Week 4. Multivariate normal distribution, 3D visualization and animation

Section 4.1.

The $m \times 1$ random vector $\mathbf{X}$ has a multivariate normal distribution if

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Omega}),$$

where $\boldsymbol{\mu}$ is the $m \times 1$ mean vector with the ith component μ_i and $\boldsymbol{\Omega}$ is the $m \times m$ covariance matrix:

$$E(\mathbf{X}) = \boldsymbol{\mu}, \quad \mathrm{cov}(\mathbf{X}) = E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'] = \boldsymbol{\Omega}$$

meaning that

$$E[(X_i - \mu_i)(X_j - \mu_j)] = \Omega_{ij}.$$

The pdf

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Omega}) = (2\pi)^{-m/2} \left|\boldsymbol{\Omega}\right|^{-1/2} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Omega}^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \quad \mathbf{x} \in R^m.$$

For the univariate normal $X \sim \mathcal{N}(\mu, \sigma^2)$ we have $m = 1$ and $\Omega = \sigma^2$ with the pdf:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$

For the bivariate $\mathbf{X}^{2\times 1}$ we have $m = 2$ with the pdf

$$f(x_1, x_2; \boldsymbol{\mu}, \boldsymbol{\Omega}) = \frac{1}{(2\pi)\sigma_1\sigma_2\sqrt{1-\rho^2}} \mathrm{e}^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]}.$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

The normalized pdf for

$$Z_1 = \frac{X_1 - \mu_1}{\sigma_1}, \quad Z_2 = \frac{X_2 - \mu_2}{\sigma_2}$$

is

$$f(z_1, z_2) = \frac{1}{(2\pi)\sqrt{1-\rho^2}} \mathrm{e}^{-\frac{1}{2(1-\rho^2)}(z_1^2 - 2\rho z_1 z_2 + z_2^2)}$$

The level set $f(z_1, z_2) = const$ is an ellipse

$$\{(z_1, z_2) : z_1^2 - 2\rho z_1 z_2 + z_2^2 = const\}.$$

Estimation using the data

R implementation: If `X` is a $n \times m$ matrix of data (rows are observations and columns are variables) function `colMeans(X)` returns m means $\overline{\mathbf{X}}^{m\times 1} = \widehat{\boldsymbol{\mu}} = \overline{\mathbf{X}}$ and `var(X)` returns a $m \times m$ covariance matrix $\widehat{\boldsymbol{\Omega}}$.

Properties of the multivariate normal distribution

Normal distribution is invariant with respect to linear transformation: if $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Omega})$ then

$$\mathbf{Y} = \mathbf{b} + \mathbf{A}\mathbf{X} \sim \mathcal{N}(\mathbf{b} + \mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Omega}\mathbf{A}')$$

where $\mathbf{b}$ is a fixed $k \times 1$ vector and $\mathbf{A}$ is a fixed $k \times m$ matrix.

Letting $k = 1$ we get the following corollary:

$$var(b + \mathbf{a}'\mathbf{X}) = \mathbf{a}'\boldsymbol{\Omega}\mathbf{a}$$

which implies

$$b + \mathbf{a}'\mathbf{X} \sim \mathcal{N}(b + \mathbf{a}'\boldsymbol{\mu}, \mathbf{a}'\boldsymbol{\Omega}\mathbf{a}).$$

A matrix $\boldsymbol{\Omega}$ may be a covariance matrix if and only if it's symmetric and nonnegative definite:

$$\boldsymbol{\Omega}' = \boldsymbol{\Omega}, \quad \mathbf{a}'\boldsymbol{\Omega}\mathbf{a} \geq \mathbf{0}$$

for every $\mathbf{a}$. The key: $var(\mathbf{a}'\mathbf{X}) = \mathbf{a}'\boldsymbol{\Omega}\mathbf{a} \geq 0$.

Correlation matrix in matrix notation is defined as

$$\mathbf{R} = \mathbf{D}^{-1/2}\boldsymbol{\Omega}\mathbf{D}^{-1/2}$$

$$\begin{bmatrix} 1 & \rho_{12} & \rho_{12} & \rho_{14} \\ & 1 & \rho_{23} & \rho_{24} \\ & & 1 & \rho_{34} \\ & & & 1 \end{bmatrix}$$

Multivariate conditional distribution and regression

If Y and $\mathbf{X}$ have multivariate normal distribution we are looking for a conditional distribution as the distribution of Y when $\mathbf{X} = \mathbf{x}$. It can be proven that $Y|\mathbf{X} = \mathbf{x}$ has a normal distribution with the conditional mean and variance given by

$$E(Y|\mathbf{X} = \mathbf{x}) = \mu_y + \boldsymbol{\omega}'_{yx}\boldsymbol{\Omega}_x^{-1}(\mathbf{x} - \boldsymbol{\mu}_x), \quad \mathrm{var}(Y|\mathbf{X} = \mathbf{x}) = \sigma_y^2(1 - \rho_{yx}^2),$$

where

$$\rho_{yx}^2 = \sigma_y^{-2}\boldsymbol{\omega}'_{yx}\boldsymbol{\Omega}_x^{-1}\boldsymbol{\omega}_{yx}$$

is the multiple coefficient of determination that tells the proportion of variance of Y explained by $\mathbf{X}$.

The difference between multivariate normal conditional regression and linear model: in the former, $\mathbf{X}$ is normally distributed bust in the lattter it's fixed. Formally,

$$\begin{aligned} \widehat{\boldsymbol{\omega}}_{yx} &= \frac{1}{n}(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}')'(\mathbf{y} - \overline{y}\mathbf{1}), \\ \widehat{\boldsymbol{\Omega}}_x &= \frac{1}{n}(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}')'(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}'). \end{aligned}$$

the covariance vector and covariance matrix with the OLS slope coefficients

$$[(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}')'(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}')]^{-1}(\mathbf{X} - \mathbf{1}\overline{\mathbf{x}}')'(\mathbf{y} - \overline{y}\mathbf{1}).$$

Popular correlation structures

Two important correlation structures/matrices:

1. *Compound symmetry*

$$\mathbf{R} = \begin{bmatrix} 1 & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho \\ \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & 1 \end{bmatrix}$$

is the correlation matrix to model cluster correlation: $cor(X_i, X_j) = \rho$. This happens when $\mathbf{X} = \mathbf{Z}+aU\mathbf{1}$ where $\mathbf{Z}$ and U are uncorrelated and $cov(\mathbf{Z}) = \sigma^2\mathbf{I}$, $var(U) = 1$. We have

$$cov(\mathbf{X}) = cov(\mathbf{Z})+cov(aU\mathbf{1}) = \sigma^2\mathbf{I}+var(aU)\mathbf{11}' = \sigma^2\mathbf{I}+a^2\mathbf{11}'.$$

All the off-diagonal elements are the same and therefore all the off-dagonal elements of matrix $\mathbf{R}$ are the same.

Kriging technique as an application of multivariate normal distribution

Prediction with spatial data

$$E(Y|\mathbf{X} = \mathbf{x}) = \mu_y + \boldsymbol{\omega}'_{yx}\boldsymbol{\Omega}_x^{-1}(\mathbf{x} - \boldsymbol{\mu}_x).$$

Kriging

From Wikipedia, the free encyclopedia

In statistics, originally in geostatistics, **kriging** or **Kriging**, also known as **Gaussian process regression**, is a method of interpolation based on Gaussian process governed by prior covariances. Under suitable assumptions of the prior, kriging gives the best linear unbiased prediction (BLUP) at unsampled locations.[1] Interpolating methods based on other criteria such as smoothness (e.g., smoothing spline) may not yield the BLUP. The method is widely used in the domain of spatial analysis and computer experiments. The technique is also known as **Wiener–Kolmogorov prediction**, after Norbert Wiener and Andrey Kolmogorov.

Example 1 ***Compound symmetry for COVID prediction.*** *The previous rate of COVID-19 cases in one hundred towns of the state was* 0.1% *(one per hundred) with SD* = 0.02%*. The new rate, one month after, jumps up to* 0.3%*. Under assumption that infection rates follow a multivariate normal distribution with compound symmetry and* $\rho = 0.5$ *predict the rate and its SD in town 100, where the new testing has not been done yet.*

Solution. We assume that the rates are random and follow a multivariate normal distribution. The expected rate is computed by formula

$$E(Y|\mathbf{X} = \mathbf{x}) = \mu_y + \boldsymbol{\omega}'_{yx}\boldsymbol{\Omega}_x^{-1}(\mathbf{x} - \boldsymbol{\mu}_x)$$

where Y is the rate in town 100, $\mathbf{X}^{99\times 1}$ are rates in other 99 towns, $\mu_y = 0.1$ in town 100 with $\sigma_y = 0.02$, $\boldsymbol{\Omega}_x$ is the 99×99 covariance matrix of rates, $\boldsymbol{\omega}_{yx}$ is the 99×1 vector of covariances betwen Y and $\mathbf{X}$ matrix, $\mathbf{x} = 0.3 \times \mathbf{1}$ and $\boldsymbol{\mu}_x = 0.1 \times \mathbf{1}$.

$$\begin{aligned} \mu_y &= 0.1 \\ \boldsymbol{\omega}_{yx} &= \mathbf{1}^{99\times 1} \times 0.02^2 \times 0.5 \\ (\boldsymbol{\Omega}_x)_{ij} &= \left[\begin{array}{c} 0.02^2 \text{ if } i = j \\ 0.02^2 \times 0.5 \text{ if } i \neq j \end{array} \right. \\ \boldsymbol{\mu}_x &= 0.1 \times \mathbf{1} \\ \mathbf{x} &= 0.3 \times \mathbf{1} \end{aligned}$$

Compute the variance of prediction

$$\mathrm{var}(Y|\mathbf{X}=\mathbf{x}) = \sigma_y^2(1-\rho_{yx}^2), \text{ where } \rho_{yx}^2 = \sigma_y^{-2}\boldsymbol{\omega}_{yx}'\boldsymbol{\Omega}_x^{-1}\boldsymbol{\omega}_{yx}.$$

See the R function `excovid`.

Use formula

$$(\mathbf{I}+\mathbf{v}\mathbf{v}')^{-1} = \mathbf{I}-\frac{1}{1+\|\mathbf{v}\|^2}\mathbf{v}\mathbf{v}'.$$

to avoid large matrix inverse.

2. *Time series analysis: autoregression* of the first order covariance matrix such as $(n=4)$:

$$\mathbf{R} = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}, \quad |\rho| < 1,$$

or in general case $R_{ij} = \rho^{|i-j|}$. This correlation matrix emerges in autoregression of the first order:

$$Y_{t+1} = \mu + \rho(Y_t - \mu) + \varepsilon_t, \quad var(\varepsilon_t) = \sigma^2.$$

Sections 4.1 and 6.6.2.

Example: Prediction of stock prices using autoregression

Example 8.39.

Historical stock prices can be downloaded from `finance.yahoo.com`. File `AMZN_ weekly.csv` contains weekly stock prices from the week of 7/27/2015 to the week of 7/29/2019 (211 data points). Use these data to estimate the **autoregression** with the specified `maxlag` order and predict Amazon.com prices one week ahead.

The autoregression of the mth order with the time series being $Y_1, Y_2, ..., Y_n$ is the regression of Y_t on its past values $Y_{t-1}, Y_{t-2}, ..., Y_{t-m}$. Specifically, the vector of the dependent variable and the matrix of m predictors, following the format of linear model, is as follows:

$$\mathbf{y}^{(n-m)\times 1} = \begin{bmatrix} Y_{m+1} \\ Y_{m+2} \\ \vdots \\ Y_n \end{bmatrix}, \quad \mathbf{X}^{(n-m)\times(m+1)} = \begin{bmatrix} 1 & Y_m & Y_{m-1} & \cdots & Y_1 \\ 1 & Y_{m+1} & Y_m & & Y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & Y_{n-1} & Y_{n-2} & \cdots & Y_{n-m} \end{bmatrix}.$$

The second column of matrix $\mathbf{X}$ is the time series shifted by 1 (lag=1), the third column is shifted by 2 (lag=2), etc. Thus the autoregression takes the form of a linear model

$$Y_t = \beta_1 + \beta_2 Y_{t-1} + \beta_3 Y_{t-2} + ... + \beta_{m+1} Y_{t-m} + \varepsilon_t, t = m+1, m+2, ..., n.$$

The `R` code for estimation of the autoregression by `lm` is found in the `amzn.r` file and the output for $m=8$ (`maxlag=8)` is shown below.

Visualization of 3D data

Graphics in R is superior compared to other computer languages such as Python and Matlab.

Crash source in R graphics

```
par(mfrow=c(1,2))
par(mfrow=c(2,1),mar=c(3,5,3,1))
par(mfrow=c(1,1),mar=c(4.5,4.5,3,1),cex.lab=1.5,cex.main=1.5)
plot(x,y)
plot(x,y,type="l",pch=16,col=2)
```
Savings graphs: (a) save as, (b) using R commands such as jpeg or pdf
```
#define your own axes/override defaults
plot(x,y,axes=F)
axis(side=1,seq(from=x.min,to=x.max,by=.1))
axis(side=2,seq(from=y.min,to=y.max,by=.25))

#plotting several lines/points on y-axis with the same x-axis
matplot(x,cbind(y1,y2,y3),type="l",col=1:3,lwd=c(3,1,2),lty=1:3)

#Adding lines/segements/points
line(x,y,lwd=3,lty=2,col=2)
points(x,y,pch=16,col=3,cex=1.5)
segments(x1,y1,x2,y2,col=2,lwd=3,lty=3)

#Graph annotations
text(.34,1,paste("R2 = ",round(r2,3)),cex=1.5,font=2)
text(.34,1,paste("R2 ",r2),cex=1.5,adj=0)
legend("bottomleft",c("Stock1","Stock2"),lty=c(2,1),lwd=c(4,2),col=c(2,1),cex=1.5,bg="gray94")
```

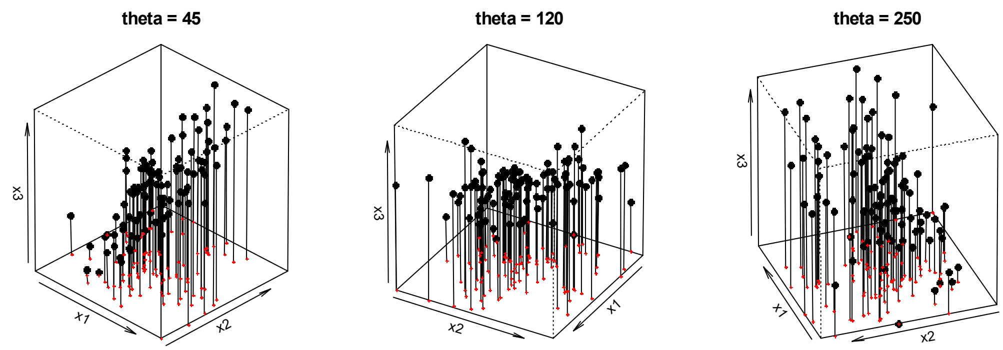
See program `rplot`

My first statistical movie

Dynamic plot of the cdf via moving threshold - the R function `cdfdyn1`.

3D scatterplots and animation

Animation of cdf: code `cdf.dyn1.r`
R program `mn3`



One hundred normally distributed 3D points viewed at different `theta` *angle. The depth of the points is achieved through projection of the points on the (x, y) plane. See the R function* `mn3`*.*

Example 2 ***Three-dimensional (3D) graphics in*** R. *(a) Generate 100 3D multivariate normal points with*

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{\Omega} = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

using the matrix square root. (b) Write an R program with arguments `nPoints, mu`*, and* `Omega` *and plot the points using the* `persp` *function. (c) Use animation by viewing the 3D plot at different angles.*

Solution. (a) If $\mathbf{Z} \sim\mathcal{N}(\mathbf{0}, \mathbf{I})$ then

$$\mathbf{X} = \boldsymbol{\mu} + \boldsymbol{\Omega}^{1/2}\mathbf{Z}$$

then

$$\mathbf{X} \sim\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Omega}).$$

Indeed

$$\begin{aligned} E(\mathbf{X}) &= \boldsymbol{\mu} + \boldsymbol{\Omega}^{1/2}E(\mathbf{Z}) = \boldsymbol{\mu}, \\ cov(\mathbf{X}) &= cov(\boldsymbol{\Omega}^{1/2}\mathbf{Z}) = \boldsymbol{\Omega}^{1/2}cov(\mathbf{Z})\boldsymbol{\Omega}^{1/2\prime} = \boldsymbol{\Omega}^{1/2}\mathbf{I}\boldsymbol{\Omega}^{1/2} = \boldsymbol{\Omega}. \end{aligned}$$

See the R code `mn3`.
When $m = 2$, bivariate normal distribution, we can create

$$\begin{bmatrix} Y \\ X \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_y \\ \mu_x \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_x^2 \end{bmatrix}\right)$$

in two steps:

1. Generate $X_i \sim \mathcal{N}(\mu_x, \sigma_x^2)$ as `rnorm(n,mean=mux,sd=sdx)`
2. Generate $Y_i \sim \mathcal{N}(\mu_y + \rho\sigma_y/\sigma_x(x - \mu_x), \sigma_y^2(1 - \rho^2))$, in R

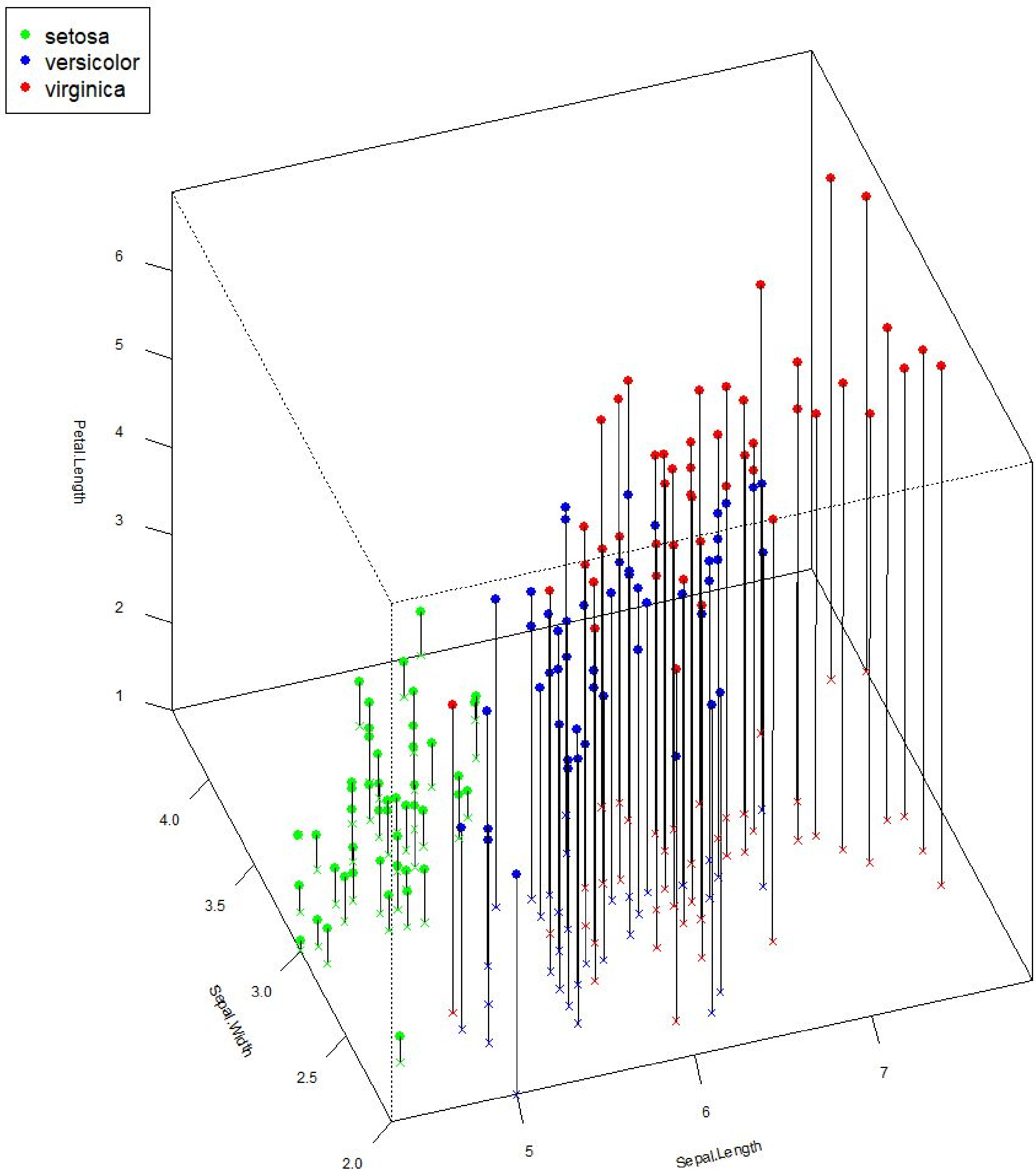
$$\mathtt{rnorm(n, mean = muy + ro * sdy/sdx * (X - mux), sd = sdy * sqrt(1 - ro\hat{}2)))}$$

3D animation in R

You have to dowaload and install Image magic third party software at

$$\mathtt{https://imagemagick.org/script/download.php}$$

Animation if R - see `iris3D` code. Fisher iris data. See `Iris.pptx`


setosa
versicolor
virginica
Petal.Length
1
2
3
4
5
6
Sepal.Width
2.0
2.5
3.0
3.5
4.0
Sepal.Length
5
6
7

Homework 4

1. (10 points). The bivariate normal distribution is given by $\mu_y = -1, \sigma_y = 0.8, \mu_x = 2, \sigma_x = 1.5$ and $\rho = -0.6$. (a) Use contour command to plot contours of the pdf. (b) Add the regression line as the conditional mean of Y on X along with $\pm\sigma_{y|x}$ line. (c) Generate 100 pairs from this distribution by generating marginal X and then normally distributed conditional Y. (d) Display the arrow with the maximum eigenvector at the center of the distribution. (e) Add contours of the estimated $\mathbf{\Omega}$ computed by `var` with different color (use `contour` with option `add=T`).

2. (10 points). The average covid rates at six towns are $(2.1, 1.7, 1.0, 1.8, 1.5, 1.2)$ with SD $= 0.1$. The GPS town locations are $(70.1, 34.3)$, $(70.4, 35.2)$, $(69.3, 36.2)$, $(72.5, 35.8)$, $(71.2, 33.8)$, $(68.7, 34.5)$. A covid outbreak is detected in town #2: the rate raised to 4.1. Assuming that the spatial correlation between towns i and j is modeled as $e^{-0.2d_{ij}}$ where d_{ij} is the distance, what is the expected 95% confidence interval for the rate of covid in town 6? Make other necessary plausible assumptions if needed.

3. (10 points). Generate 200 trivariate normally distributed random vectors with the mean vector 2, 2, 4 and covariance matrix
$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & 1 \\ -1 & 1 & 4 \end{bmatrix}$$
and 300 trivariate normally distributed random vectors with the mean vector $1, -2, 1$ and covariance matrix
$$\begin{bmatrix} 4 & 1 & .5 \\ 1 & 1 & -.1 \\ .1 & -.1 & 2 \end{bmatrix}.$$
Create animation with the theta angle running from 1 to 360°. Use different colors to show the two groups. Submit as a *.pptx file. Consult `iris3D` code.