Package 'MandalaR'

November 2, 2022

Type Package

Title Building Mandalas from Parametric Equations of Classical Curves

Version 0.1.0

Author Luciane Ferreira Alcoforado

Maintainer Luciane Ferreira Alcoforado <lucianea@id.uff.br>

Description Provides an algorithm for creating mandalas. From the perspective of classic mathematical curves and rigid movements on the plane, the package allows you to select curves and produce mandalas from the curve. The algorithm was developed based on the book by Alcoforado et. al. entitled ``Art, Geometry and Mandalas with R" (2022) in press by the USP Open Books Portal.

Depends R (>= 3.2)

Imports ggplot2

License GPL-3

URL https://lucianealcoforado.shinyapps.io/Mandala/

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-02 19:42:41 UTC

R topics documented:

`f_factor` *Mandalar: package for building mandalas from parametric equations of classical curves*

Description

Function to reduce points

Usage

```
f_factor(x, y, k)
```

Arguments

`x`	is a vector length n with coordinate x of point
`y`	is a vector length n with coordinate y of point
`k`	is a vector with factor of decrease or increase points

Value

Returns a dataframe with the original points plus the respective changed points.

Examples

```
x=c(1,1)
y=c(0,1)
k=c(0.5)
f_factor(x,y,k)
```

`f_rotacao` *Mandalar: package for building mandalas from parametric equations of classical curves*

Description

Function to rotate points by one or more angles

Usage

```
f_rotacao(x, y, rotacao)
```

Arguments

`x`	is a vector length n with coordinate x of point
`y`	is a vector length n with coordinate y of point
`rotacao`	is a vector of length k with angles in radians to rotate the point (x,y)

Details

If x and y dimension is n and rotation dimension is k, then function f_rotacao will return a dataframe with two columns and (n+1)k rows

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Examples

```
x=c(1,1)
y=c(0,1)
rotacao=c(pi/3, pi/2, pi)
f_rotacao(x,y,rotacao)
```

`f_trans` *creates a dataframe containing the points for the espiral hiperbolica mandala*

Description

Function to translation points by shifts on the x-axis or y-axis

Usage

```
f_trans(x, y, t, d)
```

Arguments

x	is a vector length n with coordinate x of point
y	is a vector length n with coordinate y of point
t	is a vector with shifts on the x or y-axis
d	is a direction translation, 1)x or 2)y

Value

Returns a dataframe with the original points plus the respective translation of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
x=c(1,1)
y=c(0,1)
t=c(-3, 3)
d=1
f_trans(x,y,t,d)
```

---

f_transxy *creates a dataframe containing the points for the espiral hiperbolica mandala*

Description

Function to translation points by shifts on the x-axis or y-axis or both

Usage

```
f_transxy(x, y, tx, ty)
```

Arguments

x	is a vector length n with coordinate x of point
y	is a vector length n with coordinate y of point
tx	is a vector with with shifts on the x-axis
ty	is a vector with with shifts on the y-axis

Value

Returns a dataframe with the original points plus the respective translation of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
x=c(1,1)
y=c(0,1)
tx=c(-1,-2)
ty=c(0,0)
f_transxy(x,y,tx,ty)
```

---

mandalar_basic *Create a mandala with algorithm basic*

Description

Function to create a mandala with the basic method

Usage

```
mandalar_basic(curve, theta, k, n, raio, a, b)
```

Arguments

curve	Either a character string or a function returning curve equation evaluated at its first argument. Curves "circle","elipse", "cardioide","limacon", "espiral1", "espiral2", "lemniscata", "deltoide" and "astroide" are recognised, case being ignored.
theta	is a vector length 2 with start angle and end angle
k	is a angle of rotations, k in (0,360) graus
n	is a number of points
raio	is a positive number for the radius of circle
a	is one of the parameters of the curves; for the ellipse is the radius on the x axis
b	is one of the parameters of the curves; for the ellipse is the radius on the y axis

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
require(ggplot2)
mandalar_basic("circle", theta = c(0,2*pi), raio=1, k = 45, n=500)
mandalar_basic("cardioide", theta = c(0,2*pi), raio=1, k = 60, n=500)
mandalar_basic("elipse", theta = c(0,2*pi), a=1, b=2, k = 30, n=500)
```

---

pastroide *creates a dataframe containing the points for the astroide mandala*

Description

Function to build a astroide

Usage

```
pastroide(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value. For astroide we do r=3.
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi)
k=45
raio = 1
n=20
pastroide(theta, raio, k, n)
```

pcardioide *creates a dataframe containing the points for the cardioide mandala*

Description

Function to build a cardioide

Usage

```
pcardioide(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi)
k=45
raio = 1
n=20
pcardioide(theta, raio, k, n)
```

pcircle *creates a dataframe containing the points for the circle mandala*

Description

Function to build point for the circle base

Usage

```
pcircle(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi) #half turn angle
raio = 1
k = 45
n=20
pcircle(theta, raio, k, n)
```

---

pdeltoide *creates a dataframe containing the points for the deltoide mandala*

Description

Function to build a deltoide

Usage

```
pdeltoide(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value. For deltoide we do r=2.
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi)
k=45
raio = 1
n=20
pdeltoide(theta, raio, k, n)
```

---

pelipse *creates a dataframe containing the points for the elipse mandala*

Description

Function to build point for the elipse base

Usage

```
pelipse(theta, a, b, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
a	is one of the parameters of the curves; for the ellipse is the radius on the x axis
b	is one of the parameters of the curves; for the ellipse is the radius on the y axis
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi) #half turn angle
a = 1
b=2
k = 90
n=20
pelipse(theta, a, b, k, n)
```

`pespiral1` *creates a dataframe containing the points for the Fermat espiral mandala*

Description

Function to build a espiral de Fermat

Usage

```
pespiral1(theta, raio, k, n)
```

Arguments

`theta`	is a vector length 2 with start angle and end angle
`raio`	is a vector length 1 with radius value
`k`	is a vector of length 1 with angles in degree to rotate the point (x,y)
`n`	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,6*pi)
k=45
raio = 1
n=20
pespiral1(theta, raio, k, n)
```

`pespiral2` *creates a dataframe containing the points for the espiral hiperbolica mandala*

Description

Function to build a espiral hiperbolica

Usage

```
pespiral2(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,6*pi)
k=45
raio = 1
n=20
pespiral2(theta, raio, k, n)
```

---

plemniscata *creates a dataframe containing the points for the lemniscata mandala*

Description

Function to build a lemniscata

Usage

```
plemniscata(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi)
k=45
raio = 1
n=20
plemniscata(theta, raio, k, n)
```

---

plimacon *creates a dataframe containing the points for the limacon mandala*

Description

Function to build a limacon

Usage

```
plimacon(theta, raio, k, n)
```

Arguments

theta	is a vector length 2 with start angle and end angle
raio	is a vector length 1 with radius value
k	is a vector of length 1 with angles in degree to rotate the point (x,y)
n	is a number of points

Value

Returns a dataframe with the original points plus the respective rotations of these points.

Author(s)

Luciane Ferreira Alcoforado

Examples

```
theta = c(0,2*pi)
k=45
raio = 1
n=20
plimacon(theta, raio, k, n)
```

plot_mandala *creates a mandala visualization*

Description

Function to plot a mandala with points in dataframe

Arguments

dt	dataframe with points x and y

Value

Returns a plot

Author(s)

Luciane Ferreira Alcoforado

Examples

```
require(ggplot2)
n=500; raio=1; t=seq(0,2*pi, length.out = n)
x1=raio*cos(t)
y1=raio*sin(t)
#pontos para os 3 círculos: translação dos pontos iniciais (x1,x=c(x1,x1-raio,x1-2*raio)
x=c(x1,x1-raio,x1-2*raio)
y=c(y1,y1,y1)
dt=data.frame(x,y,z="circulo")
rotacao = (pi/8)*(1:16); n=length(x); xt1=x; yt1=y
dt=f_rotacao(x=dt$x, y=dt$y, rotacao)
plot_mandala(dt)
```

Index