DiFrank, Annie

Course#: ST537

HW#:1

Submission Date: 01/16/25

St537_HW1

2025-01-16

Problem 1

Part 1: Defining data

```
#define mean vector for our data (mu = E(X), our random vector)
mu <- c(4,3,2,1)
mu

## [1] 4 3 2 1

#define the variance-covariance matrix, which quantifies the variability of
the random vector
sigma <- matrix(c(2,0,2,1,
                0,1,1,0,
                2,1,8,-2,
                1,0,-2,3),
                nrow=4, byrow = TRUE)
sigma

##      [,1] [,2] [,3] [,4]
## [1,]    2    0    2    1
## [2,]    0    1    1    0
## [3,]    2    1    8   -2
## [4,]    1    0   -2    3

#define the matrix A, the coefficient matrix / transformation matrix
A <- matrix(c(1,2,0,0,
              2,1,0,0,
              0,0,1,-2,
              0,0,2,-1),
            nrow=4, byrow = TRUE)
A

##      [,1] [,2] [,3] [,4]
## [1,]    1    2    0    0
## [2,]    2    1    0    0
## [3,]    0    0    1   -2
## [4,]    0    0    2   -1
```

Part 2: Answering a-d

(a) **Find E(X) and E(AX)**

```
#E(X) is simply mu. It summarizes the central tendancy of X.
E_X <- mu
E_X

## [1] 4 3 2 1
```

```
#E(AX) is computed as A*mu, using matrix multiplication (inner product).
E(AX) is the expected value of a transformed random vector AX (X is
transformed by matrix A)
E_AX <- A %*% mu
E_AX

##      [,1]
## [1,]   10
## [2,]   11
## [3,]    0
## [4,]    3
```

As A is a 4x4 matrix and mu is a 4x1 matrix, we see the expected dimensions of 4x1.

(b) **Find cov(AX)**

The covariance of AX Is computed as: cov(AX)=AΣA^T (where sigma is the cov(X), the variance-covariance matrix of X). This matrix will show how the transformation matrix A affects the variability structure of X. We know that E(AX)=A·E(X), allowing us to use this simplified cov(AX) equation.

```
cov_AX <- A %*% sigma %*% t(A)
cov_AX

##      [,1] [,2] [,3] [,4]
## [1,]    6    6    2    7
## [2,]    6    9    1    8
## [3,]    2    1   28   32
## [4,]    7    8   32   43
```

(c) **Suppose we have a random sample of size n = 10 from a distribution with mean vector μ, and the variance-covariance matrix Σ as given above. Find the variance-covariance matrix of the sample mean vector.**

The variance-covariance matrix of the sample mean vector is: cov(Xbar)= Σ/n. This quantifies the variability and relationships between the components of the sample mean. As n increases, cov(Xbar) decreases, as larger samples lead to more precise estimates of the population mean. It tells us how reliable the sample mean vector Xbar is as an estimator of the true mean vector mu.

```
n= 10
cov_sample_mean <- sigma/n
cov_sample_mean

##      [,1] [,2] [,3] [,4]
## [1,]  0.2  0.0  0.2  0.1
## [2,]  0.0  0.1  0.1  0.0
## [3,]  0.2  0.1  0.8 -0.2
## [4,]  0.1  0.0 -0.2  0.3
```

(d) **Suppose now that X follows a multivariate normal distribution with mean vector μ, and the variance- covariance matrix Σ, as given above. Find the distribution of the sample mean vector.**

If X follows a multivar normal distribution, then the sample mean vector also does. so the mean of Xbar is just mu. The variance-covariance matrix is scaled down by n. We calculated this in step 4.

```
#mean vector of the sample mean
mu

## [1] 4 3 2 1

dist_samplemean_sigma <- cov_sample_mean

#distribution of the sample mean vector:
mu

## [1] 4 3 2 1

dist_samplemean_sigma

##      [,1] [,2] [,3] [,4]
## [1,]  0.2  0.0  0.2  0.1
## [2,]  0.0  0.1  0.1  0.0
## [3,]  0.2  0.1  0.8 -0.2
## [4,]  0.1  0.0 -0.2  0.3
```

Problem 2

```
#install.packages("HSAUR3")
library("HSAUR3")

## Warning: package 'HSAUR3' was built under R version 4.3.3

#?skulls
head(skulls)

##     epoch  mb  bh  bl nh
## 1 c4000BC 131 138  89 49
## 2 c4000BC 125 131  92 48
## 3 c4000BC 131 132  99 50
## 4 c4000BC 119 132  96 44
## 5 c4000BC 136 143 100 54
## 6 c4000BC 138 137  89 56

skulls <- skulls
```

(a) **Suppose we want to estimate the population mean vector of all the 4 variable for skulls with epoch c3300BC. Write down the population, parameter,the sample and the statistic you will use to answer the question above.**

Population: All skull measurements for observations that have epoch = c3300BC

Parameter:

```
μ = [μmb
    μbh
    μbl
    μnh]
#This is the mean vector, E(X). The parameter is a 4x1 vector
```

Sample: The subset of skull measurements corresponding to epoch c3300BC. This is 30 observations.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

sample <- skulls %>% filter(epoch == "c3300BC")
```

Statistic: the sample mean vector, an estimate of the population mean vector. This can be calculated using colMeans.

```
Xbar = [Xbar_mb
        Xbar_bh
        Xbar_bl
        Xbar_nh]
```

(b) From the skulls data set, provide an estimator and the corresponding estimate (numeric value) of the parameter mentioned above.

```
#Xbar is the estimator
sample_mean_vector <- colMeans(sample[, c("mb","bh","bl","nh")])

sample_mean_vector

##        mb        bh        bl        nh
## 132.36667 132.70000  99.06667  50.23333
```

(c) Now suppose we want to estimate the population variance-covariance matrix. Which estimator will you use? Also provide a numeric estimate.

The estimator is the sample variance-covariance matrix, S.

```
S<- cov(sample[,2:5]) #just shorthand to select the cols

S
```

```
##            mb        bh         bl        nh
## mb 23.136782  1.010345  4.7678161 1.8425287
## bh  1.010345 21.596552  3.3655172 5.6241379
## bl  4.767816  3.365517 18.8919540 0.1908046
## nh  1.842529  5.624138  0.1908046 8.7367816
```

(d**) Compute the variance-covariance matrix of the estimator of the population mean obtained in part b.**

The variance-covar. matrix of the sample mean vector Xbar is cov(Xbar) = S/n

```
n= nrow(sample)
cov_sample_mean_vector <- S/n
cov_sample_mean_vector
```

```
##             mb         bh          bl          nh
## mb 0.77122605 0.03367816 0.158927203 0.061417625
## bh 0.03367816 0.71988506 0.112183908 0.187471264
## bl 0.15892720 0.11218391 0.629731801 0.006360153
## nh 0.06141762 0.18747126 0.006360153 0.291226054
```

(e) Provide an estimate for the parameter vector (μmb − μnh, μbh − μnh)T and obtain the estimated covariance matrix of the estimator.

The vector of contrasts can be written as

```
Aμ = (1 0 0 -1
      0 1 0 -1) multiplied by the mean vector:
     (μmb
     μbh
     μbl
     μnh)

This is a 2x4 matrix times a 4x1 matrix. Result will be a 2x1 matrix.
```

Numeric estimate:

```
#define coefficient/transformation matrix A
A <- matrix(c(1,0,0,-1,
              0,1,0,-1),
            nrow = 2, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0   -1
## [2,]    0    1    0   -1
```

```
#estimate A\mu by A Xbar
A %*% sample_mean_vector
```

```
##          [,1]
## [1,] 82.13333
## [2,] 82.46667
```

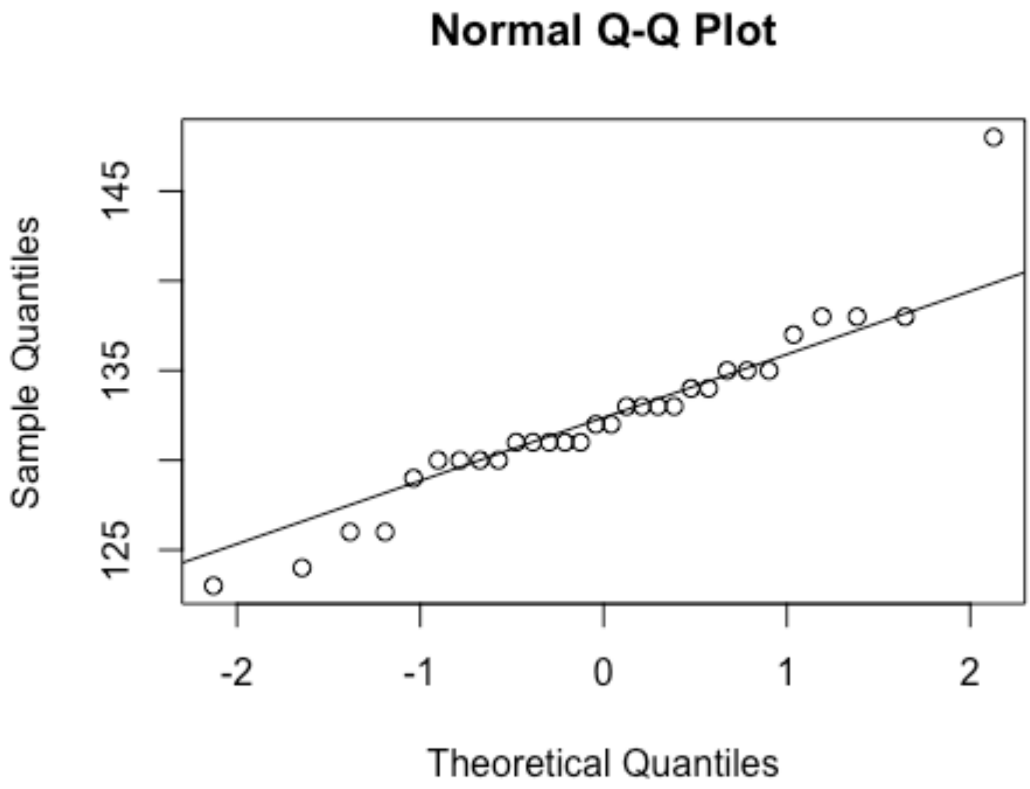
```
#estimate the variance-covariance matrix
A %*% (S/n) %*% t(A)
```

```
##            [,1]       [,2]
## [1,] 0.93961686 0.07601533
## [2,] 0.07601533 0.63616858
```

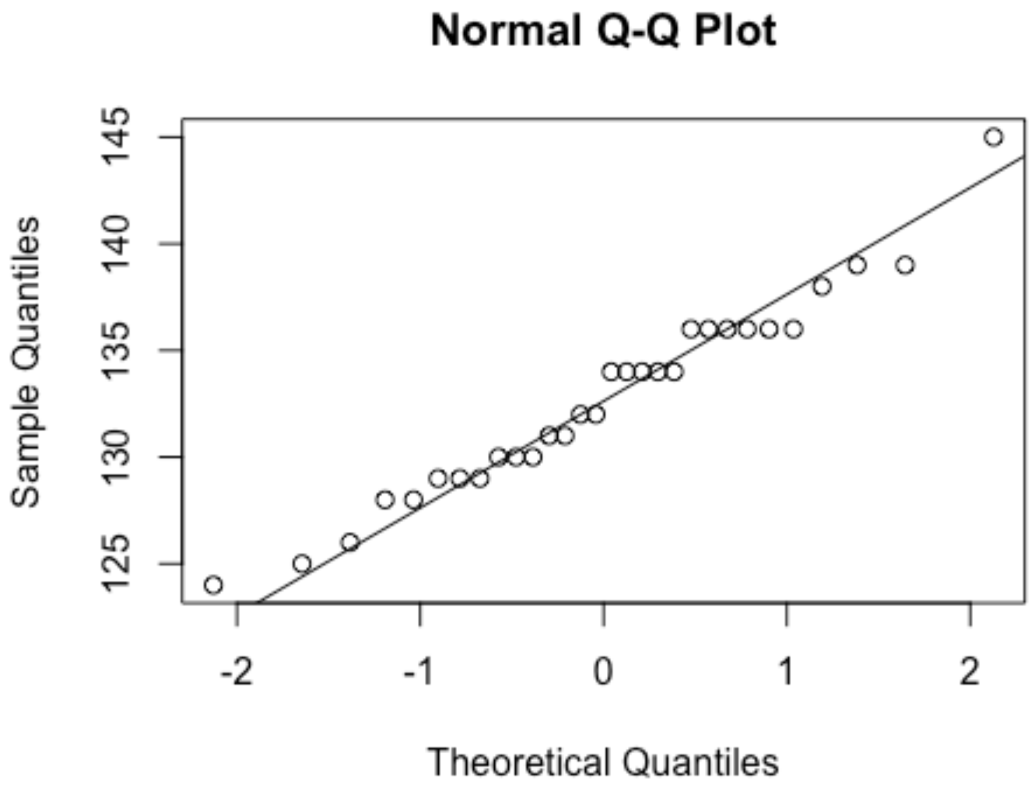
Problem 3

(a) Create normal Q-Q plots for the individual variables. Do you see any unusual patterns?

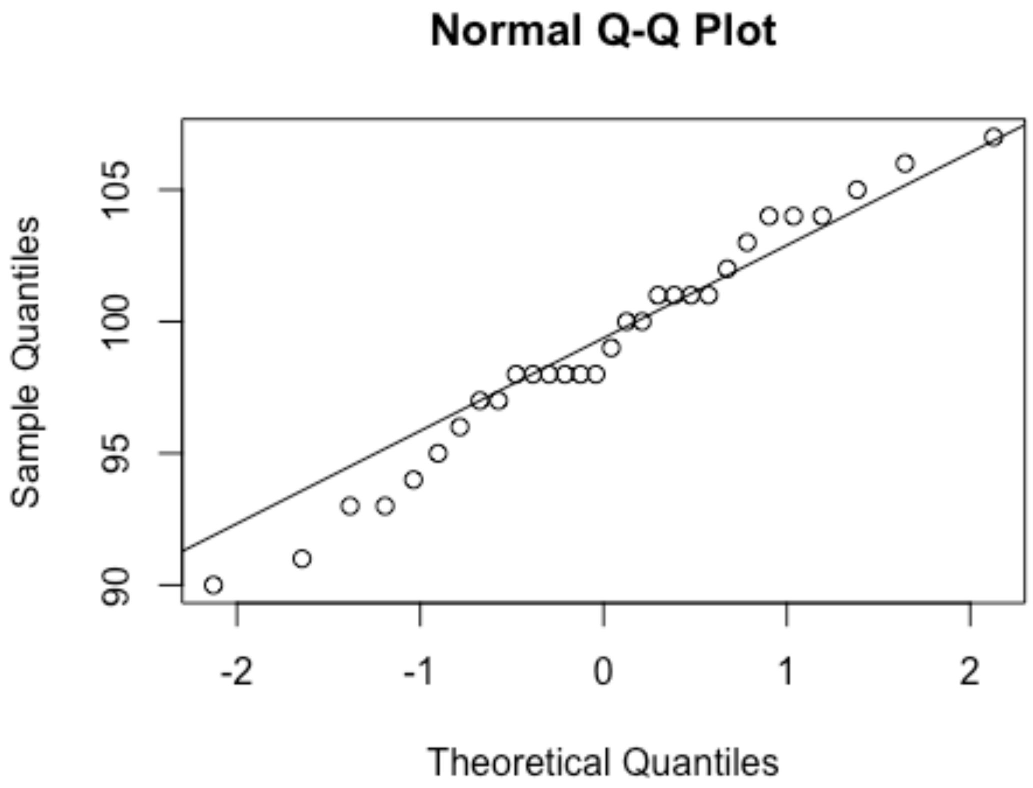
```
qqnorm(sample$mb)
qqline(sample$mb)
```



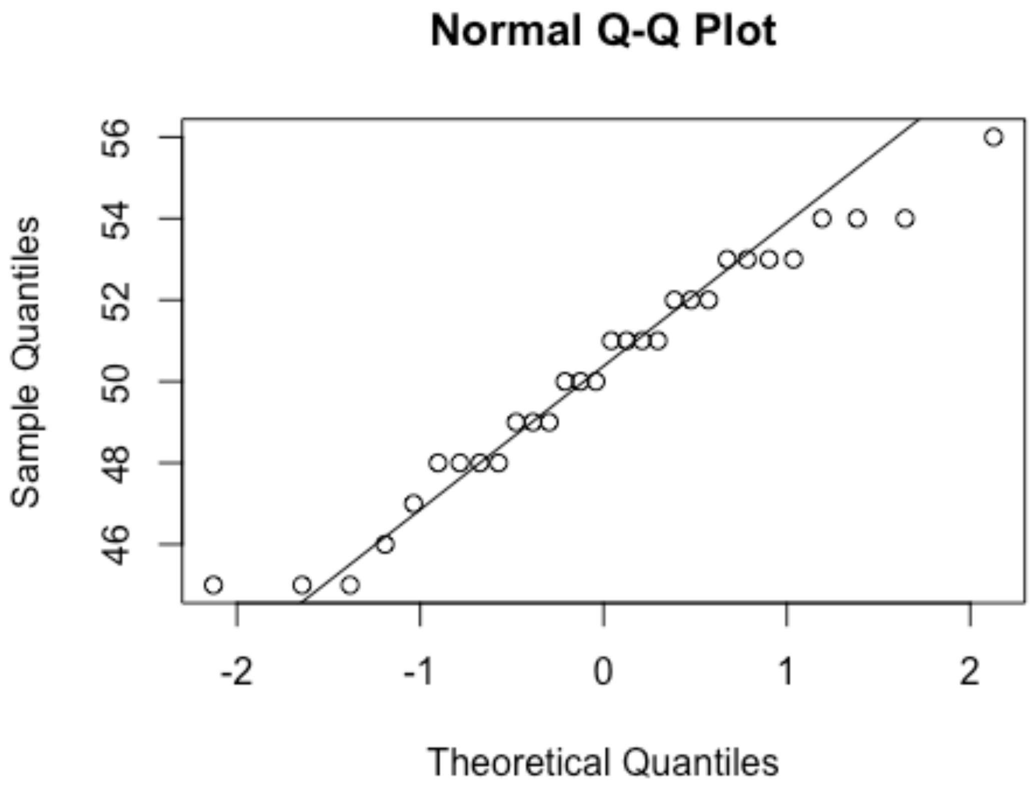
```
#mb looks to have a  a high outlier

qqnorm(sample$bh)
qqline(sample$bh)
```



```r
#only minor departures from normality

qqnorm(sample$bl)
qqline(sample$bl)
```



```
#pretty normal but looks a bit skewed toward the bottom left end

qqnorm(sample$nh)
qqline(sample$nh)
```



```
#This variable seems to have a low and high outlier.
```

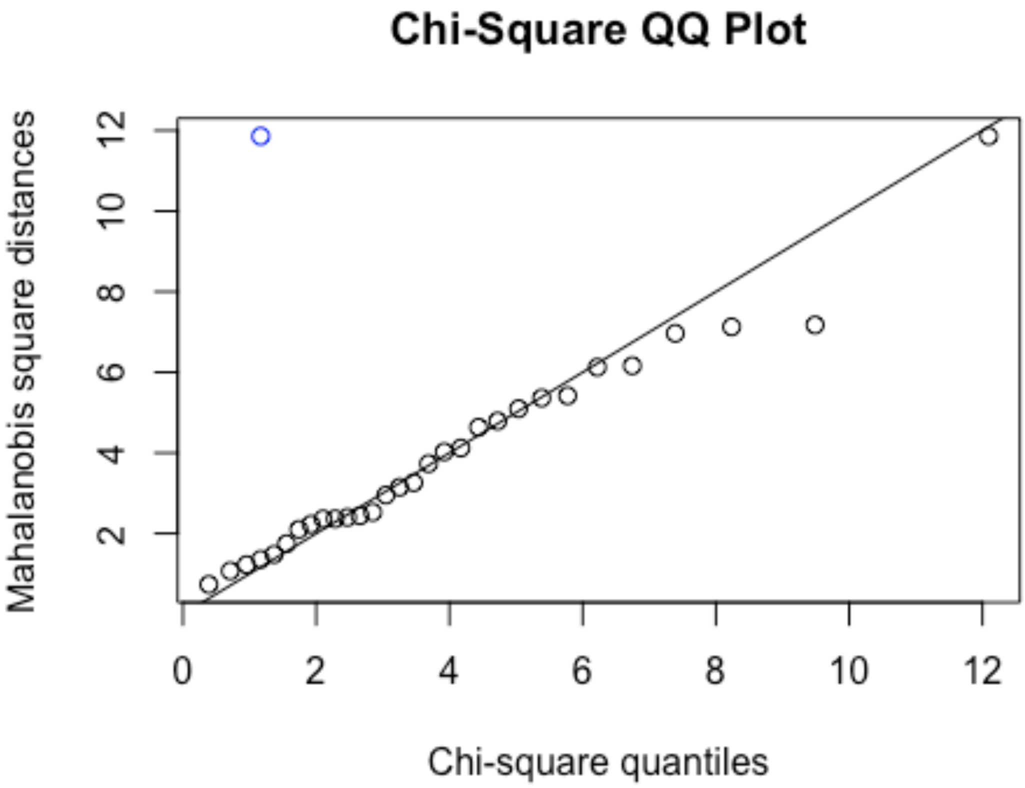
Overall, the plots are reasonably normal.

(b) create a chi-squared plot and comment on whether assumption of normality is reasonable.

Unfortunately your function you wrote in the class notes didn’t work for me for some reason.

```
#compute mahalanobis distances

#already computed covariance matrix S
#already computed mean vector sample_mean_vector
mahal_dists <- mahalanobis(sample[,2:5],center = sample_mean_vector, cov = S)

#create chi2 plot
qqplot(
  qchisq(ppoints(nrow(sample)), df = 4),
  mahal_dists,
  main = "Chi-Square QQ Plot",
  xlab = "Chi-square quantiles",
  ylab = "Mahalanobis square distances"
)
```

```r
abline(0, 1)

#potential outliers
threshold <- qchisq(0.975, df = 4)
outliers <- which(mahal_dists >threshold)

#circle the outliers
chisq_quantiles <- qchisq(ppoints(nrow(sample)), df = 4)
points(chisq_quantiles[outliers], mahal_dists[outliers],col="blue",pch = 21)
```



It looks like only one outlier from the 97.5% threshold. So for the most part there is a linear pattern/ suggesting normal distribution.

(c) Create a new dataset where you add the z-scores for each variable as well as the sample Mahalanobis distances as columns. Consider the two data points with highest distance values (these are not necessarily outliers), and comment on their z-scores.

```r
#Creating new dataset
sample$z <- scale(sample[,2:5])
sample$mahal_dist <- mahal_dists

#view 2 points with highest distances
```

```
pts<- sample[order(-sample$mahal_dist), ][1:2, ]
pts
```

```
##      epoch  mb  bh  bl nh       z.mb       z.bh       z.bl       z.nh
## 34 c3300BC 148 129 104 51  3.2501254 -0.7961768  1.1350161  0.2593766
## 47 c3300BC 138 129 107 53  1.1711539 -0.7961768  1.8252286  0.9360113
##    mahal_dist
## 34  11.858586
## 47   7.174712
```

The top distance point, around 11.9, has a high Z score for mb (3.25) which seems to skew it/ make it such a high distance because the remaining z scores for the other 3 measurements aren't crazy. The second highest distance point, around 7.2, has Z scores around within 1 standard deviation except the bl variable has around a 1.8 z score.

(d) Apply Shapiro-Wlk test on each of the variables and comment on the results.

```
shapiro.test(sample$mb)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  sample$mb
## W = 0.92675, p-value = 0.04029
```

```
shapiro.test(sample$bh)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  sample$bh
## W = 0.97146, p-value = 0.5799
```

```
shapiro.test(sample$bl)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  sample$bl
## W = 0.97728, p-value = 0.7495
```

```
shapiro.test(sample$nh)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  sample$nh
## W = 0.96335, p-value = 0.3762
```

The variables bh, bl, and nh have a non-significant p-value, meaning we can't deny the null that they full a normal distribution. The variable mb does have a significant (<0.05) p-value, giving evidence to suggest the distribution is not normal for this variable.

Problem 4

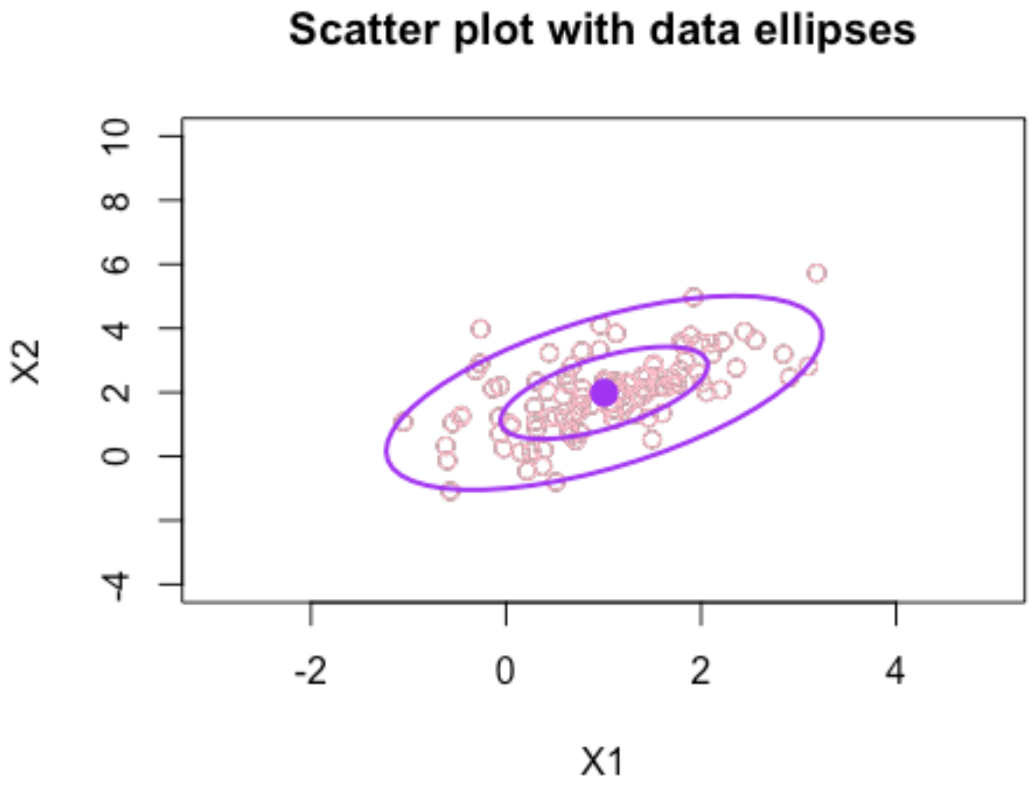
Perform the following tasks related to bivariate normal distribution. You need to install the R package mnormt, and then load the package by calling library(mnormt) command. The function rmnorm() in the mnormt library generates random samples from a multivariate normal. Use ?rmnorm() to open the documentation and read how to use it.

(a) Generate 100 data points from a bivariate normal distribution such that E(X1) = 1, E(X2) = 2, var(X1) = 1, var(X2) = 2 and cov(X1,X2) = 1. [Hint: you need to write the mean vector μ and covariance matrix Σ to use them in the R function.]

```r
#install.packages("mnormt")
library(mnormt)

#mean vector
mu <- c(1,2)
#cov matrix
sigma <- matrix(c(1,1,1,2), nrow = 2)
#generate random data points
set.seed(123)
data <- rmnorm(100,mean = mu, varcov = sigma)
X1 <- data[,1]
X2 <- data[,2]
```

(b) Create a scatter plot and overlap data ellipses (50% and 95%). What pattern would you expect to see in the scatterplot?

```r
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

plot(
  X1,
  X2,
  xlab = "X1",
  ylab = "X2",
  main = "Scatter plot with data ellipses",
  ylim = c(-4, 10),
  xlim = c(-3, 5)
)
dataEllipse(
  X1,
  X2,
  levels = c(0.5, 0.95),
  add = TRUE,
  col = c("pink", "purple")
)
```



It has an elliptical shape, but there looks to be a potential outlier in the bottom left corner.

(c) Consider a new variable Y = X1+X2. Write down the distribution of Y.

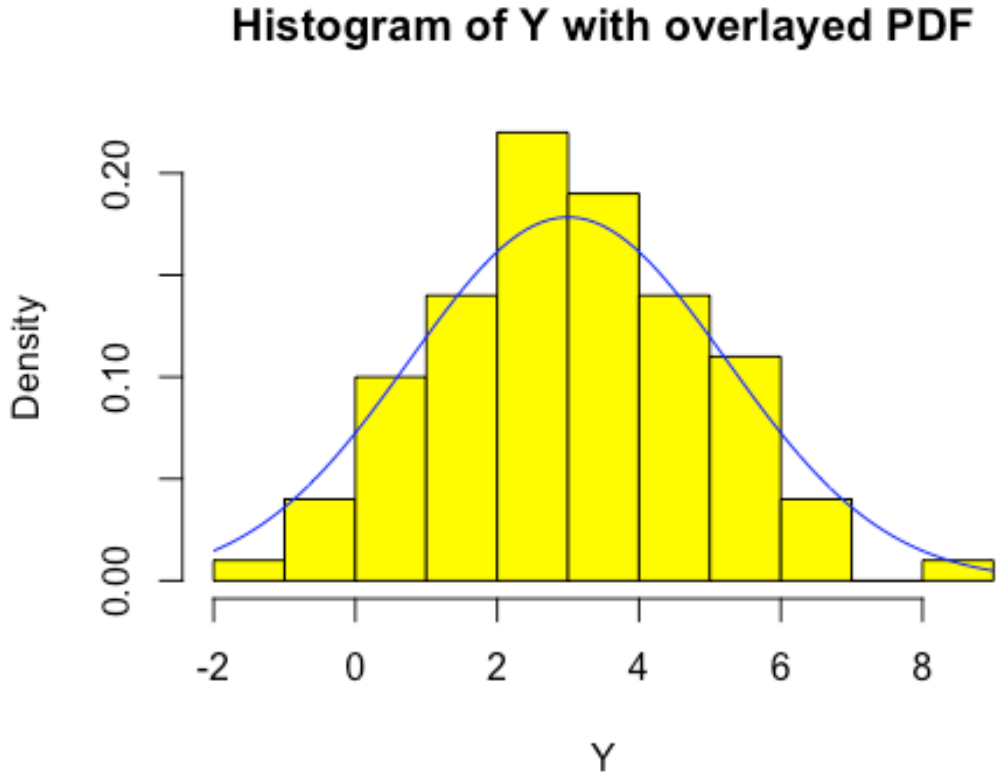
Mean: E(Y) = E(X1) + E(X2) = 1 + 2 = 3

Variance:Var(Y) = Var(X1) + Var(X2) + 2(Cov(X1,X2) =1 + 2 + 2 x 1 = 5

Y ~ N(3,5)

(d) From the sample generated in (a), compute observations of Y, draw a histogram of these values, and overlay the PDF of the distribution you obtained in part (c) on the histogram. What pattern do you see? Does your expectation match with what you see in the plot? Explain.

```
#compute y values
Y <- X1+X2

#create histogram
hist(Y, breaks = 15, probability = TRUE, col = "yellow",main = "Histogram of
Y with overlayed PDF", xlab = "Y")
#overlay pdf
curve(dnorm(x,mean=3,sd=sqrt(5)),
      add = TRUE, col = "blue")
```



I see a generally looking normal distribution, which was the expected shape. This is because Y is derived from a bivariate normal distribution, and any linear combination of normal variables is also normally distributed. The top of the bell curve is around Y=3, the theoretical normal distribution. The PDF also follows the bell shaped curve, as expected.