Package 'SnakesAndLaddersAnalysis'

April 5, 2018

Type Package

Title Play and Analyse the Game of Snakes and Ladders

Version 2.1.0

Author Hector haffenden <haffendenh@cardiff.ac.uk>

Maintainer Hector Haffenden <haffendenh@cardiff.ac.uk>

Description Plays the game of Snakes and Ladders and has tools for analyses. The tools included allow you to find the average moves to win, frequency of each square, importance of the snakes and the ladders, the most common square and the plotting of the game played.

License GPL-2

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-05 14:27:15 UTC

R topics documented:

AverageMovesToWin   *The Average Moves to Win in a Game of Snakes and Ladders*

Description

This function uses the function SnakesAndLaddersGame to estimate the average number of moves to win given any amount of players and allows you to change the number of trials used.

Usage

```
AverageMovesToWin(number.of.players = 2, number.of.trials = 1000)
```

Arguments

number.of.players

A single numeric value that represents the number of players you wish to run the trial with

number.of.trials

A single numeric value that represents the number of trials you wish to run. The more trials the more accurate the estimation.

Details

Function uses a simple for loop to run many trials of the SnakesAndLaddersGame, counts the number of moves and outputs an average.

Value

average.number.of.moves

A numeric value which represents the average number of moves to win

Warning

I have not put a limit on the amount of time that this function will run. This is to allow millions of trials.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
AverageMovesToWin(1)

AverageMovesToWin(100)
```

```
FrequencyOfSquaresByPercentage
```
Finds the Frequency of each Square by Percentage in the Game of Snakes and Ladders

Description

This function uses the function SnakesAndLaddersGame to estimate the frequency of each square dependent on the number of players.

Usage

```
FrequencyOfSquaresByPercentage(number.of.players = 2,
              number.of.trials = 100, vector = FALSE)
```

Arguments

number.of.players

      A single numeric value that represents the number of players you wish to run the trial with.

number.of.trials

      A single numeric value that represents the number of trials you wish to run. The more trials the more accurate the estimation.

vector   Takes values TRUE or FALSE, if TRUE will display results as a vector. This is sometimes a much nicer display of results.

Details

Function uses the SnakesAndLaddersGame function to find and display the frequency of how often a square is landed on, given a set number of players. it is also very useful to plot this data to see the distribution of how often a square is landed on.

Value

Fequency.of.squares

      A numeric vector or matrix which represents how often a square is landed on in the game of Snakes and Ladders.

Warning

I have not put a limit on the amount of time that this function will run. This is to allow many of trials.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
FrequencyOfSquaresByPercentage(1)

FrequencyOfSquaresByPercentage(10)


x <- FrequencyOfSquaresByPercentage(2)
plot(x)
```

ImportenceOfTheLadders

Finds the Importence of the Ladders during the game of Snakes and Ladders

---

Description

This function uses the SnakesAndLaddersGame function to estimate the importence of the ladders dependent on the number of players by looking at which ladders the winners of each game lands on.

Usage

```
ImportenceOfTheLadders(number.of.players = 2)
```

Arguments

number.of.players

A single numeric value that represents the number of players you wish to run the trial with.

Details

Function uses the SnakesAndLaddersGame function to find and display the ladders used by the winners. As the number of players increase we see a steady state forming of the same ladders. This results in us working out which ladders are key to land on if you want to win.

Value

most.importent.ladders

A data frame which shows an estimation of the most important ladders. The higher the frequency the more importent the ladder.

Warning

I have not put a limit on the amount of time that this function will run. This is to allow as many players as you want.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
ImportenceOfTheLadders(1)

ImportenceOfTheLadders(100)

ImportenceOfTheLadders(1000)
```

ImportenceOfTheSnakes *Finds the Importence of the Snakes during the game of Snakes and Ladders*

Description

This function uses the SnakesAndLaddersGame function to estimate which snakes are the most powerful dependent on the number of players by looking at which snakes the losers of each game lands on.

Usage

```
ImportenceOfTheSnakes(number.of.players = 2)
```

Arguments

number.of.players

A single numeric value that represents the number of players you wish to run the trial with.

Details

Function uses the SnakesAndLaddersGame function to find and display the snakes used by the losers. As the number of players increase we see a steady state forming of the same snakes. This results in us working out which snakes are key to avoid if you want to win.

Value

most.powerful.snakes

A data frame which shows an estimation of the most powerful snakes. The higher the frequency the more powerful the snake.

Warning

I have not put a limit on the amount of time that this function will run. This is to allow as many players as you want.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
ImportenceOfTheSnakes(1)

ImportenceOfTheSnakes(100)

ImportenceOfTheSnakes(1000)
```

| MostCommonSquare | *Finds the Most Common Square during the Game of Snakes and Ladders* |

Description

This function uses the SnakesAndLaddersGame function to estimate the most common square in a game of snakes and ladders.

Usage

```
MostCommonSquare(number.of.players = 2, number.of.trials = 500)
```

Arguments

number.of.players

A single numeric value that represents the number of players you wish to run the trial with.

number.of.trials

A single numeric value tha represents the number of trials you wish to run.

Details

Function uses the SnakesAndLaddersGame function to find and display the most common square. The more trials the more accurate the estimate. There may be more than one if the trials are low enough.

Value

most.common.square

A print out of the most common squares landed on in a given number of trials.

Warning

I have not put a limit on the amount of time that this function will run. This is to allow as many players and trials as you want.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
MostCommonSquare(1)

MostCommonSquare(100, 10)
```

PlotMyGame   *Plots the game of Snakes and Ladders*

Description

This function plots the result from the SnakesAndLaddersGame Function.

Usage

```
PlotMyGame(output.from.SnakesAndLaddersGame.function, legend = TRUE)
```

Arguments

output.from.SnakesAndLaddersGame.function
              The output from the SnakesAndLaddersGame function.

legend   Takes value TRUE or FALSE, if TRUE, displays the legend for the plot.

Details

Function that takes output from the SnakesAndLaddersGame function and plots it with an appropriate legend and nice colours to represent each players game. Only accepts number of players between 1 and 7.

Value

A plot   Plots the players position on each roll.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Examples

```
PlotMyGame(SnakesAndLaddersGame(1))

PlotMyGame(SnakesAndLaddersGame(3))

PlotMyGame(SnakesAndLaddersGame(7))
```

SnakesAndLaddersGame   *Plays the Game of Snakes and Ladders*

Description

This is a function that will run the game of snakes and ladders for as many players as you want.

Usage

```
SnakesAndLaddersGame(number.of.players = 2)
```

Arguments

```
number.of.players
```
                          a single numeric value representing the number of players. Must be greater than
                          0.

Details

We use matrices here to make computations very fast and allowing millions of iterations.

Value

`game.output`   A matrix which displays the entire game of snakes and ladders.

Warning

Do not put too many players in as your computer may crash. I have not put a limit on run time as people with more powerful computers will want to run it for longer.

Note

An assumption that has been made is that everyone will get the same amount of rolls, irrespective of if someone has already won. This is to keep the game fair. This sometimes results in more than one player winning. This is rare when the player count is below 7 but as it grows the liklyhood of this increases dramatically. This is useful in some parts of the analysis.

The starting position for all players is 0.

The ladders used were:

1 to 38

4 to 14

9 to 31

21 to 42

28 to 84

36 to 44

51 to 67

71 to 91

80 to 100

The snakes used were:

16 to 6

47 to 26

49 to 11

56 to 53

62 to 19

64 to 60

87 to 24

93 to 73

95 to 75

98 to 78

The only check i have put in this function is to ensure that the number of players is greater than 0. This is because I want it to run at maximum efficieny and I have only included this check to ensure that the code does not go on for ever, which it would if number.of.players = 0 was inputted.

Author(s)

Hector Haffenden haffendenh@cardiff.ac.uk

Sam Davies

References

Dr Andreas Artemiou: Programming and Statistics.

Examples

```
SnakesAndLaddersGame(1)
SnakesAndLaddersGame(100)

# Run some analysis on this, consistant, output.
x <- SnakesAndLaddersGame(7)
```

Index