HW 2

Ran Zhang

1/23/2019

Question 1a & 1b Load the data in R and assign as data;

```
data <- read.table("https://www.stat.ncsu.edu/people/maity/courses/st537-S2019/data/T4-
3.DAT",header = F)
```

Remove the twi two outliers #9 and #16 rows;

```
new_data <- data[-c(9,16),]
```

Attach the new data;

```
attach(new_data)
```

Perform the Shapiro-Wilk tests;

```
apply(new_data[, 1:4], 2, shapiro.test)
```

```
## $V1
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98469, p-value = 0.9439
##
##
## $V2
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96636, p-value = 0.4871
##
##
## $V3
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96717, p-value = 0.507
##
##
## $V4
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96598, p-value = 0.4779
```

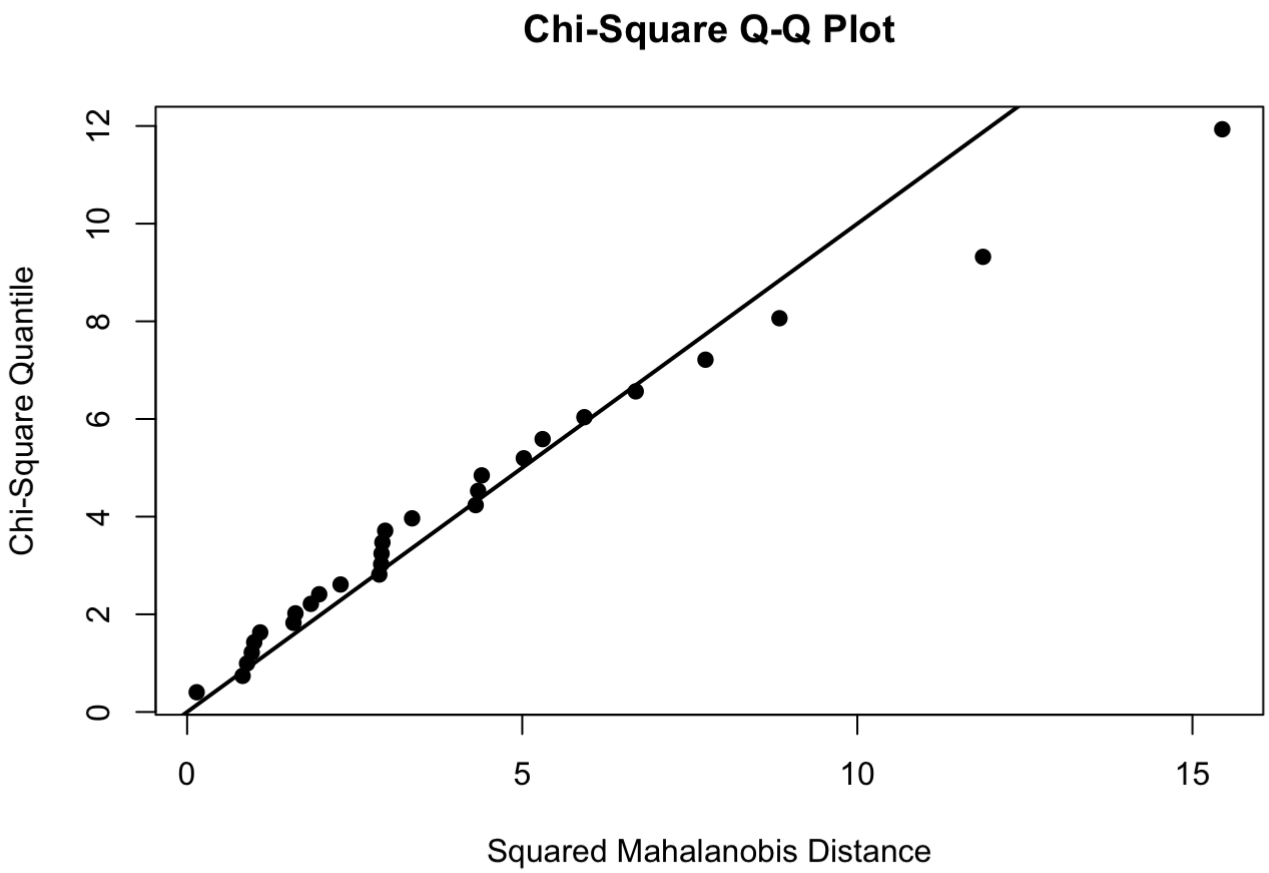
Load the MVN library;

```
library(MVN)
```

```
## sROC 0.1-2 loaded
```

Perform Roystpn’s tests on new data and create a chi-square plot;

```
mvn(new_data[, 1:4], mvnTest = "royston", multivariatePlot = "qq")
```



```
## $multivariateNormality
##      Test        H   p value MVN
## 1 Royston 1.098338 0.6271166 YES
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9847    0.9439    YES
## 2 Shapiro-Wilk    V2        0.9664    0.4871    YES
## 3 Shapiro-Wilk    V3        0.9672    0.5070    YES
## 4 Shapiro-Wilk    V4        0.9660    0.4779    YES
##
## $Descriptives
##     n     Mean  Std.Dev Median  Min  Max    25th    75th       Skew
## V1 28 1865.929 262.1619 1857.5 1325 2403 1711.50 2049.75 0.08994538
## V2 28 1697.964 244.8618 1663.0 1170 2301 1593.25 1847.50 0.39091767
## V3 28 1488.643 253.1536 1466.0 1002 2087 1307.25 1617.25 0.49661284
## V4 28 1710.250 277.9986 1674.5 1176 2234 1528.75 1876.25 0.25921958
##      Kurtosis
## V1 -0.5084972
## V2  0.1961808
## V3  0.0516768
## V4 -0.6484407
```

Question 1c Shaprio-Wilk test shows that all test p-value is far larger than the significance level, which represents all follow normal distribution without rejecting hypothesis test.

Royston’s test also show that the p-value is higher than the significane level, which proves that all 4 variables follow normal distribution.

As the most (above 95%) points are located in the eclipses pattern, it suggest that the distribution of normality.

Question 2a Load the library of HSAUR3;

```
library(HSAUR3)
```

```
## Loading required package: tools
```

Attach the data;

```
attach(skulls)
```

Extract the c4000BC epoch from the data;

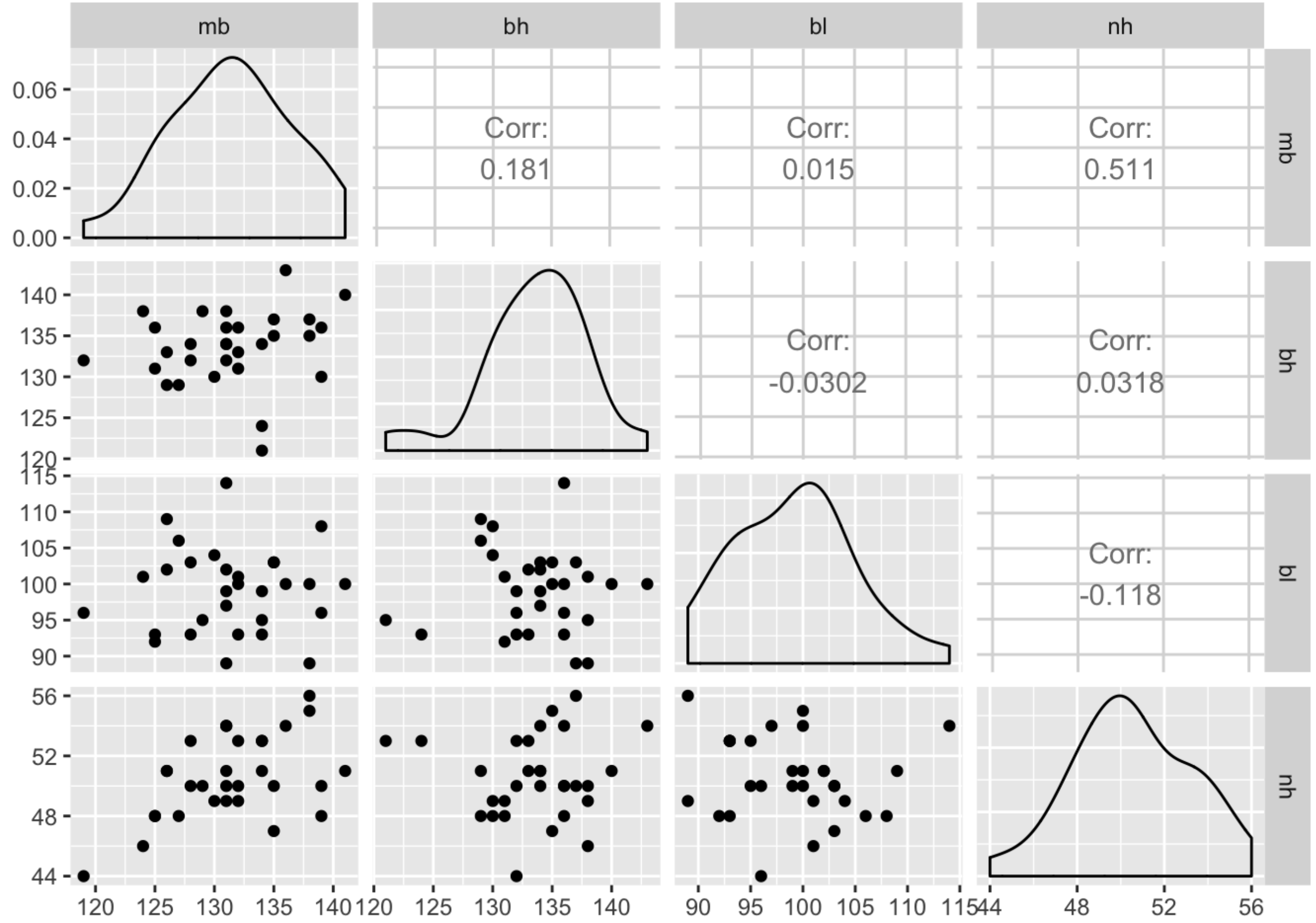
```
c4000BC <- skulls[epoch=="c4000BC",2:5]
c4000BC
```

	mb <dbl>	bh <dbl>	bl <dbl>	nh <dbl>
1	131	138	89	49
2	125	131	92	48
3	131	132	99	50
4	119	132	96	44
5	136	143	100	54
6	138	137	89	56
7	139	130	108	48
8	125	136	93	48
9	131	134	102	51
10	134	134	99	51

1-10 of 30 rows Previous **1** 2 3 Next

Create the pairs-plot;

```
library(ggplot2)
library(GGally)
ggpairs(c4000BC)
```



mb has positive correlations to the other variables, however, bl has a negative correlation to the nh. Besides, the mb and nh has a strong positive linear relationship here.

Question 2b x: data matrix;

```
x <- c4000BC
```

Number of variables;

```
p <- ncol(x)
```

sample size;

```
n <- nrow(x)
```

xbar and s;

```
xbar <- colMeans(x)
s <- cov(x)
```

Mahalanobis distance;

```
x.cen <- scale(x, center=T, scale =F)
d2 <- diag(x.cen %*% solve(s) %*% t(x.cen))
```

chi-square quantiles;

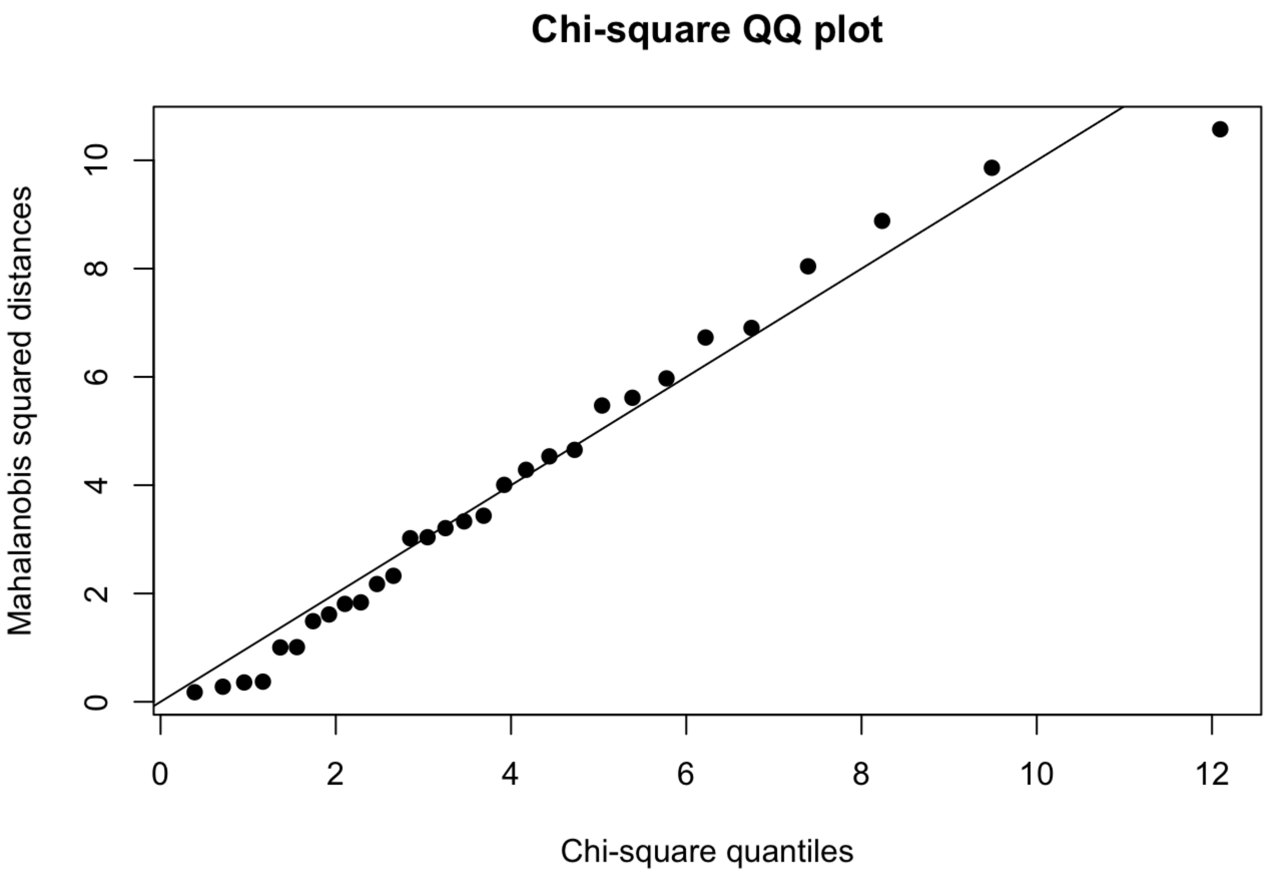
```
qchi <- qchisq((1:n - 0.5)/n, df=p)
```

sorted d^2 value;

```
sortd <- sort(d2)
```

Plot the chi-square and add the line;

```
plot(qchi, sortd, pch=19, xlab="Chi-square quantiles", ylab = "Mahalanobis squared dista
nces", main="Chi-square QQ plot")
abline(0,1)
```



As we draw the line of 45 degree, we found that almost all points are laid closely to the line, except the point in the right corner. As that point is not very influencial to the whole trendline, so we can still think the variables are normally distributed.

Question 2c Calculate the z scores for each variable;

```
z1 <- scale(c4000BC[,1],scale=T)
z2 <- scale(c4000BC[,2],scale=T)
z3 <- scale(c4000BC[,3],scale=T)
z4 <- scale(c4000BC[,4],scale=T)
```

Attach the Mahalanobis distances to the z score table;

```
c4000BC_1 <- cbind(z1,z2,z3,z4,d2)
colnames(c4000BC_1) <- c("z1","z2","z3","z4","d2")
```

Sort by d2;

```
c4000BC_sort <- c4000BC_1[order(d2 ,decreasing=TRUE), ]
```

Look for the first two rows of data;

```
head(c4000BC_sort,2)
```

```
##              z1         z2         z3        z4        d2
## 29 -0.07148545  0.5370268  2.5207795 1.2544602 10.573099
## 12  0.51339549 -2.8193907 -0.7080841 0.8925967  9.861683
```

For this sorting data, the first two data are 29 and 12. For data 29, z1 and z2 are good, z3 is a bit of far but z4 is extremely far from the original point. For data 12, z1, z3 and z4 are good, but the z2 is relatively far from the original point.

Question 2d Perform the univariate Shapiro-Wilk test;

```
apply(c4000BC[,1:4], 2, shapiro.test)
```

```
## $mb
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98136, p-value = 0.8603
##
##
## $bh
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.95664, p-value = 0.2536
##
##
## $bl
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.97314, p-value = 0.6282
##
##
## $nh
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.97481, p-value = 0.6772
```

Perform the Royston tests;

```
mvn(c4000BC[,1:4], mvnTest = "royston")
```

```
## $multivariateNormality
##      Test        H  p value MVN
## 1 Royston 2.752767 0.603866 YES
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    mb        0.9814    0.8603    YES
## 2 Shapiro-Wilk    bh        0.9566    0.2536    YES
## 3 Shapiro-Wilk    bl        0.9731    0.6282    YES
## 4 Shapiro-Wilk    nh        0.9748    0.6772    YES
##
## $Descriptives
##     n      Mean  Std.Dev Median Min Max   25th   75th        Skew
## mb 30 131.36667 5.129249    131 119 141 128.00 134.75 -0.16642216
## bh 30 133.60000 4.469051    134 121 143 131.25 136.00 -0.64720446
## bl 30  99.16667 5.884423    100  89 114  95.00 102.75  0.31717217
## nh 30  50.53333 2.763473     50  44  56  49.00  53.00 -0.08670975
##      Kurtosis
## mb -0.4879548
## bh  0.8488047
## bl -0.2756768
## nh -0.4538837
```

Question 3a Load the library;

```
library(mnormt)
```

Generate 100 points and named as z_1;

```
data3 <- rmnorm(100, mean=rep(1,2), varcov=cbind(c(1,1),c(1,2)))
data3
```

```
##              [,1]        [,2]
##  [1,]  0.43430773  1.79040013
##  [2,]  0.60044104 -0.30333007
##  [3,]  1.00886159  1.43672859
##  [4,]  0.03971547 -0.78248181
##  [5,]  1.65553642  2.60457853
##  [6,] -0.21969737  0.44427293
##  [7,] -0.38811623  0.16987002
##  [8,]  0.73543903  1.60913688
##  [9,]  0.80335398  0.52016191
## [10,] -0.94541542 -0.16482017
## [11,]  0.39337877 -1.05263814
## [12,]  1.50542238  2.61911011
## [13,]  1.15397979  1.46270127
## [14,] -0.09781131  0.96590251
## [15,]  0.04647084  1.18519298
## [16,]  1.11898287  1.00783272
## [17,]  1.74157424  3.88576585
## [18,]  0.26697753 -0.15804803
## [19,] -0.33130638 -2.72852704
## [20,]  1.58998559  0.49868537
## [21,]  0.31531514 -0.17630466
## [22,]  1.18046185  0.72039988
## [23,] -0.24021425 -0.68558953
## [24,]  1.43553813  1.49962841
## [25,]  1.59004845  1.55898953
## [26,]  2.32735015  2.92879983
## [27,] -0.37494686 -0.36240324
## [28,]  0.69184260  0.71440172
## [29,]  1.65498929  2.22107577
## [30,]  1.17362705  0.18688345
## [31,]  1.72449608  1.26325169
## [32,] -0.76300673 -1.21269417
## [33,]  2.18035048  2.21459730
## [34,] -0.16709626  1.75403291
## [35,]  0.65069716  0.69281102
## [36,]  0.59434368  0.69024791
## [37,]  0.13586939  0.57509192
## [38,]  2.99404720  0.60374793
## [39,]  2.81453791  2.46385160
## [40,]  1.09163536  1.94623339
## [41,]  2.06899715  1.24486976
## [42,]  2.10082197  2.22919344
## [43,]  0.26400196  0.79487795
## [44,]  0.54557665  2.08706117
## [45,]  2.17616794  1.51847130
## [46,]  2.18976744  4.37885789
## [47,]  1.84305246  1.71627791
## [48,]  1.99583974  4.40564296
## [49,]  1.40725276  0.88992723
## [50,]  1.52342951  1.64969731
## [51,]  3.55819593  1.17490518
## [52,]  2.35189559  4.59032626
##  [53,]  0.09726990  0.64102105
##  [54,] -0.03250759  0.94317115
##  [55,]  2.00830366  1.63851997
##  [56,]  1.22698921  1.29965203
##  [57,]  2.35242285  2.11339977
##  [58,]  1.93620023  2.04358806
##  [59,] -0.15985931  0.97859391
##  [60,]  1.62520096  1.46422831
##  [61,]  0.68109466  0.21949378
##  [62,]  0.52956931 -0.95339210
##  [63,]  1.09063733  1.07419032
##  [64,]  1.31580791  0.08215835
##  [65,]  1.90426622  1.54547391
##  [66,]  0.31994915  1.01846083
##  [67,]  0.78048977  0.24765232
##  [68,]  0.85842133  1.51138403
##  [69,]  1.92348365  1.73860022
##  [70,]  1.22190459  1.94497974
##  [71,]  0.03296801  1.17021856
##  [72,]  2.20906368  2.26739672
##  [73,]  0.02654688  0.46987155
##  [74,]  0.84481076  1.86482555
##  [75,] -0.54635779 -0.59663283
##  [76,]  2.04739985  1.15754601
##  [77,]  0.15041638 -0.33847299
##  [78,]  2.06909377  0.77473209
##  [79,]  2.98567876  3.30771363
##  [80,]  2.05878223  1.24726729
##  [81,]  0.21501922 -0.67521897
##  [82,]  0.68324170 -0.51918291
##  [83,]  0.15812222 -0.47232293
##  [84,]  1.85947720  4.12864636
##  [85,]  1.62949287  1.60269094
##  [86,] -0.80042110 -1.43440484
##  [87,]  0.43589356  0.19831476
##  [88,]  0.21591413  0.06962816
##  [89,] -0.52261780 -0.53440060
##  [90,] -0.60451989  1.13355074
##  [91,]  1.58352274  0.71428173
##  [92,]  1.31476461 -0.64459560
##  [93,]  0.92728979  1.29681172
##  [94,] -0.80696922 -0.96941787
##  [95,]  1.50170933  1.51517041
##  [96,]  1.04872155 -0.13038910
##  [97,]  0.34988796  0.65523991
##  [98,] -0.02208992  0.38539593
##  [99,]  1.60347434  1.65680741
## [100,] -0.14973653 -1.26413408
```

Question 3b Load the library;

```
library(car)
```

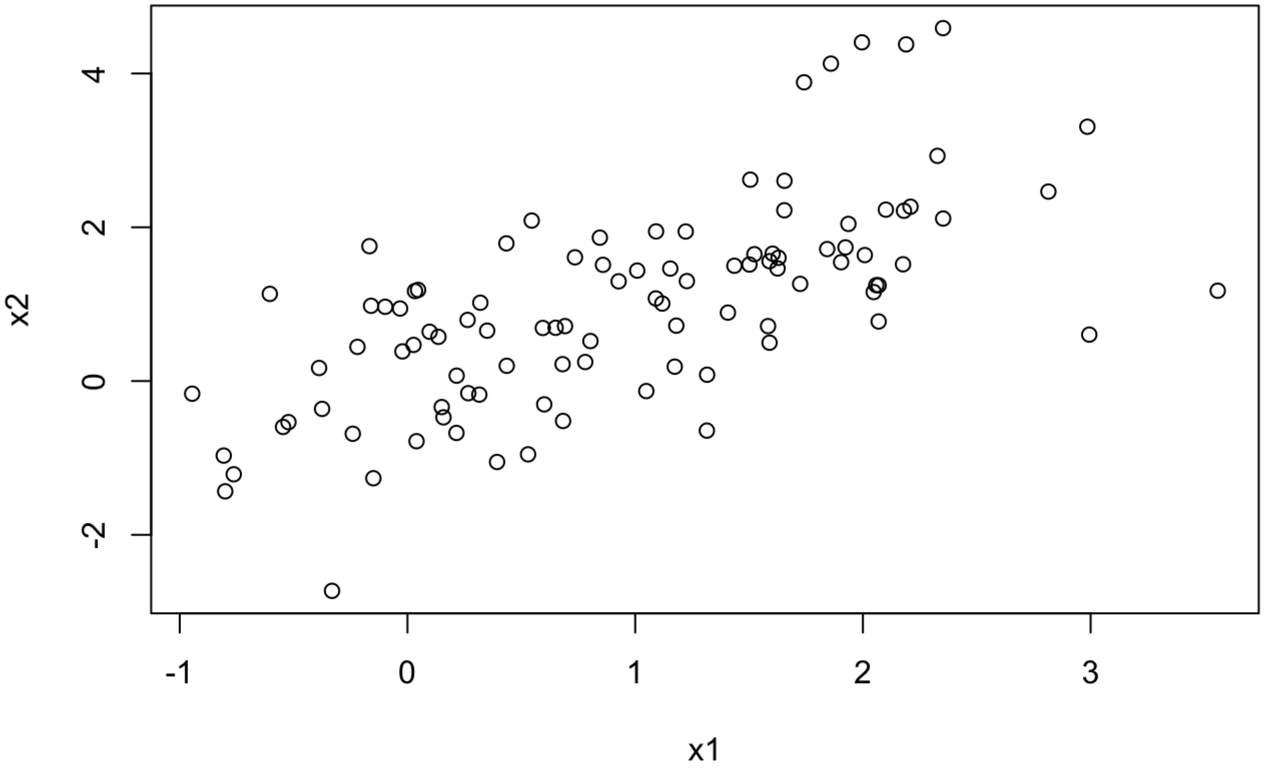
```
## Loading required package: carData
```

Assign the value as x1 and x2;

```
x1 <- data3[,1]
x2 <- data3[,2]
```

Make a scatter plot of x1 and x2;

```
plot(x1,x2)
```



Overlay data ellipse (50%, 95%);

```
dataEllipse(x1,x2,xlim=c(-2,5), ylim=c(-3,8), pch=20, col=c("red", "green"), ellipse.lab
el = c(0.5,0.95), levels=c(0.5,0.95), fill=TRUE, fill.aplpha=0.1)
```

```
## Warning in plot.window(...): "fill.aplpha" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "fill.aplpha" is not a graphical
## parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "fill.aplpha"
## is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "fill.aplpha"
## is not a graphical parameter
```

```
## Warning in box(...): "fill.aplpha" is not a graphical parameter
```

```
## Warning in title(...): "fill.aplpha" is not a graphical parameter
```

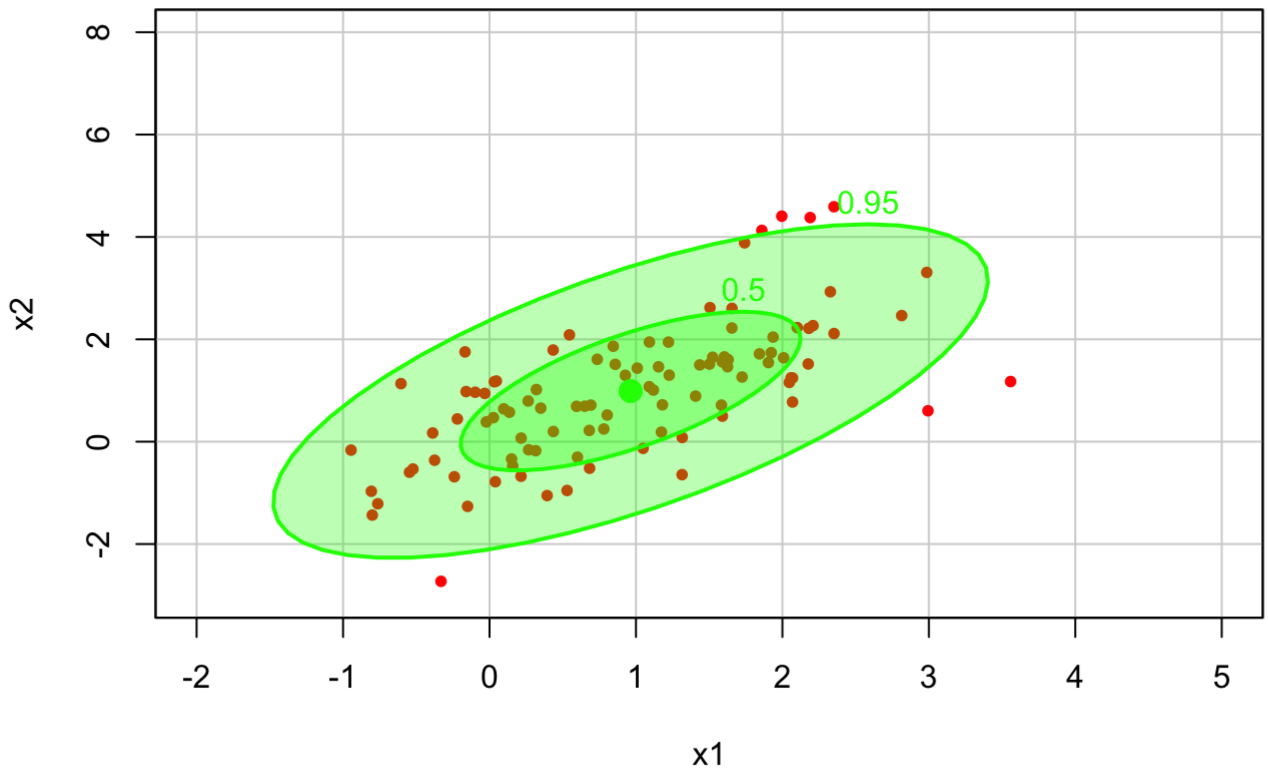
```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "fill.aplpha" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "fill.aplpha" is not
## a graphical parameter
```

```
## Warning in text.default(x, y, label, adj = adj, col = col, ...):
## "fill.aplpha" is not a graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "fill.aplpha" is not
## a graphical parameter
```

```
## Warning in text.default(x, y, label, adj = adj, col = col, ...):
## "fill.aplpha" is not a graphical parameter
```



It is an ellipse pattern;

Question 3c Y = X1 + X2 The mean of Y is the sum of expect value of X1 and X2

```
miu_Y <- 1 + 2
miu_Y
```

```
## [1] 3
```

The variance of Y is the sum of variance of X1, X2 and the double covariance;

```
Var_Y <- 1 + 2 + 1*2
Var_Y
```

```
## [1] 5
```

Y is follows the normal distribution with mean of 3 and variance of 5, Y ~ N(3,5)

Question 3d As the sample used on part a, the sample y is;

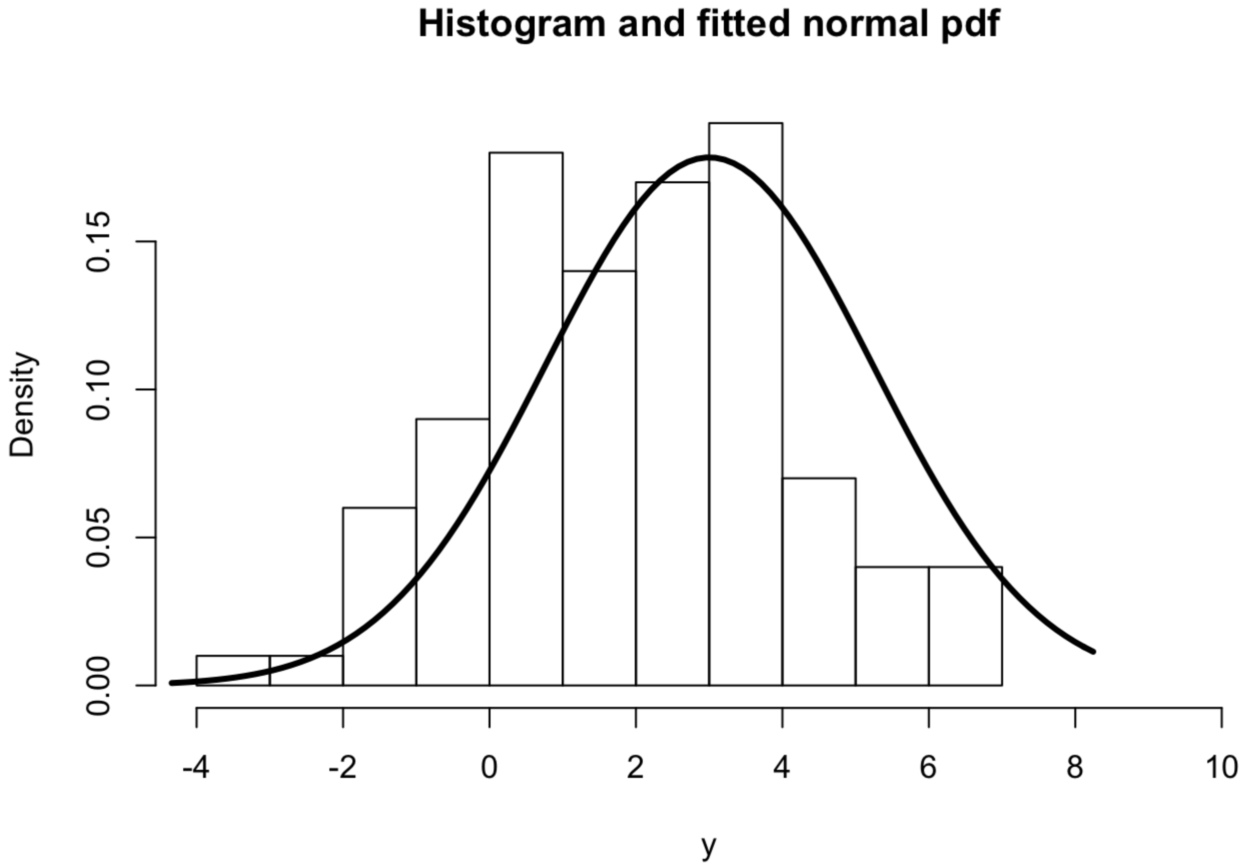
```
y <- x1 + x2
```

Sample mean and standard deviation of y;

```
xbar <- mean(y)
std <- sd(y)
```

Plot a hist and overlay by a normal dis;

```
xx <- seq(xbar-3*std, xbar+3*std, len = 101)
yy <- dnorm(xx, mean = 3, sd = sqrt(5))
hist(y, probability = TRUE, xlim = range(-4,10), main="Histogram and fitted normal pdf")
lines(xx, yy, lwd=3)
```



I am expecting to see a normal distribution with a bell shape, and the curve mostly matches my expectation but not exactly match. However, the sample distribution is not exactly same as the estimate because of the random sampling distribution. Samples are selected randomly, so it could be not exactly as the estimate distribution from the population.

Question 4a See attachment;

Question 4b Generate 100 random points;

```
data4 <- rmnorm(100, mean=c(1,2,3), varcov=cbind(c(2,1,1),c(1,1,1),c(1,1,3)))
```

Compute the sample correlation matrix

```
cor1 <- cor(data4)
cor1
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.6379390 0.3848404
## [2,] 0.6379390 1.0000000 0.5313038
## [3,] 0.3848404 0.5313038 1.0000000
```

Find the Z_1, Z_2 and Z_3;

```
Z_1 <- sapply(data4[,1], function(x) (x-1)/sqrt(2))
Z_2 <- sapply(data4[,2], function(x) (x-2)/sqrt(1))
Z_3 <- sapply(data4[,3], function(x) (x-3)/sqrt(3))
```

Compute the covariance matrix;

```
cov2 <- cov(cbind(Z_1,Z_2,Z_3))
cov2
```

```
##           Z_1       Z_2       Z_3
## Z_1 0.9064848 0.5979936 0.3809422
## Z_2 0.5979936 0.9693360 0.5438488
## Z_3 0.3809422 0.5438488 1.0809267
```

Compare the covriance and correlation;

```
identical(cor1, cov2)
```

```
## [1] FALSE
```

They are close but different, as the samples are picked from the population randomly, random samples give different data cannot be idealy exact match the thorey. However, as the simulation times increased, the result of them should be closer and closer.