

Lab Session-1: Normality & Beyond with R Code

MATH350 – Statistical Inference

STATISTICS + MACHINE LEARNING + DATA SCIENCE

Dr. Tanujit Chakraborty, Ph.D. from ISI Kolkata.

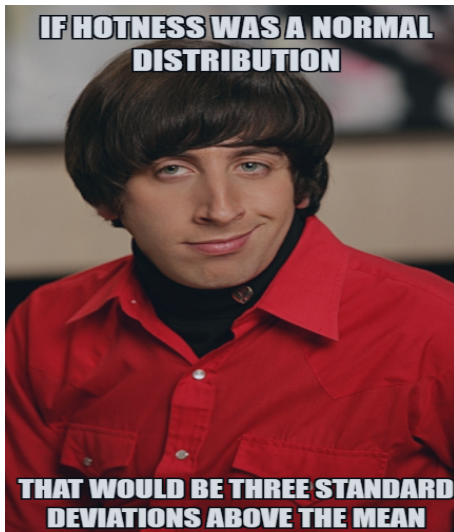
Assistant Professor in Statistics at Sorbonne University.

Mail me at tanujitisi@gmail.com

Course Webpage: <https://www.ctanujit.org/SI.html>

Code available at <https://github.com/tanujit123/MATH350>

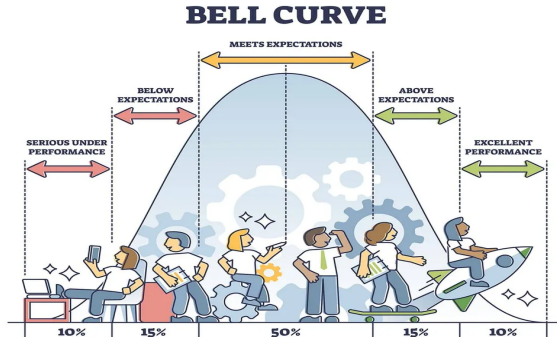
Let's Start the Journey to Normality





- Normality: A Brief History
- Univariate Normal Distribution
- Drawbacks and Skew Normal Distribution
- Multivariate data: Iris Data
- Data Visualisation
- Multivariate Normal Distribution
- Multivariate t-distribution
- Skew normal and Skew t distribution

Normality is a paved road. It is easy to walk but no flowers grow on it. — Vincent Van Gogh.



By Dr. Saul McLeod (2019)

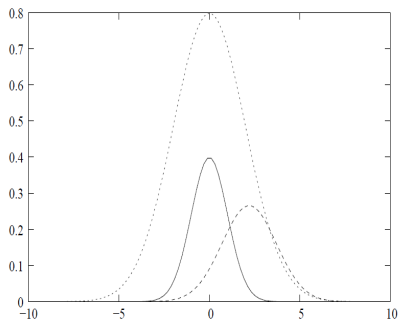
Normality is a myth; there never was, and there never will be a normal distribution — Roy C. Geary (1947; Biometrika, vol. 34, 248).

*Everybody believes in the exponential law of errors (*the normal distribution*), the experimenters, because they think it can be proved by mathematicians; and the mathematicians, because they believe that it has been established by observations — E.T. Whittaker and G. Robinson (1967).*

*... *the statisticians knows* ... that in nature there never was a normal distribution, there never was a straight line, yet with normal and linear assumptions, known to be false he can often derive results which match to a useful approximation, those found in real world — George W. Box (1976, Journal of American Statistical Association, vol. 71, 791-799).*

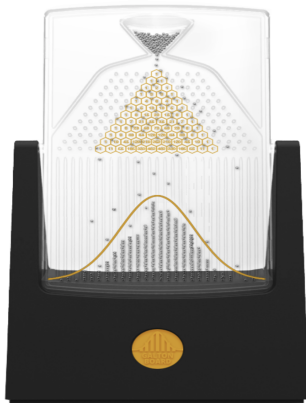
A random variable X is said to be normally distributed with mean μ and variance σ^2 , if the probability density function of X is the following (for $-\infty < \mu < \infty$ and $\sigma > 0$)

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



Probability Density Function of Normals

- Sir Francis Galton, Charles Darwin's half-cousin, invented the 'Galton Board' in 1874 to demonstrate that the normal distribution is a natural phenomenon.
- It specifically shows that the binomial distribution approximates a normal distribution with a large enough sample size.



Picture of Galton Board

Gambling Question: A 17th century gambler, the Chevalier de Mere, asked Pascal for an explanation of his unexpected losses in gambling.

The famous correspondence between Pascal and Fermat was instigated in 1654, and they were mainly interested to calculate the following binomial sum:

$$\sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k}$$

The problem was not difficult when n is small.

Within few years the following problem arises in a sociological study, where the following computation was necessary:

$$n = 11,429, i = 5745, j = 6128$$

$$\sum_{k=i}^j \binom{n}{k} p^k (1-p)^{n-k}$$

Original Problem: The problem is to test the hypothesis that male and female births are equally likely against the actual birth in London over 82 years from 1629 - 1710. It is observed that the relative number of male births varies from a low of $7765/15,448 = 0.5027$ in 1703 to a high of $4748/8855 = 0.5362$ in 1661. Given that 11,429 is the average number of births in London over 82 years, and 5745 and 6128 are two limits.

Using the following recurrence relation

$$\binom{n}{x+1} = \binom{n}{x} \binom{n-x}{x+1}$$

and some involved rational approximation it has been obtained

$$P(5747 \leq X \leq 6128 \mid p = 1/2) = \sum_{i=5745}^{6128} \binom{11,429}{i} \left(\frac{1}{2}\right)^i$$

$$\approx 0.292$$

Using the following recurrence relation

$$\binom{n}{x+1} = \binom{n}{x} \binom{n-x}{x+1}$$

and some involved rational approximation it has been obtained

$$P(5747 \leq X \leq 6128 \mid p = 1/2) = \sum_{i=5745}^{6128} \binom{11,429}{i} \left(\frac{1}{2}\right)^i$$

$$\approx 0.292$$

De Moivre began the search for this approximation in 1721 ,
and in 1733 it has been proved that

$$\binom{n}{\frac{n}{2} + x} \left(\frac{1}{2}\right)^n \approx \frac{2}{\sqrt{2\pi n}} e^{-2x^2/n}$$

and

$$\sum_{|x-n/2| \leq a} \binom{n}{x} \left(\frac{1}{2}\right)^n \approx \frac{4}{\sqrt{2\pi}} \int_0^{a/\sqrt{n}} e^{-2y^2} dy.$$

Eventually using the second approximation one gets

$$\sum_{k=i}^j \binom{n}{k} p^k (1-p)^{n-k} \approx \Phi\left(\frac{j-np}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{i-np}{\sqrt{np(1-p)}}\right)$$

where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-x^2/2} dx$$

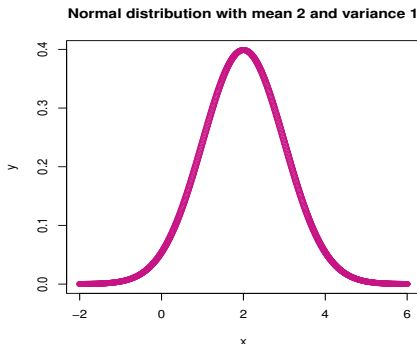
which is the cumulative distribution function (CDF) of the standard normal distribution.

Gauss (1809) made the following assumptions and deduce the normal distribution as an error distribution:

- 1 Small errors are more likely than large errors.
- 2 For any real numbers ϵ , the likelihood of errors of magnitudes ϵ and $-\epsilon$ are equal.
- 3 In the presence of several measurements of the same quantity, the most likely value of the quantity being measured is their average.

Univariate normal with mean 2 and variance 1.

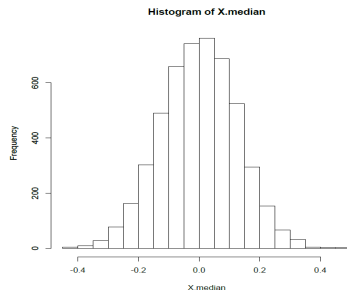
```
x <- seq(-2, 6, by = .01)
y <- dnorm(x, mean = 2,
sd = 1)
plot(x,y,col = "mediumvioletred")
```



Simulating a sample median. Let $X_1, \dots, X_{99} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1)$. The sample median is the 50th largest value among X_1, \dots, X_{99} . Compute the sample medians from 5000 simulations of X_1, \dots, X_{99} . What is the mean of these 5000 sample medians? What is their standard deviation? Plot a histogram of the 5000 values - what does the sampling distribution of the sample median look like?

R Code:

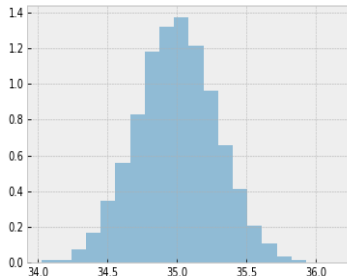
```
X.median = numeric(5000)
for(i in 1:5000) {
  X = rnorm(99, mean = 0, sd = 1)
  X.median[i] = median(X)
}
print(mean(X.median))
print(sd(X.median))
hist(X.median)
```



Lindeberg-Levy CLT:

Suppose $\{X_1, X_2, \dots\}$ is a sequence of independent identically distributed random variables with mean μ and variance $\sigma^2 < \infty$, then as $n \rightarrow \infty$

$$\frac{\sqrt{n}}{\sigma} \left(\frac{1}{n} \sum_{i=1}^n X_i - \mu \right) \rightarrow N(0, 1)$$



CLT in Practice

What will happen if the data indicate that the parent distribution

- ① is not symmetric?
- ② is heavy tail?
- ③ is not unimodal?

What will happen if error distribution is not normal during regression modeling?

In Distribution Theory:

- 1 **Skew Normal Distribution** (A Azzalini, Scandinavian Journal of Statistics 1985)
- 2 Power Normal Distribution (RD Gupta, Test 2008)
- 3 Geometric Skew-Normal Distribution (D Kundu, Sankhya 2014), etc.

In Regression Theory:

- 1 Box-Cox Transformation (Box, Cox, JRSS Series-B 1964)
- 2 Generalized linear model (Nelder, Wedderburn, JRSS Series-A 1972)
- 3 Semiparametric and Nonparametric Approaches (see ESLR/ISLR Book), etc.

Goal:

- 1 Generate a non-symmetric class of distributions which have support on the whole real line.
- 2 Normal distribution is a special member.
- 3 It should not have too many parameters.

Construction:

- Suppose X and Y are two independent standard normal random variables, and λ is any real number. Therefore

$$P(X < \lambda Y) = P(X - \lambda Y < 0) = \frac{1}{2}$$

as $X - \lambda Y$ is a normal random variable with mean 0 , and variance $1 + \lambda^2$.

- On the other hand

$$P(X < \lambda Y) = \int_{-\infty}^{\infty} \Phi(\lambda y) \phi(y) dy$$

where

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad \text{and} \quad \Phi(x) = \int_{-\infty}^x \phi(u) du.$$

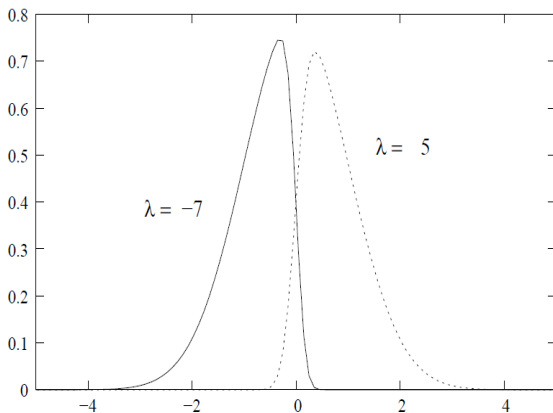
- Therefore,

$$\frac{1}{2} = \int_{-\infty}^{\infty} \Phi(\lambda y) \phi(y) dy.$$

Since $\Phi(\lambda y) \phi(y) \geq 0$, the function

$$f(x; \lambda) = 2\phi(x)\Phi(\lambda x)$$

is a proper probability density function, and it is called skew-normal probability density function with parameter λ and we will denote it by $\text{SN}(\lambda)$.



(1) The $SN(0)$ density is the $N(0, 1)$ density.

(2) As $\lambda \rightarrow \infty$,

$$f(x; \lambda) \rightarrow \sqrt{\frac{2}{\pi}} e^{-x^2/2}; \quad x > 0$$

(3) If Z is a $SN(\lambda)$ random variable, then $-Z$ is a $SN(-\lambda)$ random variable.

(4) The PDF of a $SN(\lambda)$ random variable is unimodal.

(5) If Z is $SN(\lambda)$ then Z^2 is χ_1^2 .

For data analysis purposes three-parameter skew normal distribution can be easily defined with the probability density function as follows:

$$f(x; \mu, \sigma, \lambda) = \frac{2}{\sigma} \phi\left(\frac{x - \mu}{\sigma}\right) \Phi\left(\frac{\lambda(x - \mu)}{\sigma}\right)$$

- Rectangular in shape - organized by rows and columns
 - ☐ Rows represent observations
 - ☐ Columns represent variables
- May or may not include:
 - ☐ Row names or numbers
 - ☐ Column headers
- Possible missing data

Example : Iris Data

This is perhaps the best known database to be found in the ML literature created by R.A. Fisher. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

You are not a data scientist..



if you don't know this flower

Iris data from '*datasets*' package in R.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

The Iris dataset comprises 150 observations (rows) on the following 5 variables (columns)

- ✿ *Sepal.Length* - length (in cm) of the flower's sepal.
- ✿ *Sepal.Width* - width (in cm) of the flower's sepal.
- ✿ *Petal.Length* - length (in cm) of the flower's petal.
- ✿ *Petal.Width* - width (in cm) of the flower's petal.
- ✿ *Species* - categorical variable represents the category of the flower.

Reading data

```
data (iris)
```

```
head (iris, n= 4)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa

Check the dataset dimension

```
dim (iris)
```

```
150 5
```

Extract the column names

```
names (iris)
```

```
"Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

Access Sepal length and Sepal width columns of observations 8 to 10

```
iris [8:10, 1:2]
```

Sepal.Length	Sepal.Width
5.0	3.4
4.4	2.9
4.9	3.1

Check the data types

```
str(iris)
```

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2
 $ Species      : Factor w/ 3 levels "setosa","versicolor"
```

Reassign factor labels

Re-code the factors “setosa”, “versicolor”, and “virginica” of *Species* variable to “1”, “2”, and “3”

```
library(car)
```

```
iris$Species <- recode(iris$Species, "'setosa' = 1; 'versicolor' = 2; 'virginica' = 3")
```

Calculate mean

```
colMeans (iris [, 1:4])
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.843333	3.057333	3.758000	1.199333

Functions that calculate means by subgroups

- ✓ *by ()*
- ✓ *aggregate ()*

by (data = iris [,1:4], INDICES = iris\$Species, FUN = colMeans)

```
iris$Species: 1
Sepal.Length Sepal.Width Petal.Length Petal.Width
          5.006          3.428          1.462          0.246

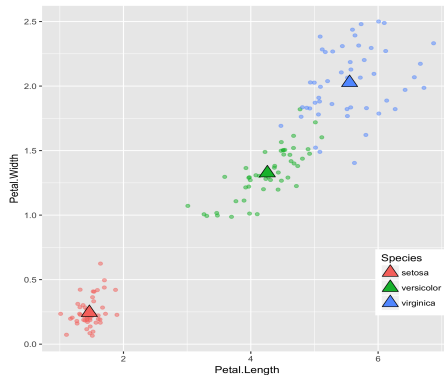
iris$Species: 2
Sepal.Length Sepal.Width Petal.Length Petal.Width
          5.936          2.770          4.260          1.326

iris$Species: 3
Sepal.Length Sepal.Width Petal.Length Petal.Width
          6.588          2.974          5.552          2.026
```

aggregate (. ~ Species, iris, mean)

	Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	1	5.006	3.428	1.462	0.246
2	2	5.936	2.770	4.260	1.326
3	3	6.588	2.974	5.552	2.026

Species	Petal.Length	Petal.Width
setosa	1.46	0.244
versicolor	4.26	1.326
virginica	5.55	2.026



```
var (iris[ , 1:4])
```

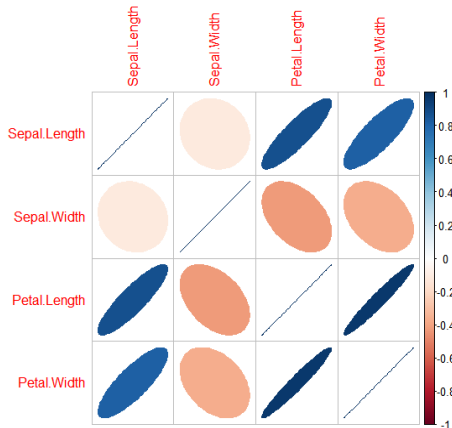
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.6856935	-0.0424340	1.2743154	0.5162707
Sepal.Width	-0.0424340	0.1899794	-0.3296564	-0.1216394
Petal.Length	1.2743154	-0.3296564	3.1162779	1.2956094
Petal.Width	0.5162707	-0.1216394	1.2956094	0.5810063

```
cor (iris[ , 1:4])
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	-0.1175698	0.8717538	0.8179411
Sepal.Width	-0.1175698	1.0000000	-0.4284401	-0.3661259
Petal.Length	0.8717538	-0.4284401	1.0000000	0.9628654
Petal.Width	0.8179411	-0.3661259	0.9628654	1.0000000

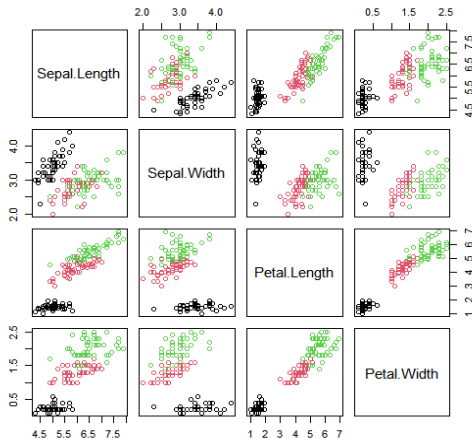
corrplot function to visualize correlation plot

```
library (corrplot)
corrplot (cor (iris [ , 1:4]), method = "ellipse")
```

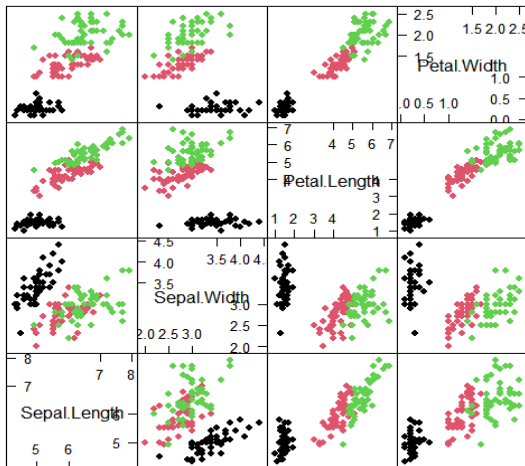


- Basic *R* plot
- *lattice* library
- *ggplot* library
- 3D ploing options

```
pairs (iris [ , 1:4], col = iris$Species)
```



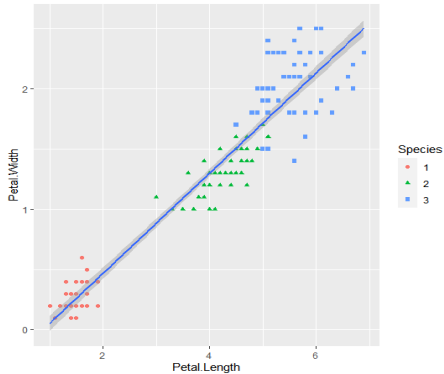
```
library(lattice)
splom ( iris[ , 1:4], col = iris$Species, pch = 16)
```



Scatter Plot Matrix

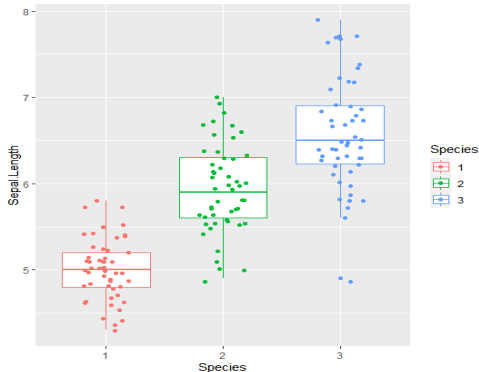
Plot petal length and petal width grouped by species with a linear trend line

```
library(ggplot2)
ggplot(data = iris) + aes(x = Petal.Length, y = Petal.Width) +
  geom_point(aes(color = Species, shape = Species)) + geom_smooth(method = lm)
```



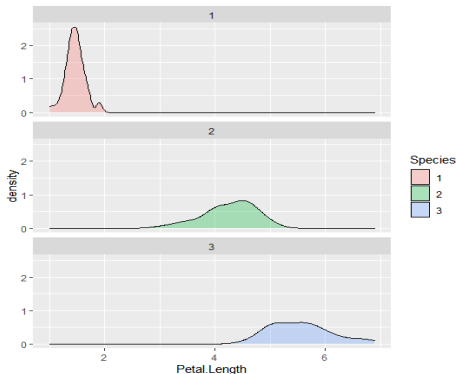
Plot the boxplot of Sepal length grouped by species and add the corresponding measurements

```
library(ggplot2)
ggplot(data = iris) + aes(x = Species, y = Sepal.Length, color = Species) +
  geom_boxplot() + geom_jitter(position = position_jitter(0.2))
```



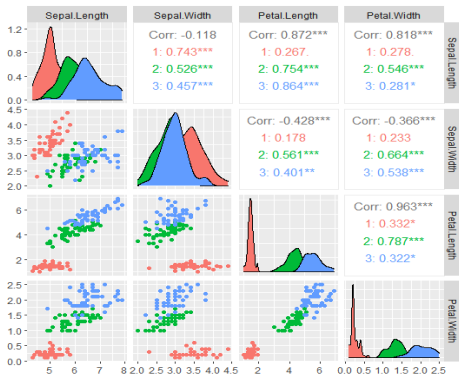
Visualize the density plot of Petal length of different species

```
library(ggplot2)
ggplot(data = iris) + aes(x = Petal.Length, fill = Species) +
  geom_density(alpha = 0.3) + facet_wrap(~ Species, nrow = 3)
```



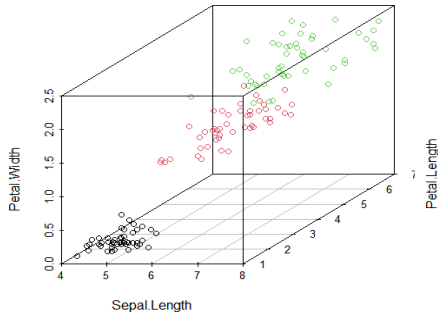
Visualize correlation plot using *ggplot*

```
library (ggplot2)
library (GGally)
ggpairs (data = iris, columns = 1:4, mapping = aes (color = Species))
```



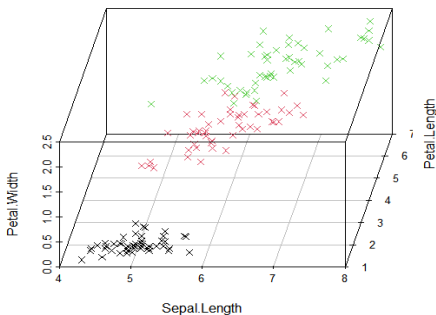
Visualize sepal length, petal length, and petal width of iris data grouped by species

```
library(scatterplot3d)
scatterplot3d(iris[, c(1, 3, 4)], color = as.numeric(iris$Species))
```



Change the angle between X axis and y axis of the previous 3D plot to 80 degrees

```
library (scatterplot3d)  
scatterplot3d(iris[, c(1, 3, 4)], color = as.numeric(iris$Species), pch = 4, angle =  
80)
```



Two must 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

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**

```
install.packages("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

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")
```

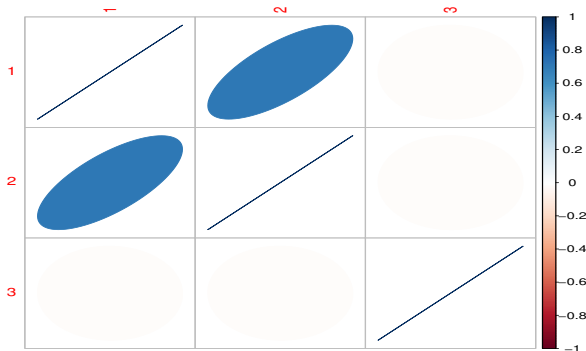


Figure: Correlation plot of the generated sample

```
install.packages("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

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
```

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

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

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

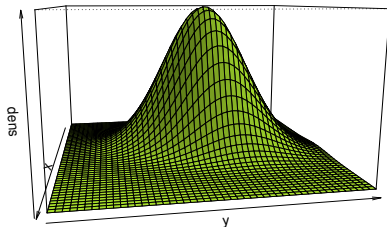
$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

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

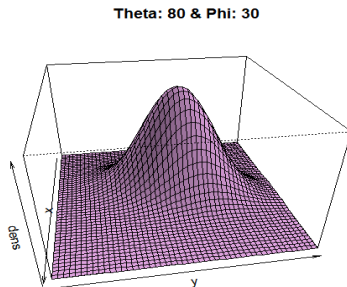
Steps:

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

`persp()` function



```
> d <- expand.grid ( seq ( -3,
6, length.out = 50 ), seq( -3, 6,
length.out = 50 ) )
> dens1 <- dmnorm (
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")
```



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

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 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
```

```
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"
```

Why check normality?

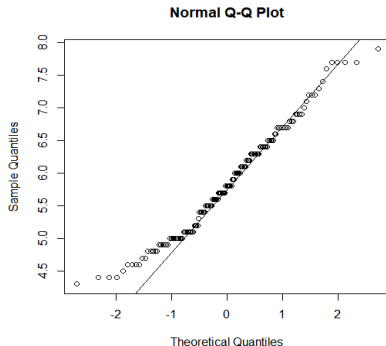
Classical statistical techniques that assume univariate or multivariate normality:

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

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

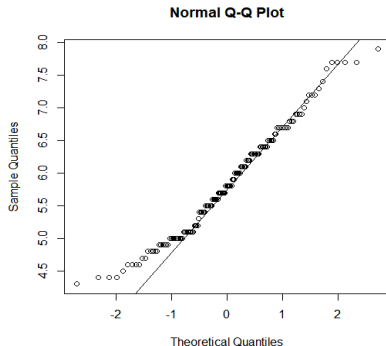
```
qqnorm ( iris [, 1] )  
qqline ( iris [, 1] )
```

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



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



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


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

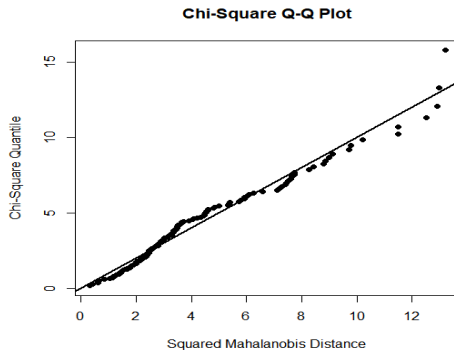
```
$multivariateNormality
      Test      Statistic      p value Result
1 Mardia Skewness  67.430508778062 4.75799820400869e-07 NO
2 Mardia Kurtosis -0.230112114481001 0.8180004651478012 YES
3              MVN              <NA>              <NA> NO
```

```
$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

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



```
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

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

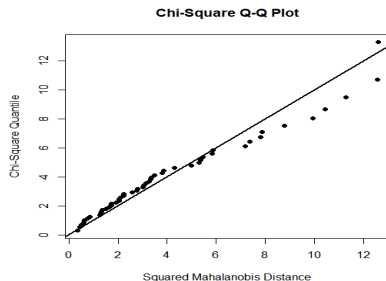
```
$multivariateNormality
      Test      Statistic      p value Result
1 Mardia Skewness 25.6643445196298 0.177185884467652 YES
2 Mardia Kurtosis 1.29499223711605 0.195322907441935 YES
3      MVN          <NA>          <NA>      YES
```

```
$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

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



Data is multivariate normal

```
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))
```

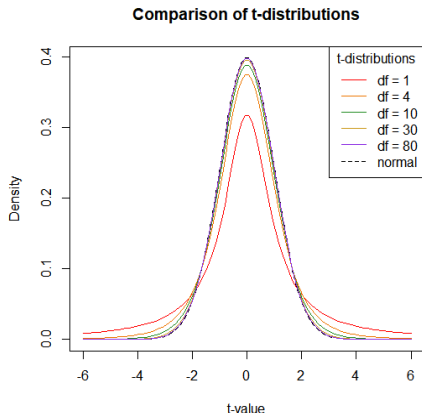


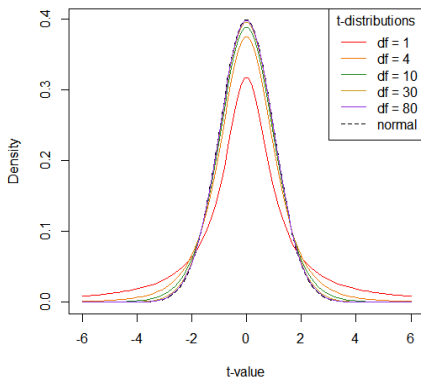
Figure: Standard Normal distribution vs t distributions

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

Comparison of t-distributions



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

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

- 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.

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

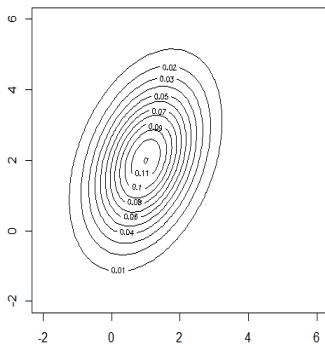
Normal distribution

```
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] <- dmvtnorm (c (x.points[i], y.points[j]),
mean = mu, sigma = sigma)
  }
}
contour(x.points,y.points,z)
```

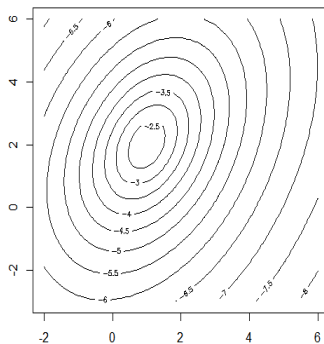
t distribution

```
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 include:

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

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

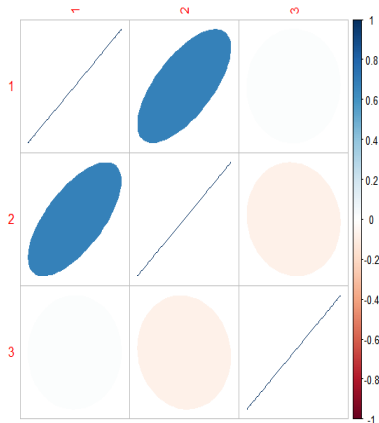
$$\Sigma = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \text{df} = 4.$$

```
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")
```

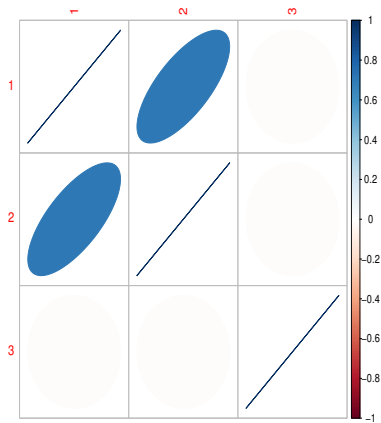
Output:

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

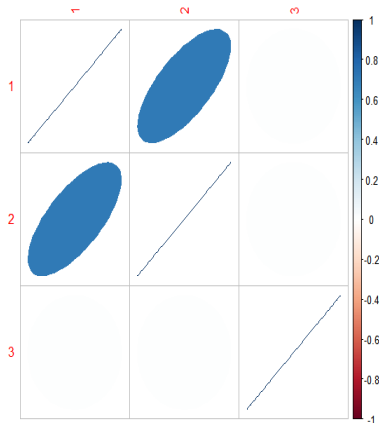
t distribution with df 4



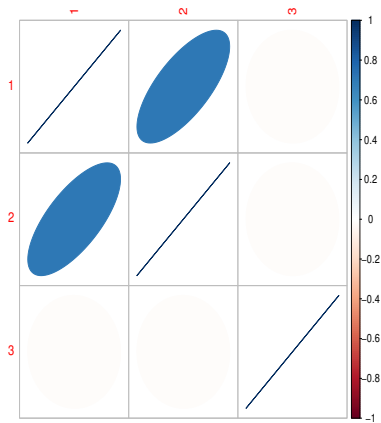
Normal distribution



t distribution with df 10



Normal 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 ()`

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

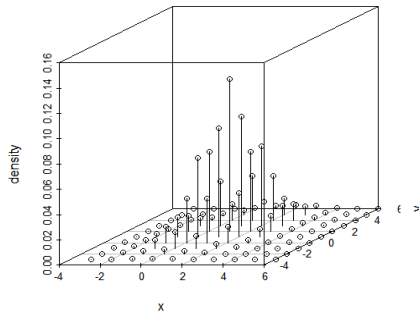
- *x* can be a vector or a matrix.
- Unlike *dmvnorm* the default calculation is in log scale

To get the densities in natural scale use

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



```
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")
```



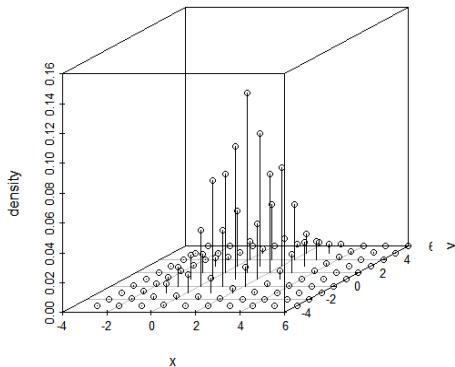


Figure: Density of multivariate t distribution with 30 degrees of freedom

```
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"

qmvot (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

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

Output:

quantile 3.960018

f.quantile -1.048671e-06

attr("message") "Normal Completion"

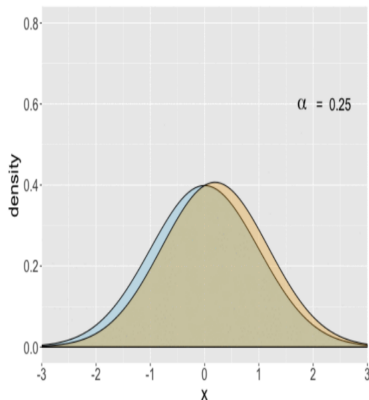
General skew-normal is denoted by $SN(\xi, \omega, \alpha)$

- ξ and ω are the location and scale parameters

Simplest form: $z \sim SN(\alpha)$

- 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

Notations: three-dimensional multivariate skew-normal distribution

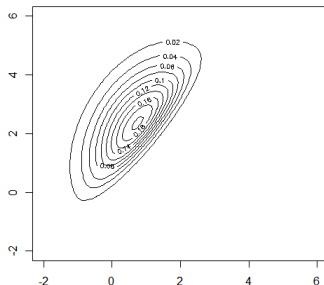
$$\text{SN}(\xi, \omega, \alpha)$$

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

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)
```



From sn library:

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

Need to specify ξ , Ω , α

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)

Generate 2000 samples from 3 dimensional skew-normal

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

```
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")
```

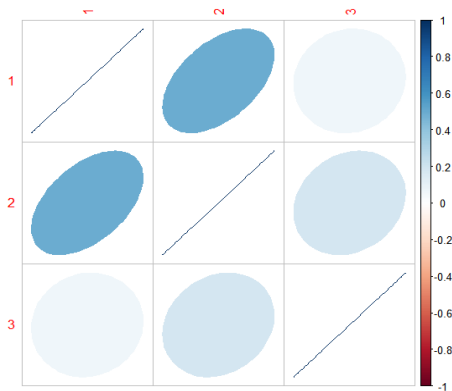


Figure: Correlation plot of skew-normal samples

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")
```

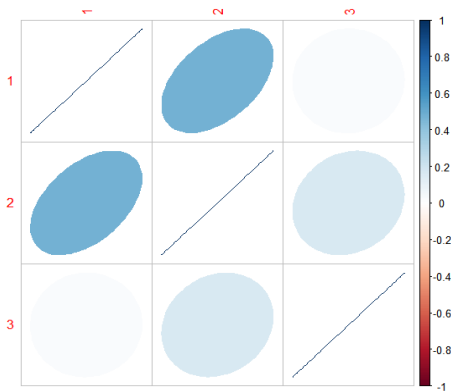


Figure: Correlation plot of skew-t samples

- 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

```
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:

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

- Not all data follows a Normal Distribution.
- Data with outliers or skewness may not be Normally distributed.
- Large samples will be closer to a Normal distribution than small samples.
- Real life data is almost NEVER EXACTLY NORMAL.



1. Saul Stahl (2006), "The evolution of normal distribution", Mathematics Magazine, vol. 79, no. 2, 96 - 113.
3. <https://www.datacamp.com/courses/multivariate-probability-distributions-in-r>
4. Kundu, Debasis. "A Journey Beyond Normality." (2014).



thank
you