Package: RoundAndRound (via r-universe)

September 17, 2024

Title Plot Objects Moving in Orbits

Version 0.0.1

Maintainer Lele Shu <lele.shu@gmail.com>

Description Visualize the objects in orbits in 2D and 3D. The packages is under developing to plot the orbits of objects in polar coordinate system. See the examples in demo.

Depends R (>= 3.0.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports geometry, methods, graphics, rgl

Repository https://shulele.r-universe.dev

RemoteUrl https://github.com/shulele/roundandround

RemoteRef HEAD

RemoteSha d115e407dfc1e6c67a86bf516ecd70642cbb111e

Contents

---

ab2c *Calculate c in Focus (c, 0)*

Description

Calculate c in Focus (c, 0)

Usage

```
ab2c(a = 1, ab)
```

Arguments

a	Semi-major (Ellipse) or Radium (Ring).
ab	Semi-major over semi-minor. ab=1 for a Ring.

Value

c in Focus (c, 0)

Examples

```
x1=PCS2CCS(a=10, ab=1.5)
x2=PCS2CCS(a=9, ab=1.2)
c1 = ab2c(a=10, ab=1.5)
c2 = ab2c(a=9, ab=1.2)
plot(x1, type='n', xlim=c(-10,10), ylim=c(-10,10), asp=1)
abline(h=0, v=0, asp=1, lty=2)
lines(x1, col=2);
points(c1, 0, col=2)
lines(x2, col=3);
points(c2, 0, col=3)
```

Arrow.pcs *Add arrows in Polar Coordinate System*

Description

Add arrows in Polar Coordinate System

Usage

```
Arrow.pcs(theta, r1 = 0, r2 = 1e+06, o1 = c(0, 0), o2 = o1,
  ab1 = 1, ab2 = ab1, ...)
```

Arguments

theta	Angle in polar coordinate system
r1, r2	Radius of start and end points of the arrow.
o1, o2	Origin
ab1, ab2	Semi-major over semi-minor. ab=1 for a Ring.
...	More options for graphics::arrows function.

Examples

```
x1=PCS2CCS(a=10, ab=1.5)
c1 = ab2c(a=10, ab=1.5)
plot(x1, type='n', xlim=c(-10,10), ylim=c(-10,10), asp=1)
abline(h=0, v=0, asp=1, lty=2)
graphics::lines(x1, col=2);
points(c1, 0, col=2) # focus
Arrow.pcs(theta = 1:12 * 30, r1=0, r2=10, ab1=1.5, length=.1, col=2, o1 = c(c1,0), o2=c(0,0))
```

Arrow3D *Plot 3D Arrow axis.* Arrow3D

Description

Plot 3D Arrow axis. Arrow3D

Usage

```
Arrow3D(len = 10, orig = c(0, 0, 0), cols = c(2:4), ...)
```

Arguments

len	Length of the arrow.
orig	Origin of the axis.
cols	Colors of axis.
...	More options of arrow3d().

---

d2r *Degree to Radian*

Description

Degree to Radian

Usage

```
d2r(x)
```

Arguments

x	Degree

Value

Radian

Examples

```
r = (1:100)/100 * 4 * pi
d = r2d(r)
rr = d2r(d)
plot(d, sin(rr));
abline(h=0 )
abline(v = 360)
```

FactSheet *This is data to be included in my package*

Description

This is data to be included in my package

Orbit.location *Calculate location of a planet* Orbit.location

Description

Calculate location of a planet Orbit.location

Usage

```
Orbit.location(t, p.orb, a = 1, theta = 0, orig = c(0, 0), ab = 1)
```

Arguments

t	Time (day).
p.orb	Period of the orbit.
a	Radius or Semi-major of the orbit.
theta	angle in PCS.
orig	Reference orgin.
ab	Semi-major over semi-minor. ab=1 for a Ring.

Value

(x,y) in Cartesian Coordinate System.

Examples

```
tday = seq(0, 365, 30)
x=Orbit.location(t=tday, p.orb = 365, a=10)
plot(PCS2CCS(0:360, a=10), type='l')
plotplanet(orig=x, rad = .51)
grid()
```

orbit.parameter *Give the orbit the parameter*

Description

Give the orbit the parameter

Usage

```
orbit.parameter(a, b = NULL, ab = NULL)
```

Arguments

a	Semi-major axis
b	Semi-minor axis
ab	Semi-major over semi-minor. ab=1 for a Ring.

Examples

```
orbit.parameter(a=1, b=1.5)
```

PCS2CCS *Convert Polar Coordinate System to Cartesian Coordinate System.*

Description

Convert Polar Coordinate System to Cartesian Coordinate System.

Usage

```
PCS2CCS(theta = 0:360, a = 1, ab = 1, orig = c(0, 0),
  rotation = 0, clockwise = FALSE)
```

Arguments

theta	angle in PCS.
a	Semi-major (Ellipse) or Radium (Ring).
ab	Semi-major over semi-minor. ab=1 for a Ring.
orig	Reference orgin. Default = c(0, 0)
rotation	Rotation of the theta=0
clockwise	Whether clockwise, Default = FALSE

Value

(x,y) in Cartesian Coordinate System.

Examples

```
x1=PCS2CCS(a=10, ab=1.5)
x2=PCS2CCS(a=9, ab=1.2)
c1 = ab2c(a=10, ab=1.5)
c2 = ab2c(a=9, ab=1.2)

plot(x1, type='n', xlim=c(-10,10), ylim=c(-10,10), asp=1)
abline(h=0, v=0, asp=1, lty=2)
lines(x1, col=2);
points(c1, 0, col=2)
```

```
lines(x2, col=3);
points(c2, 0, col=3)

# Test 2
x1=PCS2CCS(a=10, ab=1.5, clockwise = FALSE, rotation=0);
x2=PCS2CCS(a=8, ab=1.5, clockwise = FALSE, rotation=45);
plot(x1, asp=1, col=terrain.colors(nrow(x1)), pch=19)
points(x2, asp=1, col=terrain.colors(nrow(x1)))
```

`plotclock` *Plot a clock*

Description

Plot a clock

Usage

```
plotclock(time = c(as.numeric(format(Sys.time(), format = "%H")),
  as.numeric(format(Sys.time(), format = "%M")),
  as.numeric(format(Sys.time(), format = "%S"))), rad = 1, ab = 1,
  orig = c(0, 0), val = 1:12, angle = c((time[1] +
  ifelse(is.na(time[2]), 0, time[2]/60) + ifelse(is.na(time[3]), 0,
  time[3]/3600)) * 30, (time[2] + ifelse(is.na(time[3]), 0, time[3]/60)) *
  6, time[3] * 6) * (-1) + 90, val.arg = list(col = "blue", cex = 1),
  arr.arg = list(col = c(1, 3, 2), lwd = ring.arg$lwd/(1:3) * 1.5, lty =
  rep(1, 3), arrlen = rep(0.1, 3), length = c(0.5, 0.8, 0.9) * rad),
  ring.arg = list(col = "gold", type = "l", lwd = 4, len.tick = rad *
  0.05), fun.plot = lines, add = F, ...)
```

Arguments

`time`	vector, c(h, m, s)
`rad`	Radius of the clock
`ab`	Semi-major over semi-minor. ab=1 for the planet
`orig`	Origin of the clock.
`val`	The labels on clock edge
`angle`	Angle of the three Arrows, default angles are calculated from the time
`val.arg`	Arguments for plot values on clock.
`arr.arg`	Arguments for plot three arrows on clock.
`ring.arg`	Arguments for plot ring on clock.
`fun.plot`	Plot function
`add`	Whether add plot to existing plot.
`...`	More options in plot function.

Examples

```
plot(0, type='n', xlim=c(-1,1)*1.5, ylim=c(-1,1)*1.5, asp=1)
plotclock(add=TRUE,fun.plot = lines, orig=c(0.5,0),
rad=.25, val=rep('', 60), time=c(NA, NA, as.numeric(format(Sys.time(), '%S'))))
plotclock(add=TRUE,fun.plot = lines, orig=c(0, .5),
rad=.25, val=rep('', 60), time=c(NA, as.numeric(format(Sys.time(), '%M')), NA) )
plotclock(add=TRUE,fun.plot = lines, orig=c(-.5,0),
rad=.25, val=rep('', 60), time=c(as.numeric(format(Sys.time(), '%H')), NA,  NA))
plotclock(add=TRUE, fun.plot=lines)
```

`plotfan` *Plot a fan*

Description

Plot a fan

Usage

```
plotfan(theta, a = 1, ab = 1, orig = c(0, 0), fun = graphics::plot,
  ...)
```

Arguments

`theta`	Angle in polar coordinate system
`a`	Radius of start and end points of the arrow.
`ab`	Semi-major over semi-minor. ab=1 for a Ring.
`orig`	Origin
`fun`	Plot function. default = plot
`...`	More options in plot function

Examples

```
n=50
theta =0:n
plotpcs(theta = theta, a=1, ab=1, asp=1)
plotfan(theta = theta * 10, a=1, ab=1, fun=polygon)
```

plotpcs *Plot in polar coordinate system*

Description

Plot in polar coordinate system

Usage

```
plotpcs(theta, a = 1, ab = 1, orig = c(0, 0), fun = graphics::plot,
  ...)
```

Arguments

theta	Angle in polar coordinate system
a	Radius of start and end points of the arrow.
ab	Semi-major over semi-minor. ab=1 for a Ring.
orig	Origin
fun	Plot function. default = plot
...	More options in plot function

Examples

```
n=50
par(mfrow=c(2,1))
plotpcs(theta = 1:n * 15, a=1:n/10, ab=1, type='l', asp=1)
plotpcs(theta = 1:n * 10, a=1:n/10, ab=1, type='l', asp=1)
xy = PCS2CCS(theta = 1:n * 10, a=1:n/10, ab=1)
xy[,1]=xy[,1]+1
points(xy, pch=19, col=terrain.colors(nrow(xy)))
```

plotplanet *Plot a planet*

Description

Plot a planet

Usage

```
plotplanet(orig = c(0, 0), rad = 1, theta = 0,
  fun = graphics::lines, cols = "gray", ab = 1, arrow = TRUE,
  arrow.len = 0.1, ...)
```

Arguments

orig	Origin
rad	Radius of the planet
theta	Angle of the Arrow inside of the planet
fun	Function to plot the planet
cols	Color of planet and arrow.
ab	Semi-major over semi-minor. ab=1 for the planet
arrow	Whether plot the arrow.
arrow.len	Length in arrow function.
...	More options in plot function.

Examples

```
a = 10;
ab=1.5
x1=PCS2CCS(a=a, ab=ab)
c1 = ab2c(a=a, ab=ab)
plot(x1, type='l', xlim=c(-10,10), ylim=c(-10,10), asp=1, col='gray')
Arrow.pcs(theta = 1:12 * 30, r1=0, r2=a, ab1=ab, length=.1, col=2, o1 = c(c1,0), o2=c(0,0))
pos = PCS2CCS(theta = 1:12 * 30, a=a, ab=ab)
plotplanet(orig = pos, arrow.len=0.1)
```

---

r2d *Radian to degree*

Description

Radian to degree

Usage

```
r2d(x)
```

Arguments

x	Radian

Value

Degree

Examples

```
r = (1:100)/100 * 4 * pi
d = r2d(r)
rr = d2r(d)
plot(d, sin(rr));
abline(h=0 )
abline(v = 360)
```

---

rotate *Roate a (x,y) with an Angle.*

Description

Roate a (x,y) with an Angle.

Usage

```
rotate(x, theta, orig = c(0, 0))
```

Arguments

x	Coordinate (x,y)
theta	An angle (degree)
orig	Pivot of rotation

Value

Rotated (x,y)

Examples

```
plot(PCS2CCS(a=2), asp=1)
grid()
x=rbind(c(1,1))
for(i in 1:12){
  points(rotate(x, 30*i, orig=c(1,0)))
}
```

SpaceObject-class *Class of planet* SpaceObject

Description

Class of planet SpaceObject

Value

Class of SpaceObject

Slots

shape Ploting function of the shape

radius Radius for sphere

Period.Rotate data.frame 1*3 c(Period.Rotate, Period.Orbit, Period.Synodic)

SpaceOrbit-class *Class of Orbit* Orbit

Description

Class of Orbit Orbit

Value

Class of SpaceOrbit

Slots

ab Shape of the object, ab=1 Sphere, ab!=1 Ellipsoid

e eccentric of the orbit

radius Radius for sphere (ab=1), or Semi-major axis for Ellipsoid (ab!=1)

period data.frame 1*3 c(Period.Rotate, Period.Orbit, Period.Synodic)

Inclination Inclination.

CenterObject Central Object.

`Status.planet` *Calculate the status of planet* `Status.planet`

Description

Calculate the status of planet `Status.planet`

Usage

```
Status.planet(t, p.orb, ab = 1, r.orb = 1, orig = c(0, 0))
```

Arguments

`t`	Time (day).
`p.orb`	Orbital Period.
`ab`	Semi-major over semi-minor. ab=1 for a Ring.
`r.orb`	Radius of the orbit.
`orig`	Reference orgin.

Value

(x,y) in Cartesian Coordinate System.

Examples

```
tday = seq(0, 365, 30)
x=Status.planet(t=tday, p.orb = 365, r.orb=10)
plot(PCS2CCS(0:360, a=10), type='l')
plotplanet(orig=x[,-1], rad = .51)
grid()
```

Index