

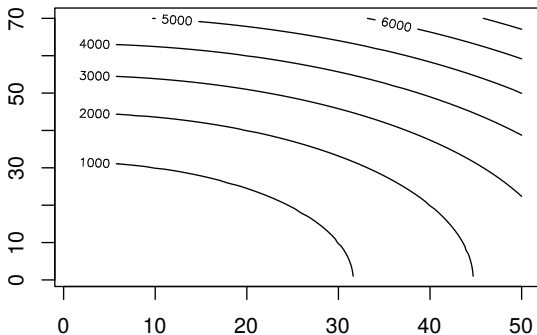
데이터 사이언티스트를 위한 R-4-1: 그래픽스

Jinseog Kim

2016-11-08

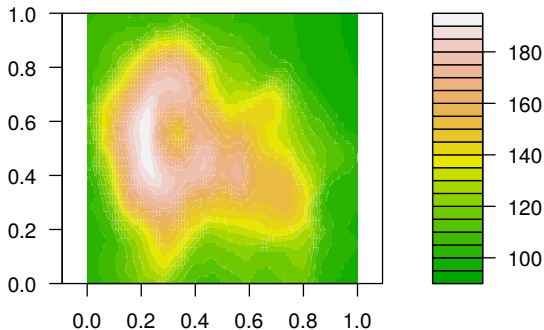
등고선 그림

```
x <- 1:50; y <- 1:70  
z <- matrix(expand.grid(x,y)$Var1^2 + expand.grid(x,y)$Var2^2,50,70)  
contour(x,y,z)
```



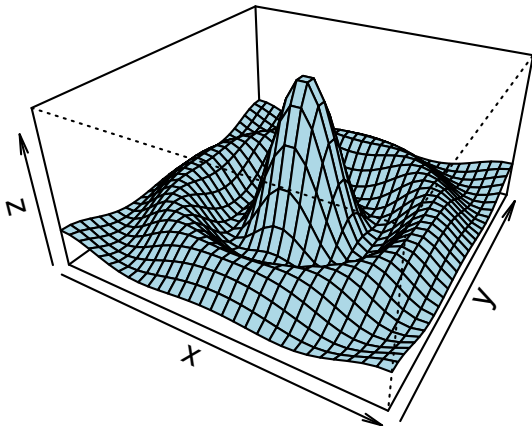
```
filled.contour(volcano, color.palette = terrain.colors, asp = 1)  
title(main = "volcano data: filled contour map")
```

volcano data: filled contour map



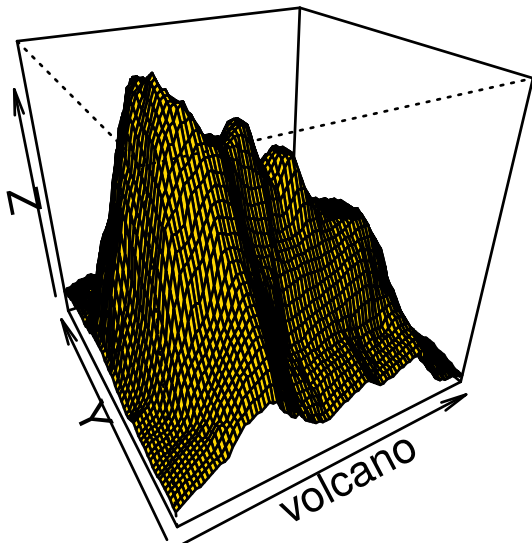
perspective plot

```
x <- seq(-10, 10, length= 30); y <- x  
f <- function(x, y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }  
z <- outer(x, y, f)  
z[is.na(z)] <- 1  
op <- par(bg = "white")  
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
```



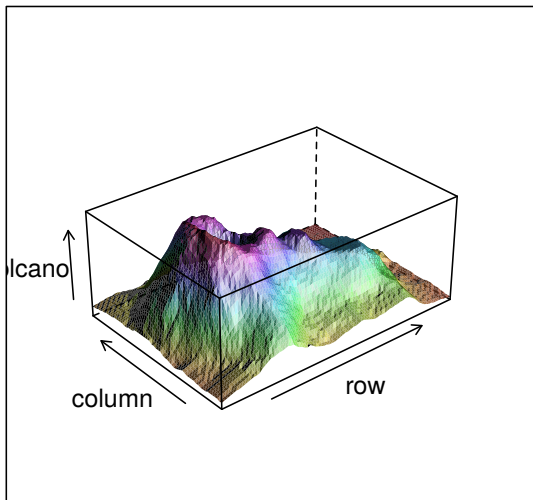
perspective plot

```
persp(volcano, col="gold",phi = 30, theta = -30)
```



wireframe

```
library(lattice)  
wireframe(volcano, shade = TRUE, aspect = c(61/87, 0.4),  
          light.source = c(10,0,10))
```



3D Visualization Using OpenGL

```
#install.packages("rgl")  
library(rgl)  
x <- 1:5/10  
y <- 1:5  
z <- x %o% y  
z <- z + .2*z*runif(25) - .1*z
```

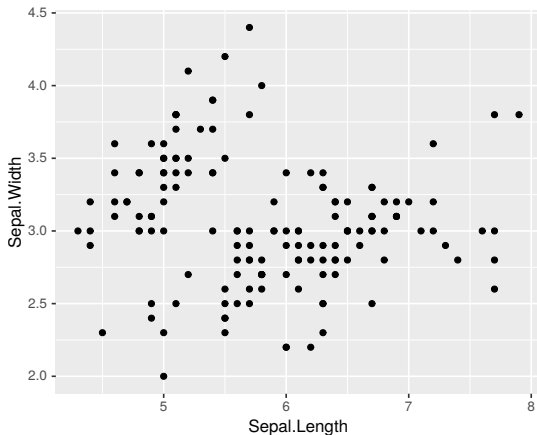
```
#install.packages("plot3d")  
persp3d(x, y, z, col="skyblue")  
open3d() #open new device  
x <- sort(rnorm(1000))  
y <- rnorm(1000)  
z <- rnorm(1000) + atan2(x, y)  
plot3d(x, y, z, col = rainbow(1000))
```

Using ggplot2 package

- 1 `qplot()` : basic plotting function
- 2 `ggplot()`: initializes a ggplot object
- 3 `aes()`: Generate aesthetic mappings
- 4 `geom_[chart]`
 - `geom_bar`
 - `geom_boxplot`
 - `geom_histogram`, `geom_density`
 - `geom_point`, `geom_line`
 - `geom_text`
- 5 '+' : adding objects to ggplot objects

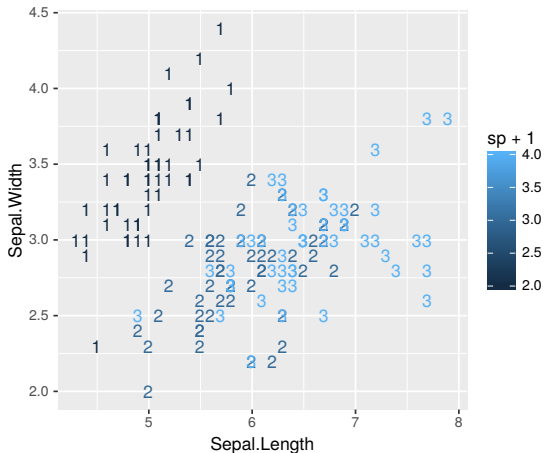
Using ggplot2 package

```
#install.packages("ggplot2")  
library(ggplot2)  
qplot(Sepal.Length, Sepal.Width, data=iris)
```



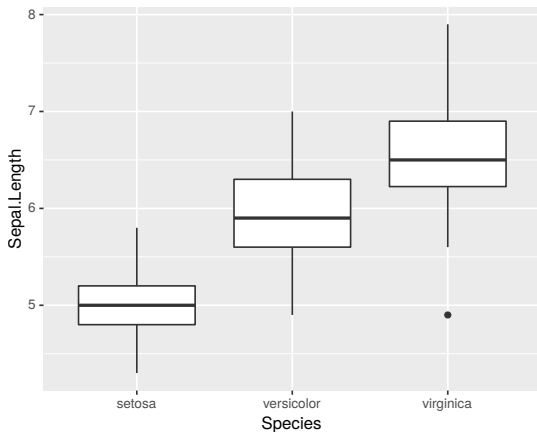
Using ggplot2 package

```
sp <- as.integer(iris$Species)
ggplot(iris, aes(Sepal.Length, Sepal.Width, label=sp, color=sp+1)) +
  geom_text()
```



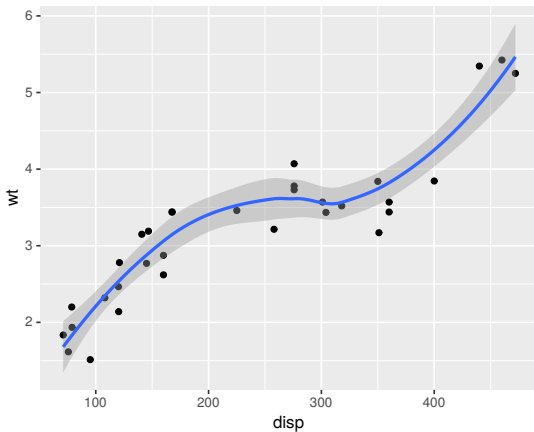
Using ggplot2 package

```
ggplot(iris, aes(y=Sepal.Length, x=Species)) +  
  geom_boxplot()
```



Using ggplot2 package

```
p <- ggplot(mtcars, aes(dis, wt))  
p <- p + geom_point()  
p <- p + geom_smooth()  
p
```



Using ggplot2 package

```
ggplot(diamonds, aes(cut, price)) +  
  geom_boxplot() +  
  coord_flip() #flip coordinates
```

