AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Biblioteka ‘rgl’

Narzędzia informatyczne w badaniach naukowych
mgr inż. Judyta Książek

Wydział Geodezji Górniczej i Inżynierii Środowiska
Katedra Geoinformacji, Fotogrametrii
i Teledetekcji Środowiska

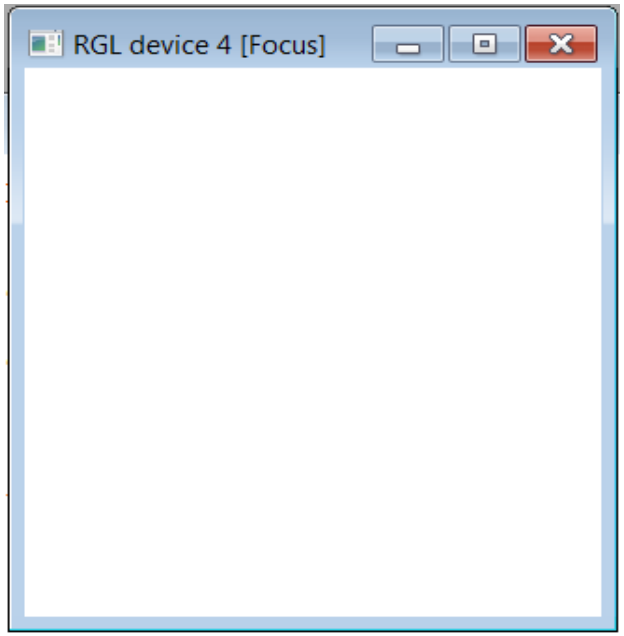
Kraków 07.02.2013

Biblioteka‘rgl’

- **Title: 3D visualization device system (OpenGL)**
- **System/urządzenie do renderowania 3D w czasie rzeczywistym**
- **Biblioteka rgl jest szeroko wykorzystywana jako sterowniki dla R do wizualizacji 3D dla innych specjalistycznych bibliotek**

open3d

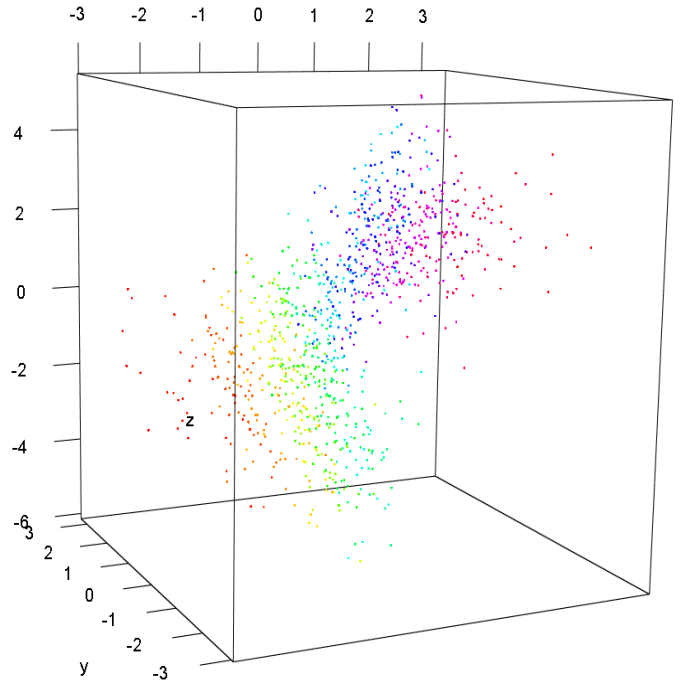
- open3d(..., params=get("r3dDefaults", envir=.GlobalEnv))
- Otwarcie urządzenia do wizualizacji 3D
- r3dDefaults
  – standardowe parametry
  – układ matematyczny



plot3d

- plot3d(x, ...)

- Rysowanie przestrzennych wykresów punktowych

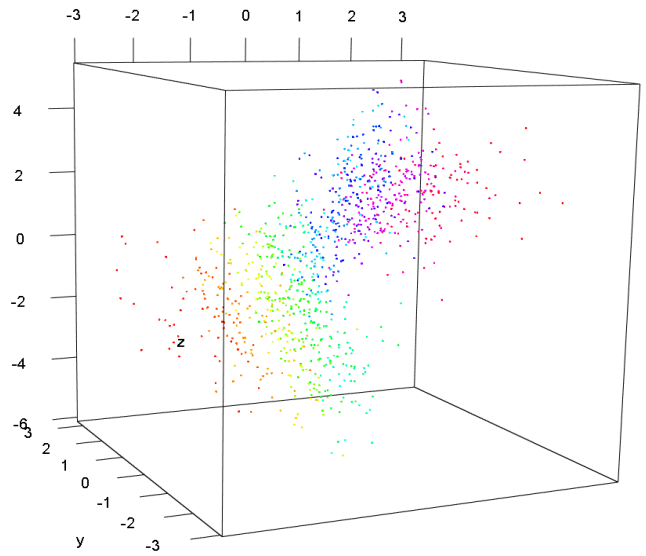


plot3d

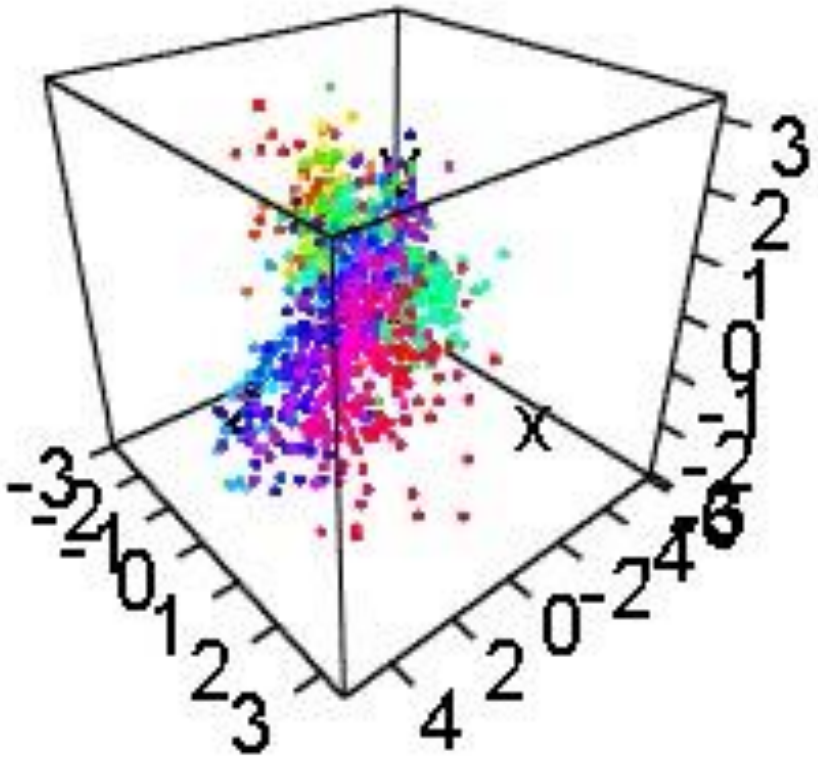
- **Przykład**

```
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x,y)
plot3d(x, y, z, col=rainbow(1000))

setwd(tempdir())
for (i in 1:900) {
rgl.viewpoint(i/2,30)
punkty<-
    paste("punkty",formatC(i,digits=1,flag="0"),".png",sep="")
rgl.snapshot(punkty) }
```



plot3d



play3d
rgl.snapshot

- play3d(f, duration = Inf, dev = rgl.cur(), ..., startTime = 0)
- Wywołanie funkcji wielokrotnie, uwzględniając czas, resetuje punkt widzenia

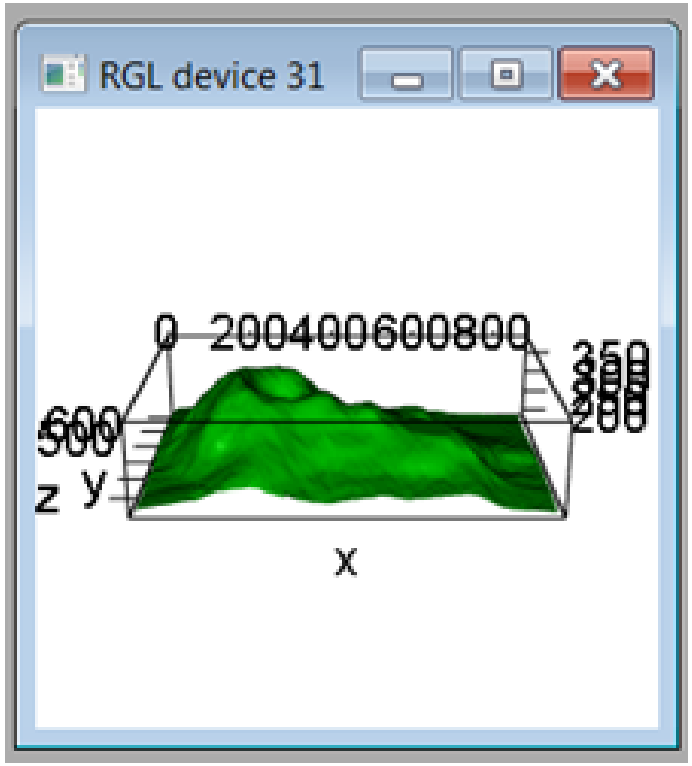
- movie3d(f,…)
- Wykonuje te same czynności, ale zapisuje wynik jako klatki dając możliwość stworzenia filmu

- rgl.snapshot(filename, fmt="png", top=TRUE)
- Zapisywanie poszczególnych klatek do pliku *.png

scene3d

- scene3d()
- Zachowanie RGL device do zmiennej
- Możliwość otwarcia później
- Przykład

- open3d()
- z <- 2 * volcano
- x <- 10 * (1:nrow(z)
- y <- 10 * (1:ncol(z))
- persp3d(x, y, z, col = "green3", aspect="iso")
- s <- scene3d()



abclines

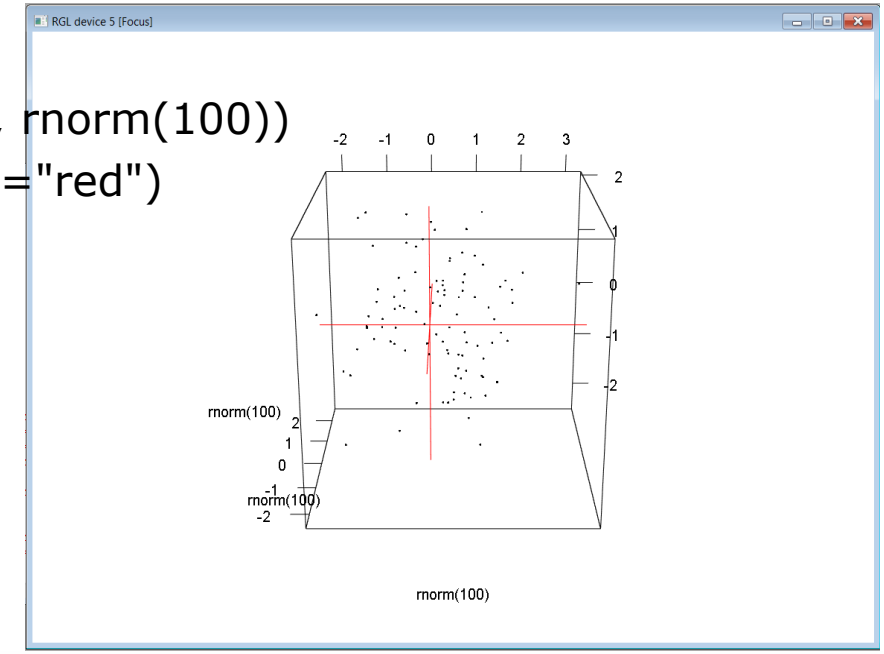
- abclines3d(x, y = NULL, z = NULL, a, b = NULL, c = NULL, ...)
- Dodanie linii do sceny.

  x,y,z – współrzędne przez które przechodzi linia

  a, b, c – współrzędne wyznaczające kierunek linii

- **Przykład**

```
plot3d(rnorm(100), rnorm(100), rnorm(100))
abclines3d(0,0,0, a=diag(3), col="red")
```



mesh3d

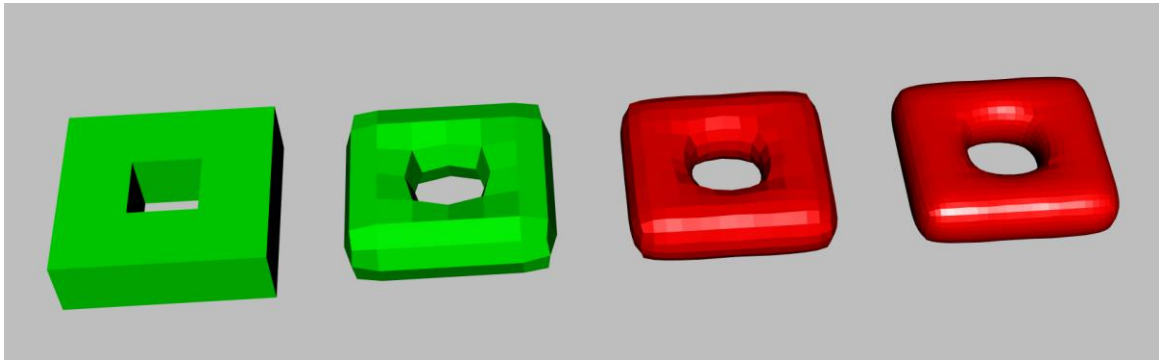
- Tworzenie powierzchni z trójkątów i czworokątów oraz zbiór przykładowych obiektów

- cube3d(trans = identityMatrix(), ...)
- tetrahedron3d(trans = identityMatrix(), ...)
- oh3d(trans = identityMatrix(), ...)
- wire3d(x, ...) – szkieletowe obiekty
- shade3d(x, override = TRUE, ...) – cieniowane obiekty

mesh3d

- **Przykład**

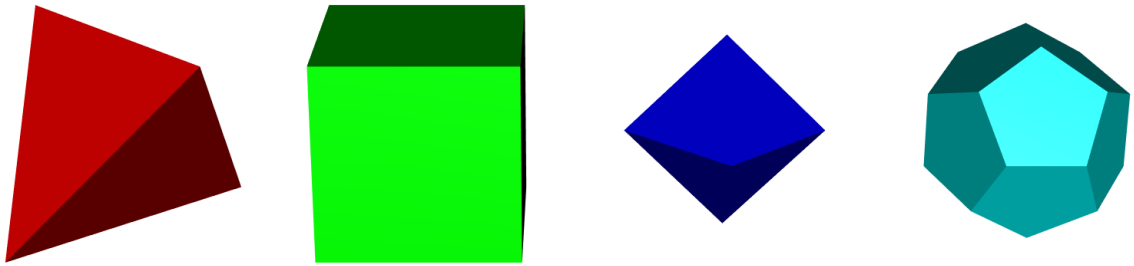
```
open3d()
bg3d("gray")
l0 <- oh3d(tran = par3d("userMatrix"), color = "green" )
shade3d( translate3d( l0, -6, 0, 0 ))
l1 <- subdivision3d( l0 )
shade3d( translate3d( l1 , -2, 0, 0 ), color="red", override = FALSE )
l2 <- subdivision3d( l1 )
shade3d( translate3d( l2 , 2, 0, 0 ), color="red", override = TRUE )
l3 <- subdivision3d( l2 )
shade3d( translate3d( l3 , 6, 0, 0 ), color="red" )
```



mesh3d

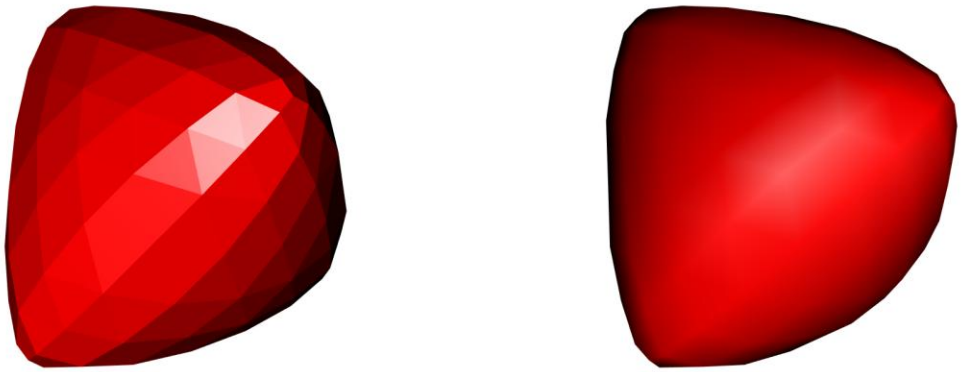
- **Przykład**

```
open3d()
shade3d( translate3d( tetrahedron3d(col="red"), 0, 0, 0) )
shade3d( translate3d( cube3d(col="green"), 3, 0, 0) )
shade3d( translate3d( octahedron3d(col="blue"), 6, 0, 0) )
shade3d( translate3d( dodecahedron3d(col="cyan"), 9, 0, 0) )
shade3d( translate3d( icosahedron3d(col="magenta"), 12, 0, 0) )
```

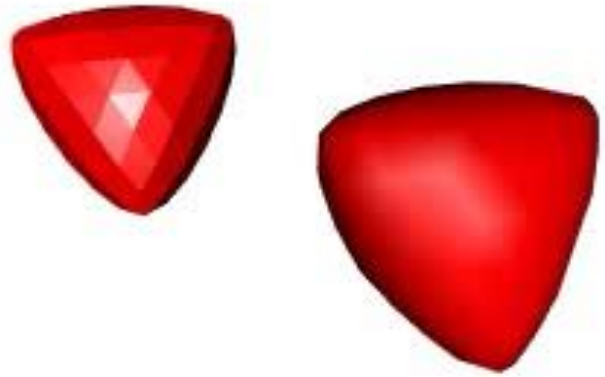


addNormals

- addNormals(x, ...)
- Dodanie normalnych w każdym wierzchołku bryły
- Wyświetlanie powierzchni obiektu bardziej gładko



addNormals



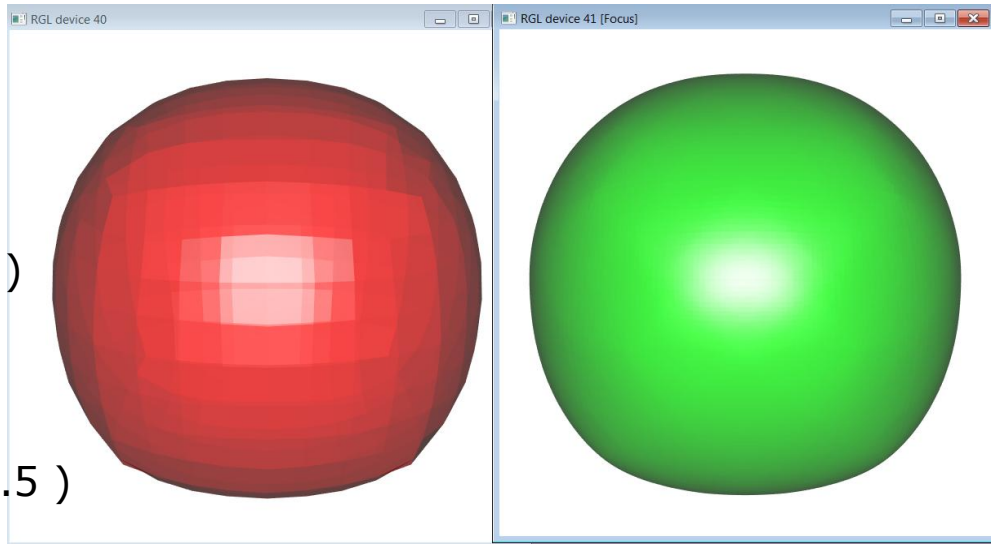
Przykład

```
open3d()
y <-
    subdivision3d(tetrahedron3d(col=
    "red"), depth=3)
shade3d(y) # bez normalych
y <- addNormals(y)
shade3d(translate3d(y, x=1, y=0,
    z=0)) # z normalnymi
setwd(tempdir())
for (i in 1:900) {
rgl.viewpoint(i/3,30)
czerwone<-
    paste("czerwone",formatC(i,digits
    =1,flag="0"),".png",sep="")
rgl.snapshot(czerwone) }
```

subdivision3d

- subdivision3d(x, ...)
- Poprawienie gładkości obiektu poprzez podział na siatkę trójkątów/kwadratów i ponowne przeliczenie
- **Przykład**

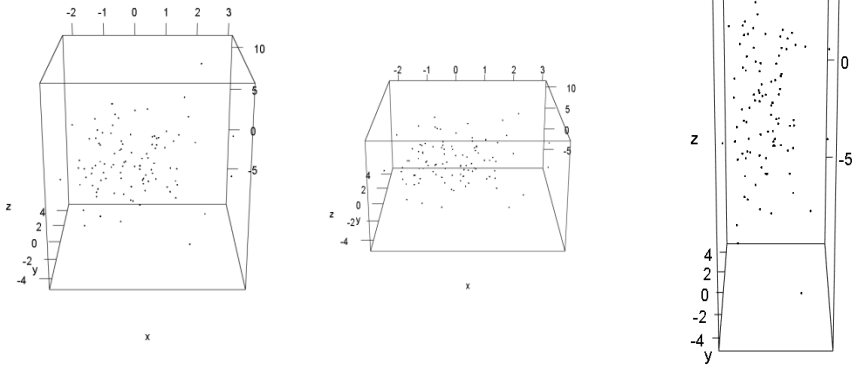
```
open3d()
shade3d( subdivision3d
    ( cube3d(), depth=3 ),
     color="red", alpha=0.5 )
open3d()
shade3d( subdivision3d
    ( cube3d(), depth=6 ),
     color="green", alpha=0.5 )
```



aspect3d

- aspect3d(x, y = NULL, z = NULL)
- Ustawienie proporcji poszczególnych osi
- Iso – równa skala dla każdej osi
- **Przykład**

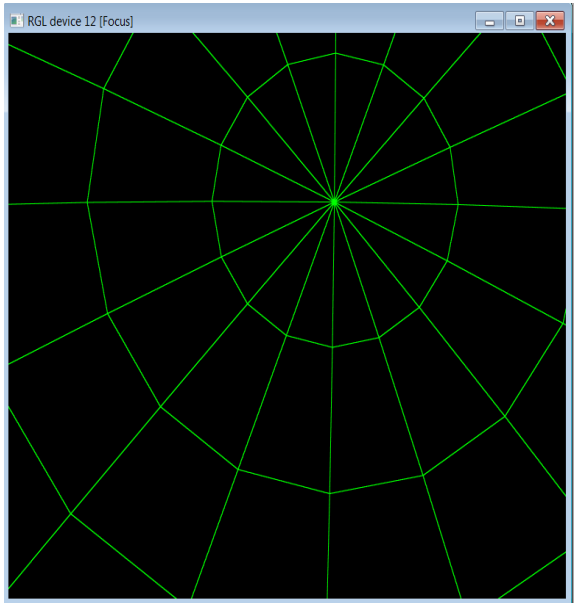
```
x <- rnorm(100)
y <- rnorm(100)*2
z <- rnorm(100)*3
open3d()
plot3d(x, y, z)
aspect3d(1,1,0.5)
open3d()
plot3d(x, y, z)
aspect3d("iso")
```



bg

- bg3d(...)
- Ustawienie tła na sceny
- Przykład (centrum przestrzeni 3D w środku sfery)

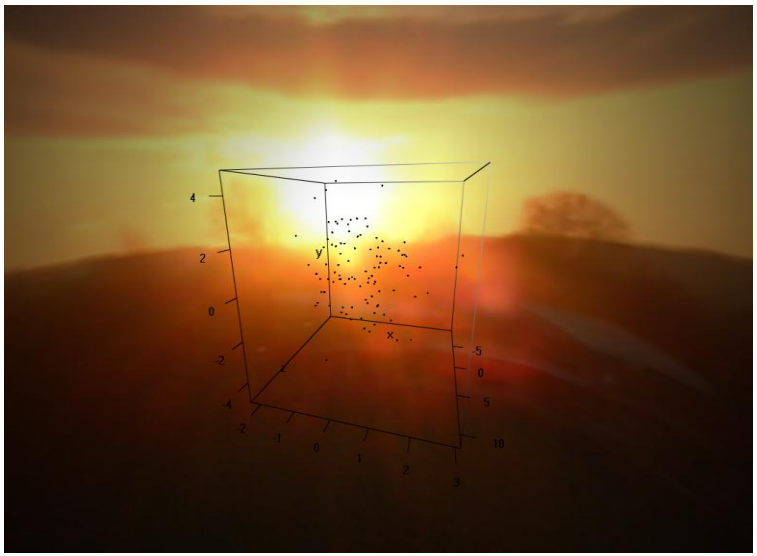
```
rgl.open()
rgl.bg(sphere=TRUE, color=c("black","green"), lit=FALSE,
    back="lines" )
```



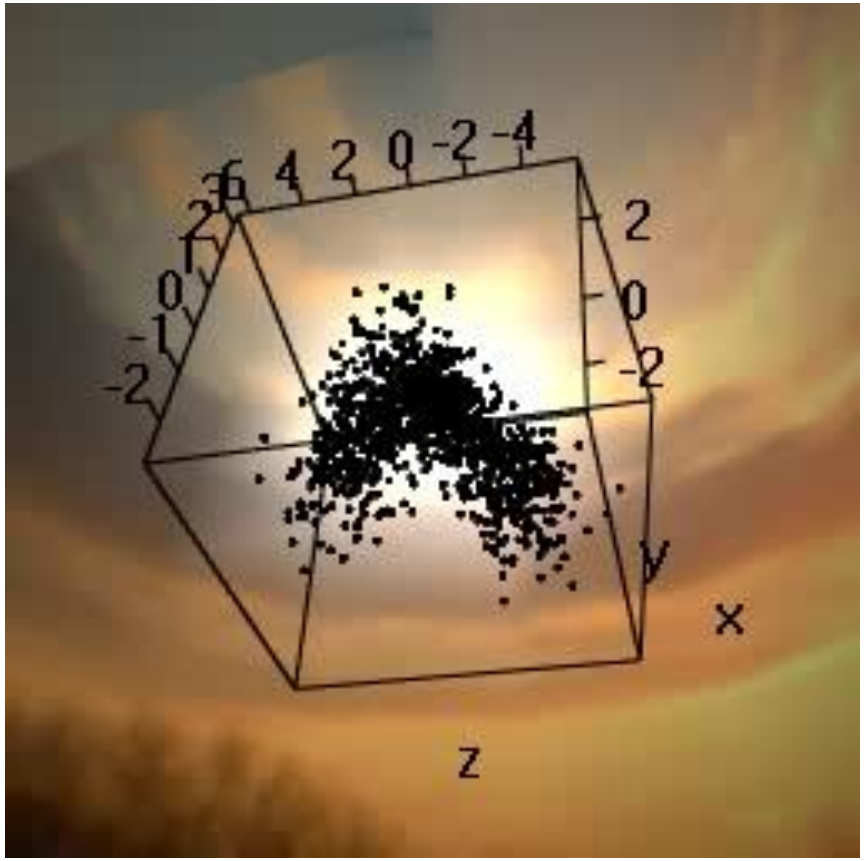
bg

- **Przykład**
- rgl.open()

```
rgl.bg(sphere=TRUE, texture=system.file("textures/sunsleep.png",
    package="rgl"), back="filled" )
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x,y)
plot3d(x, y, z)
for (i in 1:900) {
rgl.viewpoint(i,-45)
tlo<- paste
    ("tlo",formatC(i,
    digits=1,flag="0"),".png",sep="")
rgl.snapshot(tlo) }
```



bg



texts

- text3d(x, y = NULL, z = NULL, texts, adj = 0.5, justify, ...)
- Dodanie tekstu do sceny
- Tekst jest zorientowany w stronę kamery
- **Przykład**

```
open3d()
famnum <- rep(1:4, 8)
family <- c("serif", "sans", "mono", "symbol")[famnum]
font <- rep(rep(1:4, each=4), 2)
cex <- rep(1:2, each=16)
text3d(font, cex, famnum, text=paste(family, font),adj = 0.5,
color="blue", family=family, font=font, cex=cex)
```

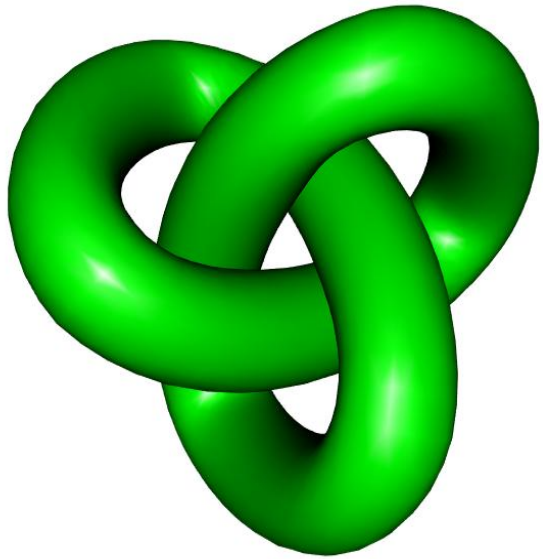
texts


serif 1 serif 2 serif 3 serif 4
sans 1 sans 2 sans 3 sans 4
serif 1 serif 2 serif 3 serif 4
mono 1 mono 2 mono 3 mono 4
sans 1 sans 2 sans 3 sans 4
mono 1 mono 2 mono 3 mono 4
symbol 1 symbol 2 symbol 3 symbol 4
symbol 1 symbol 2 symbol 3 symbol 4

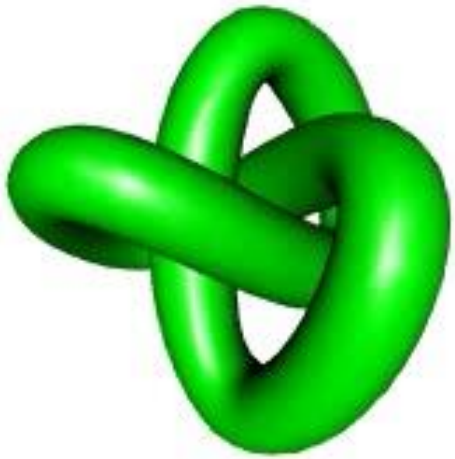
cylinder3d

- cylinder3d(center,…)
- Konwertuje opis przestrzeni w obiekcie mesh3D tworząc cylindryczną tubę wokół zadanej krzywej.
- **Przykład -** trefoil knot

- open3d()
- theta <- seq(0, 2*pi, len=25)
- knot <- cylinder3d(cbind(sin(theta) +2*sin(2*theta), 2*sin(3*theta), cos(theta)-2*cos(2*theta)),
- e1=cbind(cos(theta)+4*cos(2*theta), 6*cos(3*theta), sin(theta)+4*sin(2*theta)),
- radius=0.8, closed=TRUE)
- shade3d(addNormals(subdivision3d (knot, depth=2)), col="green")



cylinder3d



Dziękuję za uwagę!!!