ORIGIN ≡ 1

Multivariate Normal Distribution and Confidence Ellipses

Multivariate statistics is largely built upon a straight-forward extension of the Normal Distribution seen in Introductory *Biostatistics*. The classic formula for the Normal Distribution looks like this:

$$f(x) = \frac{1}{\sqrt{2\cdot\pi\cdot\sigma^2}}\cdot e^{\frac{-\left(\frac{x-\mu}{\sigma^2}\right)}{2}}$$

where f(x) refers to the probability density function (as accessed by dnorm() in R), μ is the parameter for population mean, and σ^2 is the population variance. In this equation, the multiplied term: $\frac{1}{\sqrt{2\cdot\pi\cdot\sigma^2}}$ may be viewed as a scaling factor, and the term in the exponent of e: $\left(\frac{x-\mu}{\sigma}\right)^2$ doing much of the work in shaping the the Normal Distribution's familiar "bell curve". The latter term can be interpreted as a description of squared of distance $(x - \mu)$ between some value of x who's probability is being assessed (along the x axis), and μ the center of the probability density distribution, "standardized" by the distribution's known variance σ^2. It should be noted that $f(x)$ only depends on this single scalar range variable x, and as such, is "one-dimensional".

The Multivariate Normal Distribution now extends this idea of a probability density function into a number p of multiple directions x_1, x_2, ... x_n. To do so, the single mean of the distribution μ becomes a vector μ specifying means for each of x_1, x_2, ... x_n, and interpreted as a point in p-dimensional space. In addition, σ^2 is replaced by covariance matrix Σ. This square matrix, by definition, contains variance for each of the x_1, x_2, ... x_n variables along the main diagonal along with covariances between each pair of x variables interpreted as indicating degree of "colinearity" of these variables in p-dimensional space. As a result of these modifications, the probability density function p(x), still a scalar value, now becomes:

$$f(x) = \frac{1}{(2\cdot\pi)^{\frac{p}{2}}\cdot(|\Sigma|)^{\frac{1}{2}}}\cdot e^{\frac{-(x-\mu)\cdot\Sigma^{-1}\cdot(x-\mu)}{2}}$$

In this formulation, the exponent part $(x-\mu)\cdot\Sigma^{-1}\cdot(x-\mu)$ is squared Mahalanobis distance, the extension of the concept of squared standardized distance $\left(\frac{x-\mu}{\sigma}\right)^2$ seen in the univariate case. As described in MTB 040, the matrix Σ^{-1} can be viewed as "correcting" the multivariate p-dimensional space for all covariances between the x variables, yielding a space of statistically equivalent distances.

In interpreting the multivariate density function, it is important to remember that f(x) is still a scalar value of probability that may be assigned to a position in p-dimensional space indicated by column vector x = $(x_1, x_2, \ldots x_n)^T$. If p = 2 dimensions, the familiar "bell curve" is now a peak in the z direction along a plane defined by x_1 and x_2, and any slice in the z direction along a diameter of the peak yields the "bell curve". For p >2 the situation becomes harder to visualize, but the geometric concepts and mathematics remain identical.

As in univariate statistics, the Multivariate Normal Distribution, designated $N_p(\mu,\Sigma)$, has wonderfully useful propterties, and is often invoked as an assumption in multivariate statistical tests. However, assessing the validity of this assumption in real data, is typically much more difficult. For one thing, a collection of data of useful size, typically yields a "sparce" distribution of points in p-dimensional space that is typically hard to visualize. Moreover, covariance (colinearity) between variables in these p dimensions greatly complicates interpretation of this non-orthogonal space itself. However, in defense of mutivariate statistics, the wealth of information embedded especially in the off diagonal elements in Σ allows for greatly expanded field of inquiry with typically greater precision.

For extended discussion of properties assignable to the Multivariate Normal Distribution, see RA Johnson & DW Wichern (JW) *Applied Multivariate Statistical Analysis 4th Edition* 1998, and especially AC. Rencher (AR) *Methods of Multivariate Analysis* 1995. Following both, useful propterties may be summarized as follows:

Linear combinations of variables with Multivariate Normal Distribution are Normally distributed.

All subsets of Multivariate Normally distributed variables are (Multivariate) Normally distributed.

However, the reverse is not necessarily true: individual or sets of variables may be (Multivariate) Normally distributed, but this doesn't ensure that the whole ensemble of variables is Multivariate Normally distributed.

Variables x_1, x_2, ... x_p are independent if their covariances in Σ are 0.

Conditional distributions of Multivariate Normally distributed variables are (Multivariate) Normally distributed.

If data are Multivariate Normally distributed then:

Sample mean vector X_{bar} is a sufficient statistic for population mean μ and is distributed $N_p(\mu,(1/n)\Sigma)$.

Sample covariance matrix S is a sufficient statistic for population covariance Σ with (n-1)·S is a distributed according to the Wishart random matrix distribution with n-1 degrees of freedom.

X_{bar} and S are independent.

Squared Mahalanobis distance $D_M = (x-\mu)\cdot\Sigma^{-1}\cdot(x-\mu)$ is distributed according to the chi-square (χ^2) distribution with p degrees of freedom.

Even if the data is NOT Multivariate Normally distributed, an extension of the Central Limit Theorem applies: With large enough sample size, sample statistics X_{bar}, S and D_M still work as reasonable approximations.

What follows here is an examination of simulated bivariate data (p=2) to get an sense of what the Multivariate Normal Distribution looks like in reality, and the use of confidence ellipses based on the χ^2 statistical distribution for D_M in characterizing the Multivariate Normal Distribution.

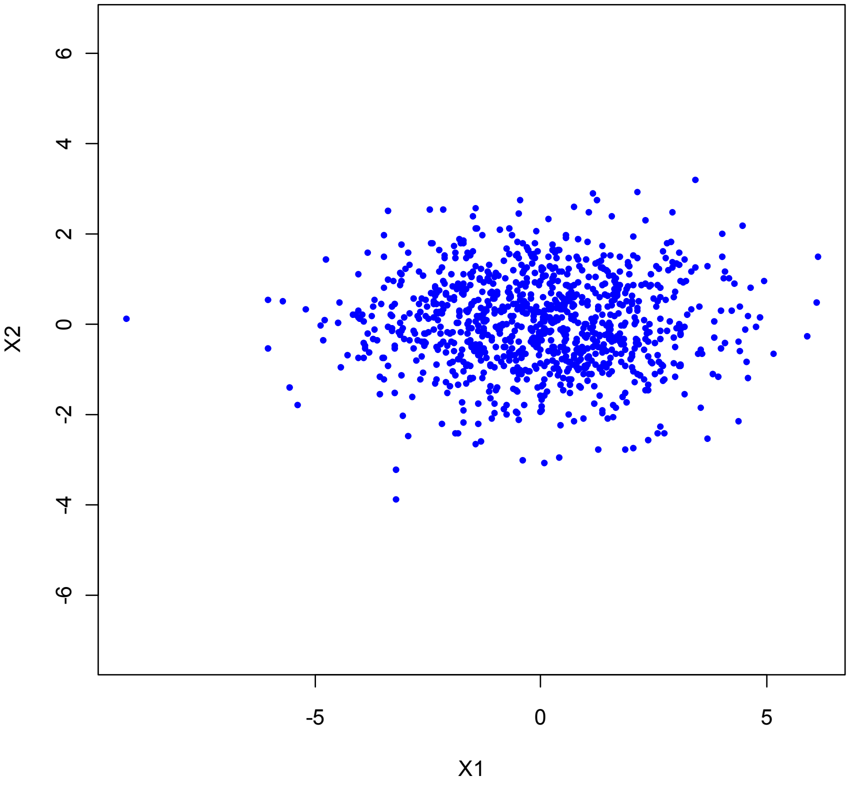
Generating Simulated Data in R:

```
#MULTIVARIATE NORMAL DISTRIBUTION AND CONFIDENCE ELLIPSE:
# CREATE DATA USING RANDOM NUMBER GENERATOR:
X1=rnorm(1000,0,2)
X2=rnorm(1000,0,1)
X=cbind(X1,X2)
#WRITE DATASET:
write.table(X,file="c:/DATA/Multivariate/randombivariate.txt")
```

< 1000 data points simulated for each of two variables X1 & X2

Plotting Data:

```
plot(X,asp=1,cex=0.5,pch=19,col='blue')
```



Reading Data:

Data is written to disk and read by MathCAd here.

M := READPRN("c:/DATA/Multivariate/randombivariate.txt")

$n := \mathrm{rows}(M) = (1000)$

$p := \mathrm{cols}(M) = (2)$

$i := 1 .. n$ $\quad j := 1 .. p$ **< index variables**

$l_{vec_i} := 1$

$I := \mathrm{identity}(n)$

Mean Vector:

$M_{bar} := \frac{1}{n} \cdot M^T \cdot l_{vec}$ $\quad M_{bar} = \begin{pmatrix} -0.0327573 \\ 0.0268316 \end{pmatrix}$ **< close to (0,0)**

Covariance Matrix:

Standard deviation:

$S := \frac{1}{n-1} \cdot M^T \cdot \left(I - \frac{1}{n} \cdot l_{vec} \cdot l_{vec}^{\;T} \right) \cdot M$ $\quad S = \begin{pmatrix} 4.0040869 & 0.0113922 \\ 0.0113922 & 1.0830404 \end{pmatrix}$ $\quad sd := \begin{pmatrix} \sqrt{S_{1,1}} \\ \sqrt{S_{2,2}} \end{pmatrix}$ $\quad sd = \begin{pmatrix} 2.0010215 \\ 1.0406923 \end{pmatrix}$

^ covariance between X1 & X2 is essentially 0. Variances for X1 & X2 occur along the main diagonal

^ close to (2,1) as originally specified

Confidence ellipses based on (JW eq. 4-8):

```
> summary(M)
       X1                 X2          
 Min.   :-9.16418   Min.   :-3.87029  
 1st Qu.:-1.40422   1st Qu.:-0.65055  
 Median : 0.03787   Median : 0.02150  
 Mean   :-0.03276   Mean   : 0.02683  
 3rd Qu.: 1.35983   3rd Qu.: 0.71904  
 Max.   : 6.11437   Max.   : 3.19310  
> sd(M)
      X1       X2 
2.001021 1.040692 
```

Making a set of ellipse points:

$\alpha := 0.05$ $\quad df := 2$ **< Set α as desired**

$c := \sqrt{qchisq(1-\alpha, df)}$ $\quad c = 2.448$ **< radius of circle**

Constructing the points:

$j := 1..100$ $\quad \theta_j := \frac{j}{100} \cdot 2 \cdot \pi$

$x1_j := c \cdot \cos(\theta_j)$ $\quad x2_j := c \cdot \sin(\theta_j)$

$x := \text{augment}(x1, x2)$ **< points on a circle of radius c**

Constructing Confidence Ellipse:

$\lambda := \text{eigenvals}(S)$ $\quad \varepsilon := \text{eigenvecs}(S)$ **< Eigenvalues and Eigenvectors of S**

$\Lambda_{sqrt} := \text{diag}(\sqrt{\lambda})$

$S_{sqrt} := \varepsilon \cdot \Lambda_{sqrt} \cdot \varepsilon^{T}$ **< Square root matrix of S**

Major axis of ellipse M:

$p_{x_1} := c \cdot \sqrt{\lambda_1} \cdot \varepsilon^{\langle 1 \rangle}$ $\quad p_{x_1} = \begin{pmatrix} 4.897984 \\ 0.019102 \end{pmatrix}$

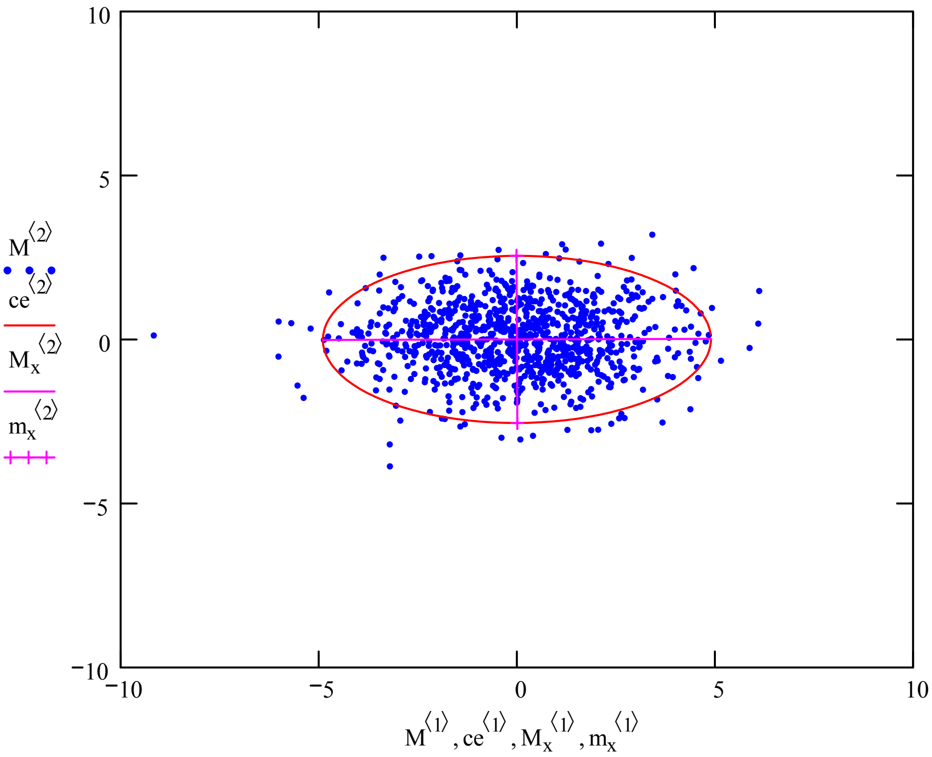
$M_x := \text{augment}(p_{x_1}, -p_{x_1})^{T}$

Minor axis of ellipse m:

$p_{x_2} := c \cdot \sqrt{\lambda_2} \cdot \varepsilon^{\langle 2 \rangle}$ $\quad p_{x_2} = \begin{pmatrix} -0.009934 \\ 2.54728 \end{pmatrix}$

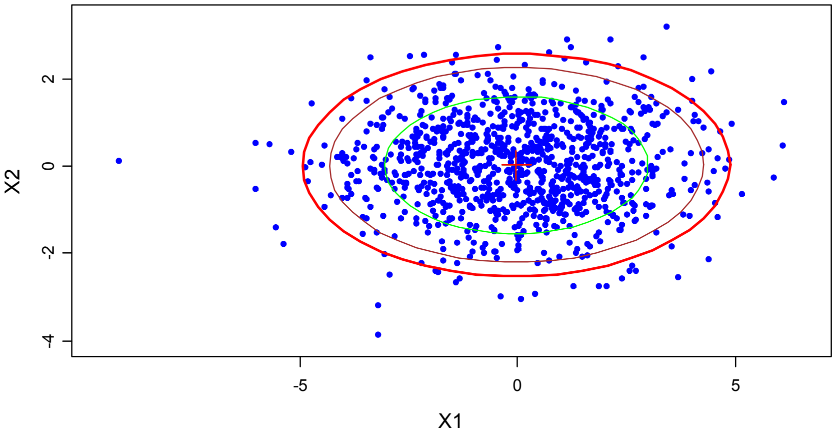
$m_x := \text{augment}(p_{x_2}, -p_{x_2})^{T}$

$ce := \left(S_{sqrt} \cdot (x)^{T}\right)^{T}$ **< points on the confidence ellipse based on S**



Plotting in R:

```
#CONFIDENCE ELLIPSE:
library(latticeExtra)
attach(M)

xyplot(X2 ~ X1,
   scales = "free",aspect='iso',
   par.settings = list(plot.symbol = list(col='blue',cex = 0.5, pch=19)),
   panel = function(x, y, ...) {
     panel.xyplot(x, y, ...)
     panel.ellipse(x,y,lwd = 1,level=0.68,col='green', ...
     panel.ellipse(x,y,lwd = 1,level=0.90,col='brown', ...)
     panel.ellipse(x,y,lwd = 2,level=0.95,col='red', ...)
   },
   auto.key = list(x = .1, y = .8, corner = c(0, 0)))
```



^ Although method for calculating Confidence Ellipses is not specified here, they appear to match explicit calculation in MathCad above.

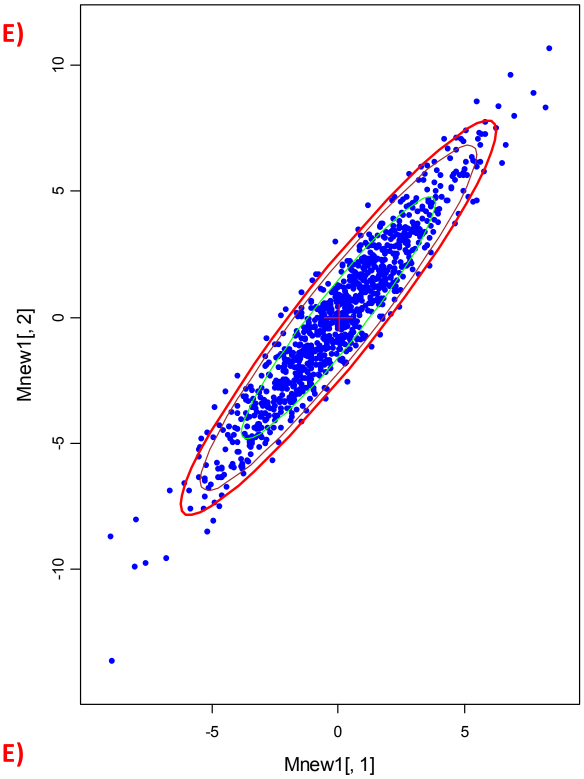
Applying correlation structure to the data:

```
#APPLYING CORRELATION BETWEEN VARIABLES IN
MATRIX M
#USING LINEAR TRANSFORMATION:
A=matrix(c(1,1.5,1.5,1),nrow=2,ncol=2,byrow=TRUE)
A

Mnew1=t(A%*%as.matrix(t(M)))


cor(Mnew1)
```

```
> A
     [,1] [,2]
[1,]  1.0  1.5
[2,]  1.5  1.0
```

```
> cor(Mnew1)
          [,1]      [,2]
[1,] 1.0000000 0.9469233
[2,] 0.9469233 1.0000000
```

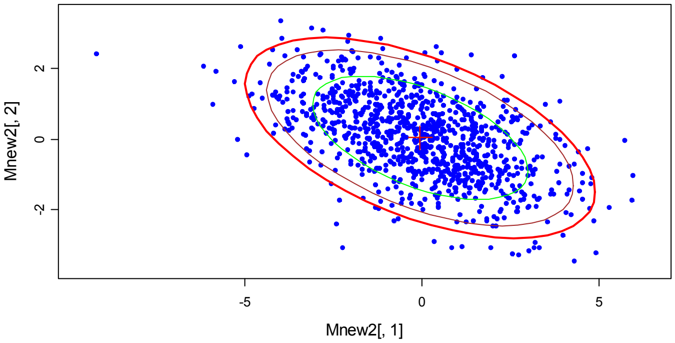
```
#USING A DIFFERENT LINEAR TRANSFORMATION:
A=matrix(c(1,-0.25,-0.25,1),nrow=2,ncol=2,byrow=TRUE)
A

Mnew2=t(A%*%as.matrix(t(M)))
cor(Mnew2)
```



```
#USING A DIFFERENT LINEAR TRANSFORMATION:
A=matrix(c(1,-0.25,-0.25,1),nrow=2,ncol=2,byrow=TRUE)
A

Mnew2=t(A%*%as.matrix(t(M)))


cor(Mnew2)
```

```
> A
      [,1]  [,2]
[1,]  1.00 -0.25
[2,] -0.25  1.00
```

```
> cor(Mnew2)
           [,1]       [,2]
[1,]  1.0000000 -0.5421729
[2,] -0.5421729  1.0000000
```



AR presents a useful discussion and a graph concering confidence ellipses calculated in this manner. Confidence ellipses generally do not match univariate confidence intervals (such as for X_1 or X_2 above) because the multivariate analysis takes in account covariance in Σ (or S) whereas univariate intervals act as if the variables are independent. As a result, there are two discordant cases:

The multivariate confidence ellipse will determine that a case is outside the confidence limit set by α whereas one or both univariate analysis will consider the same case to be within a confidence intervals. This results because covariance between th variables has not been considered in the univariate analyses.

The multivariate confidence ellipse will determine that a case resides within the confidence ellipse whereas one of both univariate analysis will consider the same case to lie outside the confidence interval. This is an example of "Rao's paradox" resulting from the fact that the univariate analyses are run on the same dataset thus have a "family-wise" α that should be considered, but is not.