Package ‘darts’

February 19, 2015

Type Package
Title Statistical Tools to Analyze Your Darts Game
Version 1.0
Date 2011-01-17
Author Ryan Tibshirani
Maintainer Ryan Tibshirani <ryantibs@gmail.com>
Description Are you aiming at the right spot in darts? Maybe not! Use
this package to find your optimal aiming location. For a better
explanation, go to
http://www-stat.stanford.edu/~ryantibs/darts/ or see the paper
``A Statistician Plays Darts”.
License GPL
LazyLoad yes
Repository CRAN
Date/Publication 2011-01-20 15:00:39
NeedsCompilation yes

R topics documented:

darts-package .......................................................... 2
drawAimSpot ............................................................ 3
drawBoard ............................................................... 4
drawHeatmap ............................................................ 5
generalEM ............................................................... 6
generalExpScores .................................................... 7
simpleEM ............................................................... 7
simpleExpScores ....................................................... 8

Index 10
Description

Are you aiming at the right spot on the dartboard? Maybe not! Use this package to compute your optimal aiming location. For a better explanation, go to http://stat.stanford.edu/~ryantibs/darts/ or read the paper "A Statistician Plays Darts".

Details

Package: darts
Type: Package
Version: 1.0
Date: 2011-01-17
License: GPL
LazyLoad: yes

Author(s)

Ryan Tibshirani <ryantibs@gmail.com>

References

http://stat.stanford.edu/~ryantibs/darts/

Examples

# An example of how to use this package to calculate my variance, and then generate a personalized heatmap instructing me where to aim

# Start with 100 scores from throws aimed at the center of the board
x = c(12,16,19,3,17,1,25,19,17,50,18,1,3,17,2,2,13,18,16,2,25,5,5, 1,5,4,17,25,25,50,3,7,17,17,3,3,3,3,3,7,3,10,25,1,19,15,4,1,1,5,1,12,17,16, 50,20,20,20,25,50,2,17,3,20,20,20,5,1,18,15,2,3,25,12,9,3,3,19,16,20, 5,5,1,4,15,16,5,20,16,2,25,6,12,25,11,25,7,2,5,19,17,17,2,12)

# Simple model

# Step 1: EM algorithm
# Get my variance in the simple Gaussian model
a = simpleEM(x,niter=100)

# Check the log likelihood
plot(1:a$niter,a$loglik,type="l",xlab="Iteration",ylab="Log likelihood")

# The EM estimate of my variance
s = a$s.final

## Step 2: Generate a heatmap

# Build the matrix of expected scores
e = simpleExpScores(s)

# Plot it
par(mfrow=c(1,2))
drawHeatmap(e)
drawBoard(new=TRUE)
drawAimSpot(e)

## General model

## Step 1: EM algorithm

# Get my variance in the general Gaussian model
aa = generalEM(x,niter=100,seed=0)

# The EM estimate of my covariance matrix
Sig = aa$Sig.final

## Step 2: Generate a heatmap

# Build the matrix of expected scores
ee = generalExpScores(Sig)

# Plot it
par(mfrow=c(1,2))
drawHeatmap(ee)
drawBoard(new=TRUE)
drawAimSpot(ee)

drawAimSpot

---

**Optimal Aiming Spot**

**Description**

Draws the optimal aiming location, i.e. the spot with the highest expected score, on top of an existing plot.
**drawBoard**

**Usage**

drawAimSpot(e, col = "blue", pch = 19, ...)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>e</code></td>
<td>A matrix of the expected scores, created by the function <code>simpleExpScores</code> or <code>generalExpScores</code>.</td>
</tr>
<tr>
<td><code>col</code></td>
<td>The color of the dot.</td>
</tr>
<tr>
<td><code>pch</code></td>
<td>The plotting &quot;character&quot; for the dot.</td>
</tr>
<tr>
<td><code>...</code></td>
<td>More arguments for drawing the dot.</td>
</tr>
</tbody>
</table>

**Author(s)**

Ryan Tibshirani

**Examples**

```r
# This will take a couple of seconds
e = simpleExpScores(s=0)

# This is what your heatmap would look like if you had perfect accuracy
par(mfrow=c(1,2))
drawHeatmap(e)
drawBoard(new=TRUE)
drawAimSpot(e)
```

---

**drawBoard**  
**Dartboard**

**Description**

Draws the dartboard, either as a new plot or on top of an existing one.

**Usage**

drawBoard(new = FALSE, lines = TRUE, numbers = TRUE, outside = TRUE, col = "black", ...)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>new</code></td>
<td>Make a new plot?</td>
</tr>
<tr>
<td><code>lines</code></td>
<td>Draw the lines separating the regions?</td>
</tr>
<tr>
<td><code>numbers</code></td>
<td>Draw the numbers?</td>
</tr>
<tr>
<td><code>outside</code></td>
<td>Draw the numbers outside (or inside) the dartboard?</td>
</tr>
<tr>
<td><code>col</code></td>
<td>The color for the lines.</td>
</tr>
<tr>
<td><code>...</code></td>
<td>More arguments for drawing the numbers (text).</td>
</tr>
</tbody>
</table>
drawHeatmap

Author(s)

Ryan Tibshirani

Examples

# Draw a new dartboard
drawBoard(new=TRUE)

drawHeatmap 

Heatmap of Expected Scores

Description

Draws a heatmap of the expected score as the aiming location varies across the dartboard, as a new plot.

Usage

drawHeatmap(e, col = heat.colors(30))

Arguments

e  A matrix of the expected scores, created by the function simpleExpScores or generalExpScores.

col  The colors to use for the heatmap.

Author(s)

Ryan Tibshirani

Examples

# This will take a couple of seconds
e = simpleExpScores(s=0)

# This is what your heatmap would look like if you had
# perfect accuracy
drawHeatmap(e)
Description

EM algorithm to estimate your variance based on your scores, in the general model.

Usage

generalem(x, Sig.init = c(10^2, 10^2, 0.1 * 10 * 10), niter = 100, seed = NULL)

Arguments

x          Scores of throws aimed at the center of the dartboard.
Sig.init   The initial guess for the covariance matrix, represented as a vector: x marginal
           variance, then y marginal variance, then x-y covariance.
niter      The number of iterations.
seed       The seed for the random number generator (the E-step is done by importance
           sampling).

Value

Sig.final  The final estimate of the covariance matrix.
Sig.init   The initial estimate of the covariance matrix.
Sig        The estimate of the covariance at each iteration.
loglik     The log likelihood at each iteration—currently not implemented (this is just an
           array of 0s).
niter      The number of iterations.

Author(s)

Ryan Tibshirani

Examples

# Scores of 100 of my dart throws, aimed at the center of the board
x = c(12,16,19,3,17,1,25,19,17,50,18,1,3,17,2,2,13,18,16,2,25,5,5,
     1,5,4,17,25,25,50,3,7,17,17,3,3,3,7,11,10,25,1,19,15,4,1,5,12,17,16,50,20,20,20,25,50,2,17,3,20,20,20,20,5,1,18,15,2,3,25,12,9,3,3,19,16,20,
     5,5,1,4,15,16,5,20,16,2,25,12,25,11,25,7,2,5,19,17,17,2,12)

# Get my variance in the general Gaussian model
a = generalem(x,niter=100,seed=0)

# The EM estimate of my covariance matrix
Sig = a$Sig.final
generalExpScores

Expected Scores for the General Model

Description

Computes the expected score as the aiming location varies across the board, using the specified
covariance matrix and the general model for dart throws.

Usage

generalExpScores(Sig)

Arguments

Sig  The covariance matrix.

Value

e  The matrix of expected scores.

Author(s)

Ryan Tibshirani

Examples

# Generate the matrix of expected scores if my covariance is
# Sig=c(15^2,30^2,2,0), in the general model. This will take a
# couple of seconds.
e = generalExpScores(Sig=c(15^2,30^2,2,0))

# Draw a heatmap!
drawHeatmap(e)

simpleEM

EM Algorithm for the Simple Model

Description

EM algorithm to estimate your variance based on your scores, in the simple model.

Usage

simpleEM(x, s.init = 100, niter = 100)
Arguments

- **x**: Scores of throws aimed at the center of the dartboard.
- **s.init**: The initial guess for the marginal variance.
- **niter**: The number of iterations.

Value

- **s.final**: The final estimate of the variance.
- **s.init**: The initial estimate of the variance.
- **s**: The estimate of the variance at each iteration of the EM algorithm.
- **loglik**: The (observed) log likelihood at each iteration of the EM algorithm.
- **niter**: The number of iterations.

Author(s)

Ryan Tibshirani

Examples

```r
# Scores of 100 of my dart throws, aimed at the center of the board
x = c(12,16,19,3,17,1,25,19,17,50,18,1,3,17,2,2,13,18,16,2,25,5,5,
1,5,4,17,25,25,50,3,7,17,17,3,3,3,7,11,10,25,1,19,15,4,1,5,12,17,16,
50,20,20,25,50,2,17,3,20,20,5,1,18,15,2,3,25,12,9,3,3,19,16,20,
5,5,1,4,15,16,5,20,16,2,25,6,12,25,11,25,7,2,5,19,17,17,2,12)

# Get my variance in the simple Gaussian model
a = simpleEM(x,niter=100)

# Check the log likelihood
plot(1:a$niter,a$loglik,type="l",xlab="Iteration",ylab="Log likelihood")

# The EM estimate of my variance
s = a$s.final
```

**simpleExpScores**  
*Expected Scores for the Simple Model*

Description

Computes the expected score as the aiming location varies across the board, using the specified variance and the simple model for dart throws.

Usage

```r
simpleExpScores(s)
```
simpleExpScores

Arguments

s  The marginal variance.

Value

e  The matrix of expected scores.

Author(s)

Ryan Tibshirani

Examples

# Generate the matrix of expected scores if my variance is 25^2,
# in the simple model. This will take a couple of seconds.
e = simpleExpScores(s=25^2)

# Draw a heatmap!
drawHeatmap(e)
Index

*Topic **package**
  darts-package, 2
*Topic
darts-package, 2

darts (darts-package), 2
darts-package, 2
drawAimSpot, 3
drawBoard, 4
drawHeatmap, 5

generalEM, 6
generalExpScores, 7

simpleEM, 7
simpleExpScores, 8