R Notes for Multivariate Analysis

Yongfu, Liao

2018-04-04

Contents

About

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.name/tinytex/.

Chapter 1

Multivariate Normal Distribution & Covariance Matrix

```
library(dplyr)
library(latex2exp)
library(ggplot2)
theme <- theme(axis.text.x = element_text(size = 7, face = "plain", angle = 30),
               axis.text.y = element_text(size = 7, face = "plain"),
               axis.title.x = element_text(size = 9, face = "bold"),
    axis.title.y = element_text(size = 9, face = "bold"))
```

1.1 Bivariate Normal Contour Map

1.1.1 `ellipse` function

```
ellipse(x, scale, centre, level, npoints = 1000)
```

- `x`: a single number, correlation of the two variables.
- `scale`: vector, **standard deviation** of the two variables.
- `centre`: vector, center of the ellipse, i.e. the mean vector of the bivariate normal distribution.
- `level`: a single number, the contour probability.
- `npoints`: number of points used to draw the contour.

`ellipse` returns a **matrix** with dimension (`npoints` $\times$ 2), which can be used to plot contour.

1.1.2 Data Generation

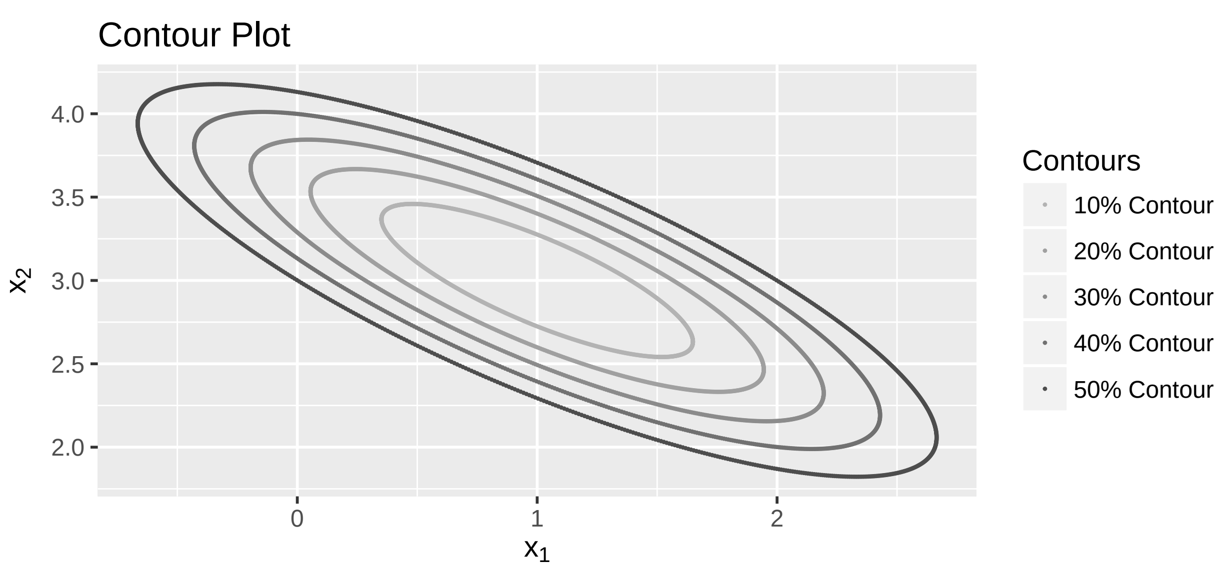
The `for` loop below is used to generate a data frame with 3 columns(variables): - Column 1: First variable of bivariate normal function (x_1) - Column 2: Second variable of bivariate normal function (x_2) - Column 3: The contour that x_1 & x_2 on the same row belongs to.

```
library(ellipse)
```

```
All_contours <- c(NA, NA, NA)
    ## Set empty start for appending ##

for (i in 1:5) {
    level <- 0.1*i
        ## Set Contour prob., prob. of obs within contour ##
    ell_data <-ellipse(-0.8, c(sqrt(2), 1), centre = c(1, 3), level = level, npoints = 800+(i-1)^3)
        ## npoints: bigger contours with more points ##
    class <- rep(paste(level*100, "% Contour", sep=""), nrow(ell_data))
        ## Assign contour class ##
    ell_data <- as.data.frame(ell_data)
        ## Change to data.frame BEFORE cbind, ##
        ## or coersion happens ##
    ell_data <- cbind(ell_data, class)

    All_contours <- rbind(All_contours, ell_data)
}

All_contours <- All_contours[-1,]
    ## Remove the empty start ##
```

1.1.3 Plotting

```
ggplot(data = All_contours) +
    geom_point(aes(x = x, y = y, color = class),
               size = 0.1) +
    scale_colour_grey(start = 0.7, end = 0.3) +
        ## Use gray scales instead of colored default ##
    labs(color = "Contours",
         title = "Contour Plot",
         x = TeX("$x_1$"), y = TeX("$x_2$")
    )
```



1.2 Multivariate Normal Functions

1.2.1 Generate density f(x)

```
library(mvtnorm)

mu <- c(1, 3) # mean vector
Sigma <- matrix(c(2, -0.8*sqrt(2), -0.8*sqrt(2), 1),
                nrow = 2) # covariance matrix

dmvnorm(x = c(2, 5), mean = mu, sigma = Sigma)
```

```
[1] 1.562995e-05
```

- `x`: Vector x in f(x), all variables of the multivariate normal distribution.
- `mean`: Mean vector(center of ellipse) of the multivariate normal distribution.
- `sigma`: Covariance matrix of the multivariate normal distribution.

`dmvnorm` returns f(x), the range of the multivariate normal function. For example, `dmvnorm(x = c(2, 5), mean = mu, sigma = Sigma)` returns the value $f(x_1 = 2, x_2 = 5)$ of the multivariate normal distribution specified by mean vector, `mu`, and covariance matrix, `Sigma`.

1.2.1.1 Example: Densities of a Contour

```
data <- All_contours %>%
    filter(class == "50% Contour")

dmvnorm(x = data[1, 1:2], mean = mu, sigma = Sigma)[[1]]
```

```
[1] 0.09378295
```

```
dmvnorm(x = data[4, 1:2], mean = mu, sigma = Sigma)[[1]]
```

```
[1] 0.09378295
```

The retured values are the same(very close), since they are on the same contour. See the section above for more details.

1.2.2 Covariance Matrix

Generater covariance and correlation Matricies:

```
library(mat2tex)
cov.mt <- cov(iris[,1:4]) ## Cov Matrix of variable 1~4
cor.mt <- cor(iris[,1:4]) ## Cor Matrix of variable 1~4
```

$$\text{Covariance matrix} = \begin{pmatrix} 0.69 & -0.04 & 1.27 & 0.52 \\ -0.04 & 0.19 & -0.33 & -0.12 \\ 1.27 & -0.33 & 3.12 & 1.30 \\ 0.52 & -0.12 & 1.30 & 0.58 \end{pmatrix}$$

$$\text{Correlation matrix} = \begin{pmatrix} 1.00 & -0.12 & 0.87 & 0.82 \\ -0.12 & 1.00 & -0.43 & -0.37 \\ 0.87 & -0.43 & 1.00 & 0.96 \\ 0.82 & -0.37 & 0.96 & 1.00 \end{pmatrix}$$

Chapter 2

Principle Component Analysis

2.1 Conversion Between Correlation & Covaraince Matrices

$$\mathbf{R} = diag(\mathbf{S})^{\frac{-1}{2}}\ \mathbf{S}\ diag(\mathbf{S})^{\frac{-1}{2}}$$