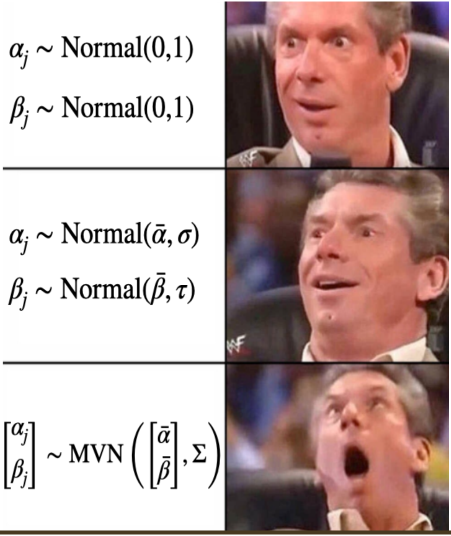
Lab Session-3: Multivariate Distributions

MATH350 – Statistical Inference

Dr. Tanujit Chakraborty
Assistant Professor in Statistics at Sorbonne University
tanujit.chakraborty@sorbonne.ae

Course Webpage: https://www.ctanujit.org/SI.html
R Code: https://github.com/tanujit123/MATH350

Multivariate Distributions!!!!!


$\alpha_j \sim \text{Normal}(0,1)$
$\beta_j \sim \text{Normal}(0,1)$
$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma)$
$\beta_j \sim \text{Normal}(\bar{\beta}, \tau)$
$\begin{bmatrix}\alpha_j \\ \beta_j\end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix}\bar{\alpha} \\ \bar{\beta}\end{bmatrix}, \Sigma\right)$
WWF
WWF

Topics for Today's Session

- Multivariate Normal Distribution
- Multivariate t-distribution
- Skew normal and Skew t distribution

Multivariate Distributions

Two most useful statistical multivariate distributions include:

- Multivariate normal distribution
- Multivariate t-distribution

Distribution	**Location Parameter**	**Scale Parameter**	**Degrees of Freedom**
Normal	mean	sigma	No
t	delta	sigma	Yes

Functions for statistical distributions in R

Normal	Multivariate Normal	t	Multivariate t
rnorm	rmvnorm	rt	rmvt
dnorm	dmvnorm	dt	dmvt
pnorm	pmvnorm	pt	pmvt
qnorm	qmvnorm	qt	qmvt

The first letter denotes

- **r** for “simulation"
- **d** for “density"
- **p** for “probability"
- **q** for “quantile"

Followed by the distribution name

- **norm**
- **mvnorm**
- **t**

The rmvnorm function

```
install.package (“mvtnorm”)
library (mvtnorm)
rmvnorm (n, mean , sigma)
```

Parameters need to be specified:

- **n** the number of samples
- **mean** the mean of the distribution
- **sigma** the variance-covariance matrix

Using rmvnorm to generate random samples

Generate 1000 samples from a 3 dimensional normal with

$$\mu = \begin{pmatrix} 1 \\ 2 \\ -5 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

```
mu1 <- c (1, 2, -5)
sigma1 <- matrix ( c ( 1,1,0,1,2,0,0,0,5 ), 3,3 )
set.seed (34)
sim_mv = rmvnorm (n = 1000, mean = mu1, sigma = sigma1)
library (”corrplot”)
corrplot (cor (sim_mv), method = ”ellipse”)
```

Plot of generated samples

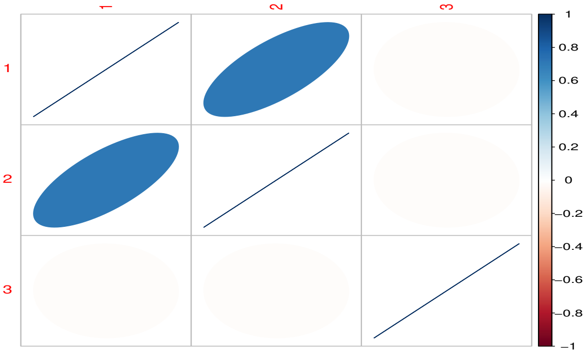


Figure: Correlation plot of the generated sample

Density using dmvnorm

```
install.package (“mvtnorm”)
library (mvtnorm)
dmvnorm (x, mean , sigma)
```

Parameters need to be specified:

- **x** can be a vector or matrix
- **mean** the mean of the distribution
- **sigma** the variance-covariance matrix

Density using dmvnorm

Compute the density at $(0, 0)$ from normal distribution with mean and variance-covariance matrix as

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

```
mu1 <- c (1, 2)
sigma1 <- matrix ( c ( 1, .5, .5, 2 ) , 2 )
dmvnorm ( x = c ( 0, 0 ), mean = mu1, sigma = sigma1)
Output: 0.03836759
```

Density at multiple points using dmvnorm

Compute the density at $x = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ from normal distribution with mean and variance-covariance matrix as

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

Density at multiple points using dmvnorm

Compute the density at $x = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ from normal distribution with mean and variance-covariance matrix as

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

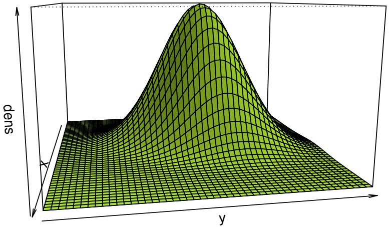
```
x <- rbind ( c ( 0, 0 ), c ( 1, 1 ), c ( 0, 1 ) )
mu1 <- c (1, 2)
sigma1 <- matrix ( c ( 1, .5, .5, 2 ) , 2 )
dmvnorm ( x = c ( 0, 0 )x, mean = mu1, sigma = sigma1)
Output: 0.03836759   0.09041010   0.06794114
```

Code for plotting bivariate densities

Steps:

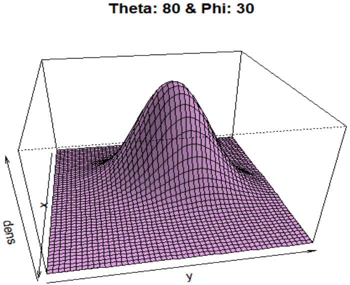
- Create grid of **x** and **y** coordinates
- Calculate density on grid
- Convert densities into a matrix
- Create perspective plot using

persp() function



Plotting bivariate densities

```
> d <- expand.grid ( seq ( -3,
6, length.out = 50 ), seq( -3, 6,
length.out = 50 ) )
> dens1 <- dmvnorm (
as.matrix ( d ), mean = c ( 1, 2
), sigma = matrix ( c( 1, .5, .5,
2 ), 2 ) )
> dens1 <- matrix(dens1,
nrow = 50 )
> persp(dens1, theta = 80, phi
= 30, expand = 0.6, shade
= 0.2, col = ″plum1″, xlab
= ″x″, ylab = ″y″, zlab =
″dens″)
```



Calculating CDF and inverse CDF

Compute the probability at $x \leq 200$ where x is distributed as a normal distribution with mean 210 and variance 100.

```
pnorm ( 200, mean = 210, sd = 10 )
Output: 0.1586553
```

What is the x_0 such that the cumulative probability at x_0 is 0.95?

```
qnorm ( p = 0.95, mean = 210, sd = 10 )
Output: 226.4485
```

Cumulative distribution using pmvnorm

Bivariate CDF at x = 2 and y = 4 for a normal with

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

```
mu1 <- c ( 1, 2 )
sigma1 <- matrix ( c ( 1, 0.5, 0.5, 2 ) , 2 )
pmvnorm ( upper = c ( 2, 4 ) , mean = mu1, sigma = sigma1)
Output:
0.79
attr(,"error")
1e-15
attr ( "msg" )
"Normal Completion"
```

Probability between two values using

Probability of $1 < x < 2$ and $2 < y < 4$ for a normal with

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

```
mu1 <- c ( 1, 2 )
sigma1 <- matrix ( c ( 1, 0.5, 0.5, 2 ) , 2 )
pmvnorm ( lower = c(1, 2), upper = c(2, 4), mean = mu1,
sigma = sigma1)
Output: [1] 0.163
```

Implementing qmvnorm to calculate quantiles

```
sigma1 <- diag ( 2 )
sigma1
```

Output: $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$

```
qmvnorm ( p = 0.95, sigma = sigma1, tail = "both")
```

Output:

```
$quantile
2.24
$f.quantile
-1.31e-06
attr(, "message")
"Normal Completion"
```

Checking normality of multivariate data

Why check normality?
Classical statistical techniques that assume univariate or multivariate normality:

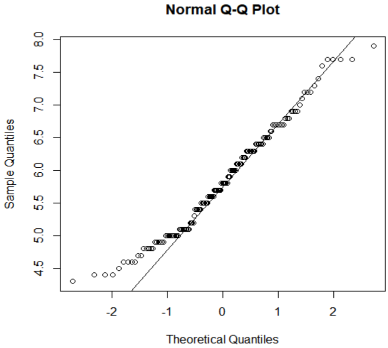
- Multivariate regression
- Discriminant analysis
- Model-based clustering
- Principal component analysis (PCA)
- Multivariate analysis of variance (MANOVA)

Univariate normality tests

Check whether “Sepal.Length" attribute of iris dataset in R follows a normal distribution.

```
qqnorm ( iris [, 1] )
qqline ( iris [, 1] )
```

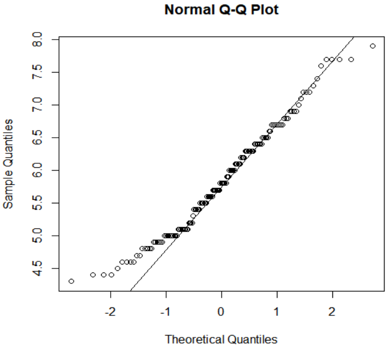
- If the values lie along the reference line the distribution is close to normal.



Univariate normality tests

Check whether “Sepal.Length" attribute of iris dataset in R follows a normal distribution.

- If the values lie along the reference line the distribution is close to normal.
- Deviation from the line might indicate the following :
  - heavier tails
  - skewness
  - outliers
  - clustered data

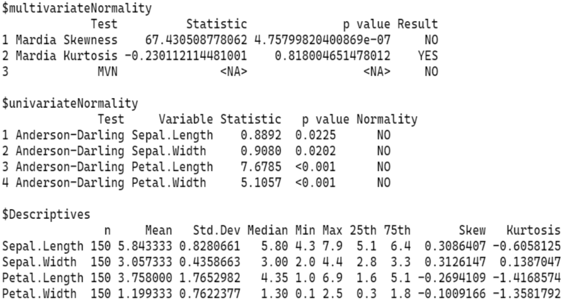


MVN library multivariate normality test functions

- Multivariate normality tests by
  - Mardia
  - Henze-Zirkler
  - Royston
- Graphical appoaches
  - chi-square Q-Q
  - perspective
  - contour plots

Using mardiaTest to check multivariate normality

```
install.packages ( "MVN" )
library ( MVN )
mvn ( iris [, 1:4 ] , subset = NULL, mvnTest = "mardia")
```

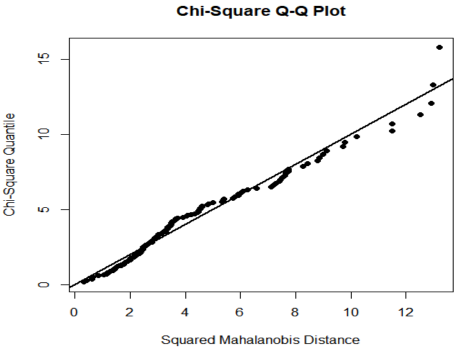


```
$multivariateNormality
             Test          Statistic              p value Result
1 Mardia Skewness    67.430508778062 4.75799820400869e-07     NO
2 Mardia Kurtosis -0.230112114481001    0.818004651478012    YES
3             MVN               <NA>                 <NA>     NO

$univariateNormality
              Test     Variable Statistic   p value Normality
1 Anderson-Darling Sepal.Length    0.8892  0.0225      NO
2 Anderson-Darling Sepal.Width     0.9080  0.0202      NO
3 Anderson-Darling Petal.Length    7.6785  <0.001      NO
4 Anderson-Darling Petal.Width     5.1057  <0.001      NO

$Descriptives
               n     Mean   Std.Dev Median Min Max 25th 75th       Skew   Kurtosis
Sepal.Length 150 5.843333 0.8280661   5.80 4.3 7.9  5.1  6.4  0.3086407 -0.6058125
Sepal.Width  150 3.057333 0.4358663   3.00 2.0 4.4  2.8  3.3  0.3126147  0.1387047
Petal.Length 150 3.758000 1.7652982   4.35 1.0 6.9  1.6  5.1 -0.2694109 -1.4168574
Petal.Width  150 1.199333 0.7622377   1.30 0.1 2.5  0.3  1.8 -0.1009166 -1.3581792
```

Iris data is not multivariate normal

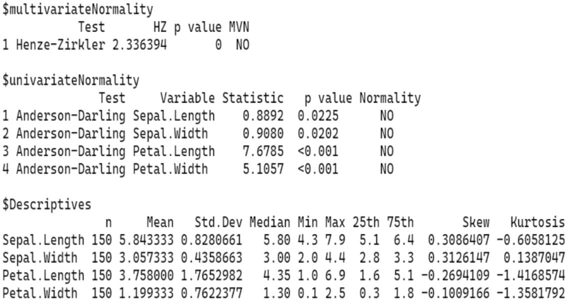
Using qqplot from mardiaTest to check multivariate normality

mvn (iris [, 1:4], subset = NULL, mvnTest = "mardia", multivariatePlot = "qq")



Using hzTest to check multivariate normality

```
install.packages ( "MVN" )
library ( MVN )
mvn ( iris [, 1:4 ] , subset = NULL, mvnTest = "hz")
```

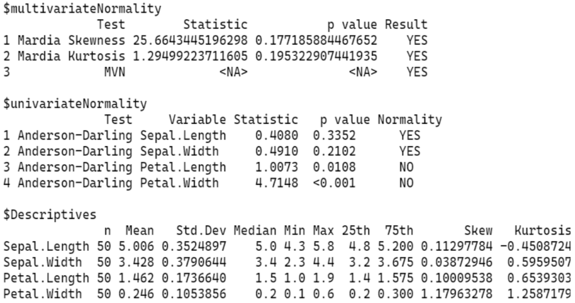


```
$multivariateNormality
           Test       HZ p value MVN
1 Henze-Zirkler 2.336394       0  NO

$univariateNormality
              Test     Variable Statistic   p value Normality
1 Anderson-Darling Sepal.Length    0.8892  0.0225      NO
2 Anderson-Darling Sepal.Width     0.9080  0.0202      NO
3 Anderson-Darling Petal.Length    7.6785  <0.001      NO
4 Anderson-Darling Petal.Width     5.1057  <0.001      NO

$Descriptives
               n     Mean   Std.Dev Median Min Max 25th 75th       Skew   Kurtosis
Sepal.Length 150 5.843333 0.8280661   5.80 4.3 7.9  5.1  6.4  0.3086407 -0.6058125
Sepal.Width  150 3.057333 0.4358663   3.00 2.0 4.4  2.8  3.3  0.3126147  0.1387047
Petal.Length 150 3.758000 1.7652982   4.35 1.0 6.9  1.6  5.1 -0.2694109 -1.4168574
Petal.Width  150 1.199333 0.7622377   1.30 0.1 2.5  0.3  1.8 -0.1009166 -1.3581792
```

Iris data is not multivariate normal

Testing multivariate normality by species

```
install.packages ( ”MVN” )
library ( MVN )
mvn (iris [iris $ Species == ”setosa”, 1:4], subset = NULL,
mvnTest = ”mardia”)
```

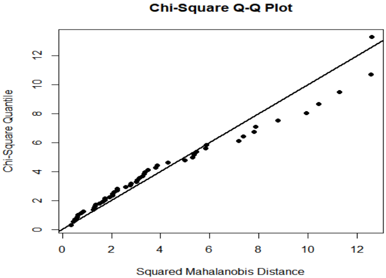


```
$multivariateNormality
             Test        Statistic           p value Result
1 Mardia Skewness 25.6643445196298 0.177185884467652    YES
2 Mardia Kurtosis 1.29499223711605 0.195322907441935    YES
3             MVN             <NA>              <NA>    YES

$univariateNormality
              Test     Variable Statistic   p value Normality
1 Anderson-Darling Sepal.Length    0.4080  0.3352      YES
2 Anderson-Darling Sepal.Width     0.4910  0.2102      YES
3 Anderson-Darling Petal.Length    1.0073  0.0108      NO
4 Anderson-Darling Petal.Width     4.7148  <0.001      NO

$Descriptives
              n  Mean   Std.Dev Median Min Max 25th  75th       Skew   Kurtosis
Sepal.Length 50 5.006 0.3524897    5.0 4.3 5.8  4.8 5.200 0.11297784 -0.4508724
Sepal.Width  50 3.428 0.3790644    3.4 2.3 4.4  3.2 3.675 0.03872946  0.5959507
Petal.Length 50 1.462 0.1736640    1.5 1.0 1.9  1.4 1.575 0.10009538  0.6539303
Petal.Width  50 0.246 0.1053856    0.2 0.1 0.6  0.2 0.300 1.17963278  1.2587179
```

Data is multivariate normal

Checking QQ plot by species

install.packages ("MVN")
library (MVN)
mvn (iris [iris $ Species == "setosa", 1:4], subset = NULL, mvnTest = "mardia", multivariatePlot = "qq")



Data is multivariate normal

Comparing univariate normal with univariate t-distributions

```
df = c ( 1, 4, 10, 30, 80 )
colour = c ( "red", "darkorange2", "forestgreen", "goldenrod3","blueviolet","black" )
plot (x, dnorm(x), type = "l", lty = 2, xlab = "t-value", ylab = "Density", main = "Comparison of t-distributions", col = "black")
for (i in 1:5) {
          lines(x, dt (x, df [i]), col = colour[i])
        }
legend ("topright", c ("df = 1", "df = 4", "df = 10", "df = 30","df = 80", "normal"), col = colour, title = "t-distributions", lty = c(1,1,1,1,1,2))
```

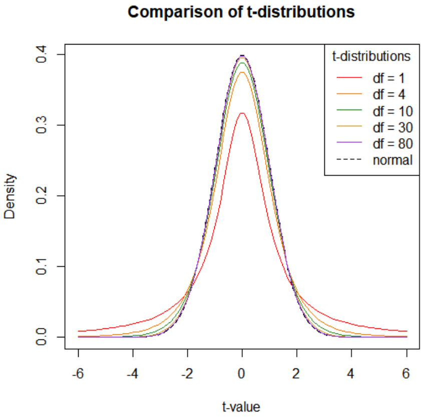
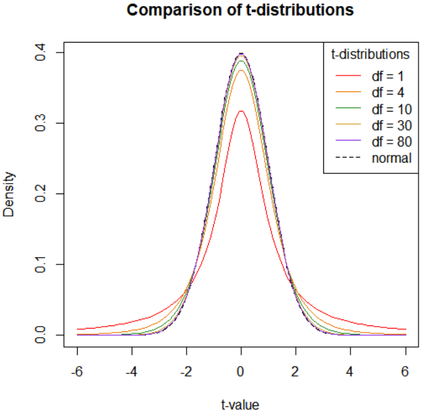


Figure: Standard Normal distribution vs t distributions

Comparing normal and t-distribution tails

Tails are fatter for the same cutoff
$P(X < 1.96 \text{ or } X > 1.96)$

Distribution	Probability
Normal	0.05
t(df=1)	0.3
t(df=8)	0.0857
t(df=20)	0.0641
t(df=30)	0.0593



Multivariate t-distribution notation

- Generalization of the univariate Student's t-distribution
- Widely used version has only one degree of freedom for all dimensions and is denoted by

$$t_{df}\left(\delta, \Sigma\right)$$

- The probability density function of the d-dimensional multivariate Student's t distribution is given by

$$f(x, \Sigma, \delta) = \frac{1}{|\Sigma|^{1/2}} \frac{1}{\sqrt{(\delta\pi)^d}} \frac{\Gamma((\delta+d)/2)}{\Gamma(\delta/2)} \left(1 + \frac{x'\Sigma^{-1}x}{\delta}\right)^{-(\nu+d)/2}.$$

where x is a $1 \times d$ vector, Σ is a $d \times d$ symmetric, positive definite matrix, and δ is a positive scalar.

Contours of bivariate normal and t-distributions

$$\mu = \delta = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

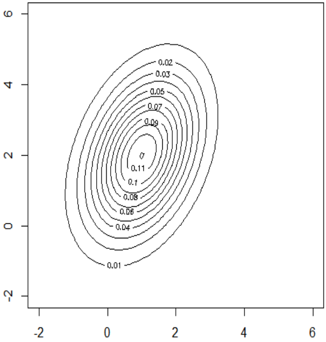
Normal distribution

```
library(mvtnorm)
x.points <- seq (-2, 6, length.out = 100)
y.points <- x.points
z <- matrix (0,nrow=100,ncol=100)
mu <- c (1,2)
sigma <- matrix (c (1,0.5,0.5,2), nrow=2)
for (i in 1:100) {
    for (j in 1:100) {
     z [i,j] <- dmvnorm (c (x.points[i], y.points[j]),
mean = mu, sigma = sigma)
    }
}
contour(x.points,y.points,z)
```

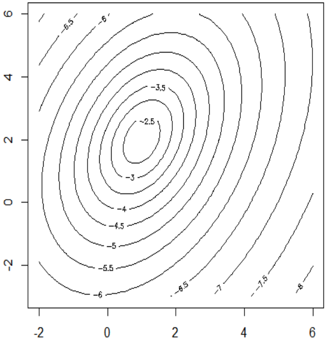
t distribution

```
library(mvtnorm)
x.points <- seq (-2, 6, length.out = 100)
y.points <- seq(-3,6,length.out = 100)
z <- matrix (0,nrow=100,ncol=100)
mu <- c (1,2)
sigma <- matrix (c (1,0.5,0.5,2), nrow=2)
for (i in 1:100) {
        for (j in 1:100) {
         ; z[i,j] <- dmvt (c (x.points[i], y.points[j]),
mean = mu, sigma = sigma)
        }
}
contour(x.points,y.points,z)
```

Normal distribution



t distribution



Functions for multivariate t-distributions

Functions include:

- rmvt (n, delta, sigma, df)
- dmvt (x, delta, sigma, df)
- qmvt(p, delta, sigma, df)
- pmvt(upper, lower, delta, sigma, df)

Generating random samples

Generate samples from 3-dimensional t distribution with $\delta = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}$,

$\Sigma = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}$, df = 4.

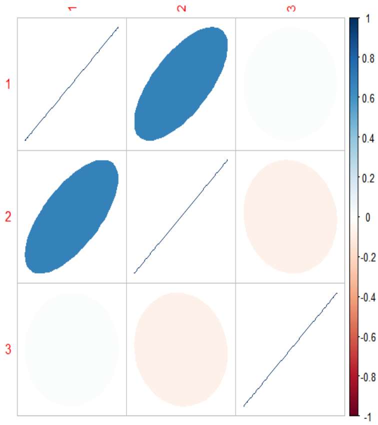
```
delta <- c(1, 2, -5)
sigma <- matrix (c (1, 1, 0, 1, 2, 0, 0, 0, 5), 3, 3)
t.sample <- rmvt (n = 2000, delta = delta, sigma = sigma, df = 4)
head (t.sample, 4) library ("corrplot")
corrplot (cor (t.sample), method = "ellipse")
```

Output:

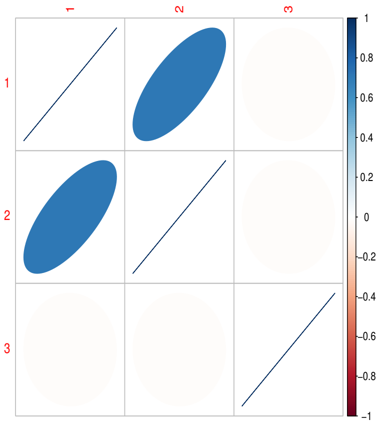
```
1.467661   0.7945283   −5.554285
1.233803   3.1037985   −5.735940
1.643157   3.5588006   −5.337057
1.078938   1.2893042   −3.737054
```

Comparing with normal samples

t distribution with df 4

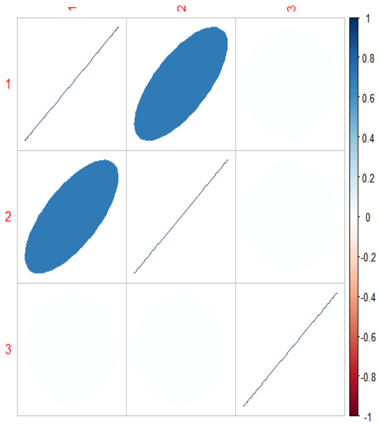


Normal distribution

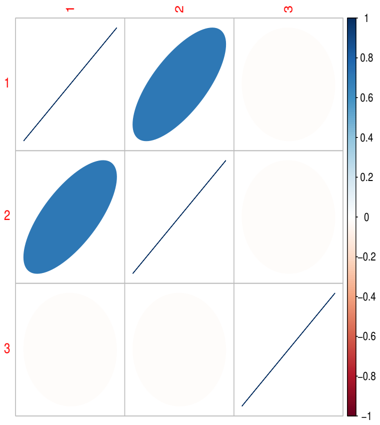


Comparing with normal samples

t distribution with df 10



Normal distribution



Example of multivariate t-distribution

- Individual stocks
  - Univariate t
- Portfolio (3 stocks)
  - Multivariate t
- Probability that all three stocks between $100-150
  - pmvt ()
- Range of values that the stocks fluctuate 95% of the time
  - qmvt ()

Density using dmvt

```
dmvt (x, delta = rep (0, p), sigma = diag (p), log = TRUE)
```

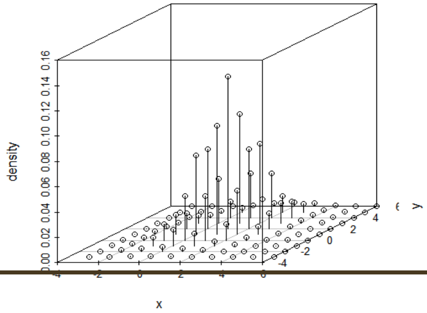
- ***x*** can be a vector or a matrix.
- Unlike ***dmvnorm*** the default calculation is in log scale

To get the densities in natural scale use

```
dmvt (x, delta = rep (0, p), sigma = diag (p), log = FALSE)
```

Calculating the density of a multivariate t-distribution on a grid

```
x <- seq (-3, 6, by = 1)
y <- seq (-3, 6, by = 1)
d <- expand.grid (x = x, y = x)
del1 <- c(1, 2); sig1 <- matrix(c(1, .5, .5, 2), 2)
dens <- dmvt (as.matrix (d), delta = del1, sigma = sig1, df = 10, log = FALSE)
library(scatterplot3d)
scatterplot3d (cbind (d, dens), type = "h", zlab = "density")
```



Effect of changing the degrees of freedom

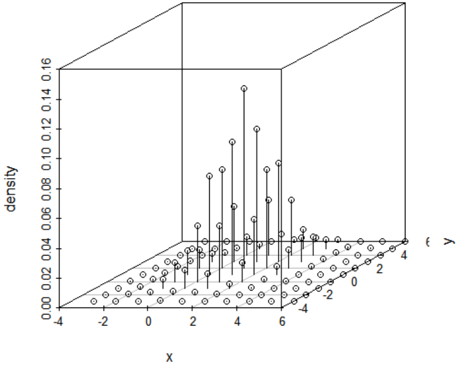


Figure: Density of multivariate t distribution with 30 degrees of freedom

Cumulative density using pmvt

```
pmvt (lower = -Inf, upper = Inf, delta, sigma, df, . . .)
```

- Calculates the cdf or volume similar to normal ***pmvnorm ()*** function

```
pmvt (lower = c(-1,-2), upper = c(2, 2), delta = c(1, 2), sigma
= diag(2), df = 6)
```

Output:

```
0.3856191
attr(,"error")
0.0001927966
attr(,"msg")
"Normal Completion"
```

Inverse cdf of t-distribution

```
qmvt (p, interval, tail, delta, sigma, df)
```

- Computes the quantile of the multivariate t-distribution.
- Computation techniques similar to ***qmvnorm ()*** function.

Calculate the 0.95 quantile for 3 degrees of freedom

```
qmvt ( p = 0.95, sigma = diag (2), tail = "both", df = 3)
```

Output:

```
quantile 3.960018
f.quantile -1.048671e-06
attr(,"message") "Normal Completion"
```

Univariate skew-normal distribution

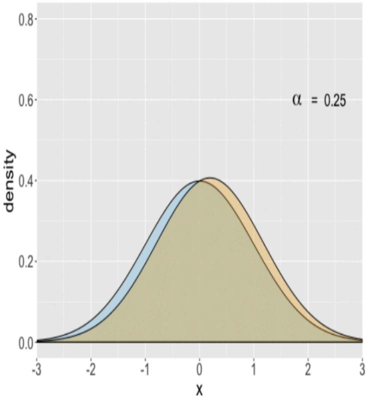
General skew-normal is denoted by SN (ξ, ω, α)

- ξ and ω are the location and scale parameters

Simplest form: z SN(α)

- α is the skewness parameter

Comparing SN (α) to a standard Normal distribution



- For $\alpha > 0$ skewed to the right
- For $\alpha < 0$ skewed to the left
- SN (0) is the same as a standard Normal

Multivariate skew-normal distribution

Notations: three-dimensional multivariate skew-normal distribution

$$\mathrm{SN}\left(\xi, \omega, \alpha\right)$$

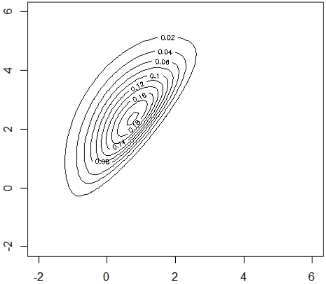
- ξ location parameter (vector of length 3)
- ω variance-covariance parameter (3 × 3 matrix)
- α skewness parameter (vector of length 3)

Bivariate skew-normal dist. contour plot

Bivariate skew-normal

$$\xi = \begin{bmatrix}1\\2\end{bmatrix}, \omega = \begin{bmatrix}1 & 0.5\\0.5 & 2\end{bmatrix}, \alpha = \begin{bmatrix}-3\\3\end{bmatrix}.$$

```
library(sn)
x.points <- seq (-2, 6, length.out = 100)
y.points <- seq (-2, 6, length.out = 100)
z <- matrix (0, nrow = 100, ncol = 100)
xi <- c (1,2); alp <- c (-3, 3)
sigma <- matrix (c (1,0.5,0.5,2), nrow=2)
for (i in 1:100) {
for (j in 1:100) {
z[i,j] <- dmsn(c(x.points[i],y.points[j]), xi =
xi, Omega = sigma, alpha = alp)
}
}
contour(x.points,y.points,z)
```



Functions for skew-normal distribution

From sn library:

- dmsn(x, xi, Omega, alpha)
- pmsn(x, xi, Omega, alpha)
- rmsn(n, xi, Omega, alpha)

Need to specify xi , Omega , alpha

Functions for skew-t distribution

From sn library:

- dmst(x, xi, Omega, alpha, nu)
- pmst(x, xi, Omega, alpha, nu)
- rmst(n, xi, Omega, alpha, nu)

Need to specify xi , Omega , alpha , nu (degrees of freedom)

Generating skew-normal samples

Generate 2000 samples from 3 dimensional skew-normal

$$\mathrm{SN}\left(\xi = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}, \omega = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \alpha = \begin{bmatrix} 4 \\ 30 \\ -5 \end{bmatrix}\right)$$

```
xi1 <- c(1, 2,-5)
Omega1 <- matrix( c(1, 1, 0, 1, 2, 0, 0, 0, 5), 3, 3)
alpha1 <- c(4, 30,-5)
skew.sample <- rmsn (n = 2000, xi = xi1, Omega = Omega1,
alpha = alpha1)
library (”corrplot”)
corrplot (cor (skew.sample), method = ”ellipse”)
```

Plot of skew-normal samples

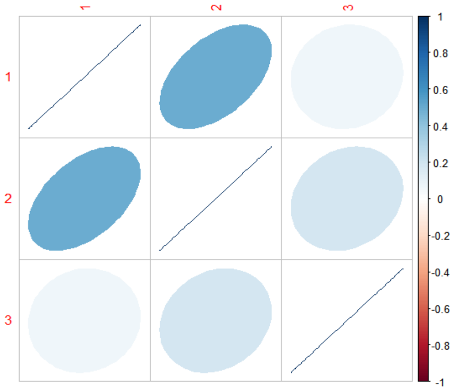


Figure: Correlation plot of skew-normal samples

Generating skew-t samples

Generate 2000 samples from 3 dimensional skew-t

$$\text{SN}\left(\xi = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}, \omega = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \alpha = \begin{bmatrix} 4 \\ 30 \\ -5 \end{bmatrix}, df = 4\right)$$

```
xi1 <- c(1, 2,-5)
Omega1 <- matrix( c(1, 1, 0, 1, 2, 0, 0, 0, 5), 3, 3)
alpha1 <- c(4, 30,-5)
skewt.sample <- rmst (n = 2000, xi = xi1, Omega = Omega1,
alpha = alpha1, nu = 4)
library (”corrplot”)
corrplot (cor (skewt.sample), method = ”ellipse”)
```

Plot of skew-t samples

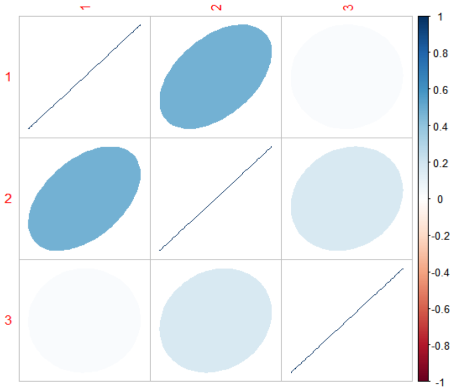


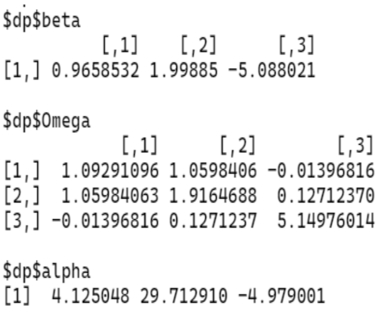
Figure: Correlation plot of skew-t samples

Estimation of parameters from data

- Need iterative algorithm to estimate the parameters of a skew-normal distribution
  - No explicit equation to calculate parameters
- Several functions in ***sn*** package, including ***msn.mle()*** function

Estimation of parameters from data

msn.mle (y = skew.sample, opt.method = ″BFGS″)



```
$dp$beta
         [,1]    [,2]      [,3]
[1,] 0.9658532 1.99885 -5.088021

$dp$Omega
            [,1]      [,2]        [,3]
[1,]  1.09291096 1.0598406 -0.01396816
[2,]  1.05984063 1.9164688  0.12712370
[3,] -0.01396816 0.1271237  5.14976014

$dp$alpha
[1]  4.125048 29.712910 -4.979001
```

Samples were generated from a skew normal distribution with parameters:

$$\xi = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}, \omega = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \alpha = \begin{bmatrix} 4 \\ 30 \\ -5 \end{bmatrix}$$

Reference

https://www.datacamp.com/courses/multivariate-probability-distributions-in-r