Multivariate Data Analysis - Assignment 1

Multivariate Data Analysis Spring 2019 (37459-2019-SPRING-CITY)

Assignment: 1

Student Name: Anuj Kapil

Student Id: 12678708

Question 1

Step 1: Download the dataset from online

```
wd <- getwd()
res <- readLines("http://users.stat.umn.edu/~kb/classes/5401/files/data/JWData5.txt")
# Figure out which lines we need. Each table reference will appear twice. Need to know the table you want and the next in the list

start <- grep("T06_12", res)[2]

end <- grep("T06_13", res)[2]

rawtable <- res[start:(end - 2)]

# Use the first line to find the dimensions of the table

infovec <- strsplit(rawtable[1], " ")[[1]]

infovec <- infovec[infovec != ""]

length <- as.numeric(infovec[2])

# Extract the rows containing the data (the last few rows)

start <- length(rawtable) - length + 1

rawdata <- rawtable[start:length(rawtable)]

# Split the row into each data point

final_data <- strsplit(rawdata, " ")

# Organise into a matrix, removing blank entries

datatable <- matrix(0, length, as.numeric(infovec[3]))

for (i in 1:length) {
  row <- final_data[[i]]
```

```
  datatable[i, ] <- as.numeric(row[row != ""])
}

# Arrange in a data frame

data.frame <- as.data.frame(datatable)
colnames(data.frame) <- c("Gender", "x1", "x2", "x3", "x4")
```

Step 2: Data Description

The dataset is about the oxygen consumption for 25 males and 25 females subjects who were asked to run on a treadmill until exhaustion. Gas contents were collected and analysed at regular intervals. The dataset includes

```
Gender: male = 1 & female = 2
x1: resting volume O2 (L/min)
x2: resting volume O2 (mL/kg/min)
x3: maximum volume O2 (L/min)
x4: maximum volume O2 (mL/kg/min)
```

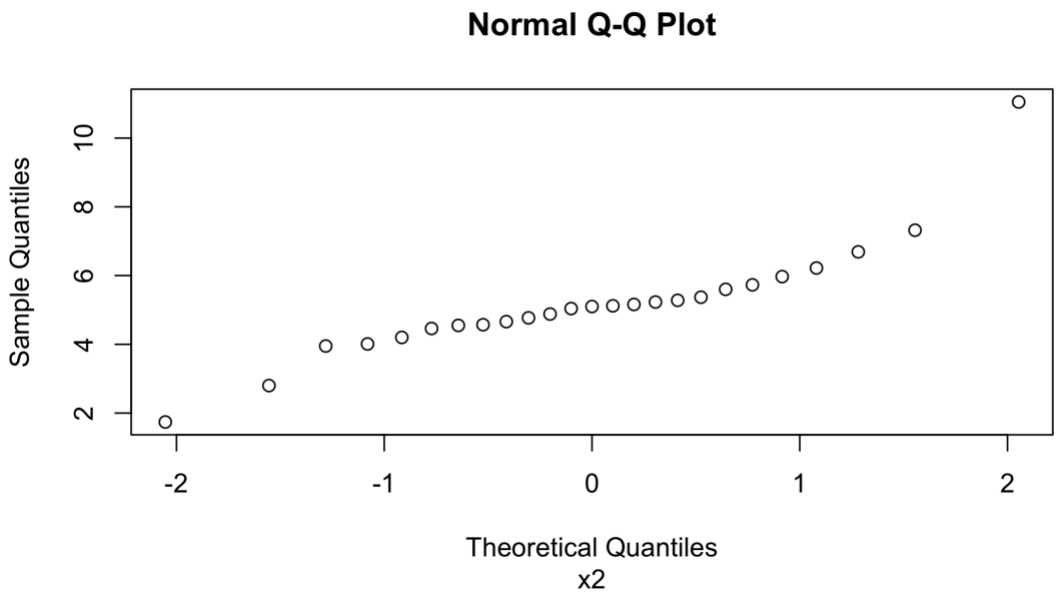
For this analysis, we are only looking at data from 25 female subjects.

```
# male and female legends
male <- 1
female <- 2

# Convert the data.frame to a data.table
setDT(data.frame)

# Filter the observation for 'female' gender only
obs.female <- data.frame[Gender==2]
```

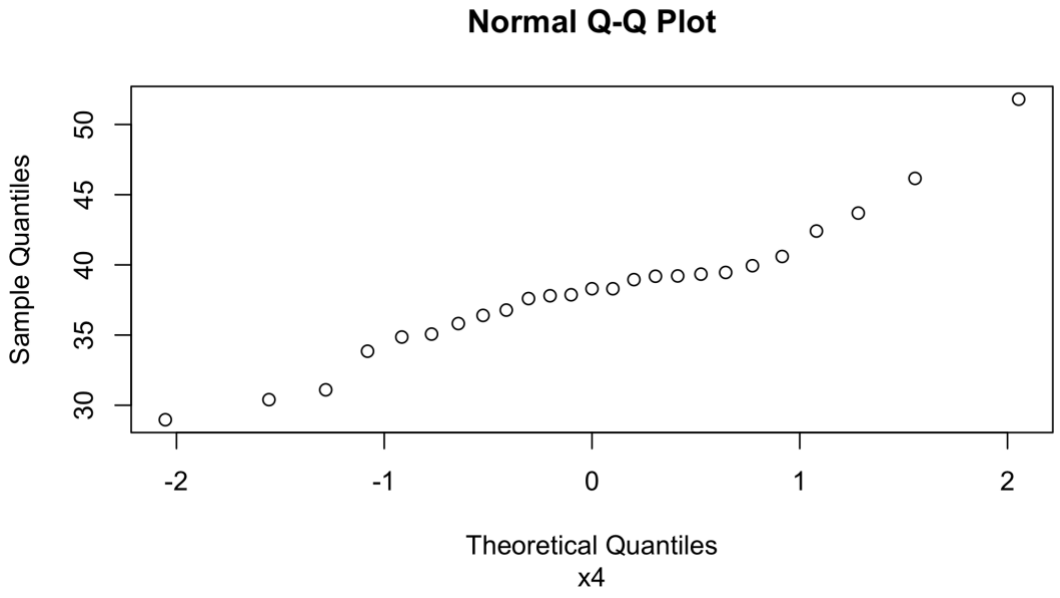
Question 1a

Both the Normal Q-Q plots suggests a right skewed distribution with bimodal distribution for the given sample of x2 and x4. However, the sample x4 tends to be more normal distribution while x2 sample does not. Plotting the density plot confirms the presence of more than one peak and right skewness. Shapiro Wilk test rejects the null hypothesis of sample x2 being univariate normal. Shapiro Wilk test does not reject the null hypothesis of sample x4 being univariate normal.

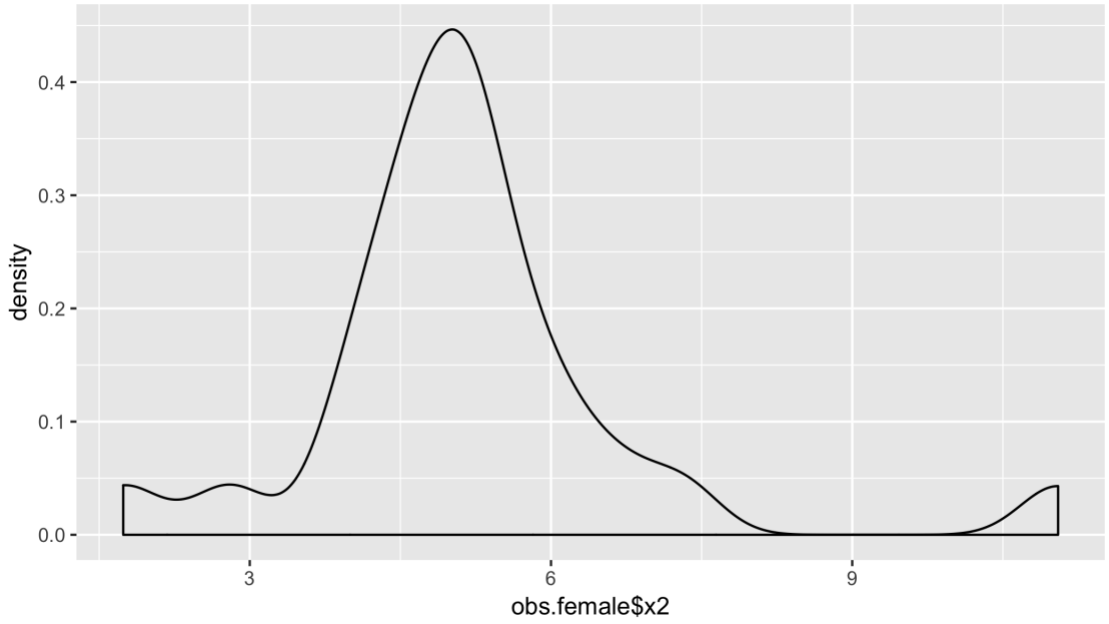
```
# Question 1a
setDF(obs.female)

# Normal Q-Q plot for x2
qqnorm(obs.female[,3], sub = colnames(obs.female)[3])
```



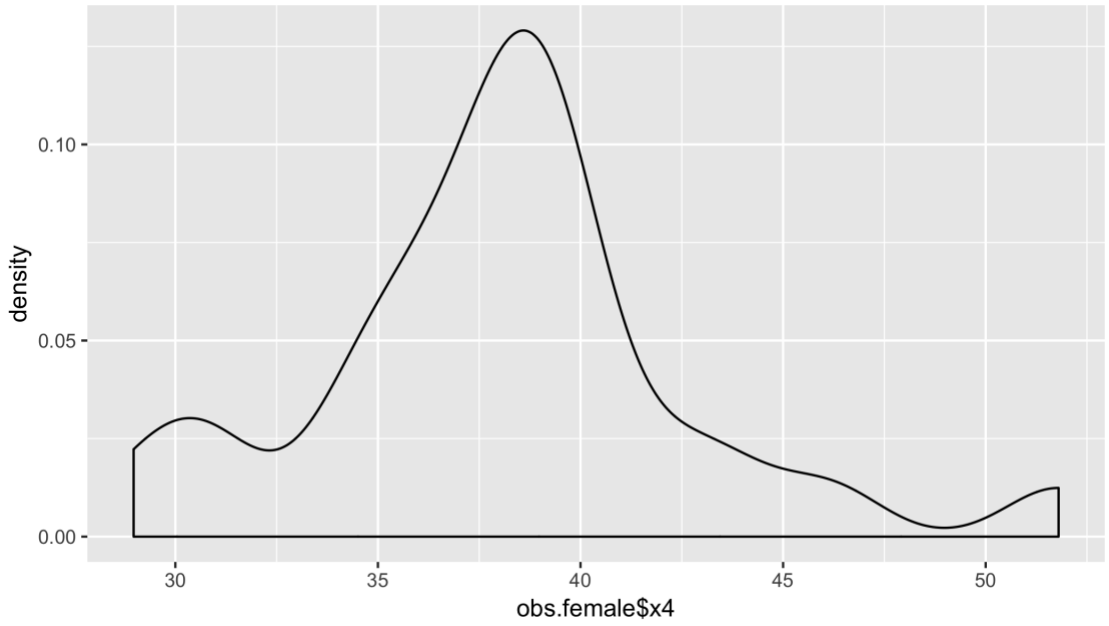

```
# Normal Q-Q plot for x4
qqnorm(obs.female[,5], sub = colnames(obs.female)[5])
```




```
setDT(obs.female)
p1<-ggplot(obs.female, aes(x=obs.female$x2)) + geom_density()
p2<-ggplot(obs.female, aes(x=obs.female$x4)) + geom_density()

# Density plot for variable x2
print(p1)
```




```
# Density plot for variable x4
print(p2)
```



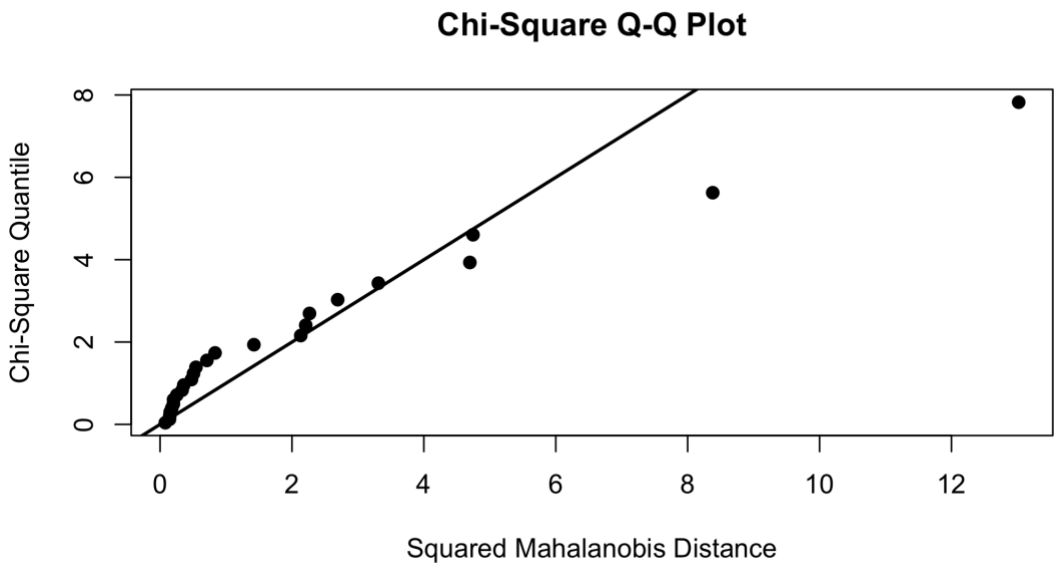

```
# Shapiro Wilk test for variable x2
shapiro.test(obs.female$x2)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  obs.female$x2
## W = 0.84752, p-value = 0.001581

# Shapiro Wilk test for variable x4
shapiro.test(obs.female$x4)

##
##  Shapiro-Wilk normality test
##
## data:  obs.female$x4
## W = 0.94409, p-value = 0.1839
```

Question 1b

Chi-square Q-Q plot shows the lower quantiles points are more dense and higher quantiles points are spread out which depicts a more right skewed distribution. Performing a Royston's H test for multivariate normality on the sample of x2 and x4 suggest that the sample dataset (x2, x4) is not multivariate normal. The univariate Shapiro Wilk test suggest that the x2 sample is not univariate normal while x4 sample is. Finally, the skewness for x2 and x4 show a positively right skewed data with x2 being more skewed than x4.

```
# Question 1b
mvtest <- mvn(obs.female[,list(x2,x4)], mvnTest='royston', multivariatePlot='
qq')
```




```
mvtest$multivariateNormality
```

```
##      Test        H      p value MVN
## 1 Royston 15.09293 0.0005281256  NO
```

```
mvtest$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    x2        0.8475    0.0016    NO
## 2 Shapiro-Wilk    x4        0.9441    0.1839    YES
```

```
mvtest$Descriptives
```

```
##     n    Mean  Std.Dev Median   Min   Max  25th  75th      Skew  Kurtosis
## x2 25  5.1788 1.667531    5.1  1.74 11.05  4.55  5.60 1.3623856 4.2103263
## x4 25 38.1548 4.822948   38.3 28.97 51.80 35.82 39.46 0.5594578 0.9846515
```

Question 1c

The intention of Chi-square test is to find out how likely an observed distribution is due to chance. It is also called a "goodness of fit" statistic, because it measures how well the observed distribution of data fits with the distribution that is expected if the variables are independent. If the test is run on a very small sample size, only the most aberrant behaviour will be identified as lack of fit. On the other hand, running the test on a very large samples invariably produces a statistically significant lack of fit, yet the departure from the specified distribution may be very small and technically unimportant to the inferential conclusions.

From the plot, we can see that our sample size is small and the marginal distribution appears normal with the exception of top two ordered squared distances which don't quite fit the line.

For further analysis, we can look at transforming the data to observations that are more bivariate normal. In practice, data are often transformed using a log transformation or a square root transformation.

Question 1d

As shown earlier, our data do not follow a multivariate normal distribution. One approach to dealing with such data is to perform a transformation on the data to make the observations more multivariate normal. Generally, the data is often transformed using a log transformation or a square root transformation to stabilise the variances of the response in each of the treatment groups when they differ significantly. These two transformations are special cases of the Box–Cox transformation. Let us compute the lambda (λ) which is a paramater that implies a particluar power family of transformations for each of the variable to transform them to near-normal. We will run a Shapiro Wilk test to check the univariate normality.

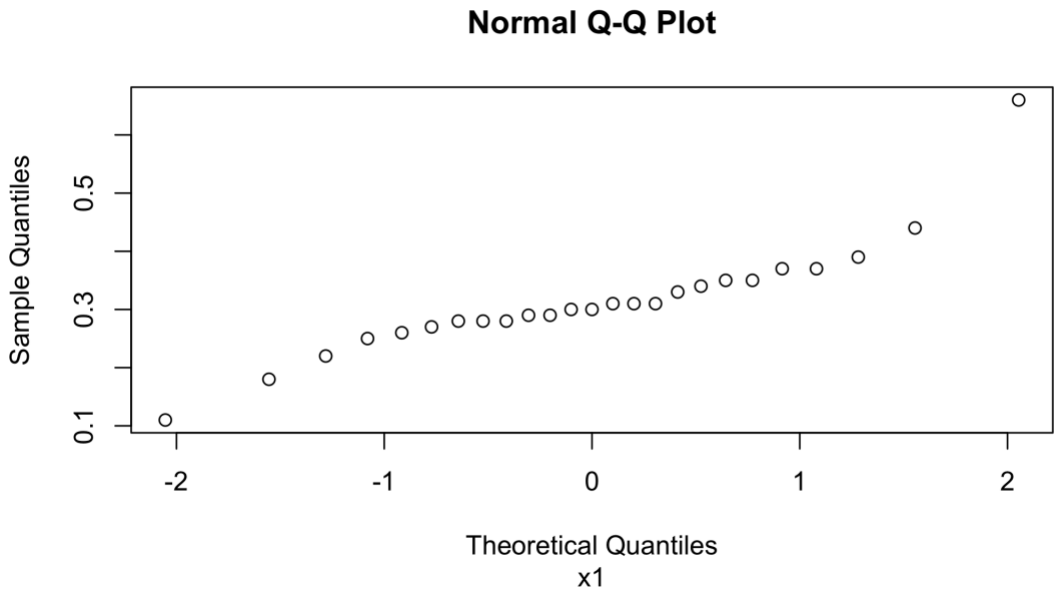
```r
# Question 1d
setDT(obs.female)

trans<-powerTransform(obs.female[,list(x1)])
obs.female.x1<-obs.female[,list(x1)]
obs.female.x1<-bcPower(obs.female.x1,trans$lambda)
print(trans)

## Estimated transformation parameter
##       x1
## 0.385718

shapiro.test(obs.female.x1[[1]])

##
##  Shapiro-Wilk normality test
##
## data:  obs.female.x1[[1]]
## W = 0.90473, p-value = 0.02329

trans<-powerTransform(obs.female[,list(x2)])
obs.female.x2<-obs.female[,list(x2)]
obs.female.x2<-bcPower(obs.female.x2,trans$lambda)
print(trans)

## Estimated transformation parameter
##       x2
## 0.407507

shapiro.test(obs.female.x1[[1]])

##
##  Shapiro-Wilk normality test
##
## data:  obs.female.x1[[1]]
## W = 0.90473, p-value = 0.02329

trans<-powerTransform(obs.female[,list(x3)])
obs.female.x3<-obs.female[,list(x3)]
obs.female.x3<-bcPower(obs.female.x3,trans$lambda)
print(trans)

## Estimated transformation parameter
##         x3
## -0.1728886

shapiro.test(obs.female.x3[[1]])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  obs.female.x3[[1]]
## W = 0.9762, p-value = 0.8011

trans<-powerTransform(obs.female[,list(x4)])
obs.female.x4<-obs.female[,list(x4)]
obs.female.x4<-bcPower(obs.female.x4,trans$lambda)
print(trans)

## Estimated transformation parameter
##         x4
## -0.1596472

shapiro.test(obs.female.x4[[1]])

##
##  Shapiro-Wilk normality test
##
## data:  obs.female.x4[[1]]
## W = 0.95903, p-value = 0.3954

obs.female.transformed <- cbind(obs.female.x1, obs.female.x2, obs.female.x3,
obs.female.x4)
```

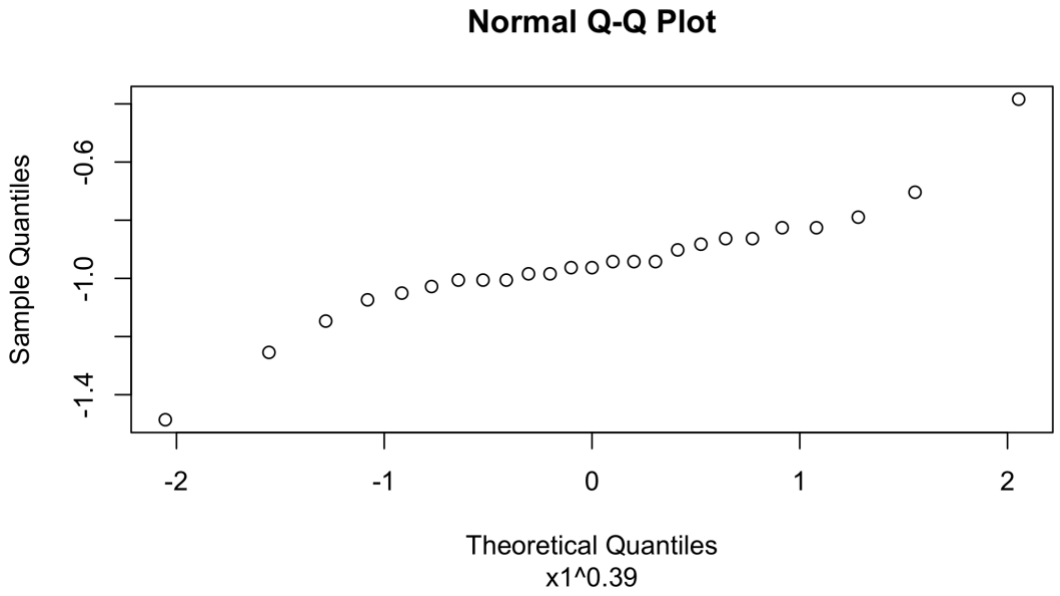
Question 1e

Looking at the Q-Q plot post transformation, we can see that in each of the charts the distribution shift toward near normal as compared to the original distribution.

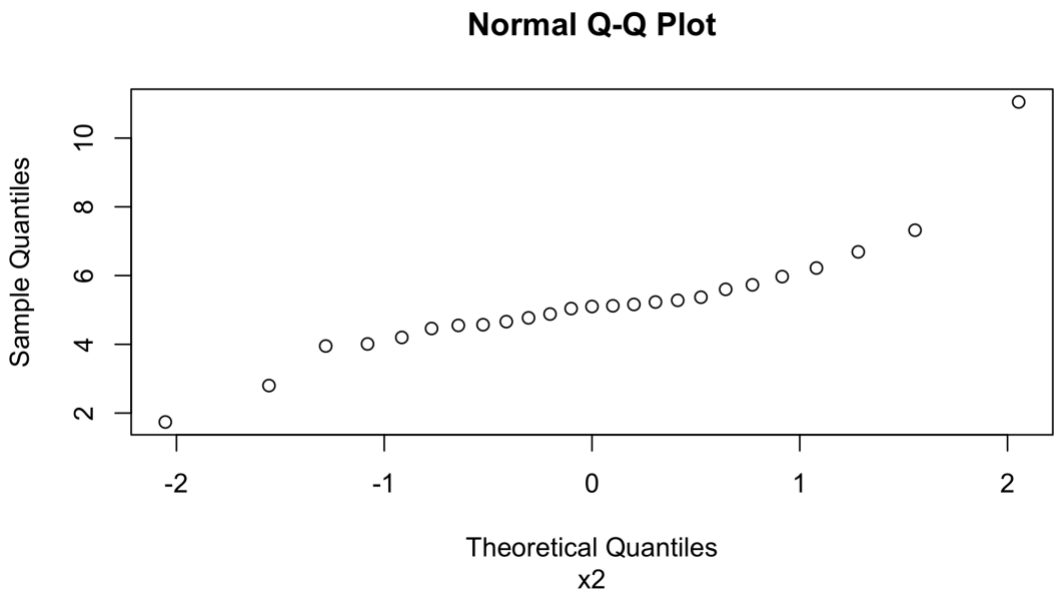
```
# Question 1e

# x1 before transformation
qqnorm(obs.female[[2]], sub = colnames(obs.female)[2])
```



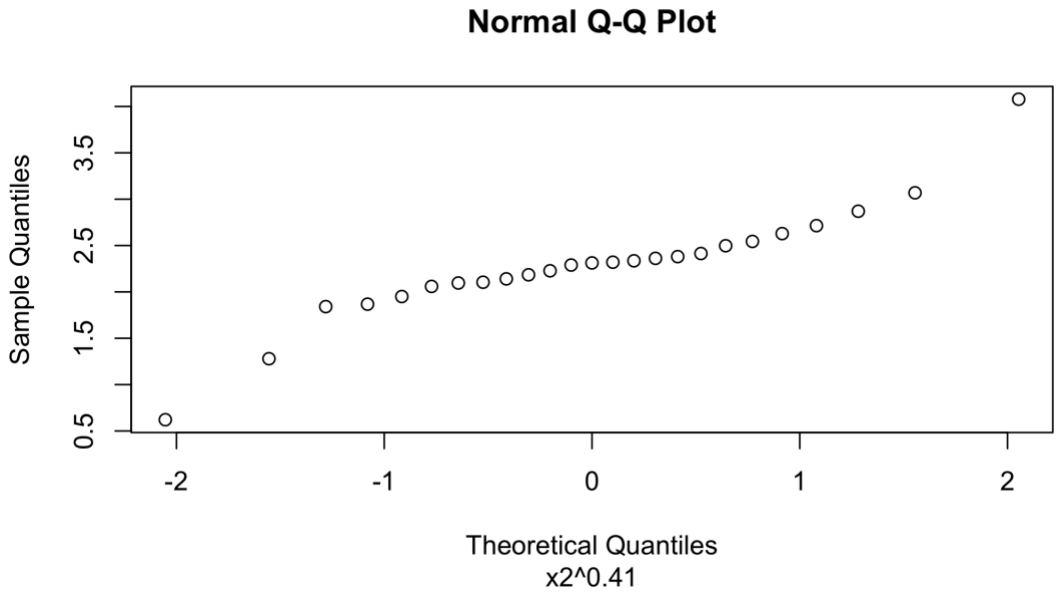

```r
# x1 after transformation
qqnorm(obs.female.x1[[1]], sub = colnames(obs.female.x1)[1])
```



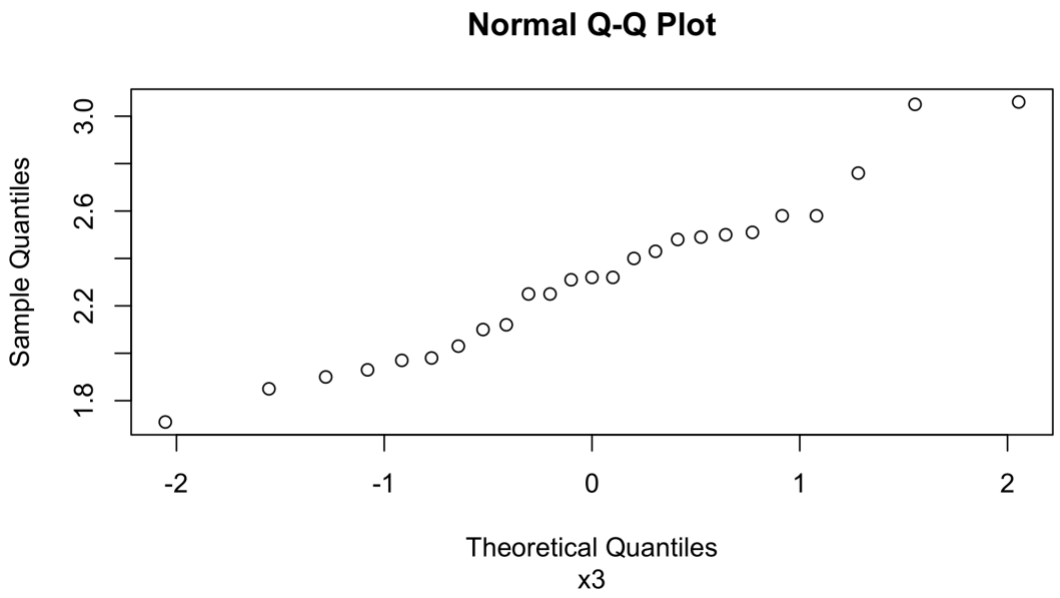

```r
# x2 before transformation
qqnorm(obs.female[[3]], sub = colnames(obs.female)[3])
```



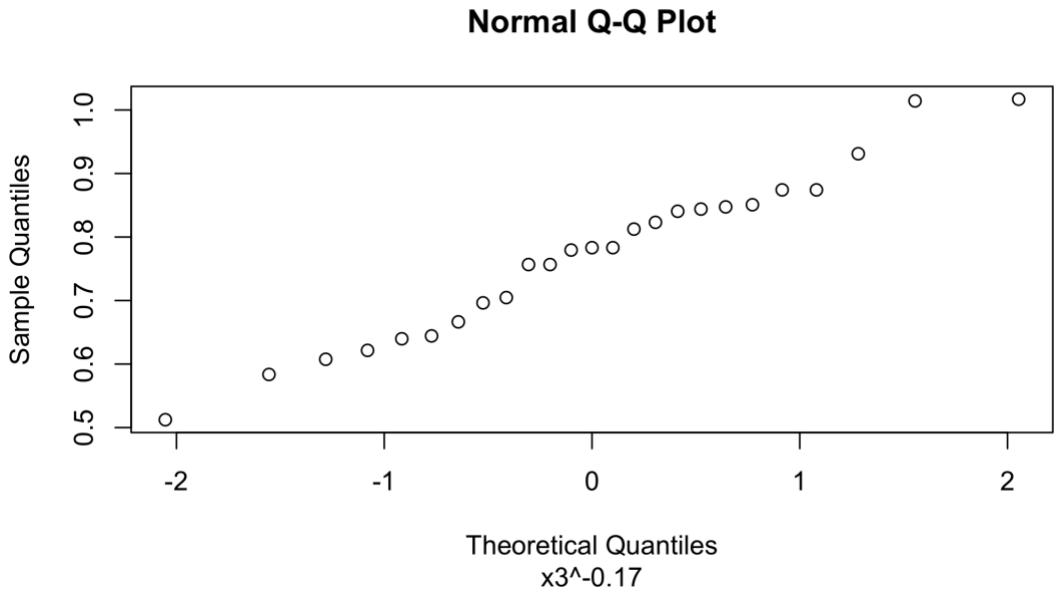

```r
# x2 after transformation
qqnorm(obs.female.x2[[1]], sub = colnames(obs.female.x2)[1])
```



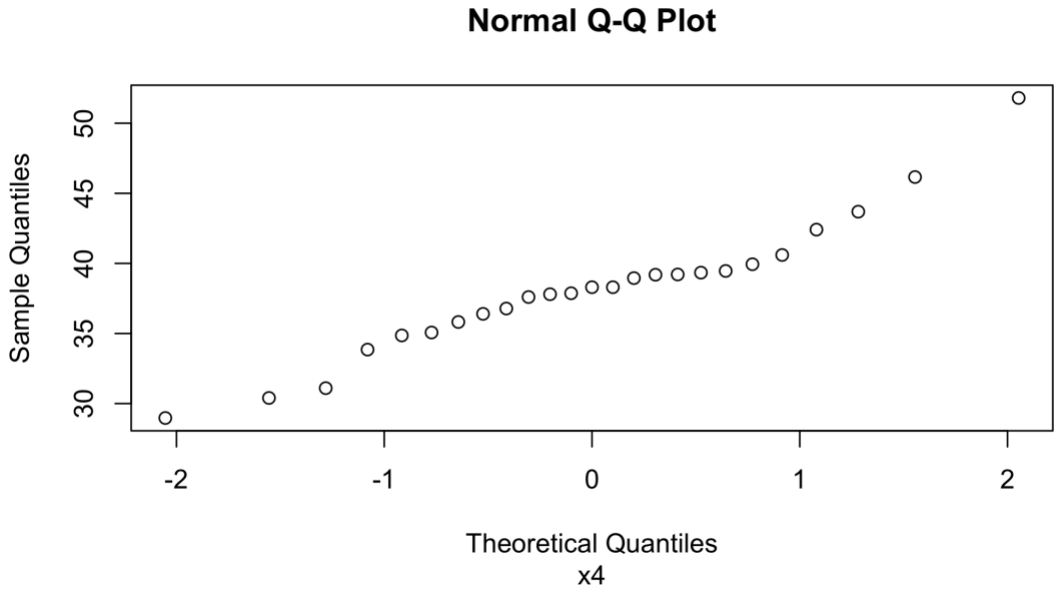

```r
# x3 before transformation
qqnorm(obs.female[[4]], sub = colnames(obs.female)[4])
```



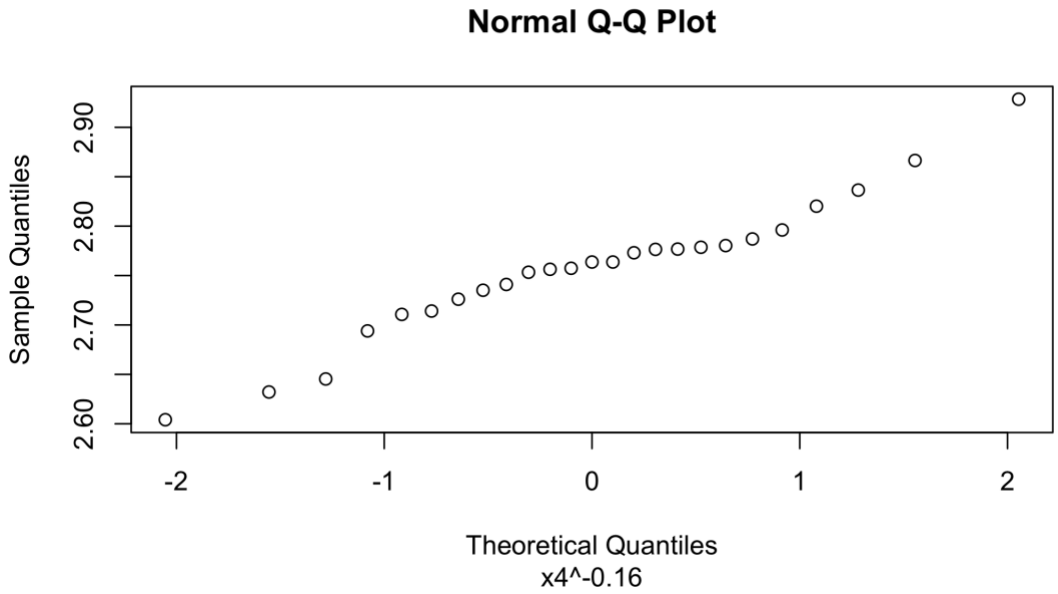

```r
# x3 after transformation
qqnorm(obs.female.x3[[1]], sub = colnames(obs.female.x3)[1])
```




```r
# x4 before transformation
qqnorm(obs.female[[5]], sub = colnames(obs.female)[5])
```




```
# x4 after transformation
qqnorm(obs.female.x4[[1]], sub = colnames(obs.female.x4)[1])
```



Question 2

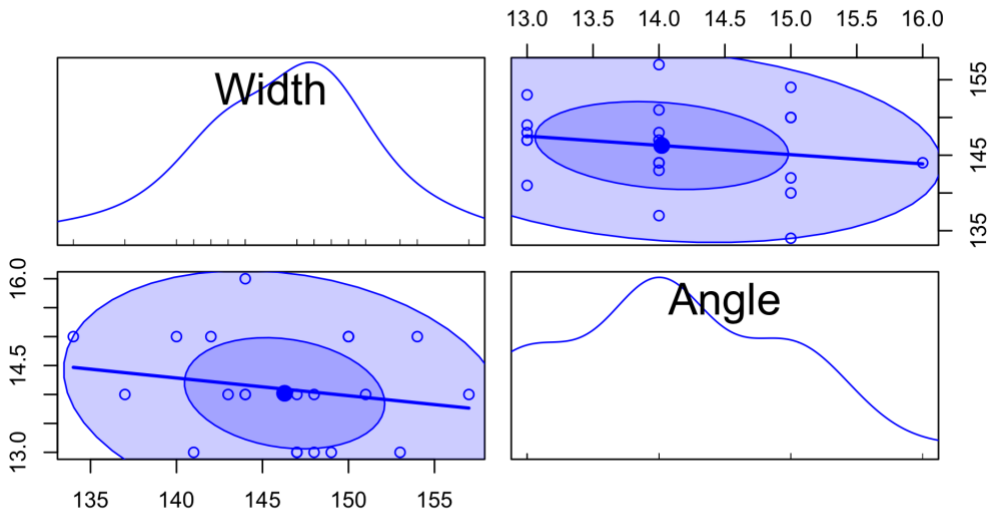
Step 1: Download the dataset from online

```
concinna <- fread('Data/Concinna.csv')
concinna[, Species:=NULL]
```

Question 2a

The one at a time confidence intervals for individual means (μ_1, μ_2) of the concinna species' 'Width' and 'Angle' data can be calculated using the t-tests . Here we are calculating 95% confidence intervals for each variable separately, ignoring the rest and plotting the ellipse format chart for bivariate control region.

```
# Question 2a
setDF(concinna)
for (i in 1:2) {
  ci <- round(t.test(concinna[, i])$conf.int, 3)
  cat(paste("The 95% CI for ", colnames(concinna)[i], " is: (", ci[1], ", ",
            ci[2], ")\n"))
}

## The 95% CI for  Width  is: ( 143.629 ,  148.752 )
## The 95% CI for  Angle  is: ( 13.691 ,  14.5 )

scatterplotMatrix(~ Width+Angle,
                  data=concinna, smooth=FALSE,  ellipse=TRUE, by.groups=TRUE,
                  diagonal="none")

## Warning in applyDefaults(diagonal, defaults = list(method =
## "adaptiveDensity"), : unnamed diag arguments, will be ignored
```



Question 2b

We will use Hotellings T^2 statistic to check whether a specific value μ_0 is a plausible value for the population mean μ. For a given mean of μ_1 = 125mm and μ_2 = 14 units for

the heikertingeri species, our null hypothesis is that the given mean is a plausible value for the population mean of concinna species. The alternative hypothesis is that the given mean is not a plausible value for the population mean of concinna species. Hotellings T^2 test rejects the null hypothesis, which mean the given mean vector μ_1 = 125mm and μ_2 = 14 does not belong to the concinna population.

```
# Question 2b
# u1 or Width is outside the 95% CI, so possibly not a concinna species
# Run the Hotellings T2 test to verify
HotellingsT2(concinna, mu = c(125,14))

##
##  Hotelling's one sample T2-test
##
## data:  concinna
## T.2 = 148.72, df1 = 2, df2 = 19, p-value = 2.485e-12
## alternative hypothesis: true location is not equal to c(125,14)
```

Question 2c

Hotellings simultaneous T^2-intervals For the given dataset, we can caluclate the sample mean $\overline{x}$ and sample covariance matrix S.
We can then calculate the simultaneous confidence interval using the below formalae for the lower and upper internval

$$a'\overline{X} - \sqrt{\frac{p(n-1)}{n(n-p)}F_{p,n-p}(\alpha)a'Sa}$$

$$a'\overline{X} + \sqrt{\frac{p(n-1)}{n(n-p)}F_{p,n-p}(\alpha)a'Sa}$$

```
# Question 2c
Xbar <- colMeans(concinna)
S <- cov(concinna)
n <- nrow(concinna)
p <- ncol(concinna)
for (i in 1:2) {
  lower <- round(Xbar[i] - sqrt(p * (n - 1)/(n - p) * qf(0.95, p, n - p)) *
                   sqrt(S[i, i]/n), 3)
  upper <- round(Xbar[i] + sqrt(p * (n - 1)/(n - p) * qf(0.95, p, n - p)) *
                   sqrt(S[i, i]/n), 3)
  cat(paste("The 95% CI for ", colnames(concinna)[i], " is: (", lower, ", ",
            upper, ")\n"))
}
```

```
## The 95% CI for  Width  is: ( 142.847 ,  149.534 )
## The 95% CI for  Angle  is: ( 13.567 ,  14.624 )
```

Question 2d

Bonferroni CI can be calculated by modifying percenatage points (confidence levels) for the t-tests.

```
# Question 2d
for (i in 1:2) {
  ci <- round(t.test(concinna[, i], conf.level = (1 - 0.05/p))$conf.int, 3)
  cat(paste("The 95% Bonferroni CI for ", colnames(concinna)[i], " is: (",
            ci[1], ", ", ci[2], ")\n"))
}
```

```
## The 95% Bonferroni CI for  Width  is: ( 143.215 ,  149.166 )
## The 95% Bonferroni CI for  Angle  is: ( 13.625 ,  14.565 )
```

Question 2e

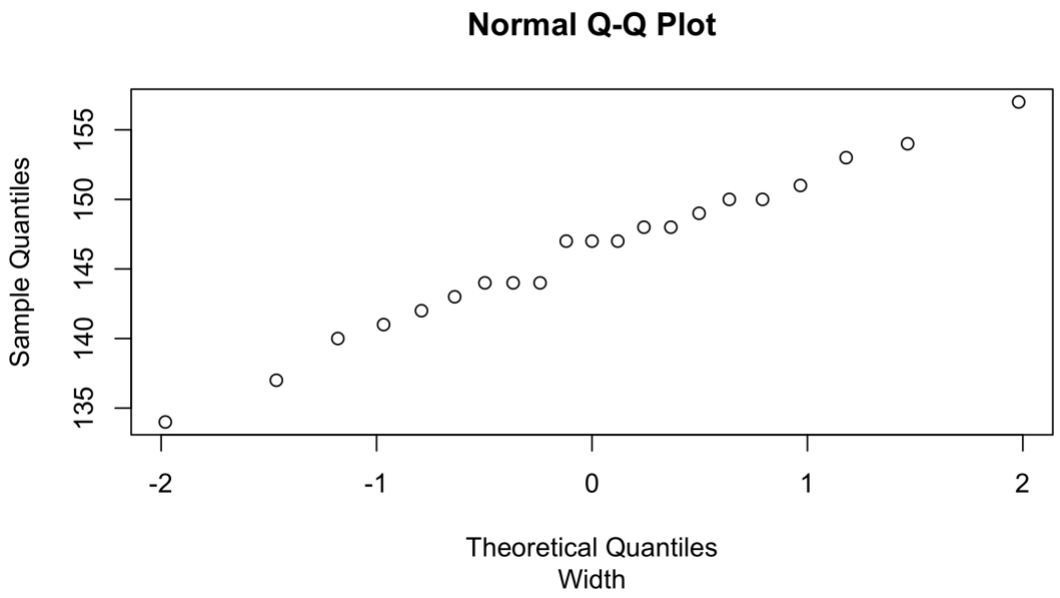
Hotelling statistics is at advantage to build simultaneous confidence statements if people are interested in confidence interval of means for each individual component with the disadvantage being that the confidence intervals are wider. The advantage for Bonferroni method is that it is a very simple and efficient method that allows many comparison statements to be made (or confidence intervals to be constructed) while still assuring an overall confidence coefficient is maintained. It uses very simple probabilistic inequality. They are easy to apply and provide a relatively short confidence intervals needed for inference.

Question 2f

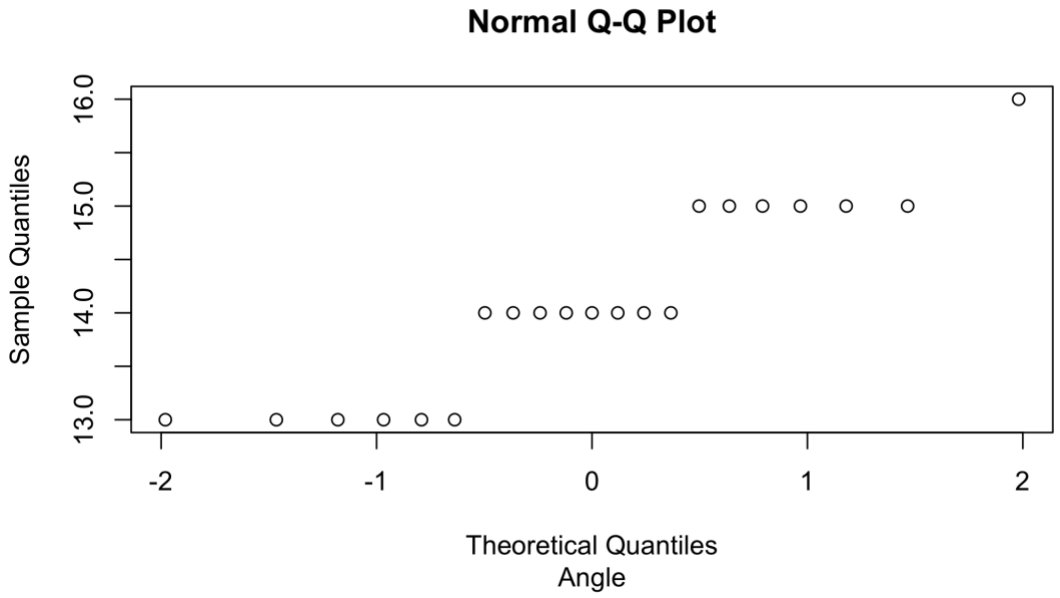
Q-Q plot for 'Width' indicates that the data is univariate normal as all the point almost fit on a straight line. Q-Q plot for 'Angle' suggests categorical/discreet data.

The Shapiro-Wilk test rejects the null hypothesis of sample data for 'Angle' being univariate normal while the test does not reject the null hypothesis for 'Width' sample data. Hence, 'Width' is univariate normal and 'Angle' is not univariate normal. Performing a Royston test suggest that the sample data ('Width', 'Angle') for the concinna species is not bivariate normal The p-value being less than .05 implies that the test rejects the null hypothesis of the sample data being multivariate normal.

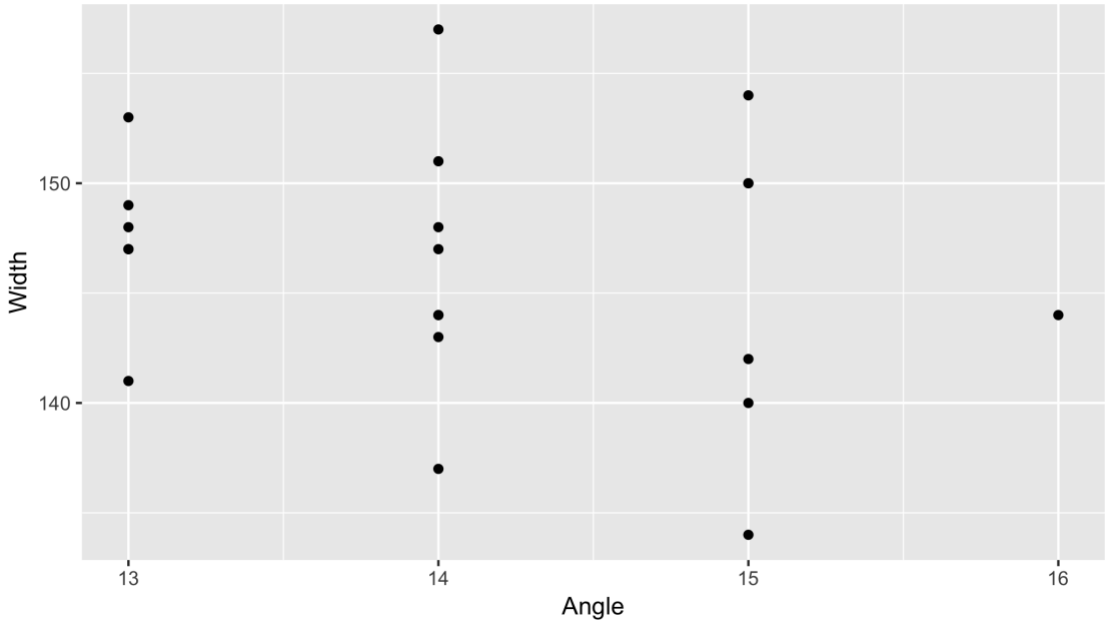
```
# Question 2f
concinna <- fread('Data/Concinna.csv')
concinna[, Species:=as.factor(Species)]

# Q-Q plot for 'Width'
qqnorm(concinna[[1]], sub = colnames(concinna)[1])
```



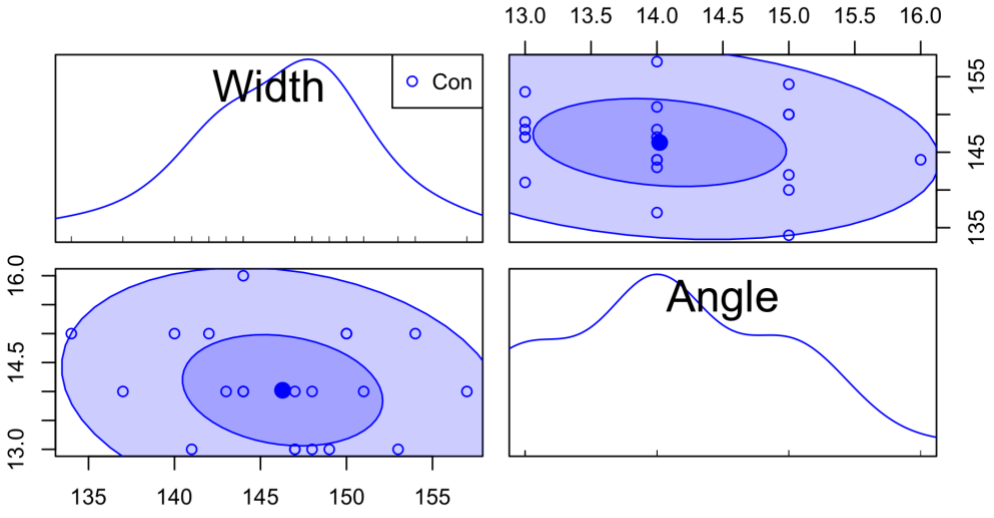

```r
# Q-Q plot for 'Angle'
qqnorm(concinna[[2]], sub = colnames(concinna)[2])
```



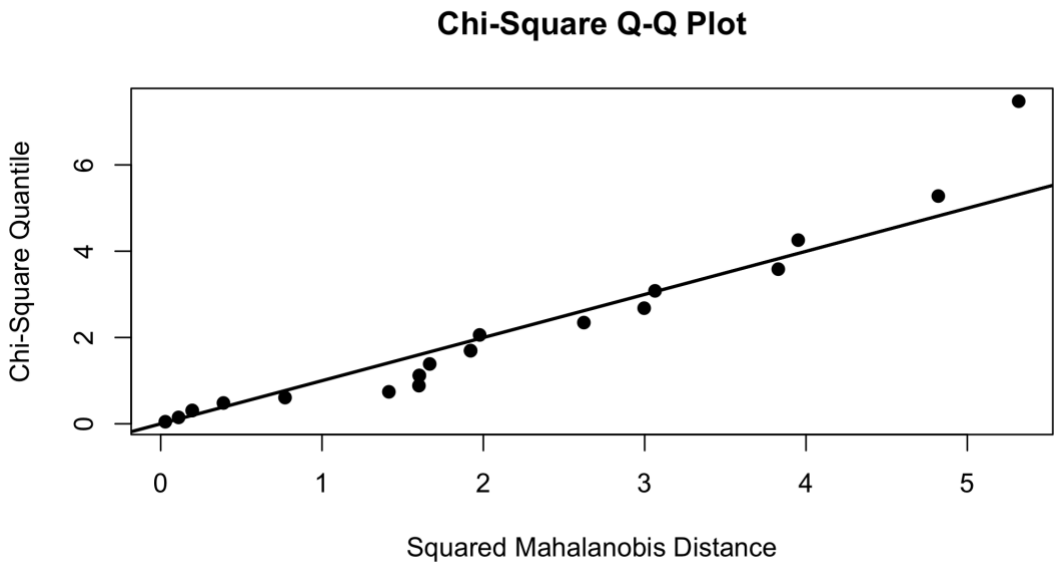

```r
# Scatter plot for 'Width' and 'Angle'
ggplot(concinna, aes(x=Angle, y=Width)) +
  geom_point()
```




```
setDF(concinna)
scatterplotMatrix(~ Width+Angle | Species,
data=concinna, smooth=FALSE, regLine=FALSE, ellipse=TRUE, by.groups=TRUE,
diagonal="none")

## Warning in applyDefaults(diagonal, defaults = list(method =
## "adaptiveDensity"), : unnamed diag arguments, will be ignored
```



```
setDT(concinna)
concinna[, Species:=NULL]
mvntest <- mvn(concinna, mvnTest='royston', multivariatePlot='qq')
```




```
mvntest$multivariateNormality
```

```
##      Test        H    p value MVN
## 1 Royston 6.941658 0.03106203  NO
```

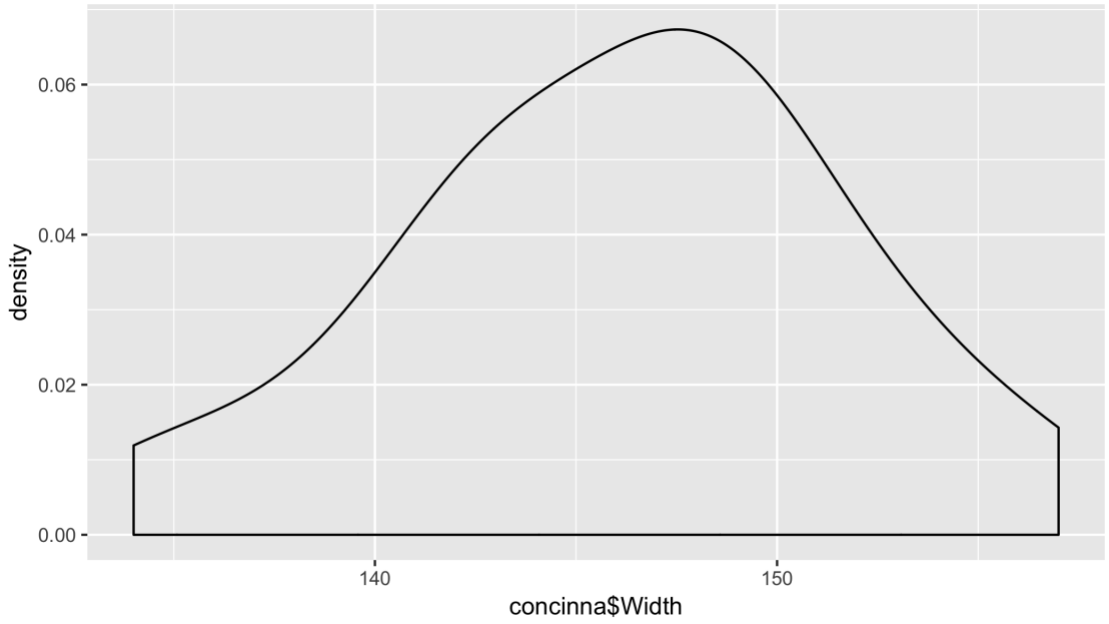
```
mvntest$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   Width      0.9881    0.9936    YES
## 2 Shapiro-Wilk   Angle      0.8669    0.0084    NO
```

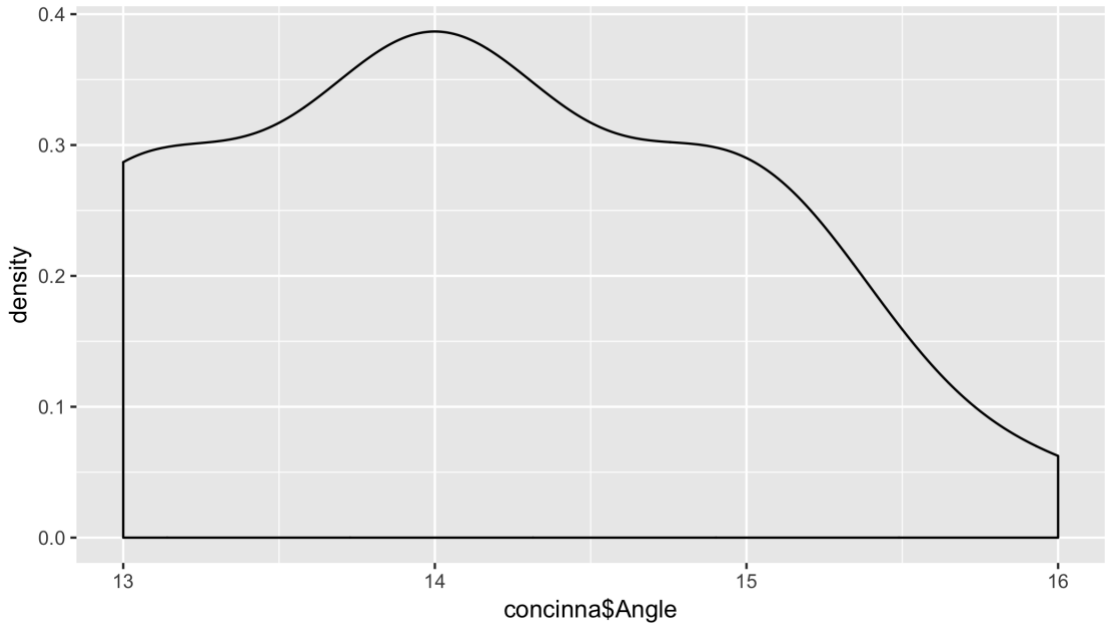
```
mvntest$Descriptives
```

```
##        n      Mean   Std.Dev Median Min Max 25th 75th       Skew
## Width 21 146.19048 5.6268912    147 134 157  143  150 -0.2111542
## Angle 21  14.09524 0.8890873     14  13  16   13   15  0.2347644
##         Kurtosis
## Width -0.5081691
## Angle -1.0324544
```

```
p1<-ggplot(concinna, aes(x=concinna$Width)) + geom_density()
p2<-ggplot(concinna, aes(x=concinna$Angle)) + geom_density()
print(p1)
```




```
print(p2)
```



Question 3

Question 3a

For a given covariance matrix, calculate the inverse of covariance matrix.

```
# Question 3a
##### Cov matrix
vcmat <- 1/5630 * matrix(c(575,-60,10,-60,300,-50,10,-50,196),nrow=3,byrow=TR
```

```
UE)

# inverse
vcmat_inv <-solve(vcmat)

# Covariance matrix
print(vcmat)

##              [,1]         [,2]         [,3]
## [1,]  0.102131439 -0.010657194  0.001776199
## [2,] -0.010657194  0.053285968 -0.008880995
## [3,]  0.001776199 -0.008880995  0.034813499

# Inverse of Covariance matrix
print(vcmat_inv)

##              [,1] [,2]          [,3]
## [1,] 1.000000e+01    2 -6.794082e-17
## [2,] 2.000000e+00   20  5.000000e+00
## [3,] 5.792468e-17    5  3.000000e+01

# Check if matrix * inverse matrix is an identity matrix
round(vcmat%*%vcmat_inv)

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

Question 3b

The conditional covariance of Y2 and Y3 given the value Y1 = y, we would take the bottom right corner of the inverse covariance matrix and then re-inverse it.

The result is equal to the covariance matrix of Y2 and Y3 conditioned on the the value for Y1 = y. The same is true for all three pairs which indicates that all the variables are conditionally independent.

Also, looking at the correlation matrix, there is no significant correlation among the 3 variables.

```
# Question 3b
# Y2 and Y3 for a given Y1
round(solve(vcmat_inv[-1,-1]), 2)
```

```
##       [,1]  [,2]
## [1,]  0.05 -0.01
## [2,] -0.01  0.03
```

```
round(vcmat[-1,-1],2)
```

```
##       [,1]  [,2]
## [1,]  0.05 -0.01
## [2,] -0.01  0.03
```

```
# Y1 and Y3 for a given Y2
round(solve(vcmat_inv[-2,-2]), 2)
```

```
##      [,1] [,2]
## [1,]  0.1 0.00
## [2,]  0.0 0.03
```

```
round(vcmat[-2,-2],2)
```

```
##      [,1] [,2]
## [1,]  0.1 0.00
## [2,]  0.0 0.03
```

```
# Y1 and Y2 for a given Y3
round(solve(vcmat_inv[-3,-3]), 2)
```

```
##       [,1]  [,2]
## [1,]  0.10 -0.01
## [2,] -0.01  0.05
```

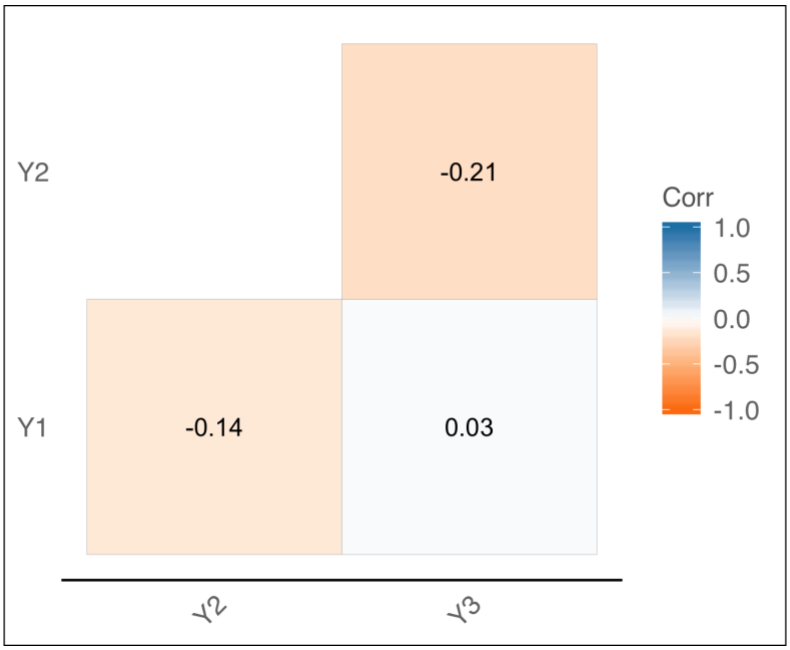
```
round(vcmat[-3,-3],2)
```

```
##       [,1]  [,2]
## [1,]  0.10 -0.01
## [2,] -0.01  0.05
```

```
# Correlation
cormat <- cov2cor(vcmat)
rownames(cormat) <- c('Y1', 'Y2','Y3')
colnames(cormat) <- c('Y1', 'Y2','Y3')
print(cormat)
```

```
##             Y1         Y2          Y3
## Y1  1.00000000 -0.1444630  0.02978777
## Y2 -0.14446302  1.0000000 -0.20619652
## Y3  0.02978777 -0.2061965  1.00000000
```

```
ggcorrplot(cormat, hc.order = FALSE, type = "lower",
           ggtheme = ggthemes::theme_gdocs,
           colors = c("#ff7f0e", "white", "#1f83b4"),
           lab = TRUE)+
  theme(panel.grid.major=element_blank())
```



Question 3c

For a given mean vector and covariance matrix, we can simulate random samples from the multivariate normal distribution in R using the 'mvrnorm' function from **MASS** package.

```
# Question 3c
#MVN
mv<-rep(0, 3)
mnd <- mvrnorm(n=1000,mv,vcmat)
```

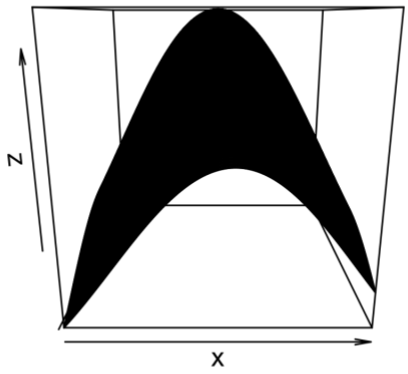
Question 3d

2D and 3D plots for the 3 pairs of multivariate normal distribution

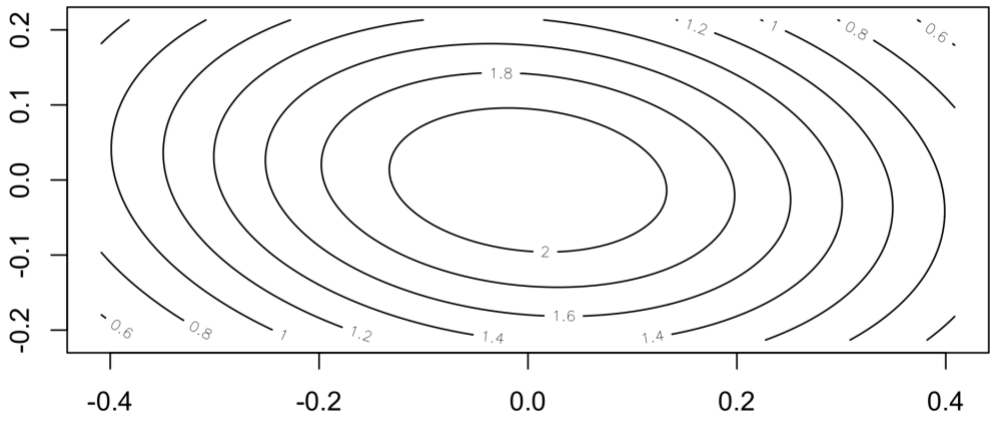
```
# Question 3d
#Pair 1 (Y1, Y2)
x <- seq(from=mv[1]-4*vcmat[1,1],to=mv[1]+4*vcmat[1,1],by=vcmat[1,1]/25)
y <- seq(from=mv[2]-4*vcmat[2,2],to=mv[2]+4*vcmat[2,2],by=vcmat[2,2]/25)
vcmat[1:2,1:2]
```

```
##             [,1]        [,2]
## [1,]  0.10213144 -0.01065719
## [2,] -0.01065719  0.05328597
```

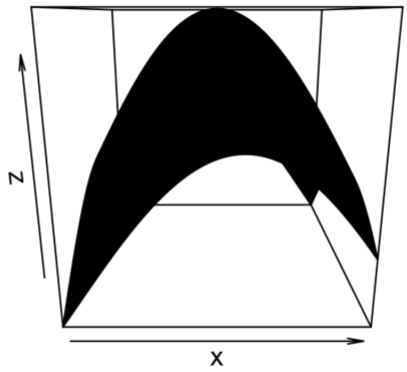
```
z <- matrix(0,201,201)
for(i in 1:201){
  for(j in 1:201){
    z[i,j]<-dmnorm(c(x[i],y[j]), mean = mv[1:2], vcmat[1:2,1:2], log = FALSE)
  }
}
persp(x,y,z, axes = TRUE,box = TRUE)
```



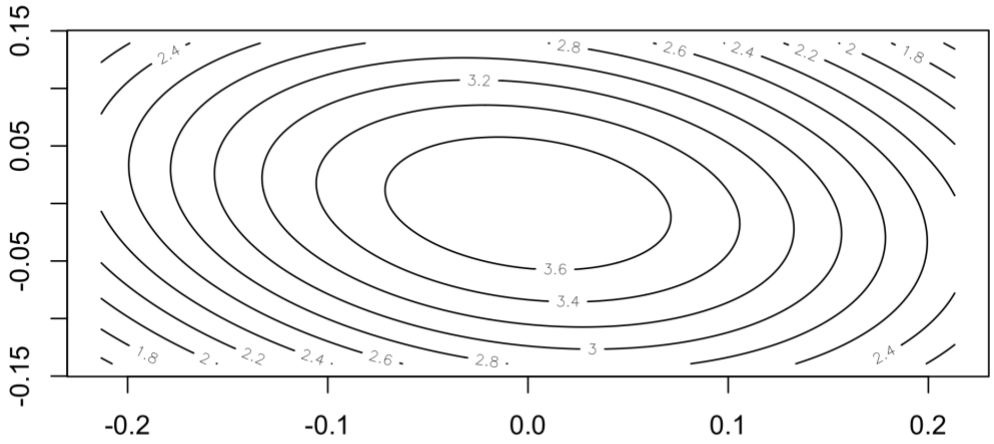

```
contour(x,y,z, axes = TRUE)
```




```r
  #Pair 2 (Y2, Y3)
x <- seq(from=mv[2]-4*vcmat[2,2],to=mv[2]+4*vcmat[2,2],by=vcmat[2,2]/25)
y <- seq(from=mv[3]-4*vcmat[3,3],to=mv[3]+4*vcmat[3,3],by=vcmat[3,3]/25)
vcmat[2:3,2:3]

##              [,1]         [,2]
## [1,]  0.053285968 -0.008880995
## [2,] -0.008880995  0.034813499

z <- matrix(0,201,201)
for(i in 1:201){
  for(j in 1:201){
    z[i,j]<-dmnorm(c(x[i],y[j]), mean = mv[2:3], vcmat[2:3,2:3], log = FALSE)
  }
}
persp(x,y,z, axes = TRUE,box = TRUE)
```

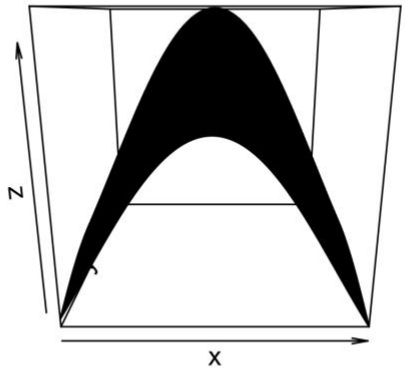



```
contour(x,y,z, axes = TRUE)
```

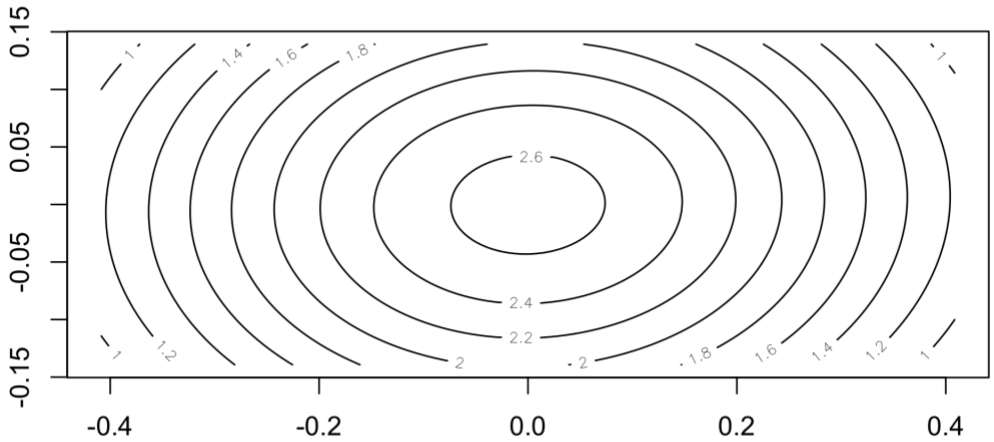



```
#Pair 3 (Y1, Y3)
x <- seq(from=mv[1]-4*vcmat[1,1],to=mv[1]+4*vcmat[1,1],by=vcmat[1,1]/25)
y <- seq(from=mv[3]-4*vcmat[3,3],to=mv[3]+4*vcmat[3,3],by=vcmat[3,3]/25)
vcmat[-2,-2]

##              [,1]        [,2]
## [1,] 0.102131439 0.001776199
## [2,] 0.001776199 0.034813499
```

```r
z <- matrix(0,201,201)
for(i in 1:201){
  for(j in 1:201){
    z[i,j]<-dmnorm(c(x[i],y[j]), mean = mv[-2], vcmat[-2,-2], log = FALSE)
  }
}
persp(x,y,z, axes = TRUE,box = TRUE)
```




```r
contour(x,y,z, axes = TRUE)
```



Question 3e

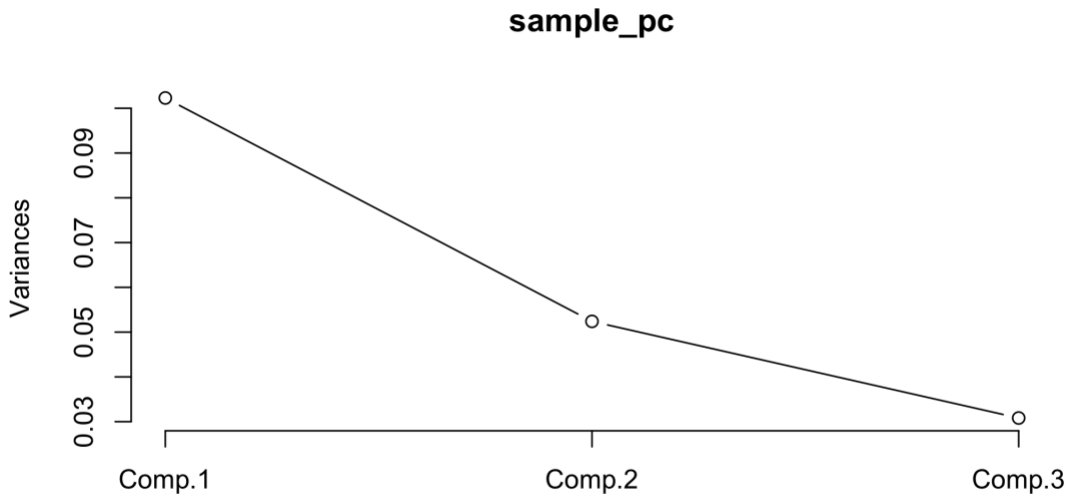
The eigenvalues and the eigenvectors for the given covariance matrix are as below:

```
# Question 3e
options(scipen = 999)
B<-eigen(vcmat)

# Eigen Values
B$values

## [1] 0.10453986 0.05453633 0.03115472

# Eigen Vectors
B$vectors

##             [,1]       [,2]       [,3]
## [1,]  0.97591370 -0.2153318 0.03499535
## [2,] -0.21190556 -0.8975461 0.38666136
## [3,]  0.05185053  0.3847638 0.92155755
```

Question 3f

For the random sample that was generated earlier using the mean vector and covariance matrix, we can transform the sample data into principal components using 'princomp' function from base-R **stats** package. It is used to explain the variance-covariance structure of a set of variables through linear combinations. The principal component 1 explains about 55% of the variance in the sample data.

```
# Question 3f
sample_pc<-princomp(mnd)
summary(sample_pc, loadings = TRUE)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3
## Standard deviation     0.3198343 0.2289065 0.1755433
## Proportion of Variance 0.5514273 0.2824584 0.1661143
## Cumulative Proportion  0.5514273 0.8338857 1.0000000
##
## Loadings:
##      Comp.1 Comp.2 Comp.3
## [1,]  0.975  0.220
## [2,] -0.214  0.897  0.386
## [3,]        -0.383  0.922

# Plot the variances by principal components
screeplot(sample_pc, npcs = 3, type = "lines")
```



Question 3g

As explained earlier, the principal component 1 explains most of the variance in the sample data (~55%). Looking at the summary statistics, the second highest variance of ~28% is explained by the second principal component. If we have to choose only two principal component, we should pick comp1 and comp2 as they together are responsible for around 83% of variance in the sample data. Ofcourse, choosing only two components will cause loss of information.

Question 3h

Calculate the least square estimates using R function and using direct formula

$$Y_2 = \beta_1 Y_1 + \epsilon_2$$

The other way to calculate β_1 is by calculating coefficients from covariances.

$$\beta_1 = \frac{Cov(Y_1, Y_2)}{Cov(Y_1, Y_1)}$$

```
# Question 3h

# Convert matrix to a data.table
mnd_df <- as.data.frame(as.table(mnd))
setDT(mnd_df)
mnd_dt <- dcast(mnd_df, Var1~Var2, value.var = 'Freq')
mnd_dt[,Var1:=NULL]

colnames(mnd_dt) <- c('Y1', 'Y2', 'Y3')
```

```r
model_1<-lm(Y2~Y1, data = mnd_dt)

model_summary <- summary(model_1)

# Coefficent of Y1
model_summary$coefficients[[2]]
```

```
## [1] -0.1069227
```

```r
# Intercept
model_summary$coefficients[[1]]
```

```
## [1] -0.01498945
```

```r
# Another way of calculating coefficent and intercept
y <- mnd[,2]
x <- mnd[,1]

a <- cov(x,x)
b <- cov(x,y)

# Coefficients from the covariance matrix
beta.hat <- b/a

# Find the intercept from the means and coefficients.
y.bar <- mean(y)
x.bar <- mean(x)
intercept <- y.bar - x.bar * beta.hat

# Coefficent of Y1
beta.hat
```

```
## [1] -0.1069227
```

```r
# Intercept
intercept
```

```
## [1] -0.01498945
```

Question 3i

Calculate the least square estimates using R function and using direct formula

$$Y_3 = \beta_1 Y_1 + \beta_2 Y_2 + \epsilon_3$$

```r
# Question 3i

model_2<-lm(Y3~Y1+Y2, data = mnd_dt)

model_summary <- summary(model_2)

#Coefficent of Y1
model_summary$coefficients[[2]]
```

```
## [1] 0.005349099
```

```r
#Coefficent of Y2
model_summary$coefficients[[3]]
```

```
## [1] -0.1604139
```

```r
# Intercept
model_summary$coefficients[[1]]
```

```
## [1] 0.005445864
```

```r
# Another way of calculating coefficent and intercept
y <- mnd[,3]
x <- mnd[,1:2]

a <- cov(x)
b <- cov(x,y)
# Coefficients from the covariance matrix
beta.hat <- solve(a, b)[, 1]

# Find the intercept from the means and coefficients.
y.bar <- mean(y)
x.bar <- colMeans(x)
intercept <- y.bar - x.bar %*% beta.hat

#Coefficent of Y1
beta.hat[1]
```

```
## [1] 0.005349099
```

```r
#Coefficent of Y2
beta.hat[2]
```

```
## [1] -0.1604139
```

```r
# Intercept
intercept[1]
```

```
## [1] 0.005445864
```

Question 3j

Calculate Var(Y2 | Y1)

```
# Question 3j
#Var(Y2|Y1)
variance_Y2_Y1 <- sum(model_1$residuals**2)/(length(model_1$residuals)-1)

#vcov(model_1)
print(variance_Y2_Y1)
```

```
## [1] 0.05037446
```

Question 3k

Calculate Var(Y3 | Y1, Y2)

```
# Question 3k
#Var(Y3|Y1, Y2)
variance_Y3_Y1Y2 <- sum(model_2$residuals**2)/(length(model_2$residuals)-1)
print(variance_Y3_Y1Y2)
```

```
## [1] 0.03291554
```

Question 4

Step 1: Download the dataset from online

```
url <- 'https://web.stanford.edu/~hastie/Papers/LARS/diabetes.data'
diabetes_orig <- fread(url, sep = '\t')
```

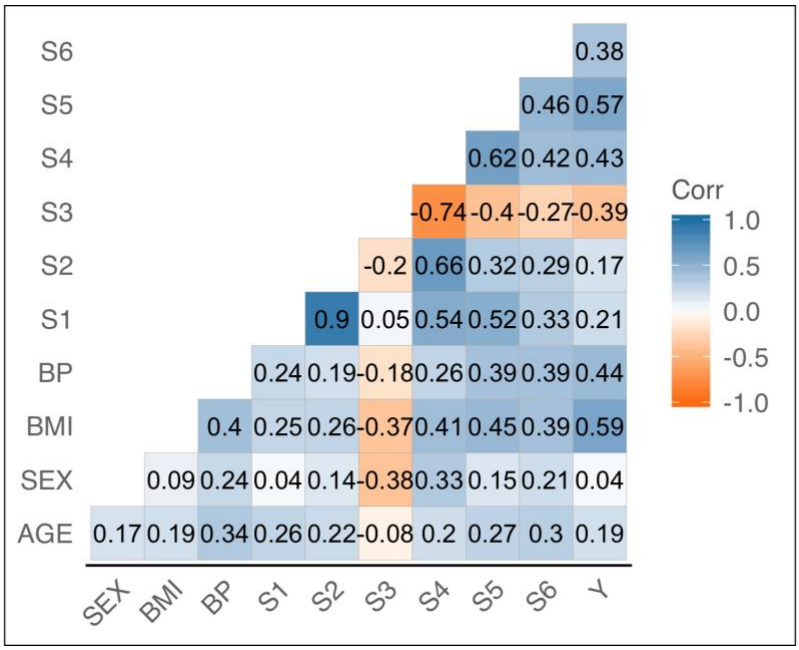
Question 4a

The data consists of ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of n = 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

Multicollinearity analysis is done by calculating correlation matrix. Blood serum measurement S1 is highly positive-correlated with the progression of diabetes. Blood serum measurement S3 is highly negative-correlated with the progression of diabetes.

```
corr <- cor(diabetes_orig, use = "pairwise.complete.obs")

ggcorrplot(corr, hc.order = FALSE, type = "lower",
```

```
          ggtheme = ggthemes::theme_gdocs,
          colors = c("#ff7f0e", "white", "#1f83b4"),
          lab = TRUE)+
  theme(panel.grid.major=element_blank())
```



Question 4b

Perfom multiple linear regression to predict the response y from the ten covariates. In the dataset the sex variable is a categorical variabe where 1 indicates female and 2 male. We would need to convert that to factor to perform meaningful regression. The the coefficients β in a linear predictor are as shown below:

```
# Question 4b
diabetes_orig$SEX <- as.factor(diabetes_orig$SEX)
model_1<-lm(Y~., data = diabetes_orig)
model_summary <- summary(model_1)
model_summary$coefficients
```

```
##                  Estimate Std. Error    t value                 Pr(>|t|)
## (Intercept) -357.42678661 67.0580722 -5.3301083 0.00000015862960998870
## AGE           -0.03636122  0.2170414 -0.1675313 0.86703063370008282007
## SEX2         -22.85964809  5.8358213 -3.9171261 0.00010416711927693297
## BMI            5.60296209  0.7171055  7.8133023 0.00000000000004296391
## BP             1.11680799  0.2252382  4.9583425 0.00000102427839221141
## S1            -1.08999633  0.5733319 -1.9011613 0.05794760536919688065
## S2             0.74645046  0.5308344  1.4061833 0.16039024001496365868
## S3             0.37200472  0.7824638  0.4754274 0.63472325577520360973
## S4             6.53383194  5.9586378  1.0965311 0.27345869366067832029
```

```
## S5           68.48312496 15.6697192  4.3704117 0.00001555899086539269
## S6            0.28011699  0.2733140  1.0248909 0.30598952619642194595
```

Question 4c

Coefficient of determination (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. In the model fitted above, the R^2 of .517 implies that 52% of the variance in the dependent variable (progression of diabetes) is explained by all the 10 independent variables and rest is residuals/errors. The best fit model is the one which has maximum R^2. In the second model below, we have eliminated the non significant independent variables from the model based on the p-value from model 1. The R^2 of model 2 is .486 implies that 49% of the variance in the dependent variable (progression of diabetes) is explained by 4 independent variables. Even though the R^2 has dropped, we have reduced the number of inputs to the model from 10 to 4. It implies these 4 variable explains most of the variability in the dependent variable which is better than taking 10 variables as input.

```
# Question 4c
model_2 <-lm(Y~SEX+BMI+BP+S5,  data = diabetes_orig)
model_summary <- summary(model_2)
model_summary$coefficients
```

```
##               Estimate Std. Error    t value
## (Intercept) -341.885373 25.9076551 -13.196307
## SEX2         -13.036008  5.4623083  -2.386538
## BMI            6.448376  0.6937284   9.295247
## BP             1.008660  0.2201177   4.582365
## S5            50.539872  5.8729376   8.605552
##                                            Pr(>|t|)
## (Intercept) 0.00000000000000000000000000001058471
## SEX2        0.01743244745221526892375862871631397865 7
## BMI         0.00000000000000000070384096234363164511 6
## BP          0.00000600743991781293661759216426232477 2
## S5          0.00000000000000013809173047597355059968 8
```

```
# R-squared for Model 2
summary(model_2)$r.squared
```

```
## [1] 0.4867715
```

```
# R-squared for Model 1
summary(model_1)$r.squared
```

```
## [1] 0.5177484
```

Question 4d

The residual standard error is the square root of the residual sum of squares divided by the residual degrees of freedom. The mean square error is the mean of the sum of squared residuals, i.e. it measures the average of the squares of the errors. Lower values (closer to zero) indicate better fit.

For both the model, there is strong relation between the disease progression (dependent variable) and sex, bmi, bp and serum S5 (independent variables). Model interpretation for model 2 is as below: For a patient being male, the disease progresses less by a factor of -13.03. Which mean the chances of disease progression is more in female than males in the given sample population. For a unit increase in BMI of patient, the disease progresses by a factor of 6.44 For a 1000 unit increase in BP of patient, the disease progresses by a factor of 1008.7 For a unit increase in S5 serum measurement, the disease progresses by a factor of 50.54

```
summary(model_2)$coefficients
```

```
##                Estimate Std. Error    t value
## (Intercept) -341.885373 25.9076551 -13.196307
## SEX2         -13.036008  5.4623083  -2.386538
## BMI            6.448376  0.6937284   9.295247
## BP             1.008660  0.2201177   4.582365
## S5            50.539872  5.8729376   8.605552
##                                            Pr(>|t|)
## (Intercept) 0.00000000000000000000000000001058471
## SEX2        0.017432447452215268923758628716313978657
## BMI         0.00000000000000000070384096234363164516
## BP          0.00000600743991781293661759216426232477
## S5          0.00000000000000001380917304759735505996 88
```

Question 4e

Final Model: In multiple regression, two or more predictor variables might be correlated with each other. This situation is referred as collinearity. There is an extreme situation, called multicollinearity, where collinearity exists between three or more variables even if no pair of variables has a particularly high correlation. This means that there is redundancy between predictor variables.

In the presence of multicollinearity, the solution of the regression model becomes unstable. Multicollinearity can assessed by computing a score called the variance inflation factor (VIF).

The smallest possible value of VIF is one (absence of multicollinearity). As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity.

For the model 1, the VIF is quite high for Serum measurements S1 to S5. Even though S5 has significant impact on the dependent variables, yet there are other multicollinear variable in the data that might cause the model to be unstable. Which is why the second model is preferred over the first model (vif is within acceptable range for model 2) even though the R^2 of model 2 is lower than model 1. By choosing model 2, we are reducing dependency on 6 input independent variable, along with some compromise on the coefficient of determination (R^2).

```
# Variable Inflation factor
vif(model_1)
```

```
##       AGE       SEX       BMI        BP        S1        S2        S3
##  1.217307  1.278071  1.509437  1.459428 59.202510 39.193370 15.402156
##        S4        S5        S6
##  8.890986 10.075967  1.484623
```

```
# R-squared
summary(model_1)$r.squared
```

```
## [1] 0.5177484
```

```
# Variable Inflation factor
vif(model_2)
```

```
##      SEX      BMI       BP       S5
## 1.066769 1.345845 1.327931 1.348473
```

```
# R-squared
summary(model_2)$r.squared
```

```
## [1] 0.4867715
```

Question 4f

Looking at the Residuals vs fitted plot below, we see that the data does not have any obvious distinct pattern. While it is slightly curved, it has equally spread residuals around the horizontal line without a distinct pattern.

For our model 2, the Q-Q plot shows pretty good alignment to the the line with a few points at the top and bottom being slightly offset. Probably not significant and a reasonable alignment.

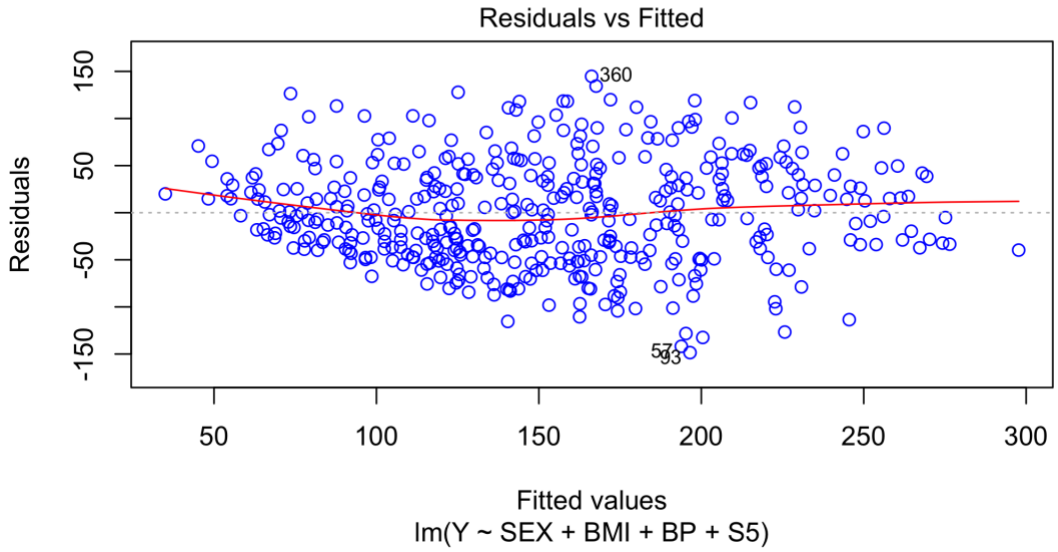
For the scale-location plot, the residuals are reasonably well spread above and below a pretty horizontal line which implies the residuals have equal variance(occupy equal space) above and below the line and along the length of the line.

Residual vs leverage plot can be used to find influential cases in the dataset. An influential case is one that, if removed, will affect the model so its inclusion or exclusion should be considered.

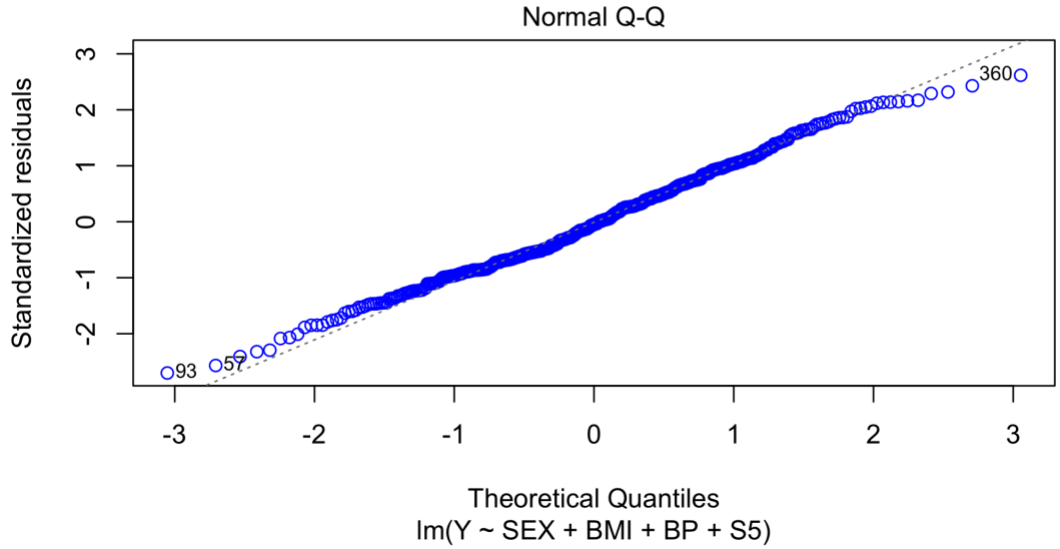
An influential case may or may not be an outlier and the purpose of this chart is to identify cases that have high influence in the model. Outliers will tend to exert leverage and therefore influence on the model.

An influential case will appear in the top right or bottom left of the chart inside a red line which marks Cook's Distance but in case of our model 2, there is no such influential case.

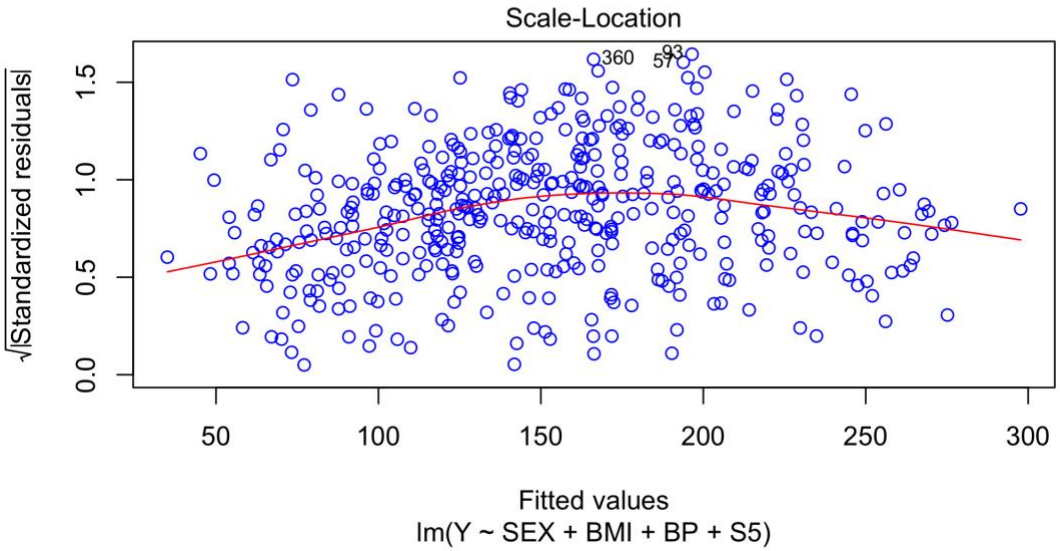
```
# Residual vs Fitted
plot(model_2, which=1, col=c("blue"))
```



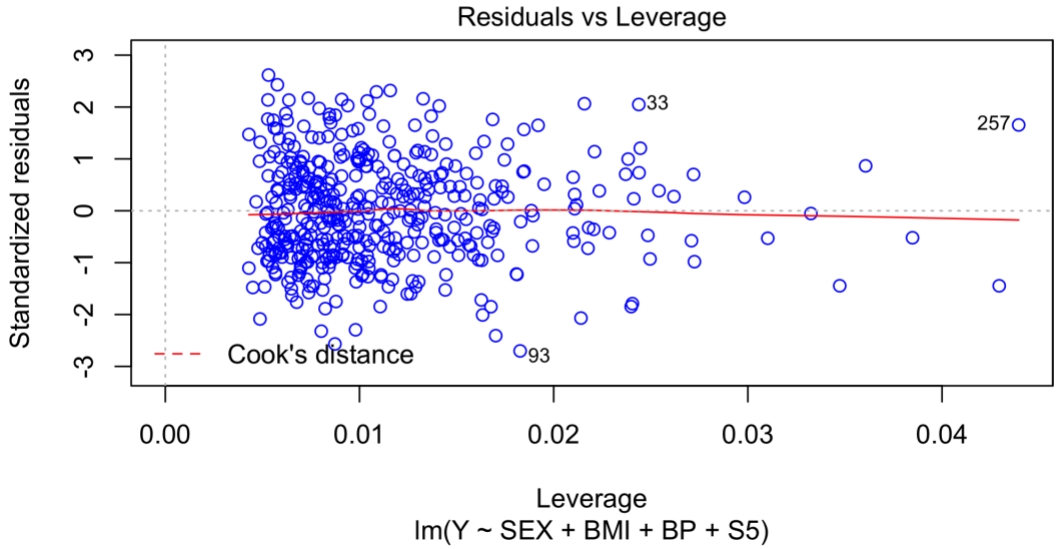

```
# Q-Q plot
plot(model_2, which=2, col=c("blue"))
```




```r
# Scale-Location
plot(model_2, which=3, col=c("blue"))
```




```r
# Residual vs Leverage
plot(model_2, which=5, col=c("blue"))
```



Question 4g

One at a time CI for coefficients. No zeroes observed in One at a time CI.

```
# Question 4g
# One at a time
confint(model_2, 'SEX2', level = .95)

##          2.5 %    97.5 %
## SEX2 -23.77167 -2.300347

confint(model_2, 'BMI', level = .95)

##        2.5 %   97.5 %
## BMI 5.084917 7.811835

confint(model_2, 'BP', level = .95)

##        2.5 %  97.5 %
## BP 0.5760387 1.44128

confint(model_2, 'S5', level = .95)

##       2.5 %   97.5 %
## S5 38.99716 62.08259
```

Question 4h

In case of large datasets with large number of predictor variables, it is often difficult to formulate an appropriate regression function immediately. Choosing the right variable

and right function is really important. Usually multiple models are tried with different subsets of predictor variables and R^2 statistic is usually considered to choose a better model. Adjusted R^2 is usually a better statistic to follow. Another approach is to follow stepwise approach for variable selection in which first step is to run simple regression with individual variables and the variable with highest correlation with dependent variable goes into the model first when trying multiple regression. Next one is the variable which has the highest significant contribution to the regression sum of squares. Next step is to perform multiple regression and the variable with no or least significance gets dropped out.

Another criteria for selecting an appropriate model is to look at the AIC (Akaike's information criteria).

Also, collinearity needs to be checked in case of multiple regression as we have done in our model 1 & 2 by calculating variable inflation factor score.