Xlib")
    sliding_puzzle()
    sliding_puzzle(z = matrix(0:11, 3, 4))
}
```

---

tagData   *Tag information of Yihui Xie's English blog*

Description

Tag data collected from Yihui Xie's Blog, containing the tag words, frequency and hyperlinks, etc.

Format

A data frame with 45 observations on the following 5 variables.

tag a character vector; the tag words

link a character vector; hyperlinks of tags

count a numeric vector; the frequency of tags in blogs (see Details)

color a character vector in hexadecimal format specifying the RGB component of tag colors

hicolor a character vector similar to color; the color when mouse hangs over the tag

Details

The count was multiplied by 4 in the data in order that the tag cloud could be more clear.

Source

<https://yihui.org/en/> (accessed on June 10, 2009)

Examples

```
hist(tagData$count/4, 10)  # extremely right skewed
# see help(tag_cloud) for the example of creating tag cloud with this data
```

---

tag_cloud *Creating Tag Cloud in R (with Flash and JavaScript)*

Description

Use R to write tag data (tag words, frequency, hyperlinks and colors, etc) into JavaScript, and the JavaScript code will generate a Flash movie. Finally the tag cloud can be created with fantastic 3D rotation effect.

Usage

```
tag_cloud(
  tagData,
  htmlOutput = "tagCloud.html",
  SWFPath = "tagcloud.swf",
  JSPath = "swfobject.js",
  divId = "tagCloudId",
  width = 600,
  height = 400,
  transparent = FALSE,
  tcolor = "333333",
  tcolor2 = "009900",
  hicolor = "ff0000",
  distr = "true",
  tspeed = 100,
  version = 9,
  bgcolor = "ffffff",
  useXML = FALSE,
  htmlTitle = "Tag Cloud",
  noFlashJS,
  target = NULL,
  scriptOnly = FALSE,
  encode = FALSE,
  reserved = FALSE
)
```

Arguments

tagData	a data.frame containing at least 3 columns: `tag`, `link` and `count`. Optional columns are `color` and `hicolor`
htmlOutput	filename of the HTML output
SWFPath	path of the SWF source file ('tagcloud.swf'); see `system.file("js", "tagcloud.swf", package = "fun")`
JSPath	path of the JavaScript file ('swfobject.js'); see `system.file("js", "swfobject.js", package = "fun")`
divId	id of the tag cloud div (HTML layer)

width, height	width and height of the tag cloud
transparent	logical; whether to use transparent backgroud for the Flash movie?

tcolor, tcolor2, hicolor, distr, tspeed
> see Details

version	the required Flash version
bgcolor	backgroud color of the Flash movie
useXML	use XML file for the tag information or just a string; this will be passed to the Flash object as a variable
htmlTitle	title of the HTML file
noFlashJS	text to show if Flash or JavaScript is not supported
target	target window of the hyperlinks; possible values are NULL, '_blank', '_top', etc
scriptOnly	print the script in the console only? (if TRUE), no HTML file will be generated
encode	encode the tag XML or not? (with URLencode) set it to be TRUE when your browser does not recognize the tag XML correctly
reserved	should reserved characters be encoded? see URLencode

Details

This function is based on the WordPress plugin "wp-cumulus". If there are any arguments you don't understand, please check the reference.

Value

NULL

Author(s)

Yihui Xie <https://yihui.org>

References

About the WordPress plugin: https://wordpress.org/plugins/wp-cumulus/

Usage of the SWFObject: http://blog.deconcept.com/swfobject/

An example of visualizing tags in my blog: https://yihui.org/en/2009/06/creating-tag-cloud-using-r-and-flas

See Also

cat, sprintf, URLencode

Examples

```
data(tagData)
htmlFile = paste(tempfile(), ".html", sep = "")
if (file.create(htmlFile)) {
    tag_cloud(tagData, htmlFile)
    if (!interactive())
        browseURL(htmlFile)
}
```

---

tower_of_hanoi   *Demonstrate the Tower of Hanoi puzzle in R*

Description

This function uses the recursive algorithm to solve the Tower of Hanoi puzzle, and demonstrates the game in animation.

Usage

```
tower_of_hanoi(n = 7)
```

Arguments

n   an integer indicating the number of disks on the rot.

Details

This function was written by Linlin Yan <linlin.yan@cos.name> in a Chinese forum (See 'References') to show the usage of recursive algorithm.

Author(s)

Linlin Yan <<linlin.yan@cos.name>>

References

Original code: <https://d.cosx.org/d/101199>

About the Tower of Hanoi: <https://en.wikipedia.org/wiki/Tower_of_Hanoi>

See Also

[barplot](#)

Examples

```
## Not run:
tower_of_hanoi(7)

## End(Not run)
```

Index