Package 'sudokuAlt'

May 9, 2026

Type Package

Title Tools for Making and Spoiling Sudoku Games

Version 0.2-1

Date 2019-12-15

Depends R (>= 3.5.0), stats

Suggests sudoku, knitr, rmarkdown

Imports graphics, magrittr

Author Bill Venables <Bill.Venables@gmail.com>

Maintainer Bill Venables <Bill.Venables@gmail.com>

Description Tools for making, retrieving, displaying and solving sudoku games. This package is an alternative to the earlier sudoku-solver package, 'sudoku'. The present package uses a slightly different algorithm, has a simpler coding and presents a few more sugar tools, such as plot and print methods. Solved sudoku games are of some interest in Experimental Design as examples of Latin Square designs with additional balance constraints.

License GPL (>= 2)

NeedsCompilation no

Encoding UTF-8

RoxygenNote 7.0.2

VignetteBuilder knitr

Repository CRAN

Date/Publication 2019-12-15 06:50:02 UTC

Contents

`as.sudoku` *Generic Sudoku Game Constructor*

Description

Construct a Sudoku Game Object

Usage

```
as.sudoku(x, ...)
```

Arguments

`x`	an n^2 x n^2 matrix object to represent the game
`...`	Other additional arguments (currently ignored)

Details

Coerce an object to one that can be used as a sudoku game. IMPORTANT: games are represented as n^2xn^2 character matrices, using 1-9 for n=2 or 3, and LETTERS[1:(n^2)] for n = 4 or 5.

Value

An object of class 'sudoku'

Author(s)

Bill Venables

Examples

```
M <- as.sudoku(matrix("", 16, 16))
M[1:4, 1:4] <- matrix(LETTERS[1:16], 4, 4, byrow = TRUE)
sM <- solve(M)
plot(sM)
```

as.sudoku.matrix *as.sudoku.matrix*

Description

Construct a Sudoku Game Object

Usage

```
## S3 method for class 'matrix'
as.sudoku(x, ...)
```

Arguments

x	An n^2 x n^2 matrix
...	other arguments (currently ignored)

Details

Coerce a matrix to an object that can be used as a sudoku game

Value

An object of class 'sudoku'

Author(s)

Bill Venables

as.sudoku.sudoku *Construct a Sudoku Game Object*

Description

Identity function for sudoku objects

Usage

```
## S3 method for class 'sudoku'
as.sudoku(x, ...)
```

Arguments

x	A sudoku object
...	other arguments (ignored)

Value

the input sudoku object

Author(s)

Bill Venables

`daysAgo` *Format a Past Date*

Description

Format a Date Relative to the Current Date

Usage

```
daysAgo(n = 0, warn = TRUE)
```

Arguments

`n`	A positive integer for how many days ago
`warn`	Issue a warning if n <= 0 or n > 30

Details

Internal function used by fetchUKGame().

Value

A character string of the form "dd/mm/yy"

Author(s)

Bill Venables

Examples

```
daysAgo()  ## today
daysAgo(7) ## a week ago
```

designGame *Sudoku Design*

Description

Take a sudoku game and represent the information as a data frame giving the row, column, square and symbol for each entry. This is a useful form if the (complete) game is to be used as an experimental design

Usage

```
designGame(g, ...)

## Default S3 method:
designGame(g, ...)

## S3 method for class 'sudoku'
designGame(g, ...)
```

Arguments

g	a sudoku game, presumably solved
...	currently ignored

Value

a data frame with four colums, `Row`, `Col`, `Square` and `Symbol`

Examples

```
set.seed(2019)
d <- seedGame(4) %>% solve() %>%
     regulariseGame(target = "b") %>%
     designGame()
rbind(head(d), tail(d))
```

emptyGame *Construct an empty game*

Description

Construct a Vacant Game Structure

Usage

```
emptyGame(n = 3)
```

Arguments

n	an integer value between 2 and 5 inclusive.

Details

Returns a vacant game structure to allow special patterns to be constructed, as shown in the example.

Value

An empty sudoku game structure

Author(s)

Bill Venables

Examples

```
g <- emptyGame(4)
diag(g) <- LETTERS[1:16]
plot(g)
g %>% solve %>% plot -> sg ## %>% imported from magrittr
```

---

fetchAUGame *Retrieve a Sudoku from the AU Site*

Description

Retrieve a Sudoku Game

Usage

```
fetchAUGame(day = 0, difficulty = c("tough", "hard", "medium", "easy"))
```

Arguments

day	non-negative integer, how many days ago? zero for today's game.
difficulty	character string, how hard would you like it?

Details

Connects to http://www.sudoku.com.au and retrieves the sudoku game from day days ago. Based on a function from a related sudoku package, sudoku::fetchSudokuUK with minor changes.

Value

The published sudoku game as a sudoku object.

Author(s)

Bill Venables

Examples

```
## Not run:
fetchAUGame() %>% solve %>% plot -> gau           ## The 'tough' game for today
fetchAUGame(3, "easy") %>% solve %>% plot -> eau  ## 'easy' game from 3 days ago

## End(Not run)
```

---

fetchUKGame *Retrieve a Sudoku from the UK Site*

Description

Retrieve a Sudoku Game

Usage

```
fetchUKGame(day = NULL)
```

Arguments

day	positive integer < 30, how many days ago? or NULL for the most recently published game.

Details

Connects to `http://www.sudoku.org.uk/DailySudoku.asp` and retrieves the sudoku game from `day` days ago. Based on a function from a related sudoku package, `sudoku::fetchSudokuUK` with minor changes.

Value

The published sudoku game as a sudoku object.

Author(s)

Bill Venables

Examples

```
## Not run:
(g0 <- fetchUKGame())  ## The game for today (according to GMT)
(g3 <- fetchUKGame(3)) ## game from 3 days ago (according to GMT)
if(require(sudoku)) {  ## the original solver
  g0a <- as.sudoku(fetchSudokuUK())
  identical(g0, g0a)   ## should be TRUE
}
g0 %>% solve %>% plot -> sg0 ## spoil the game

## End(Not run)
```

`makeGame` *Make a New Sudoku Game*

Description

Construct a Random Sudoku Game

Usage

```
makeGame(n = 3, gaps = ceiling(3 * n^4/4), maxit = 25)
```

Arguments

`n`	Size of the game, n^2 x n^2
`gaps`	Number of holes to leave for the solution
`maxit`	Number of tries before giving up.

Details

Construcs a sudoku game for given n, 2 <= n <= 5. n = 5 can be problematical.

Value

a sudoku game

Author(s)

Bill Venables

Examples

```
set.seed(54321)
makeGame() %>% solve %>% plot -> sg
originalGame(sg)
```

`originalGame` *Retrieve the Original from a Solved Game*

Description

Retrieve the Original from a Solved Game

Usage

```
originalGame(x)
```

Arguments

`x`	a sudoku object

Details

Convenience function for accessing an original from a solved game. If the game is unsolved, the object itself is returned.

Value

The original sudoku game corresponding to the solution, or object itself if the game is unsolved

Author(s)

Bill Venables

Examples

```
set.seed(666)
seedGame() %>% solve %>% plot -> sg ## %>% imported from magrittr
originalGame(sg)
```

`plot.sudoku` *Plot a Sudoku Game*

Description

Plot a Sudoku Game

Usage

```
## S3 method for class 'sudoku'
plot(
  x,
  ...,
  cex = 1.5 - (n - 3)/2,
  colSolution = "grey",
  colGame = "fire brick"
)
```

Arguments

x	The sudoku game
...	additional arguments
cex	Character expansion factor
colSolution	colour to be used for the solution (if present)
colGame	colour to be used for the original game

Details

Present a graphical display of a sudoku game and its solution if the game is solved

Value

The sudoku game x, invisibly.

Author(s)

Bill Venables

Examples

```
set.seed(20191)
makeGame(4, gaps = 0) %>% plot(cex=1) -> sg
```

---

print.sudoku *Print a Sudoku Object*

Description

Print a Sudoku Object

Usage

```
## S3 method for class 'sudoku'
print(x, ...)
```

Arguments

x	The sudoku game object
...	extra arguments (ignored)

Details

Prints a sudoku object in an easily recognisable form.

Value

the object, invisibly

Author(s)

Bill Venables

`regulariseGame` *regulariseGame*

Description

Put a solved sudoku game into a canonical form

Usage

```
regulariseGame(g, ...)

## S3 method for class 'sudoku'
regulariseGame(g, target = c("block", "col", "row"), ...)

## Default S3 method:
regulariseGame(g, ...)
```

Arguments

g	a solved sudoku game
...	additional arguments to methods (currently not used)
target	character; which section do you want to be in sorted order?

Details

If a solved sudoku game is to be used as an experimental design it is sometimes useful to re-arrange the symbols so that either the first row, first column or top left block symbols are in sorted order. This function accomplishes this task.

Value

a regularised solved sudoku game

Examples

```
set.seed(1234)
g <- makeGame() %>%
     solve() %>%
     regulariseGame(target = "b") %>%
     plot()
plot(originalGame(g))
```

`seedGame` *Starting Point to Make a Random Sudoku Game*

Description

Generate a random sudoku game starting point

Usage

```
seedGame(n = 3)
```

Arguments

`n`	Size of the game, n^2 x n^2

Details

Generates a game with one instance of each symbol in random positions.

Value

A sparse unsolved sudoku game

Author(s)

Bill Venables

Examples

```
set.seed(2345)
g <- seedGame(3)
sg <- solve(g) ## a completed random game
plot(sg)
```

`solve.sudoku` *Solve a Sudoku Puzzle*

Description

Solve a Sudoku Puzzle

Usage

```
## S3 method for class 'sudoku'
solve(a, ...)
```

Arguments

`a`	A sudoku game object to be solved
`...`	Extra arguments (curently ignored)

Details

An alternative front end to `solveGame` as a method for the base generic function `solve`.

Value

a solved game, or NULL if no solution exists.

Author(s)

Bill Venables

Examples

```
set.seed(1234)
makeGame(3, gaps = 59) %>% solve %>% plot -> sg
originalGame(sg)

g <- emptyGame(4)  # construct a patterned game
diag(g) <- LETTERS[1:16]
g %>% solve %>% plot -> sg
sg
```

`solveGame` *Solve a Sudoku Game*

Description

Solve a Sudoku Game

Usage

```
solveGame(game)
```

Arguments

`game`	The game to be solved

Details

Given a sudoku game to be solved, find the solution. IMPORTANT: games are represented as n^2 x n^2 character matrices, using 1-9 for n=2 or 3, and LETTERS[1:(n^2)] for n = 4 or 5.

Value

A solved sudoku game object if one found, or NULL if no solution exists. The original game is attached as an attribute if the game is solved.

Author(s)

Bill Venables

Examples

```
set.seed(1234)
makeGame(3, gaps = 60) %>% solve %>% plot -> sg
(g <- originalGame(sg))

g <- emptyGame(4)  # construct a patterned game
diag(g) <- LETTERS[1:16]
sg <- solve(g)
plot(sg)
```

Index