STA437/STA1005
Fall 2005
Assignment 1

1) 2.9.15 p52

Let $\mathbf{x}$ be $N_3(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu}'$=[5,10, 2] and $\Sigma = \begin{bmatrix} 4 & 1 & -1 \\ 1 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}$.

a) What is the conditional distribution of x_2 and x_3 given x_1?

b) What is the conditional distribution of $(x_1 + x_2, x_1 - x_2)$ given x_3?

c) What is the distribution of $A\mathbf{x}$, where $A = A = \begin{bmatrix} \frac{1}{2} & -1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$?

(5 marks)

2) 2.9.27 p55

Let $\mathbf{x}$ be $N_3(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu}'$=[5,10, 2] and $\Sigma = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 1 \end{bmatrix}$.

Find the distribution of

a) $x_1 - x_2$,

b) $x_1 + x_2 + x_3$

c) (x_1, x_2), the joint distribution of x_1, x_2.

d) ($x_1, x_2 \mid x_3$), the conditional distribution of x_1, x_2 given x_3.

(5 marks)

3) Let $\mathbf{x}$ be $N_3(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu}'$=[-3,1,4] and $\Sigma = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$.

Which of the following variables are independent? Why?

a) X_1 and X_2

b) X_2 and X_3

c) (X_1, X_2) and X_3

d) X_2 and X_2-2.5 X_1 - X_3

(5 marks)

4) The files, assign1dat1.txt, assign1dat2.txt and assign1dat3.txt (on course web page) contain generated multivariate data on three variables. I have also included a few outliers in these data sets. Use the methods we have discussed in lectures to identify these outlier. Test whether data are from a multivariate normal distribution.
(Use data file assign1dat1.txt if your last name begins with A-E, assign1dat2.txt if your last name begins with F-N, and assign1dat3.txt if your last name begins with O-Z.)
(10 marks)

Question 4 Sol

For data set1 observation number 950 is an outlier. See R-code below.

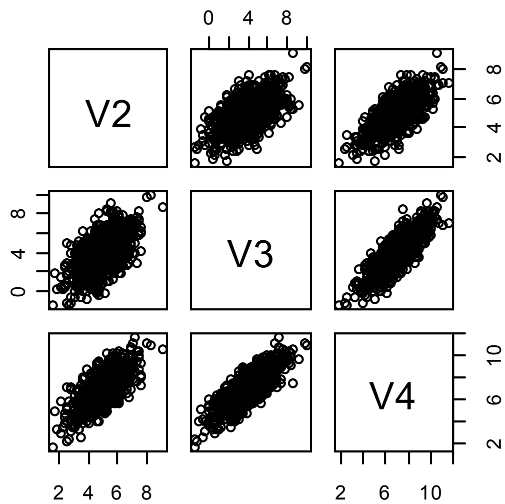
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat1.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
> # rm=k
> # x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an outlier"
> The_most_extreme_obs_is
[1] 950
> q
[1] 25.10855
> f
[1] 22.33503
> n
[1] 1000
> p
[1] 3
>
```

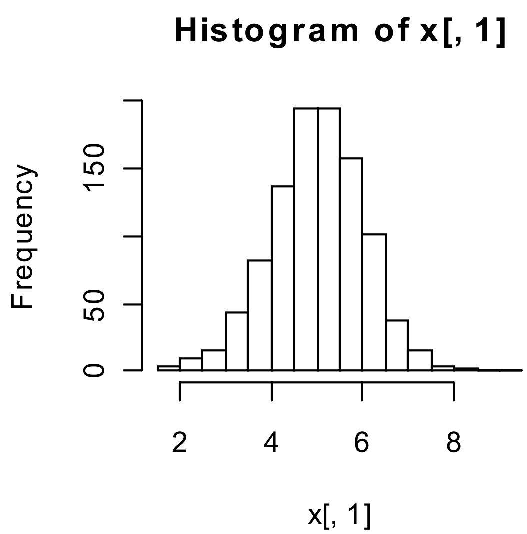
Deleting this observation (No 950), no more outliers.
See R-code below.

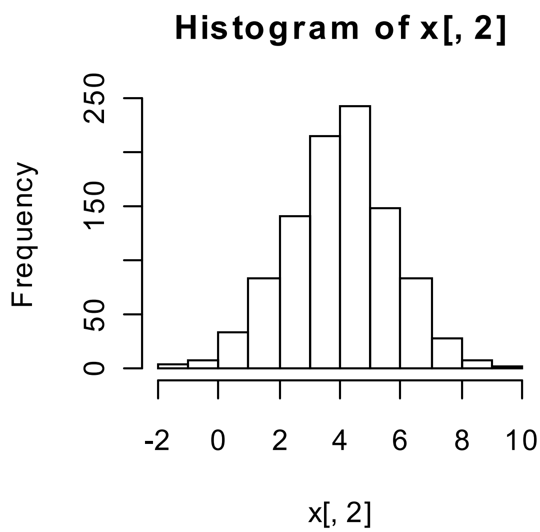
Histograms are approx bell-shaped. qq plots are approximately linear. Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling, Cramer-von Mises normality tests have p-values > 0.05.

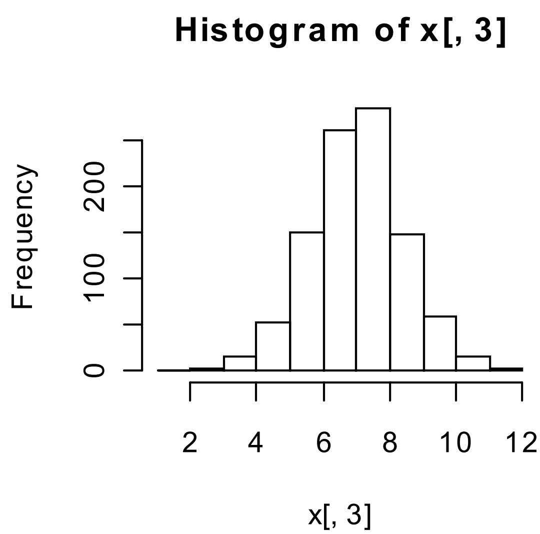
In summary statistics, skewness , kurtosis are approx. 0, also mean and median are approximately equal, all these suggesting univariate normality.
Scatter plot matrix shows elliptical shapes suggesting bivariate normality of components.

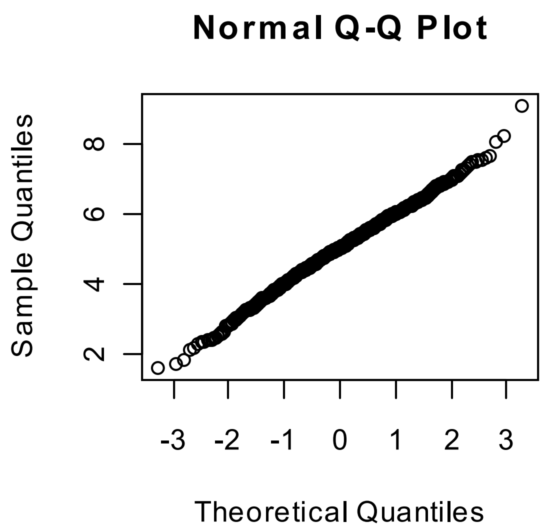
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat1.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
>  rm=950
> x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an not outlier"
> The_most_extreme_obs_is
[1] 454
> q
[1] 17.00927
> f
[1] 22.33277
> n
[1] 999
> p
[1] 3
>
```


V2
V3
V4
0
4
8
2
4
6
8
0
4
8
10
6
2
2
4
6
8
2
6
10

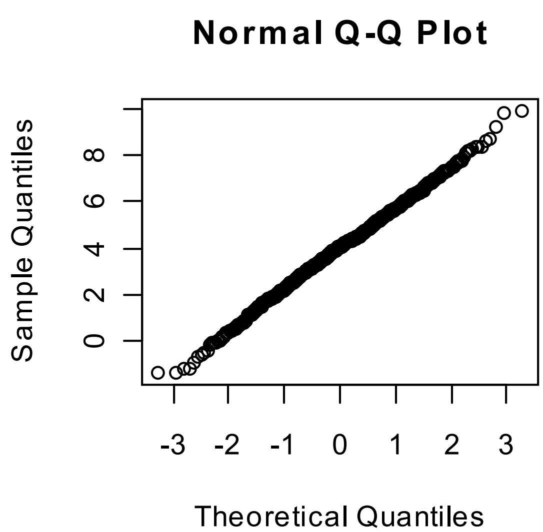

Histogram of x[, 1]
Frequency
0
50
150
2
4
6
8
x[, 1]


Histogram of x[, 2]
Frequency
0
50
150
250
-2
0
2
4
6
8
10
x[, 2]

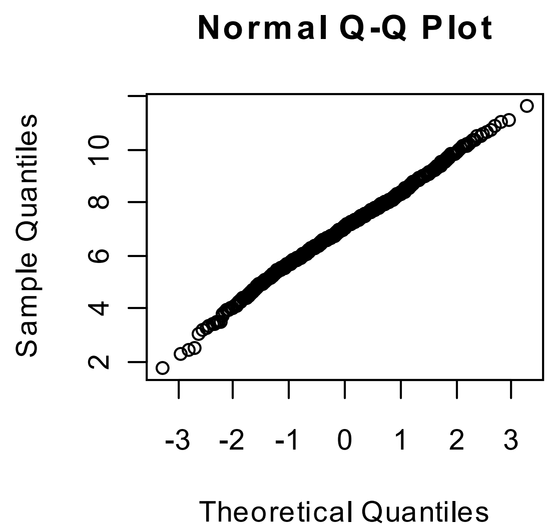

Histogram of x[, 3]
Frequency
0
100
200
2
4
6
8
10
12
x[, 3]



Q-Q plot for x1



Q-Q plot for x2



Q-Q plot for x3

> library(fBasics) # Download the package fBasics if you don't have it

```
> basicStats(x[,1])
                     Value
nobs          999.00000000
NAs             0.00000000
Minimum         1.59692000
Maximum         9.09115000
1. Quartile     4.37636500
3. Quartile     5.69202500
Mean            5.01342264
Median          5.03321000
Sum          5008.40922000
SE Mean         0.03234815
LCL Mean        4.94994445
UCL Mean        5.07690084
Variance        1.04535655
Stdev           1.02242679
Skewness       -0.12449503
Kurtosis        0.30901974
> basicStats(x[,2])
                     Value
nobs          999.00000000
NAs             0.00000000
Minimum        -1.38939000
Maximum         9.91260000
1. Quartile     2.86609500
3. Quartile     5.12728500
Mean            3.98827007
Median          4.04433000
Sum          3984.28180000
SE Mean         0.05515056
LCL Mean        3.88004570
UCL Mean        4.09649444
Variance        3.03854316
Stdev           1.74314175
Skewness       -0.05204742
```

```
Kurtosis          0.18612338
> basicStats(x[,3])
                       Value
nobs            999.00000000
NAs               0.00000000
Minimum           1.73340000
Maximum          11.65360000
1. Quartile       6.12520000
3. Quartile       7.92695000
Mean              7.02560340
Median            7.06240000
Sum            7018.57780000
SE Mean           0.04434246
LCL Mean          6.93858825
UCL Mean          7.11261856
Variance          1.96428754
Stdev             1.40153043
Skewness         -0.11576629
Kurtosis          0.40507867
>
> ks.test(x[,1], "pnorm", m=mean(x[,1]), sd=sqrt(var(x[,1])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 1]
D = 0.0265, p-value = 0.4836
alternative hypothesis: two.sided

Warning message:
cannot compute correct p-values with ties in: ks.test(x[, 1], "pnorm", m = mean(x[, 1]), sd =
sqrt(var(x[,
> shapiro.test(x[,1])

        Shapiro-Wilk normality test

data:  x[, 1]
W = 0.9972, p-value = 0.07475

> library(nortest) # Package for normality tests,
> # Download the package nortest if you don't have it
> ad.test(x[,1])

        Anderson-Darling normality test

data:  x[, 1]
A = 0.7296, p-value = 0.05699

> cvm.test(x[,1])

        Cramer-von Mises normality test

data:  x[, 1]
W = 0.1135, p-value = 0.0735

> ks.test(x[,2], "pnorm", m=mean(x[,2]), sd=sqrt(var(x[,2])))
```

```
        One-sample Kolmogorov-Smirnov test

data:  x[, 2]
D = 0.0211, p-value = 0.7668
alternative hypothesis: two.sided

> shapiro.test(x[,2])

        Shapiro-Wilk normality test

data:  x[, 2]
W = 0.9984, p-value = 0.4628

> ad.test(x[,2])

        Anderson-Darling normality test

data:  x[, 2]
A = 0.49, p-value = 0.2204

> cvm.test(x[,2])

        Cramer-von Mises normality test

data:  x[, 2]
W = 0.0925, p-value = 0.1425

> ks.test(x[,3], "pnorm", m=mean(x[,3]), sd=sqrt(var(x[,3])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 3]
D = 0.0236, p-value = 0.6367
alternative hypothesis: two.sided

Warning message:
cannot compute correct p-values with ties in: ks.test(x[, 3], "pnorm", m = mean(x[, 3]), sd =
sqrt(var(x[,
> shapiro.test(x[,3])

        Shapiro-Wilk normality test

data:  x[, 3]
W = 0.9972, p-value = 0.0762

> ad.test(x[,3])

        Anderson-Darling normality test

data:  x[, 3]
A = 0.6741, p-value = 0.07813

> cvm.test(x[,3])

        Cramer-von Mises normality test
```

```
data:  x[, 3]
W = 0.1044, p-value = 0.09784

>
```

For data set 2

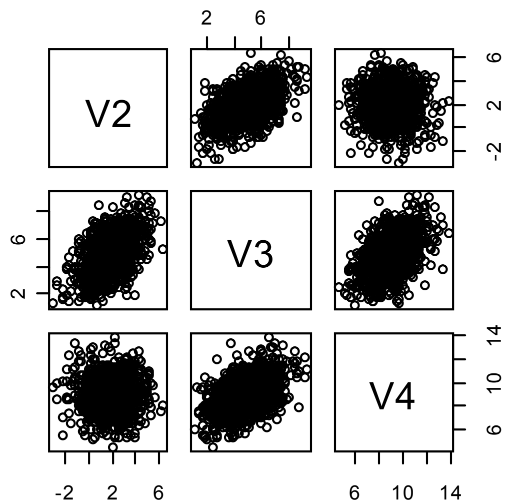
Observation No: 99 is an outlier.

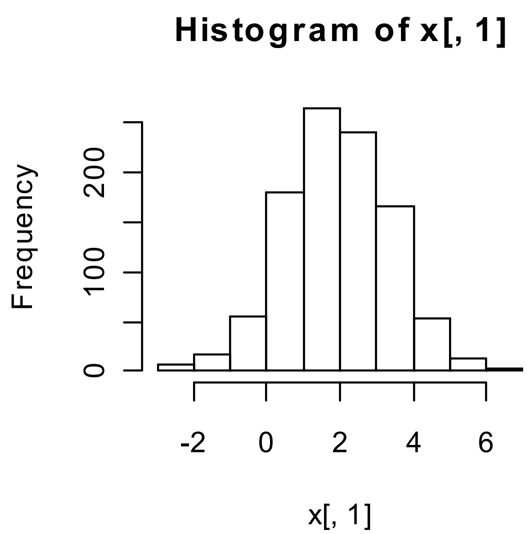
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat2.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
> # rm=950
> #x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an outlier"
> The_most_extreme_obs_is
[1] 99
> q
[1] 23.05472
> f
[1] 22.33503
> n
[1] 1000
> p
[1] 3
>
```

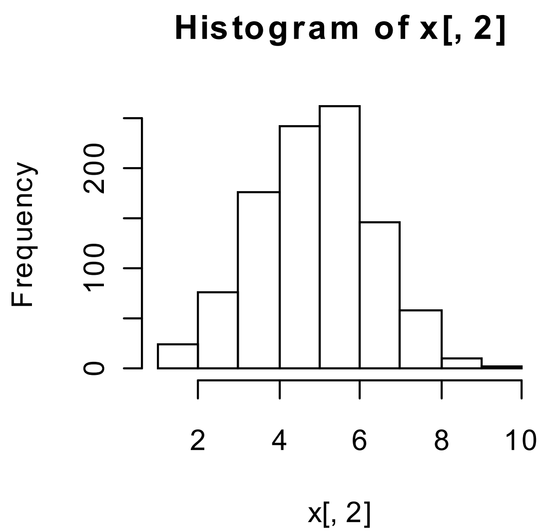
Deleting this observation (No 99), no more outliers.
See R-code below.
Histograms are approx bell-shaped. qq plots are approximately linear. Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling, Cramer-von Mises normality tests have p-values > 0.05.

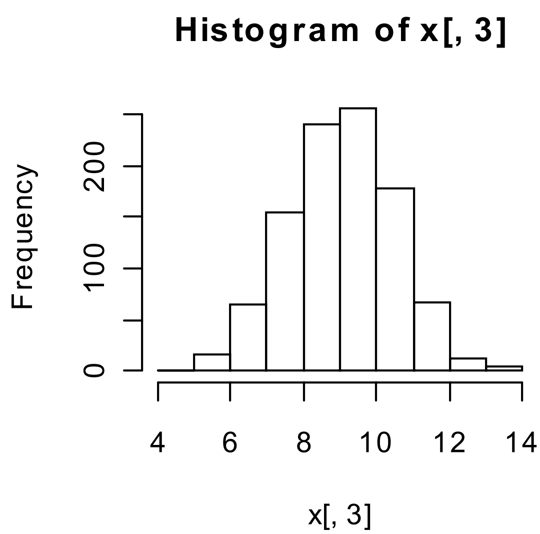
In summary statistics, skewness , kurtosis are approx. 0, also mean and median are approximately equal, all these suggesting univariate normality.
Scatter plot matrix shows elliptical shapes suggesting bivariate normality of components.

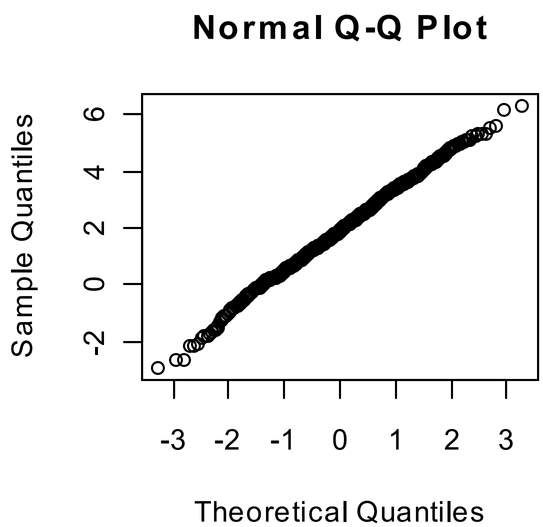
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat2.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
>  rm=99
> x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an not outlier"
> The_most_extreme_obs_is
[1] 530
> q
[1] 19.56559
> f
[1] 22.33277
> n
[1] 999
> p
[1] 3
>
```


V2
V3
V4
2
6
-2
2
6
6
10
14

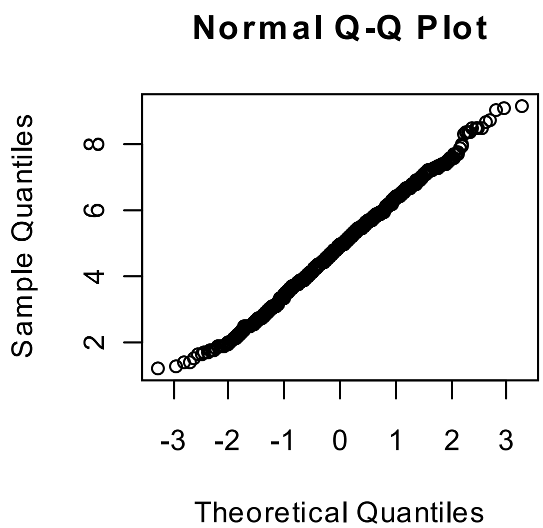

Histogram of x[, 1]
Frequency
0
100
200
-2
0
2
4
6
x[, 1]


Histogram of x[, 2]
Frequency
0
100
200
2
4
6
8
10
x[, 2]

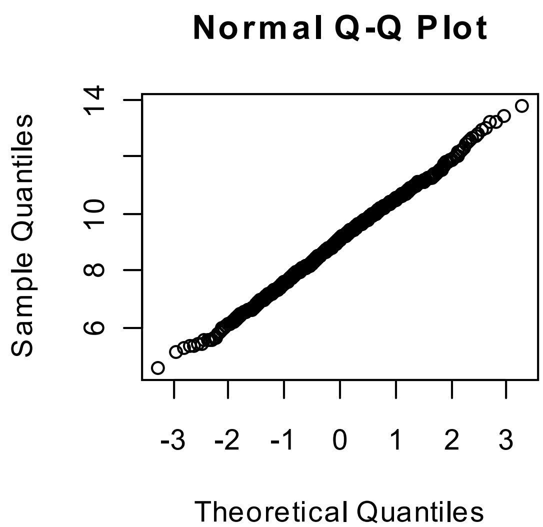

Histogram of x[, 3]
Frequency
0
100
200
4
6
8
10
12
14
x[, 3]



Q-Q plot for x1



Q-Q plot for x2



Q-Q plot for x3

```
> basicStats(x[,1])
                     Value
nobs          999.00000000
NAs             0.00000000
Minimum        -2.94612802
Maximum         6.31869131
1. Quartile     0.93269216
3. Quartile     2.92485326
Mean            1.92121793
Median          1.90158009
Sum          1919.29671584
SE Mean         0.04525996
LCL Mean        1.83240234
UCL Mean        2.01003353
Variance        2.04641525
Stdev           1.43052971
Skewness       -0.07416149
Kurtosis        0.02109943
> basicStats(x[,2])
                     Value
nobs          9.990000e+02
NAs           0.000000e+00
Minimum       1.198343e+00
Maximum       9.209430e+00
1. Quartile   3.909375e+00
3. Quartile   5.895518e+00
Mean          4.908370e+00
Median        4.943197e+00
Sum           4.903461e+03
SE Mean       4.547089e-02
LCL Mean      4.819140e+00
UCL Mean      4.997599e+00
Variance      2.065534e+00
Stdev         1.437197e+00
Skewness      6.913554e-03
Kurtosis     -2.857781e-01
```

```
> basicStats(x[,3])
                     Value
nobs           999.00000000
NAs              0.00000000
Minimum          4.56810100
Maximum         13.79616900
1. Quartile      8.06761000
3. Quartile     10.07330450
Mean             9.05447616
Median           9.10847500
Sum           9045.42168600
SE Mean          0.04602301
LCL Mean         8.96416319
UCL Mean         9.14478914
Variance         2.11599944
Stdev            1.45464753
Skewness        -0.03342650
Kurtosis        -0.11226557
>
```

> ks.test(x[,1], "pnorm", m=mean(x[,1]), sd=sqrt(var(x[,1])))

One-sample Kolmogorov-Smirnov test

data: x[, 1]
D = 0.0181, p-value = 0.8972
alternative hypothesis: two.sided

> shapiro.test(x[,1])

Shapiro-Wilk normality test

data: x[, 1]
W = 0.9982, p-value = 0.3588

> library(nortest) # Package for normality tests,
> # Download the package nortest if you don't have it
> ad.test(x[,1])

Anderson-Darling normality test

data: x[, 1]
A = 0.3472, p-value = 0.4791

> cvm.test(x[,1])

Cramer-von Mises normality test

data: x[, 1]
W = 0.0419, p-value = 0.646

> ks.test(x[,2], "pnorm", m=mean(x[,2]), sd=sqrt(var(x[,2])))

One-sample Kolmogorov-Smirnov test

data: x[, 2]
D = 0.0185, p-value = 0.8848

```
alternative hypothesis: two.sided

> shapiro.test(x[,2])

        Shapiro-Wilk normality test

data:  x[, 2]
W = 0.997, p-value = 0.06105

> ad.test(x[,2])

        Anderson-Darling normality test

data:  x[, 2]
A = 0.4499, p-value = 0.2756

> cvm.test(x[,2])

        Cramer-von Mises normality test

data:  x[, 2]
W = 0.0592, p-value = 0.3868

> ks.test(x[,3], "pnorm", m=mean(x[,3]), sd=sqrt(var(x[,3])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 3]
D = 0.0213, p-value = 0.7539
alternative hypothesis: two.sided

> shapiro.test(x[,3])

        Shapiro-Wilk normality test

data:  x[, 3]
W = 0.9984, p-value = 0.5074

> ad.test(x[,3])

        Anderson-Darling normality test

data:  x[, 3]
A = 0.4508, p-value = 0.2743

> cvm.test(x[,3])

        Cramer-von Mises normality test

data:  x[, 3]
W = 0.0733, p-value = 0.2529

>
```

Data set 3

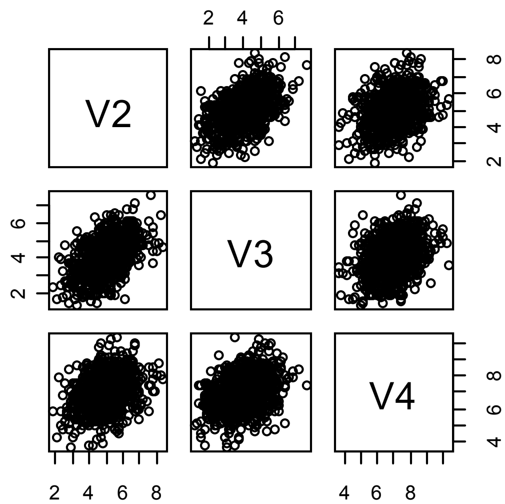
Observation no 72 is an outlier. See R code below.

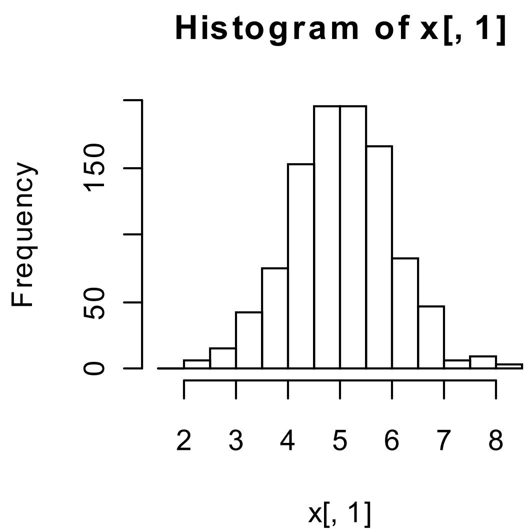
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat3.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
> # rm=99
> # x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an outlier"
> The_most_extreme_obs_is
[1] 72
> q
[1] 25.14942
> f
[1] 22.33503
> n
[1] 1000
> p
[1] 3
>
```

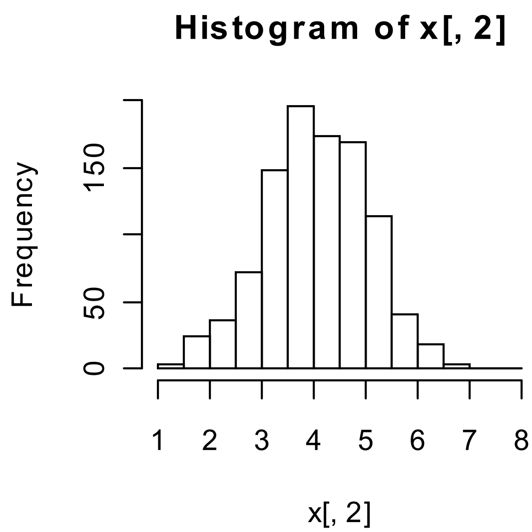
Deleting this observation (No 72), no more outliers.
See R-code below.
Histograms are approx bell-shaped. qq plots are approximately linear. Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling, Cramer-von Mises normality tests have p-values > 0.05.

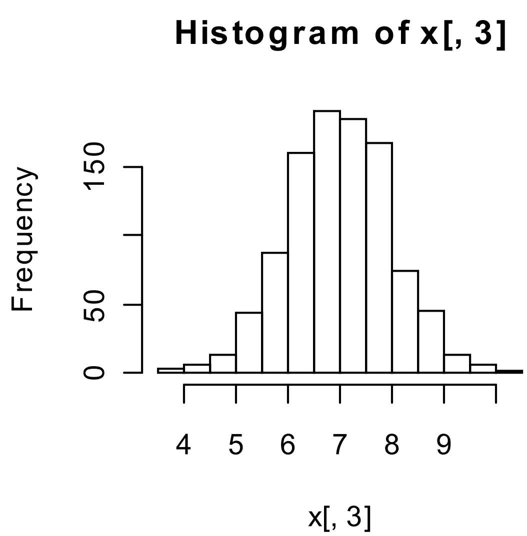
In summary statistics, skewness , kurtosis are approx. 0, also mean and median are approximately equal, all these suggesting univariate normality.
Scatter plot matrix shows elliptical shapes suggesting bivariate normality of components.

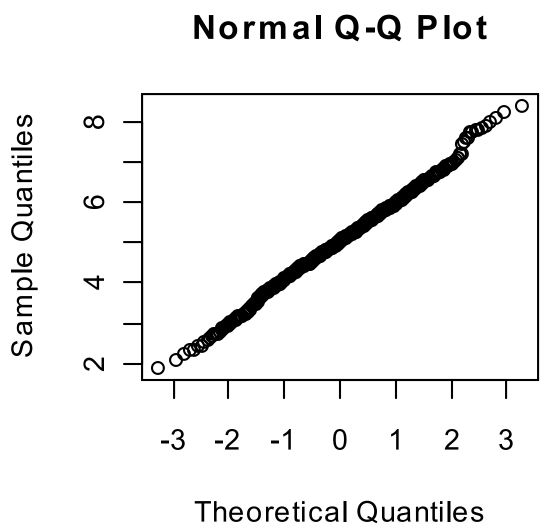
```
> # Detecting outliers in multivariate data
> # Example 3.2.1 p59
> alpha=0.05
> x=read.table('c:/MYDOCU~1/assign1dat3.txt', header=0)
> x=data.matrix(x)
> # If you only want to use some variables, e.g
> # if there are 4 variables and we want to use only
> # the last three varibles, add this command:
> x=x[,2:4]
>
> # If you want to remove one observation, say observation k, the add
> #the following lines
> rm=72
>  x=rbind(x[1:(rm-1),],x[(rm+1):n,])
> # rbind forms a matrix by putting the matrices one under the other
>
> xb=apply(x,2,mean) #mean vector of x
> p=ncol(x)
> n=nrow(x)
> xb = matrix(1,nrow=n, ncol=p)%*%diag(xb)
> s=t(x-xb)%*%(x-xb)/(n-1)
> d=(x-xb)%*%solve(s)%*%t(x-xb)
> d=c(diag(d))
> q=max(d)
> # to find which element of d is the max
> i=1
> while (d[i]<q) i=i+1
> The_most_extreme_obs_is=i
> ca=(p/(n-p-1))*qf(1-alpha/n,p,n-p-1)
> # note on p60 alpha/n instead of 1-alpha/n
> f=(ca/(1+ca))*(((n-1)**2)/n)
> ifelse(q > f, c( 'The most extreme obs is an outlier'), c( 'The most extreme obs is an not outlier'))
[1] "The most extreme obs is an not outlier"
> The_most_extreme_obs_is
[1] 676
> q
[1] 14.58173
> f
[1] 22.33277
> n
[1] 999
> p
[1] 3
```


V2
V3
V4
2 4 6
2 4 6 8
4 6 8
2 4 6
2 4 6 8

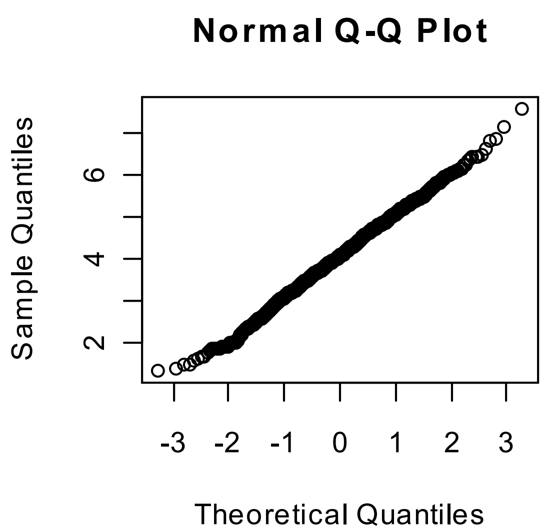

Histogram of x[, 1]
Frequency
0
50
150
2 3 4 5 6 7 8
x[, 1]


Histogram of x[, 2]
Frequency
0
50
150
1
2
3
4
5
6
7
8
x[, 2]

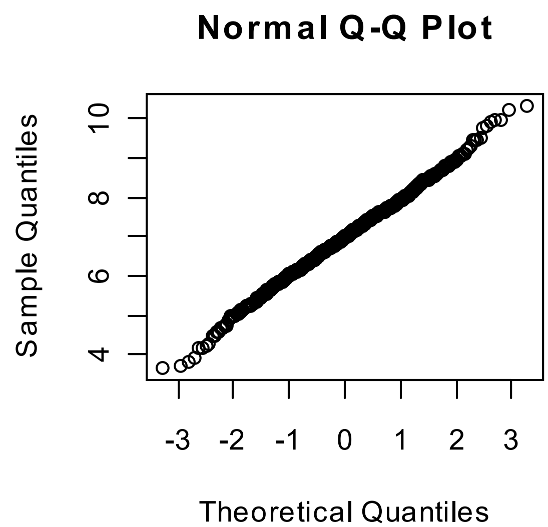

Histogram of x[, 3]
Frequency
0
50
150
4
5
6
7
8
9
x[, 3]



Q-Q plot for x1



Q-Q plot for x2



Q-Q plot for x3

```
> library(fBasics) # Download the package fBasics if you don't have it

Rmetrics, (C) 1999-2005, Diethelm Wuertz, GPL
fBasics: Markets, Basic Statistics, Date and Time
> basicStats(x[,1])
                   Value
nobs        9.990000e+02
NAs         0.000000e+00
Minimum     1.873596e+00
Maximum     8.396393e+00
1. Quartile 4.426514e+00
3. Quartile 5.674709e+00
Mean        5.031314e+00
Median      5.033883e+00
Sum         5.026283e+03
SE Mean     3.116183e-02
LCL Mean    4.970164e+00
UCL Mean    5.092465e+00
Variance    9.700887e-01
Stdev       9.849308e-01
Skewness    1.922805e-02
Kurtosis    3.358707e-01
> basicStats(x[,2])
                    Value
nobs         999.00000000
NAs            0.00000000
Minimum        1.30146700
Maximum        7.58141700
1. Quartile    3.40365500
3. Quartile    4.75213650
Mean           4.05859799
Median         4.04996800
Sum         4054.53938800
SE Mean        0.03160785
LCL Mean       3.99657251
UCL Mean       4.12062346
Variance       0.99805719
Stdev          0.99902812
Skewness      -0.04240620
Kurtosis      -0.09236178
> basicStats(x[,3])
                    Value
nobs         999.00000000
NAs            0.00000000
```

```
Minimum          3.63634400
Maximum         10.34301400
1. Quartile      6.30969700
3. Quartile      7.62500950
Mean             6.97812230
Median           6.98860300
Sum           6971.14418000
SE Mean          0.03172494
LCL Mean         6.91586705
UCL Mean         7.04037755
Variance         1.00546565
Stdev            1.00272910
Skewness        -0.02716786
Kurtosis         0.31698054
>
> ks.test(x[,1], "pnorm", m=mean(x[,1]), sd=sqrt(var(x[,1])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 1]
D = 0.0217, p-value = 0.7322
alternative hypothesis: two.sided

> shapiro.test(x[,1])

        Shapiro-Wilk normality test

data:  x[, 1]
W = 0.9972, p-value = 0.07829

> library(nortest) # Package for normality tests,
> # Download the package nortest if you don't have it
> ad.test(x[,1])

        Anderson-Darling normality test

data:  x[, 1]
A = 0.5623, p-value = 0.1455

> cvm.test(x[,1])

        Cramer-von Mises normality test

data:  x[, 1]
W = 0.0711, p-value = 0.27

> ks.test(x[,2], "pnorm", m=mean(x[,2]), sd=sqrt(var(x[,2])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 2]
D = 0.0208, p-value = 0.7829
alternative hypothesis: two.sided

Warning message:
cannot compute correct p-values with ties in: ks.test(x[, 2], "pnorm", m = mean(x[,
2]), sd = sqrt(var(x[,
> shapiro.test(x[,2])

        Shapiro-Wilk normality test

data:  x[, 2]
W = 0.9982, p-value = 0.3573

> ad.test(x[,2])

        Anderson-Darling normality test

data:  x[, 2]
A = 0.3361, p-value = 0.5056
```

```
> cvm.test(x[,2])

        Cramer-von Mises normality test

data:  x[, 2]
W = 0.0396, p-value = 0.6893

> ks.test(x[,3], "pnorm", m=mean(x[,3]), sd=sqrt(var(x[,3])))

        One-sample Kolmogorov-Smirnov test

data:  x[, 3]
D = 0.0183, p-value = 0.8928
alternative hypothesis: two.sided

> shapiro.test(x[,3])

        Shapiro-Wilk normality test

data:  x[, 3]
W = 0.998, p-value = 0.2818

> ad.test(x[,3])

        Anderson-Darling normality test

data:  x[, 3]
A = 0.3425, p-value = 0.491

> cvm.test(x[,3])

        Cramer-von Mises normality test

data:  x[, 3]
W = 0.0455, p-value = 0.5803

>
```