

Package ‘TaoTeProgramming’

February 19, 2015

Type Package

Title Illustrations from Tao Te Programming

Version 1.0

Date 2014-06-22

Author Pat Burns

Maintainer Pat Burns <patrick@burns-stat.com>

Description Art-like behavior based on randomness

License Unlimited

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-22 23:15:59

R topics documented:

TaoTeProgramming-package	2
bendplot	2
butterflies	4
canvas	5
chessboard	5
mountscene	6
P.backcolor	7
polyhull	8
rectent02	9
roads	10
safesample	11
treeColor	12
Index	14

TaoTeProgramming-package

Illustrations from the book 'Tao Te Programming'

Description

The functions to create the actual illustrations plus the underlying random-based functions that they rely upon.

Details

Package:	TaoTeProgramming
Type:	Package
Version:	1.0
Date:	2014-06-08
License:	Unlimited

Author(s)

Pat Burns

Maintainer: Pat Burns <patrick@burns-stat.com>

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

bendplot

Bend plots

Description

Random bend plots. Describing them is hard, seeing them is easy.

Usage

```
bendplot(num = 20000, xdelta = 0.1, ydelta = 0.2,
  sd = 1, lwd = 2, color = "black", seed = NULL)
bendplotmulticol(num = 20000, xdelta = 0.1, ydelta = 0.2,
  sd = 1, lwd = 2, color = colors(), cnum = 100, seed = NULL)
bend(num = 20000, xdelta = 100, ydelta = 200, sd = 1)
```

Arguments

<code>num</code>	number of data points
<code>xdelta</code>	horizontal scaling.
<code>ydelta</code>	vertical scaling
<code>sd</code>	standard deviation of normal deviates.
<code>lwd</code>	a single number giving the line width.
<code>color</code>	one or more colors.
<code>seed</code>	an integer giving the argument for <code>set.seed</code> .
<code>cnum</code>	an integer giving the number of colors to use.

Value

`bend` returns a two-column matrix.

Side effects

`bendplot` and `bendplotmultcol` create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[set.seed](#).

Examples

```
# one color
bendplot(xdelta=150, ydelta=150,
  color=grep("green", colors(), value=TRUE), seed=2)

# multiple colors
bendplotmultcol(xdelta=150, ydelta=150,
  color=grep("green", colors(), value=TRUE), seed=2)

bendplotmultcol(xdelta=100, ydelta=200, sd=.001,
  color=grep("sienna", colors(), value=TRUE))
```

butterflies*Draw butterflies*

Description

Draw one butterfly, or multiple butterflies.

Usage

```
butterflies(num = c(100, 10), scale = c(0.03, 0.1),
  color = grep("^gray", colors(), value = TRUE), seed = NULL)
butterfly(loc, color, scale = 0.1, rand = 0.1)
```

Arguments

num	a vector giving the number of butterflies to draw on each scale.
scale	the scales at which to draw butterflies.
color	colors of the butterflies.
seed	an integer giving the argument for <code>set.seed</code> .
loc	length two vector giving the location of a single butterfly.
rand	control for how spreadout the wings can be.

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
# a specific picture
butterflies(color=colors(), seed=7)
# different each time
butterflies(color=colors())
```

canvas	<i>Blank plot</i>
--------	-------------------

Description

Create a blank canvas with coordinates that go from 0 to 1.

Usage

```
canvas()
```

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

Examples

```
canvas()
```

chessboard	<i>Distorted Chessboard</i>
------------	-----------------------------

Description

Draw a distorted chessboard.

Usage

```
chessboard(color = "black", seed = NULL)
```

Arguments

color	a single color.
seed	an integer giving the argument for <code>set.seed</code> .

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
# a specific picture
chessboard(sample(colors(), 1), seed=7)
# different each time
chessboard(sample(colors(), 1))
```

mountscene

Draw mountains, hills and beaches

Description

Draws a variety of scenes depending on some subtle changes.

Usage

```
mountscene(levels = 5, vlen = 200, tilt = 0.2, df = 7,
  color = NULL, box = TRUE, seed = NULL)
hillscene(num = 100, vlen = 200, tilt = 0.2, df = 7,
  color = NULL, seed = NULL)
mountain(values, color, df = 7)
```

Arguments

levels	number of levels of objects in the scene.
vlen	the number of random points along a level.
tilt	the slope of the level. This is in fractions of the vertical dimension – the left side is at <code>tilt</code> below the average level and the right side is <code>tilt</code> above.
df	degrees of freedom of the smoothing of the random points.
color	colors to fill in below each level.
box	logical value: if TRUE, a frame is drawn around the picture.
seed	an integer giving the argument for <code>set.seed</code> .
num	the number of hills.
values	numeric vector that is smoothed.

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
# hills
hillscene(color=grep("green|yellow", colors(), value=TRUE))

# waves
hillscene(tilt=0,
  color=grep("aqua", colors(), value=TRUE))
```

P.backcolor

Tao Te Programming illustrations

Description

Functions to create the actual illustrations that appear in the book.

Usage

```
P.backcolor(filename = "backcolor.png")
```

Arguments

filename	either a character string for the name of the pdf or png file, or NULL to send the plot to the current graphics device.
----------	---

Value

nothing useful.

Side effects

either a file is created or changed, or a plot is sent to the current graphics device.

The exception is P. template which merely has the form of a function of this sort.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[bendplot](#), [butterflies](#), [chessboard](#), [mountscene](#), [polyhull](#), [roads](#), [treeColor](#).

Examples

```
P.backcolor(NULL)

# see all the specific plot functions
ls("package:TaoTeProgramming", pattern='^P')

# view a random plot
get(sample(ls("package:TaoTeProgramming", pattern='^P')))(NULL)
```

polyhull	<i>Draw Irregular Convex Polygons</i>
----------	---------------------------------------

Description

Creates a random montage of convex polygons.

Usage

```
polyhull(num = 100, lambda = 20, size = 0.05,
         color = grep("^gray", colors(), value = TRUE), seed = NULL)
quadtile(dims = c(10, 10), color = grep("^gray", colors(), value = TRUE),
         seed = NULL)
```

Arguments

num	number of polygons.
dims	dimensions for a certain matrix.
lambda	control for the random number of sides to the polygons.
size	the maximum radius of a polygon.
color	vector of colors for the polygons.
seed	an integer giving the argument for <code>set.seed</code> .

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
polyhull(color=grep("red", colors(), value=TRUE))

polysort(color=grep("green", colors(), value=TRUE))

polytile(color=grep("purple", colors(), value=TRUE))

quadtile(color=grep("orange", colors(), value=TRUE))
quadtile(color=grep("orange", colors(), value=TRUE),
  dims=c(2,20))

quadtilebalance(color=grep("turquoise", colors(), value=TRUE))

tritile(color=grep("sea", colors(), value=TRUE))
```

rectent02

Low entropy plot

Description

A campaign that failed: trying to minimize entropy of an image.

Usage

```
rectent02(x = array(sample(c(TRUE, FALSE), 4e+06, replace = TRUE), c(2000, 2000)),
  iterations = 100, inner = 20, fraction = c(0.01, 0.1),
  seed = NULL, verbose = TRUE)
entropy(xmat)
```

Arguments

<code>x</code>	a matrix of logicals.
<code>xmat</code>	a matrix of logicals.
<code>iterations</code>	the number of major iterations.
<code>inner</code>	the number of iterations in the inner loop.
<code>fraction</code>	controls the size of the inner region that is optimized.
<code>seed</code>	an integer giving the argument for <code>set.seed</code> .
<code>verbose</code>	logical value. If <code>TRUE</code> , then information on the progress of the optimization is printed.

Value

rectent02 and rectent01 return a matrix like the input x.

entropy returns a number.

Side effects

modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[set.seed.](#)

Examples

```
## Not run:
entmat1 <- rectent02()
entmat2 <- rectent02(entmat1, iterations=500)
image(entmat2)
image(xor(entmat2, entmat1))

## End(Not run)
```

roads

Draw linear forms

Description

A few ways of using lines to get possibly interesting images.

Usage

```
roads(num = 100, hprob = 0.6, color = "black", lwd = 1, seed = NULL)
```

Arguments

num	the number of objects (lines).
hprob	probability that a line is horizontal.
color	vector of colors.
lwd	vector of line widths.
seed	an integer giving the argument for <code>set.seed.</code>

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
roads(color=grep("dark", colors(), value=TRUE), lwd=2:3)

waves(color=grep("wheat", colors(), value=TRUE), lwd=2)

waves01(color=grep("orchid", colors(), value=TRUE), lwd=2:5)

waves02(color=colors(), lwd=2)

# one color only
sticks(color="violet", lwd=2)
```

safesample

Safely Sample a Single Item

Description

Single random selection from the input vector.

Usage

```
safesample(x)
```

Arguments

x a vector.

Details

This works around the infelicity of the `sample` function that can give the wrong result if `x` has length 1. See Circle 8.2.33 of 'The R Inferno'.

Value

a random element of the input `x`.

Side effects

modify the random seed.

References

<http://www.burns-stat.com/documents/books/the-r-inferno/>

See Also

[sample](#).

Examples

```
safesample(2:6) # like 'sample(2:6, 1)'

safesample(4) # still works as intended

#compare:
sample(4)
```

treeColor

Draw trees

Description

Draw trees.

Usage

```
treeColor(branches = 30, trunkColor = NULL, branchColor = NULL,
  seed = NULL)
tree(branches = 30, seed = NULL)
```

Arguments

<code>branches</code>	integer giving the number of branches.
<code>trunkColor</code>	a vector of colors, the default is the browns.
<code>branchColor</code>	a vector of colors, the default is the greens.
<code>seed</code>	an integer giving the argument for <code>set.seed</code> .

Value

nothing useful.

Side effects

create a plot on the current graphics device, and modify the random seed.

References

<http://www.burns-stat.com/documents/books/tao-te-programming/>

See Also

[canvas](#), [set.seed](#).

Examples

```
# a specific picture
treeColor(seed=7)
# different each time
treeColor()
```

Index

*Topic **distribution**

safesample, 11

*Topic **hplot**

bendplot, 2

butterflies, 4

canvas, 5

chessboard, 5

mountscene, 6

P.backcolor, 7

polyhull, 8

roads, 10

treeColor, 12

*Topic **package**

TaoTeProgramming-package, 2

bend (bendplot), 2

bendplot, 2, 8

bendplotmultcol (bendplot), 2

bundle (treeColor), 12

butterflies, 4, 8

butterfly (butterflies), 4

canvas, 4, 5, 6, 7, 9, 11, 13

chessboard, 5, 8

entropy (rectent02), 9

hillscene (mountscene), 6

mountain (mountscene), 6

mountscene, 6, 8

P.backcolor, 7

P.bend01 (P.backcolor), 7

P.bend03 (P.backcolor), 7

P.bend04 (P.backcolor), 7

P.bend05 (P.backcolor), 7

P.butterflies01 (P.backcolor), 7

P.butterflies02 (P.backcolor), 7

P.chess (P.backcolor), 7

P.chess02 (P.backcolor), 7

P.chess03 (P.backcolor), 7

P.frontcolor (P.backcolor), 7

P.hillscene01 (P.backcolor), 7

P.mountscene01 (P.backcolor), 7

P.mountscene02 (P.backcolor), 7

P.mountscene03 (P.backcolor), 7

P.mountscene04 (P.backcolor), 7

P.polyhull01 (P.backcolor), 7

P.polysort01 (P.backcolor), 7

P.polytile01 (P.backcolor), 7

P.polytile02 (P.backcolor), 7

P.polytile03 (P.backcolor), 7

P.polytile04 (P.backcolor), 7

P.polytile05 (P.backcolor), 7

P.quadtile01 (P.backcolor), 7

P.quadtile02 (P.backcolor), 7

P.quadtile03 (P.backcolor), 7

P.quadtile04 (P.backcolor), 7

P.roads01 (P.backcolor), 7

P.roads02 (P.backcolor), 7

P.sticks01 (P.backcolor), 7

P.sticks02 (P.backcolor), 7

P.sticks03 (P.backcolor), 7

P.sticks04 (P.backcolor), 7

P.template (P.backcolor), 7

P.tree01 (P.backcolor), 7

P.tree02 (P.backcolor), 7

P.tree03 (P.backcolor), 7

P.tritile01 (P.backcolor), 7

P.tritile02 (P.backcolor), 7

P.tritile03 (P.backcolor), 7

P.wave01 (P.backcolor), 7

P.wave02 (P.backcolor), 7

P.wavescene01 (P.backcolor), 7

P.wavescene02 (P.backcolor), 7

P.wavescene03 (P.backcolor), 7

polyhull, 8, 8

polysort (polyhull), 8

polytile (polyhull), 8

quadtile (polyhull), 8
quadtilebalance (polyhull), 8

rectent01 (rectent02), 9
rectent02, 9
roads, 8, 10

safesample, 11
sample, 12
set.seed, 3, 4, 6, 7, 9–11, 13
sticks (roads), 10

TaoTeProgramming
 (TaoTeProgramming-package), 2
TaoTeProgramming-package, 2
tree (treeColor), 12
treeColor, 8, 12
tritile (polyhull), 8

waves (roads), 10
waves01 (roads), 10
waves02 (roads), 10