

STAT3006 Assignment 2

Chee Kitt Win

9/19/2021

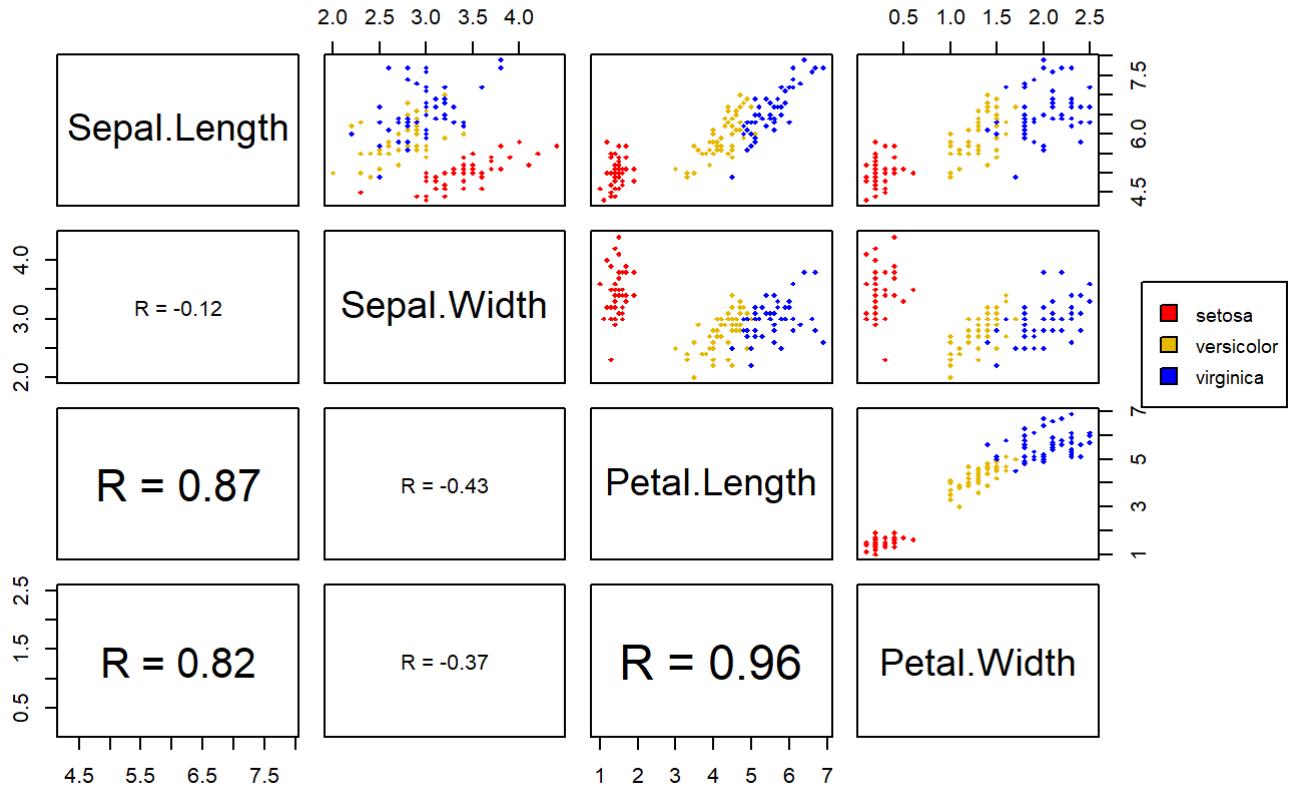
LOAD DATA

```
artificial = read.csv("C:\\\\Users\\\\Owner\\\\Desktop\\\\UQ Year 3 Sem 2 Courses\\\\STAT3006\\\\Assignment 2\\\\artificial2021.csv")
library(lattice)
data(iris)
# plot(iris)
# iris$Sepal.Length
# iris$Sepal.Width
# iris$Petal.Length
# iris$Petal.Width
# iris$Species
```

1. i) DRAFTSMAN PLOT OF IRIS DATA

```
# Correlation panel
panel.cor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), digits=2)
  txt <- paste0("R = ", r)
  cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
# Customize upper panel
my_cols <- c("#FF0000", "#E7B800", "#0000FF")
upper.panel<-function(x, y){
  points(x,y, pch = 18, col = my_cols[iris$Species], cex = 0.6)
}
# Create the plots
pairs(iris[,1:4],
      lower.panel = panel.cor,
      upper.panel = upper.panel,
      main = "Figure 1: Draftsman plot of R's inbuilt Iris dataset which consists of \n 150 observations of 3 different species of Irises, each with 50 observations. \n R denotes the correlation coefficients of the corresponding explanatory variables.",
      cex.main = 0.8,
      oma = c(3,3,7,8)
)
legend(x = "right", xpd = TRUE, fill = c("#FF0000", "#E7B800", "#0000FF"), inset=c(-0.03,0),
       legend = c(levels(iris$Species)),cex = 0.6)
```

Figure 1: Draftsman plot of R's inbuilt Iris dataset which consists of 150 observations of 3 different species of Irises, each with 50 observations. R denotes the correlation coefficients of the corresponding explanatory variables.



1. ii) CORRELATION MATRICES OF EACH SPECIES

The correlation matrices for the setosa, versicolor, and virginica species in this order are given below.

Measurements and species for which absolute value of the sample correlation is 0.7 or greater:

Setosa species: Sepal length and sepal width

Versicolor species: Sepal length and petal length

Versicolor species: Petal length and petal width

Virginica species: Petal length and sepal length

Attributes for which the sample correlation is not significant at the 0.05 level:

Setosa species: Petal length and sepal width

The sample correlation is 0.177. Conducting a t-test yielded a p value of 0.217 which is not significant at the 0.05 level. Therefore there is no evidence to suggest that the true correlation is not equal to zero. More information below.

```
cor.iris = cor(iris[,1:4])
species <- split(iris, iris$Species)
cor.setosa = cor(species[["setosa"]][,1:4])
cor.setosa
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length 1.0000000 0.7425467 0.2671758 0.2780984
## Sepal.Width 0.7425467 1.0000000 0.1777000 0.2327520
## Petal.Length 0.2671758 0.1777000 1.0000000 0.3316300
## Petal.Width 0.2780984 0.2327520 0.3316300 1.0000000
```

```
cor.versicolor = cor(species[["versicolor"]][,1:4])
cor.versicolor
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length 1.0000000 0.5259107 0.7540490 0.5464611
## Sepal.Width 0.5259107 1.0000000 0.5605221 0.6639987
## Petal.Length 0.7540490 0.5605221 1.0000000 0.7866681
## Petal.Width 0.5464611 0.6639987 0.7866681 1.0000000
```

```
cor.virginica = cor(species[["virginica"]][,1:4])
cor.virginica
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length 1.0000000 0.4572278 0.8642247 0.2811077
## Sepal.Width 0.4572278 1.0000000 0.4010446 0.5377280
## Petal.Length 0.8642247 0.4010446 1.0000000 0.3221082
## Petal.Width 0.2811077 0.5377280 0.3221082 1.0000000
```

Testing the sample correlation of petal length and sepal width of sentosa irises for significance.

```
cor.test(species[["setosa"]][,2], species[["setosa"]][,3], method = "pearson")
```

```
##
## Pearson's product-moment correlation
##
## data: species[["setosa"]][, 2] and species[["setosa"]][, 3]
## t = 1.2511, df = 48, p-value = 0.217
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.1058851 0.4345536
## sample estimates:
## cor
## 0.1777
```

1. iii) ARE THE CLASSES MULTIVARIATE NORMAL?

Mardia's test tests for multivariate normality based on skewness and kurtosis.

Applying Mardia's test to the data, we find that the classes pass the skewness and kurtosis test (retaining the null hypothesis that they are multivariate normal). However, univariate tests for petal width and petal length for some of the classes have been found to give a significant p-value (rejecting the null hypothesis that they are univariate normal). The virginica class passed all the tests, but petal width was borderline significant with a p-value of 0.0508.

####Reference: ####Mardia, K. V. (1970). Measures of Multivariate Skewness and Kurtosis with Applications. Biometrika, 57(3), 519–530. <https://doi.org/10.2307/2334770> (<https://doi.org/10.2307/2334770>)

In conclusion, the classes are unlikely to be multivariate normal. However, considering the small sample size, and given that the only problems seem to be petal width and petal length, we can make the case that the data is “normal enough” depending on our objective. For instance, with clustering, we might find that a Gaussian mixture model works well as long as the means are reasonably well spread out. For hypothesis tests, it depends on the test and how robust it is to non-normality.

```
library(MVN)

## Warning: package 'MVN' was built under R version 4.1.1

mvn(data = species$setosa[,1:4], mvnTest = "mardia")

## $multivariateNormality
##           Test      Statistic      p value Result
## 1 Mardia Skewness 25.6643445196298 0.177185884467652    YES
## 2 Mardia Kurtosis 1.29499223711605 0.195322907441935    YES
## 3       MVN          <NA>          <NA>    YES
##
## $univariateNormality
##           Test      Variable Statistic      p value Normality
## 1 Anderson-Darling Sepal.Length   0.4080  0.3352      YES
## 2 Anderson-Darling Sepal.Width   0.4910  0.2102      YES
## 3 Anderson-Darling Petal.Length  1.0073  0.0108       NO
## 4 Anderson-Darling Petal.Width   4.7148 <0.001       NO
##
## $Descriptives
##           n   Mean Std.Dev Median Min Max 25th 75th      Skew      Kurtosis
## Sepal.Length 50 5.006 0.3524897   5.0 4.3 5.8 4.8 5.200 0.11297784 -0.4508724
## Sepal.Width  50 3.428 0.3790644   3.4 2.3 4.4 3.2 3.675 0.03872946  0.5959507
## Petal.Length 50 1.462 0.1736640   1.5 1.0 1.9 1.4 1.575 0.10009538  0.6539303
## Petal.Width  50 0.246 0.1053856   0.2 0.1 0.6 0.2 0.300 1.17963278  1.2587179
```

```
mvn(data = species$versicolor[,1:4], mvnTest = "mardia")
```

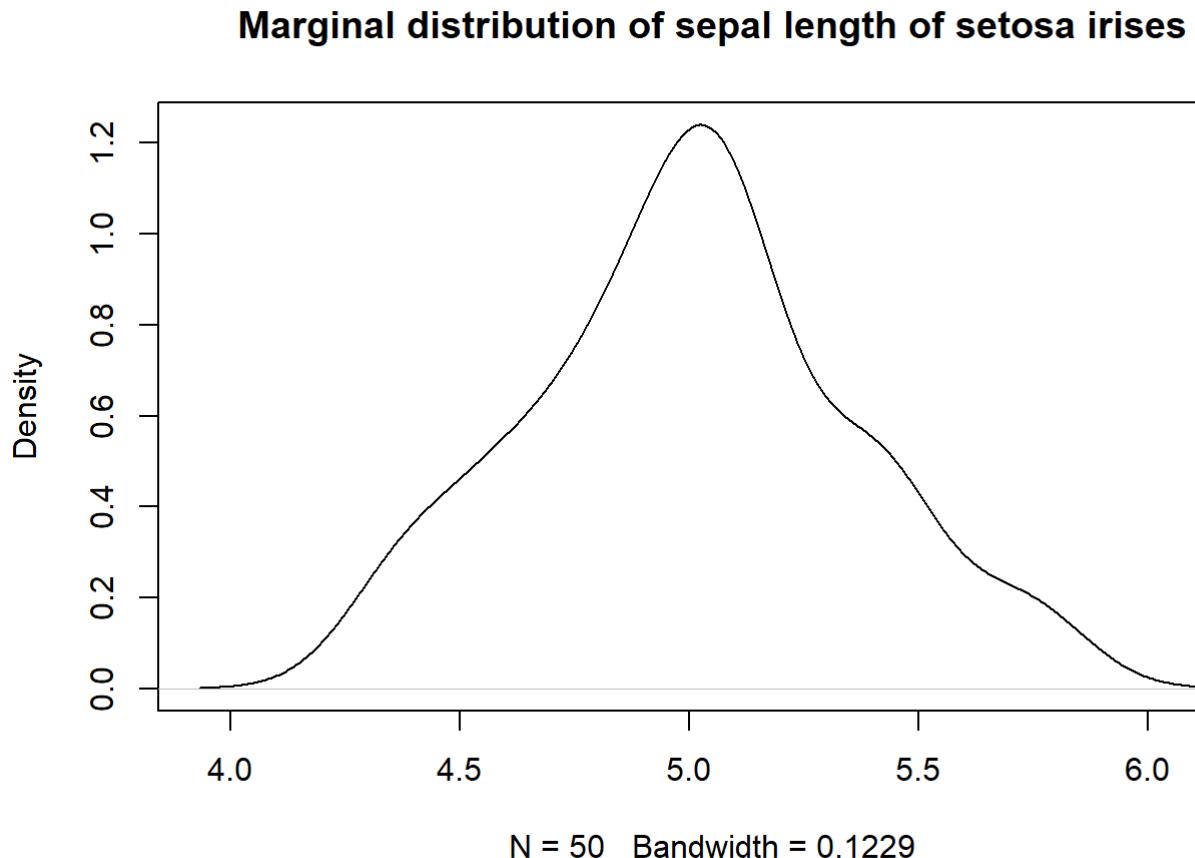
```
## $multivariateNormality
##              Test      Statistic      p value Result
## 1 Mardia Skewness  25.1850115362466 0.194444483140265    YES
## 2 Mardia Kurtosis -0.57186635893429 0.567412516528727    YES
## 3          MVN           <NA>           <NA>    YES
##
## $univariateNormality
##              Test      Variable Statistic      p value Normality
## 1 Anderson-Darling Sepal.Length   0.3608     0.4333    YES
## 2 Anderson-Darling Sepal.Width   0.5598     0.1406    YES
## 3 Anderson-Darling Petal.Length  0.5551     0.1446    YES
## 4 Anderson-Darling Petal.Width   0.9569     0.0144     NO
##
## $Descriptives
##             n  Mean Std.Dev Median Min Max 25th 75th      Skew
## Sepal.Length 50 5.936 0.5161711  5.90 4.9 7.0 5.600 6.3 0.09913926
## Sepal.Width  50 2.770 0.3137983  2.80 2.0 3.4 2.525 3.0 -0.34136443
## Petal.Length 50 4.260 0.4699110  4.35 3.0 5.1 4.000 4.6 -0.57060243
## Petal.Width  50 1.326 0.1977527  1.30 1.0 1.8 1.200 1.5 -0.02933377
##                 Kurtosis
## Sepal.Length -0.6939138
## Sepal.Width  -0.5493203
## Petal.Length -0.1902555
## Petal.Width  -0.5873144
```

```
mvn(data = species$virginica[,1:4], mvnTest = "mardia")
```

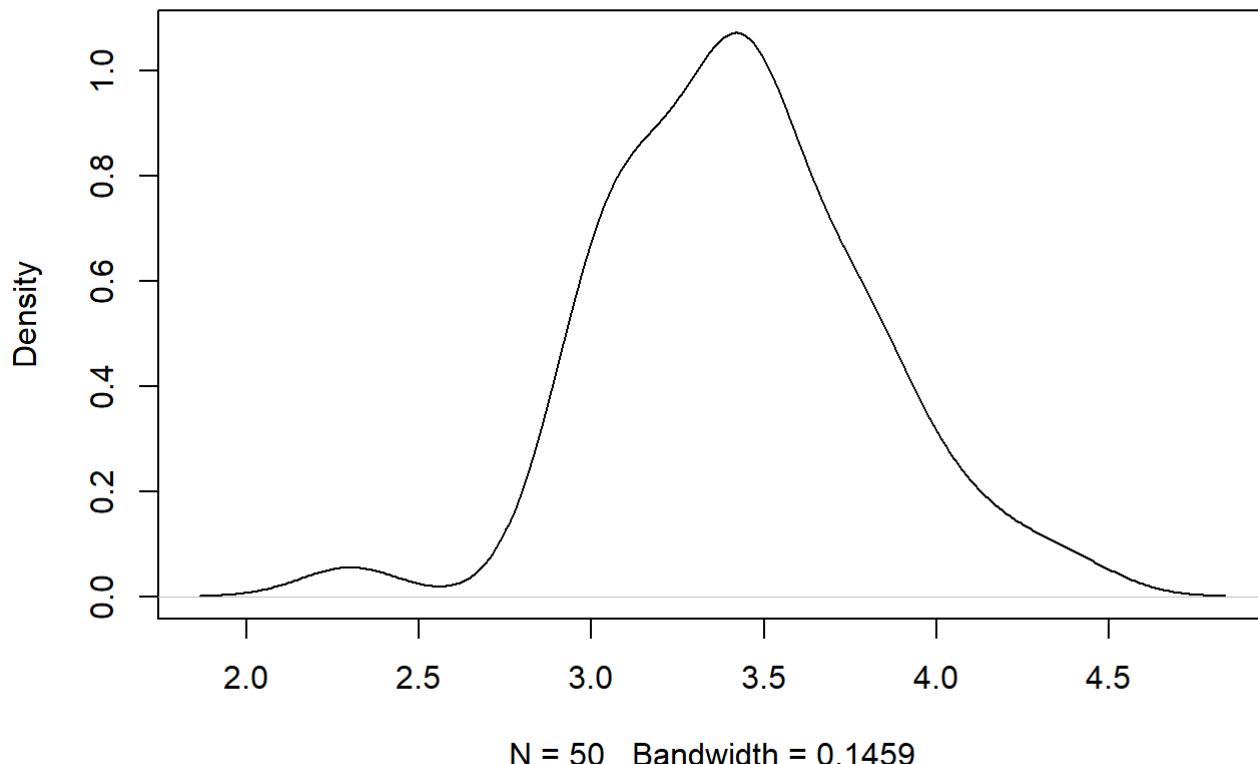
```
## $multivariateNormality
##              Test      Statistic      p value Result
## 1 Mardia Skewness  26.2705981752915 0.157059707690356    YES
## 2 Mardia Kurtosis 0.152614173978342 0.878702546726567    YES
## 3          MVN           <NA>           <NA>    YES
##
## $univariateNormality
##              Test      Variable Statistic      p value Normality
## 1 Anderson-Darling Sepal.Length   0.5516     0.1475    YES
## 2 Anderson-Darling Sepal.Width   0.6182     0.1018    YES
## 3 Anderson-Darling Petal.Length  0.6090     0.1074    YES
## 4 Anderson-Darling Petal.Width   0.7388     0.0508    YES
##
## $Descriptives
##             n  Mean Std.Dev Median Min Max 25th 75th      Skew
## Sepal.Length 50 6.588 0.6358796  6.50 4.9 7.9 6.225 6.900 0.1110286
## Sepal.Width  50 2.974 0.3224966  3.00 2.2 3.8 2.800 3.175 0.3442849
## Petal.Length 50 5.552 0.5518947  5.55 4.5 6.9 5.100 5.875 0.5169175
## Petal.Width  50 2.026 0.2746501  2.00 1.4 2.5 1.800 2.300 -0.1218119
##                 Kurtosis
## Sepal.Length -0.2032597
## Sepal.Width   0.3803832
## Petal.Length -0.3651161
## Petal.Width  -0.7539586
```

1. iv) SAMPLE MARGINAL DISTRIBUTIONS FOR EACH DIMENSION FOR EACH CLASS

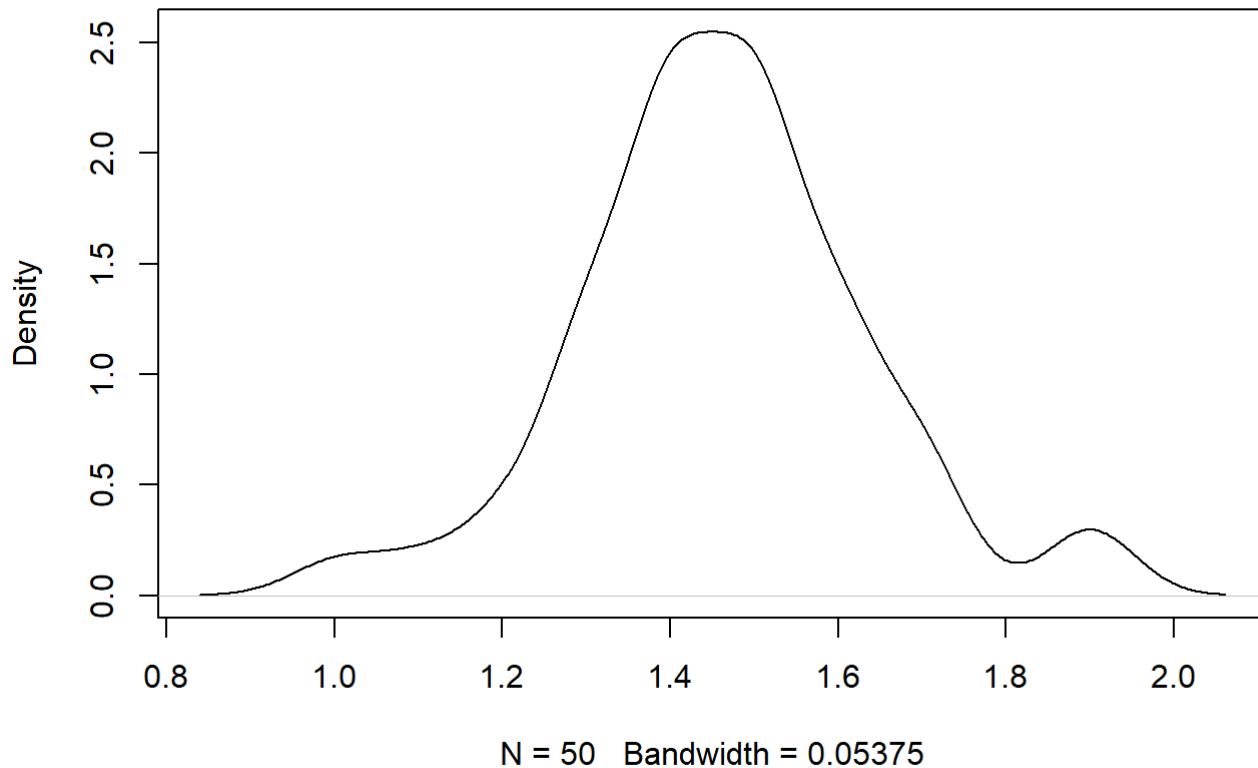
```
plot(density(species$setosa$Sepal.Length), main = "Marginal distribution of sepal length of setosa irises")
```



```
plot(density(species$setosa$Sepal.Width), main = "Marginal distribution of sepal width of setosa irises")
```

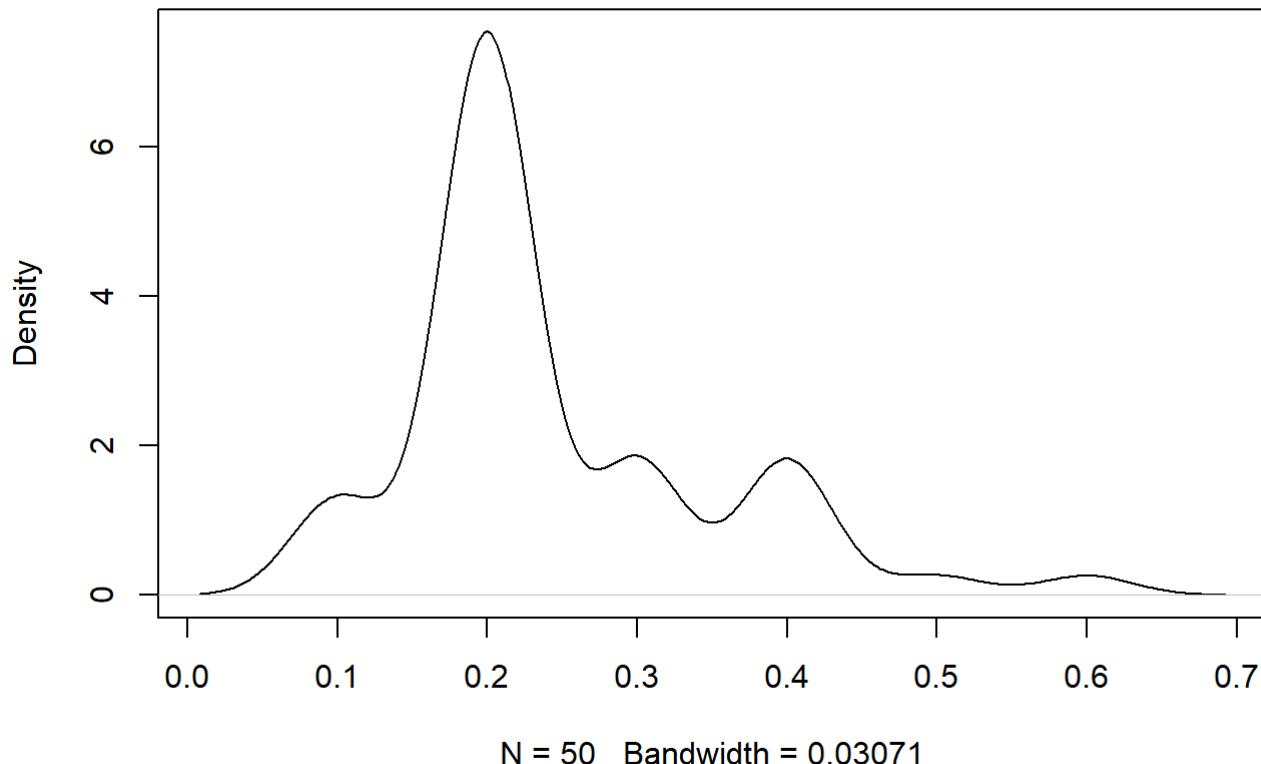
Marginal distribution of sepal width of setosa irises

```
plot(density(species$setosa$Petal.Length), main = "Marginal distribution of petal length of setosa irises")
```

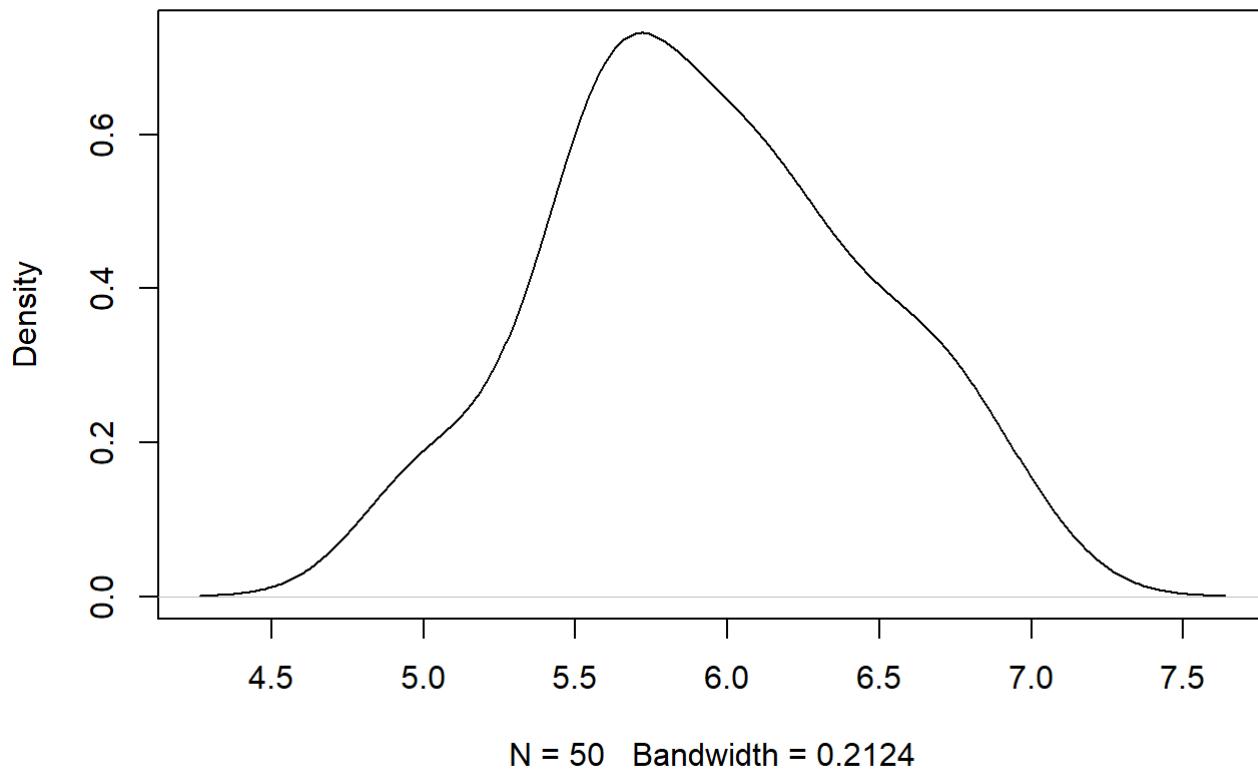
Marginal distribution of petal length of setosa irises

```
plot(density(species$setosa$Petal.Width), main = "Marginal distribution of petal width of setosa irises")
```

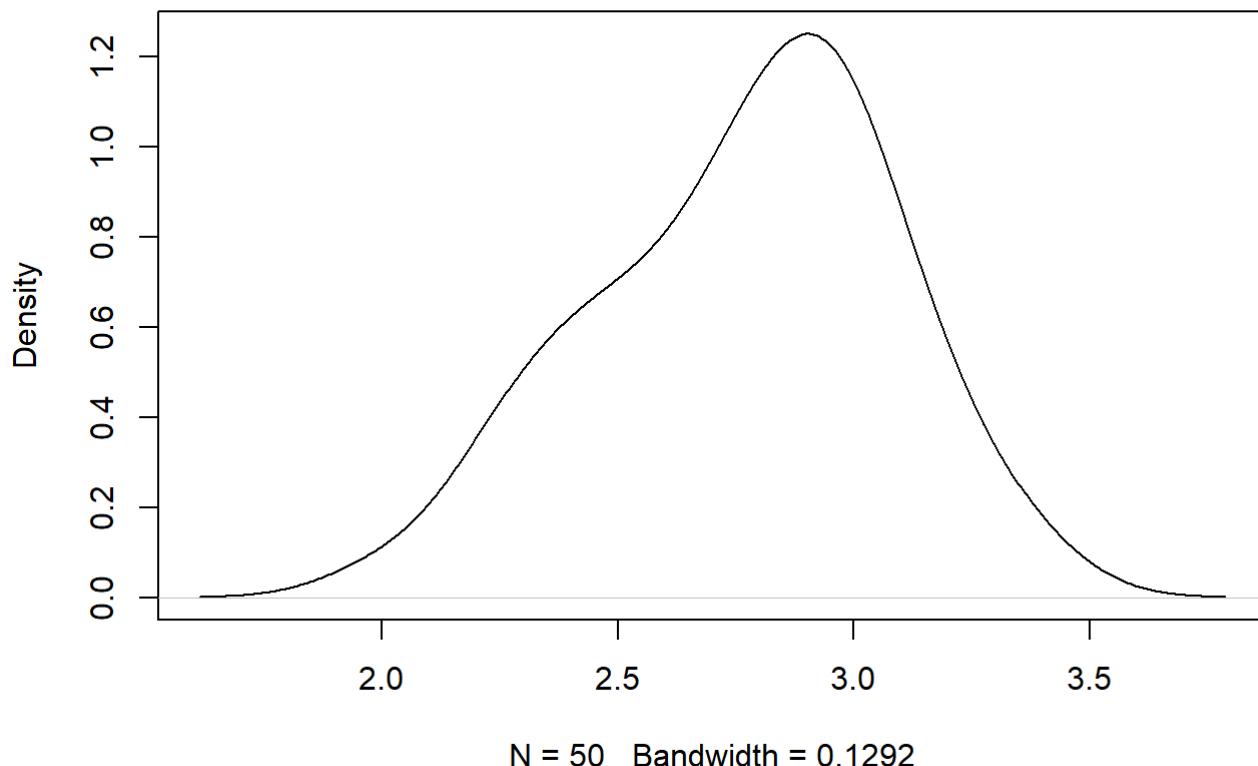
Marginal distribution of petal width of setosa irises



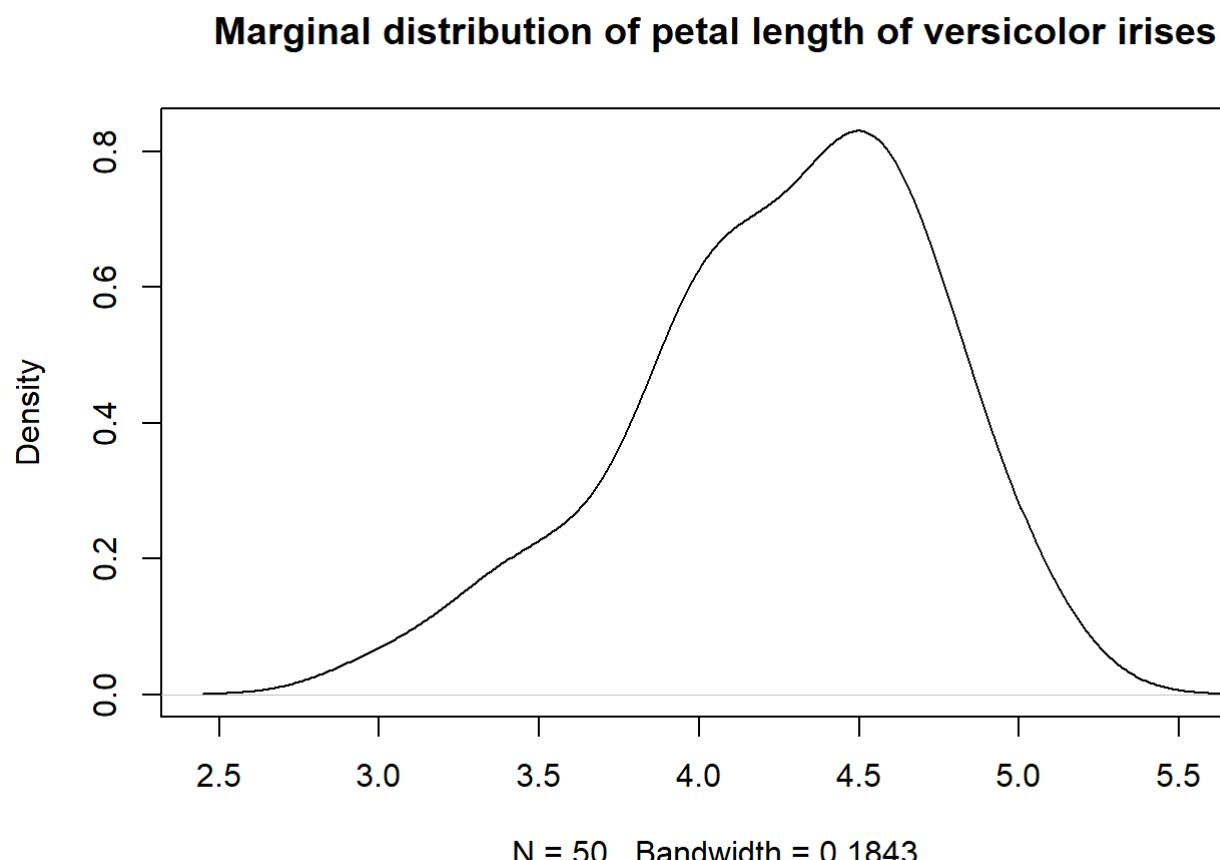
```
plot(density(species$versicolor$Sepal.Length), main = "Marginal distribution of sepal length of versicolor irises")
```

Marginal distribution of sepal length of versicolor irises

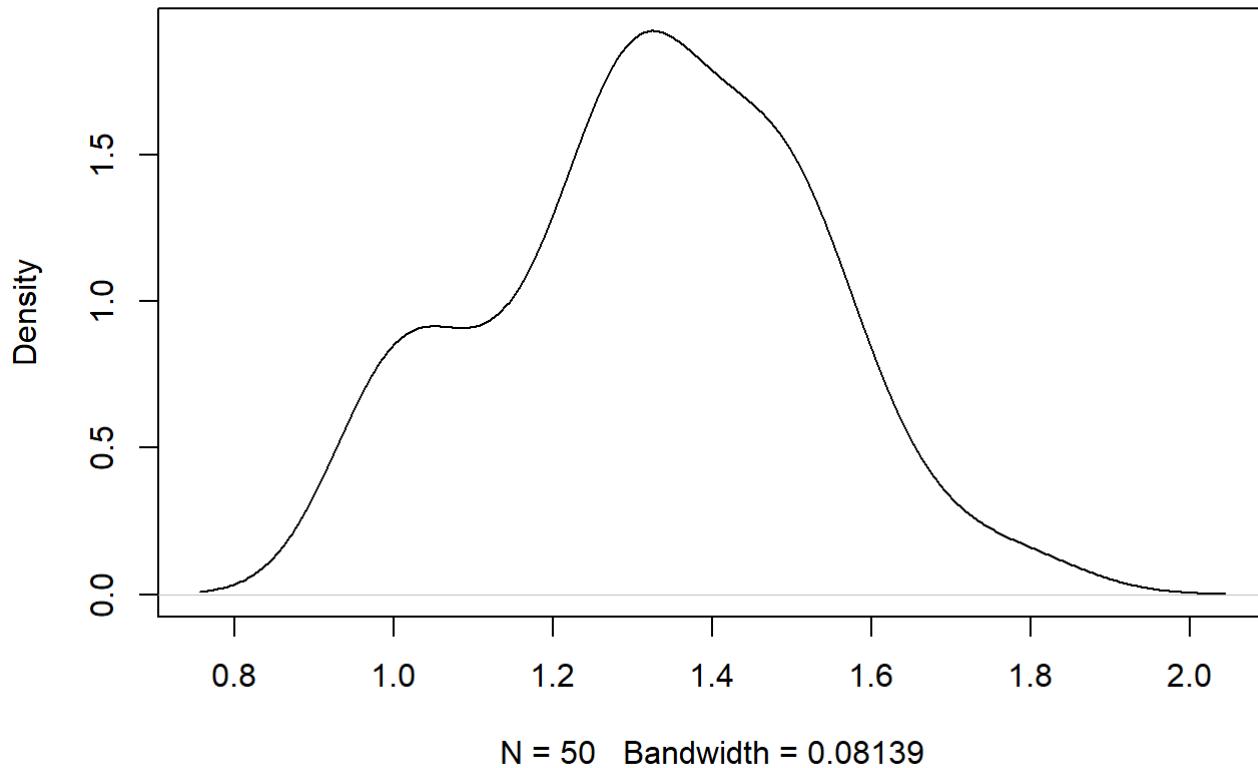
```
plot(density(species$versicolor$Sepal.Length), main = "Marginal distribution of sepal length of versicolor irises")
```

Marginal distribution of sepal width of versicolor irises

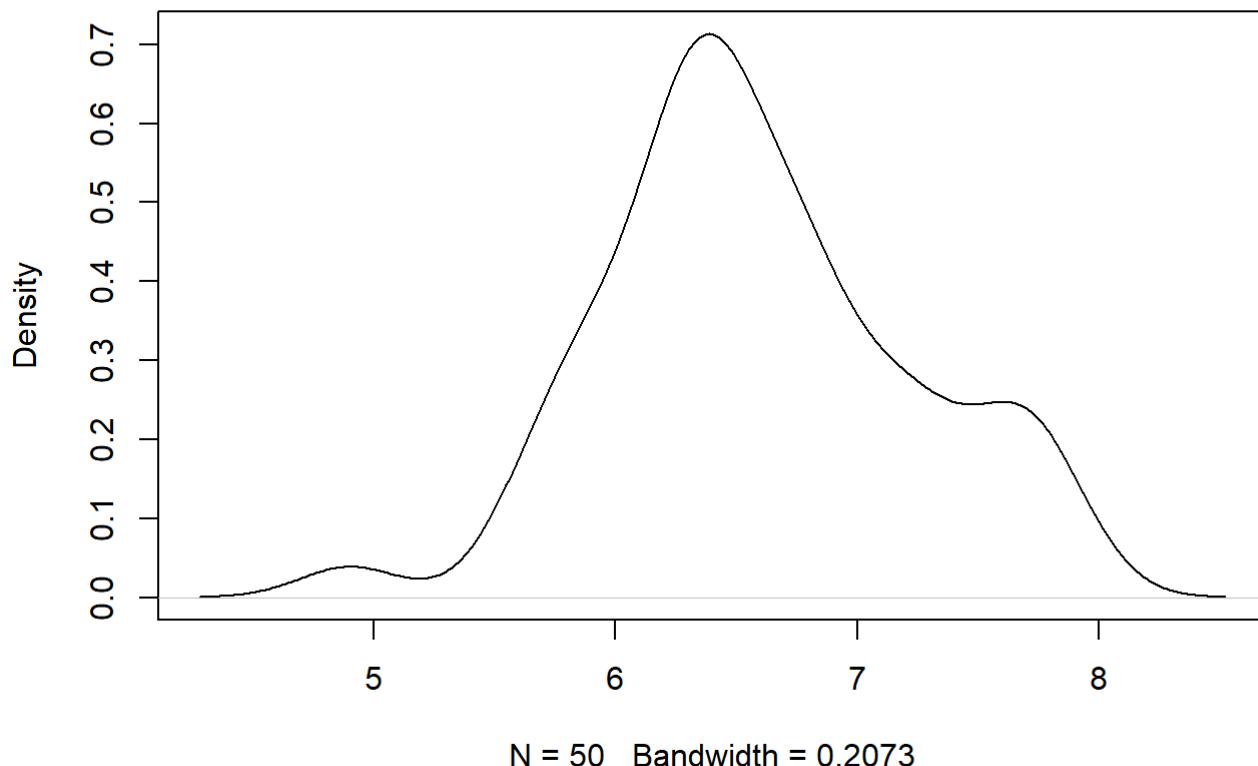
```
plot(density(species$versicolor$Petal.Length), main = "Marginal distribution of petal length  
of versicolor irises")
```



```
plot(density(species$versicolor$Petal.Width), main = "Marginal distribution of petal width of  
versicolor irises")
```

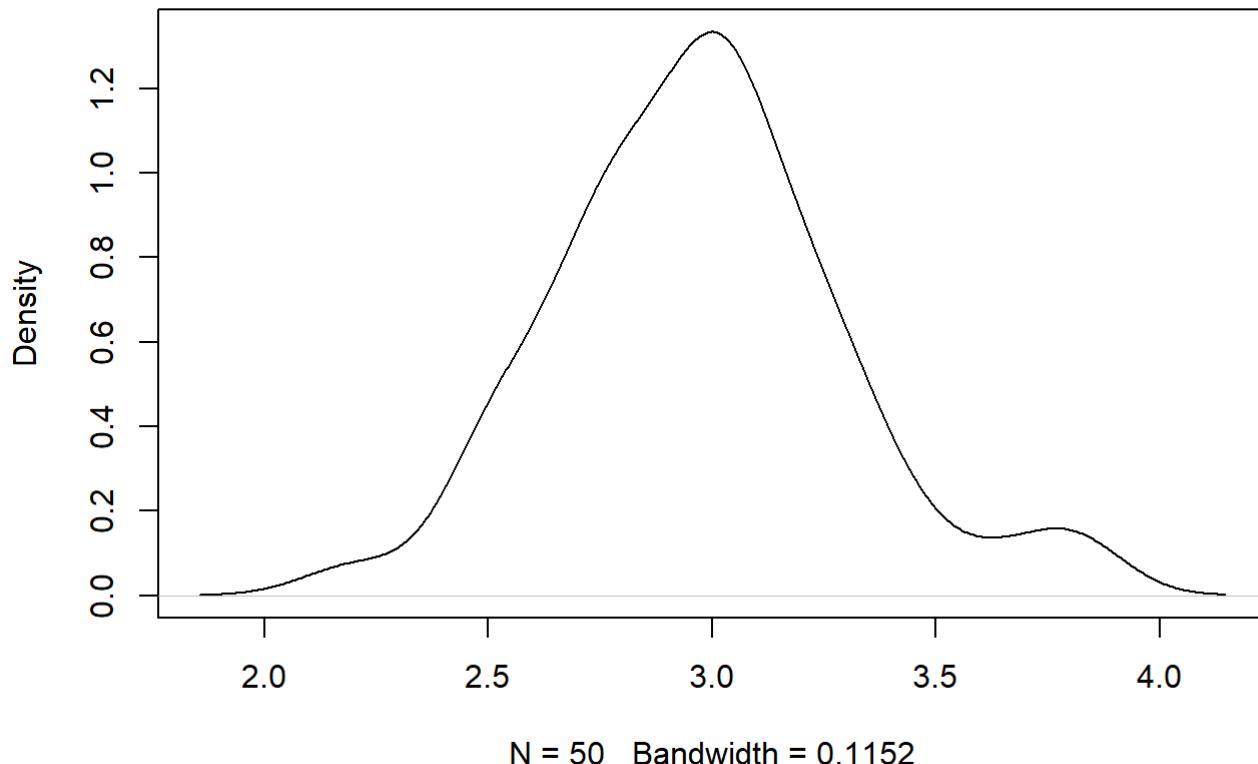
Marginal distribution of petal width of versicolor irises

```
plot(density(species$virginica$Sepal.Length), main = "Marginal distribution of sepal length o  
f virginica irises")
```

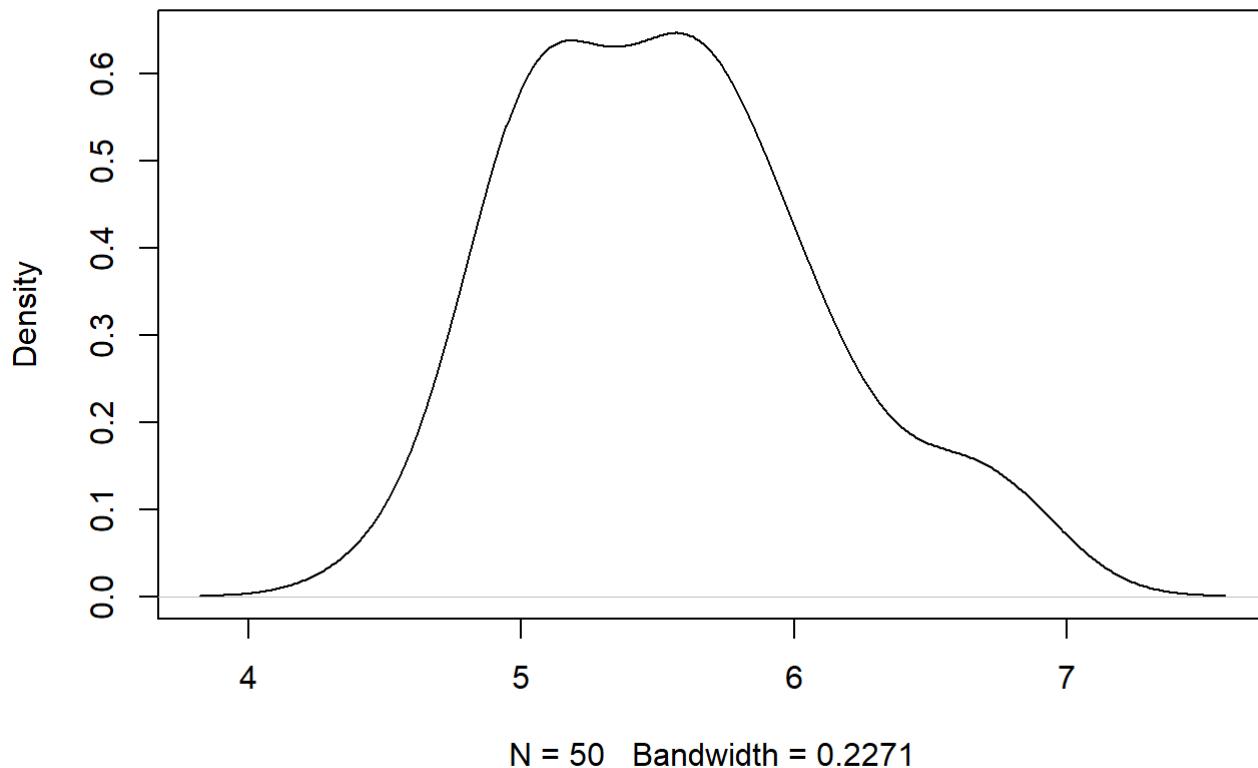
Marginal distribution of sepal length of virginica irises

```
plot(density(species$virginica$Sepal.Width), main = "Marginal distribution of sepal width of virginica irises")
```

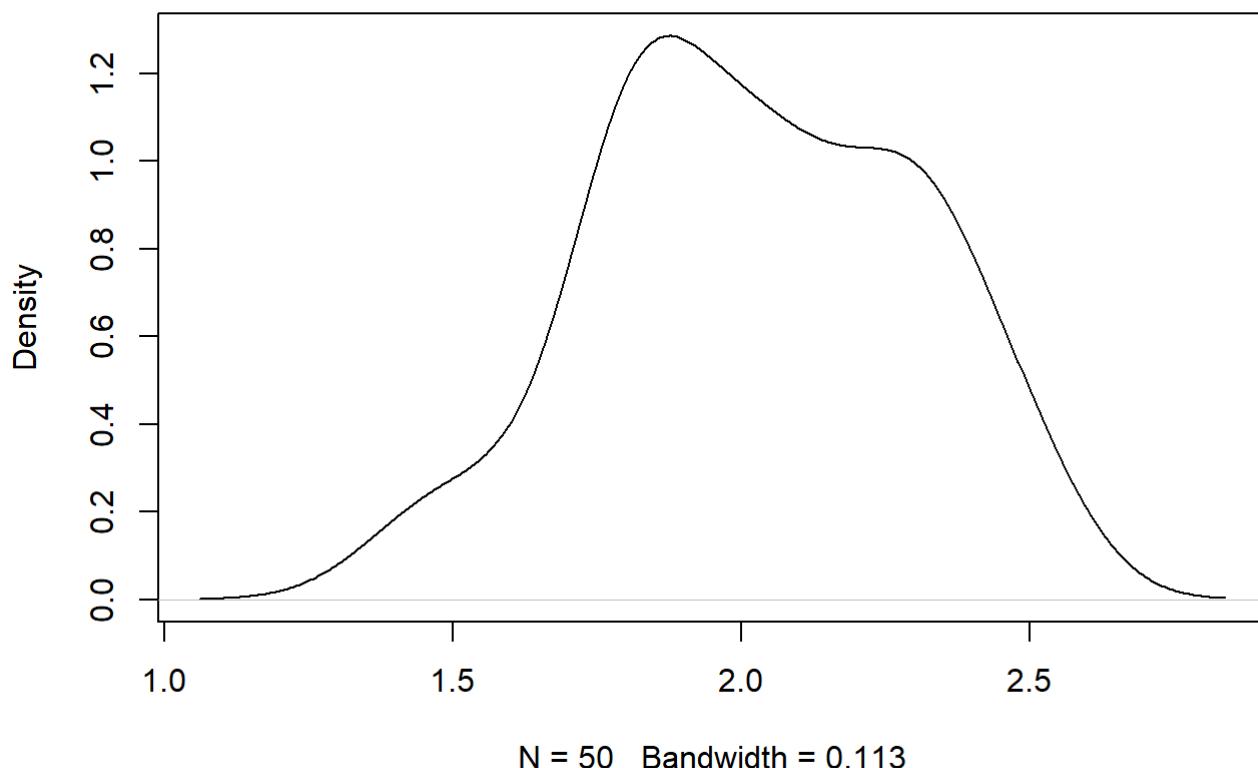
Marginal distribution of sepal width of virginica irises



```
plot(density(species$virginica$Petal.Length), main = "Marginal distribution of petal length of virginica irises")
```

Marginal distribution of petal length of virginica irises

```
plot(density(species$virginica$Petal.Length), main = "Marginal distribution of petal length of virginica irises")
```

Marginal distribution of petal width of virginica irises

1. iv) FITTING A MULTIVARIATE NORMAL DISTRIBUTION TO EACH CLASS USING MLE

To do this we find the maximum likelihood estimates of the mean vectors and covariance matrices for each class. This has been done below. We now have multivariate normal models for each class.

A caveat is that we have assumed that the observations are multivariate normal. This means that their marginal distributions should be univariate normal. However, looking at the above plots, the marginal distributions agree with the results in the previous section, i.e most of them look normal, but there are one or two which are definitely not normal, e.g petal width for setosa irises.

The marginal distributions are as follows (the mathematical form for each is simply the univariate Gaussian pdf with the mean and variance given below):

Setosa_sepal_length ~ N(5.006,0.121764)
 Setosa_sepal_width ~ N(3.428,0.140816)
 Setosa_petal_length ~ N(1.462,0.029556)
 Setosa_petal_width ~ N(0.246,0.010884)
 Versicolor_sepal_length ~ N(5.936,0.261104)
 Versicolor_sepal_width ~ N(2.770,0.09650)
 Versicolor_petal_length ~ N(4.260,0.21640)
 Versicolor_petal_width ~ N(1.326,0.038324)
 Virginica_sepal_length ~ N(6.588,0.396256)
 Virginica_sepal_width ~ N(2.974,0.101924)
 Virginica_petal_length ~ N(5.552,0.298496)
 Virginica_petal_width ~ N(2.026,0.073924)

```
# MLE of the mean vectors for setosa, versicolor and virginica irises
mu_setosa = colMeans(species[["setosa"]][,1:4])
print(mu_setosa)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##      5.006       3.428      1.462       0.246
```

```
mu_versicolor = colMeans(species[["versicolor"]][,1:4])
print(mu_versicolor)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
##      5.936       2.770       4.260       1.326
```

```
mu_virginica = colMeans(species[["virginica"]][,1:4])
print(mu_virginica)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
##      6.588       2.974       5.552       2.026
```

```
# MLE of the covariance matrix for setosa, versicolor and virginica irises
cov_setosa = (49/50)*cov(species[["setosa"]][,1:4])
cov_setosa
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.121764   0.097232   0.016028   0.010124
## Sepal.Width      0.097232   0.140816   0.011464   0.009112
## Petal.Length     0.016028   0.011464   0.029556   0.005948
## Petal.Width      0.010124   0.009112   0.005948   0.010884
```

```
cov_versicolor = (49/50)*cov(species[["versicolor"]][,1:4])
cov_versicolor
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.261104   0.08348    0.17924    0.054664
## Sepal.Width      0.083480   0.09650    0.08100    0.040380
## Petal.Length     0.179240   0.08100    0.21640    0.071640
## Petal.Width      0.054664   0.04038    0.07164    0.038324
```

```
cov_virginica = (49/50)*cov(species[["virginica"]][,1:4])
cov_virginica
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.396256   0.091888   0.297224   0.048112
## Sepal.Width      0.091888   0.101924   0.069952   0.046676
## Petal.Length     0.297224   0.069952   0.298496   0.047848
## Petal.Width      0.048112   0.046676   0.047848   0.073924
```

1. v) CONDITIONAL DISTRIBUTION FOR SEPAL LENGTH AND WIDTH CONDITIONED ON SAMPLE MEANS FOR PETAL LENGTH AND WIDTH (VIRGINICA CLASS)

The conditional distributions are also multivariate normal, and the means and covariance matrices are given below.

```

mu1 = c(6.588,2.974)
mu2 = c(5.552,2.026)
sigma11 = matrix(c(0.396256, 0.091888, 0.091888, 0.101924), nrow=2,ncol=2)
sigma22 = matrix(c(0.298496, 0.047848, 0.047848, 0.073924), nrow=2,ncol=2)
sigma12 = matrix(c(0.297224, 0.069952, 0.048112, 0.046676), nrow=2,ncol=2)
sigma21 = matrix(c(0.297224, 0.048112, 0.069952, 0.046676), nrow=2,ncol=2)

library(matlib)

```

```
## Warning: package 'matlib' was built under R version 4.1.1
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.1
```

```

# Conditional distribution for sepal length and sepal width conditioned on the sample means f
or petal length and petal width for virginica irises
# Mean
cond_mu = mu1
cond_mu

```

```
## [1] 6.588 2.974
```

```

# Covariance matrix
cond_sigma = sigma11 - sigma12%*%solve(sigma22)%*%sigma21
cond_sigma

```

```

## [,1]      [,2]
## [1,] 0.10029528 0.02198365
## [2,] 0.02198365 0.06654914

```

```

set.seed(5)
slsw_c_muplpw <- mvrnorm(5000, mu = cond_mu, Sigma = cond_sigma)

# Conditional distribution for petal length and petal width conditioned on the sample means f
or sepal length and sepal width for virginica irises
# Mean
cond2_mu = mu2
cond2_mu

```

```
## [1] 5.552 2.026
```

```

# Covariance matrix
cond2_sigma = sigma22 - sigma21%*%solve(sigma11)%*%sigma12
cond2_sigma

```

```

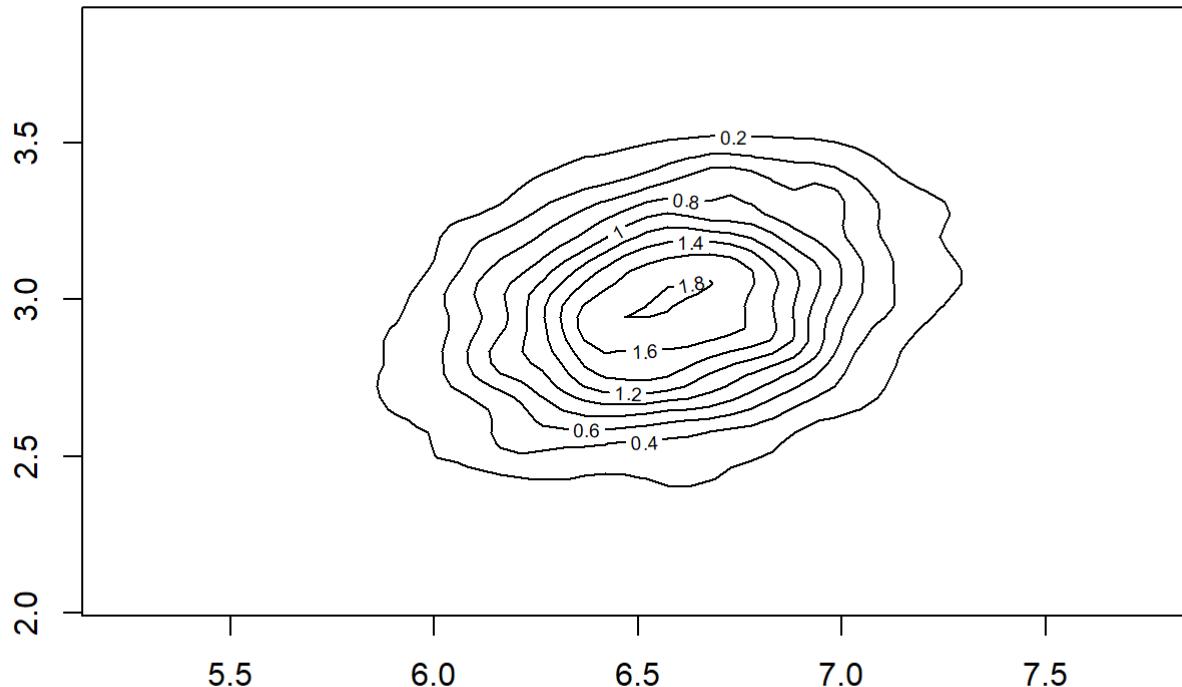
## [,1]      [,2]
## [1,] 0.07554087 0.01130692
## [2,] 0.01130692 0.05243268

```

```
set.seed(5)
plpw_c_muslsw = mvrnorm(5000, mu = cond2_mu, Sigma = cond2_sigma)

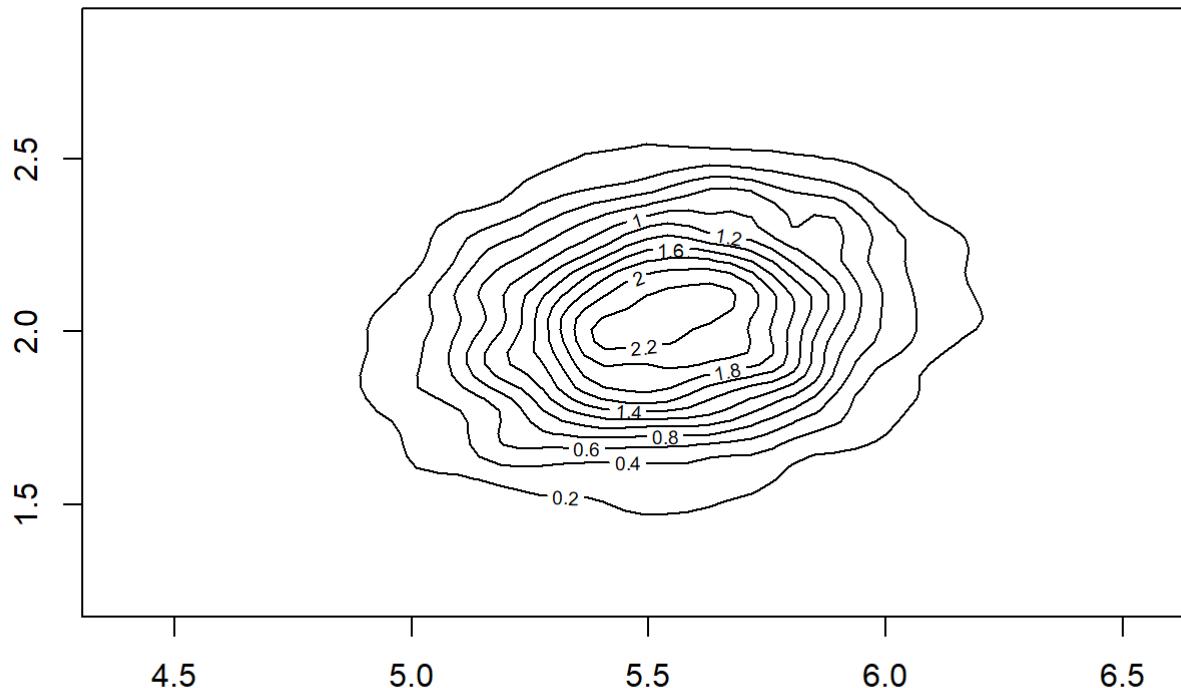
contour(kde2d(slsw_c_muplpw[,1], slsw_c_muplpw[,2], n = 50), main = "Contour plot of sepal length and sepal width conditioned \n on sample mean of petal length and pepal width (Virginica irises)", cex.main = 0.8)
```

**Contour plot of sepal length and sepal width conditioned
on sample mean of petal length and pepal width (Virginica irises)**



```
contour(kde2d(plpw_c_muslsw[,1], plpw_c_muslsw[,2], n = 50), main = "Contour plot of petal length and petal width conditioned \n on sample mean of sepal length and sepal width (Virginica irises)", cex.main = 0.8)
```

**Contour plot of petal length and petal width conditioned
on sample mean of sepal length and sepal width (Virginica irises)**



1. vi) REPRESENTING THE OBSERVATIONS ON THE PLOT OF A CONDITIONAL DISTRIBUTION.

1. vii) RESIDUAL MATRIX AND PLOTS

Based on the plots, it is hard to tell. There is too little data. If there was no other information to go by, I would say that it is not impossible that they are normally distributed.

```
# MLE of the mean vectors for setosa, versicolor and virginica irises
mu_setosa = colMeans(species[["setosa"]][,1:4])
print(mu_setosa)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##      5.006        3.428     1.462       0.246
```

```
mu_versicolor = colMeans(species[["versicolor"]][,1:4])
print(mu_versicolor)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##      5.936        2.770     4.260       1.326
```

```
mu_virginica = colMeans(species[["virginica"]][,1:4])
print(mu_virginica)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
##       6.588      2.974      5.552      2.026
```

```
# MLE of the covariance matrix for setosa, versicolor and virginica irises
cov_setosa = (49/50)*cov(species[["setosa"]][,1:4])
cov_setosa
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.121764    0.097232   0.016028   0.010124
## Sepal.Width      0.097232    0.140816   0.011464   0.009112
## Petal.Length     0.016028    0.011464   0.029556   0.005948
## Petal.Width      0.010124    0.009112   0.005948   0.010884
```

```
cov_versicolor = (49/50)*cov(species[["versicolor"]][,1:4])
cov_versicolor
```

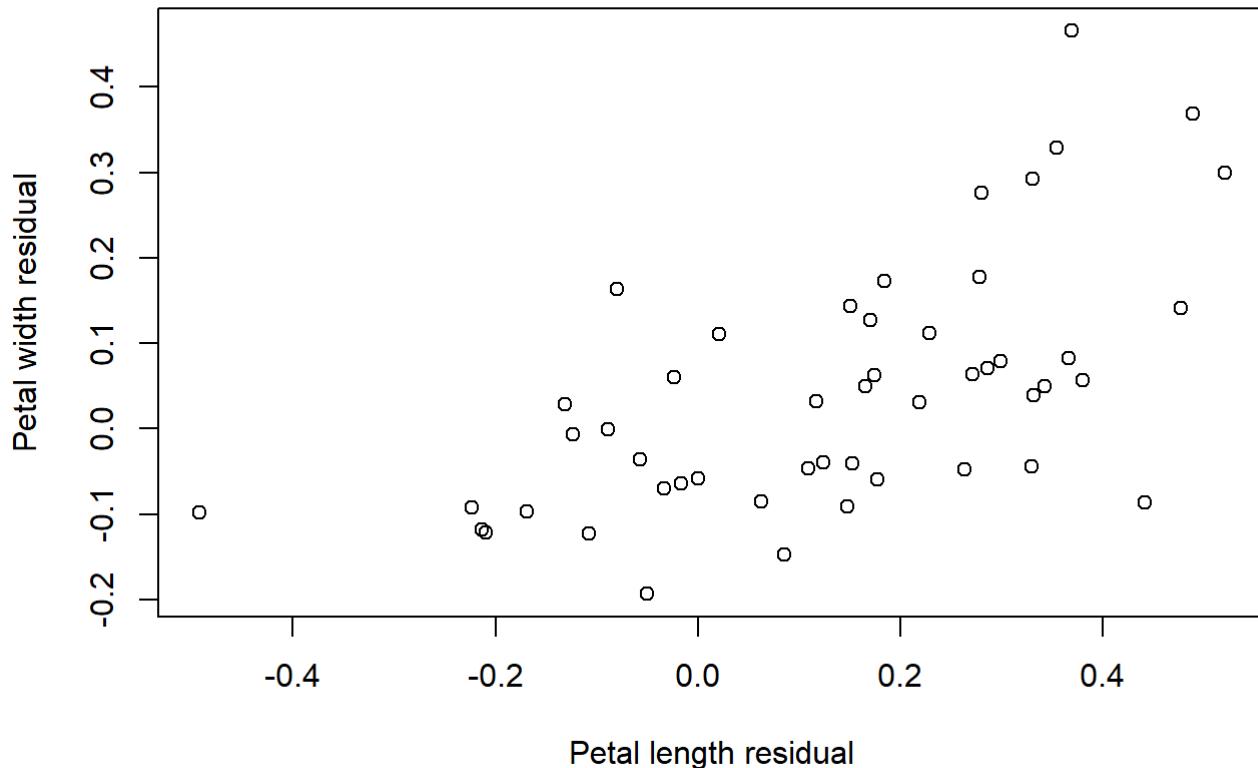
```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.261104    0.08348    0.17924    0.054664
## Sepal.Width      0.083480    0.09650    0.08100    0.040380
## Petal.Length     0.179240    0.08100    0.21640    0.071640
## Petal.Width      0.054664    0.04038    0.07164    0.038324
```

```
cov_virginica = (49/50)*cov(species[["virginica"]][,1:4])
cov_virginica
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length     0.396256    0.091888   0.297224   0.048112
## Sepal.Width      0.091888    0.101924   0.069952   0.046676
## Petal.Length     0.297224    0.069952   0.298496   0.047848
## Petal.Width      0.048112    0.046676   0.047848   0.073924
```

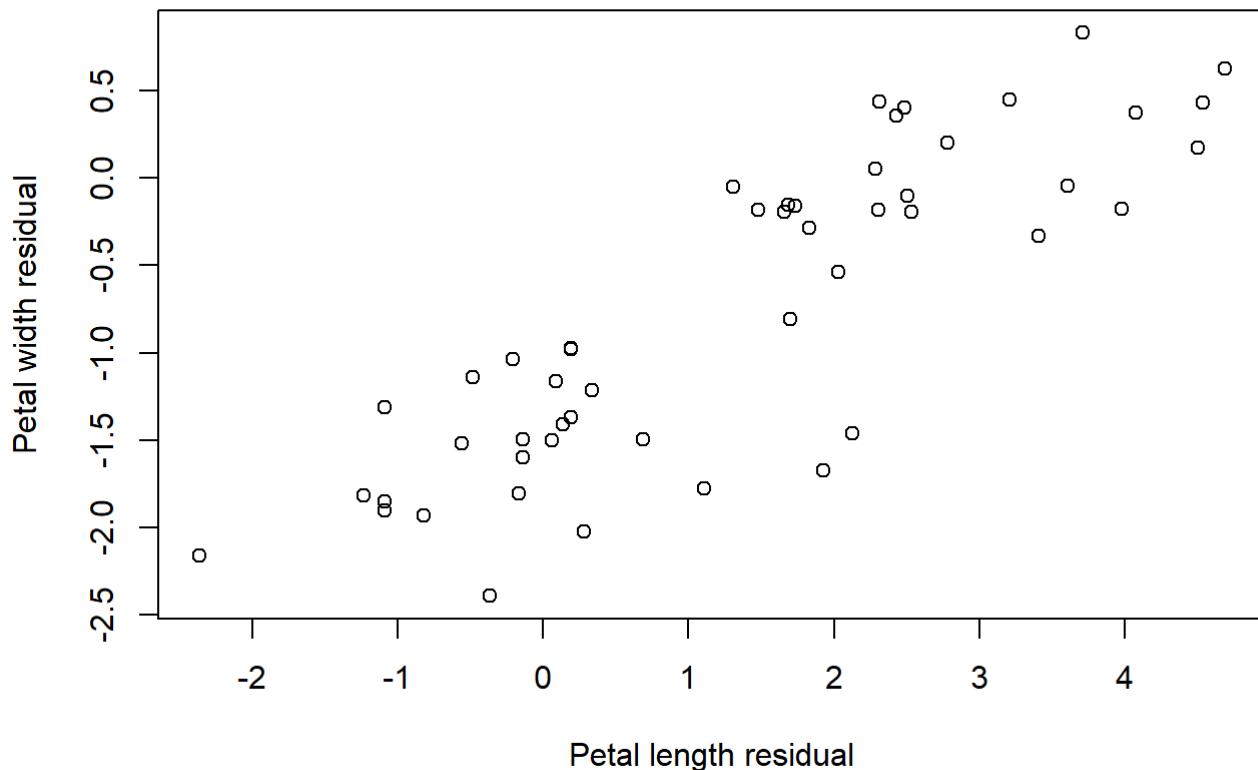
```
res_setosa = matrix(NA, nrow = 2, ncol = 50)
for (i in 1:50){
  res_setosa[,i] = t(species$setosa[i,3:4]) - matrix(colMeans(species$setosa[,3:4])) + cov_setosa[c(3,4),c(1,2)]%*%solve(cov_setosa[c(1,2),c(1,2)])%*%t((species$setosa[,1:2]-colMeans(species$setosa[,1:2]))[i,])
}
plot(t(res_setosa),xlab = "Petal length residual", ylab = "Petal width residual", main = "Setosa residuals")
```

Setosa residuals



```
res_virginica = matrix(NA, nrow = 2, ncol = 50)
for (i in 1:50){
  res_virginica[,i] = t(species$virginica[i,3:4]) - matrix(colMeans(species$virginica[,3:4]))
+  cov_virginica[c(3,4),c(1,2)]%*%solve(cov_virginica[c(1,2),c(1,2)])%*%t((species$virginica[,1:2]-colMeans(species$virginica[,1:2]))[i,])
}
plot(t(res_virginica), xlab = "Petal length residual", ylab = "Petal width residual", main =
"Virginica residuals")
```

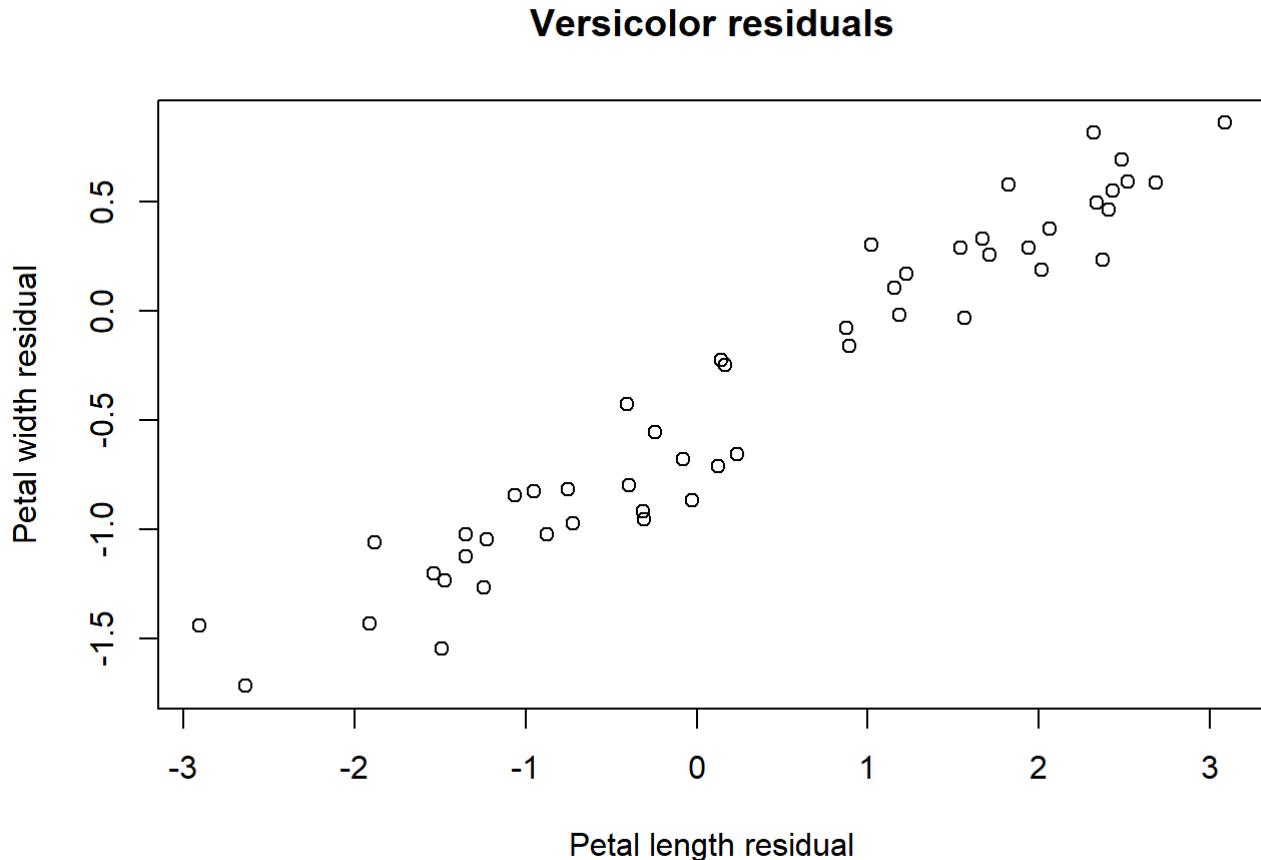
Virginica residuals



```

res_versicolor = matrix(NA, nrow = 2, ncol = 50)
for (i in 1:50){
  res_versicolor[,i] = t(species$versicolor[i,3:4]) - matrix(colMeans(species$versicolor[,3:4])) + cov_versicolor[c(3,4),c(1,2)]%*%solve(cov_versicolor[c(1,2),c(1,2)])%*%t((species$versicolor[,1:2]-colMeans(species$versicolor[,1:2]))[i,])
}
plot(t(res_versicolor),xlab = "Petal length residual", ylab = "Petal width residual", main = "Versicolor residuals")

```



1. viii) a) MAHALANOBIS DISTANCE BETWEEN EACH PAIR OF SPECIES

Mahalanobis distance for:

Setosa and Versicolor: 9.384397

Setosa and Virginica: 13.258847

Virginica and Versicolor: 4.105733

Assumptions: Common covariance matrix estimated by pooling. Given the calculated covariance matrices for each class in the earlier sections and also just from a visual test looking at the spread of the data, I think this is reasonable.

```
library(HDMD)
```

```
## Warning: package 'HDMD' was built under R version 4.1.1
```

```
## Loading required package: psych
```

```
## Warning: package 'psych' was built under R version 4.1.1
```

```
##  
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:matlib':  
##  
##     tr
```

```
library(Morpho)
```

```
## Warning: package 'Morpho' was built under R version 4.1.1
```

```
Mahala = pairwise.mahalanobis(iris[,1:4], grouping = t(iris[,5]), cov = (50/49)*covW(iris[,1:4], t(iris[,5])), inverted = FALSE)
```

```
## Warning in if (dim(cov) != c(p, p)) stop("cov matrix not of dim = (p,p)\n"): the  
## condition has length > 1 and only the first element will be used
```

```
Mahala
```

```
## $means  
##      1   2   3   4  
## setosa  5.006 3.428 1.462 0.246  
## versicolor 5.936 2.770 4.260 1.326  
## virginica  6.588 2.974 5.552 2.026  
##  
## $cov  
##           Sepal.Length Sepal.Width Petal.Length Petal.Width  
## Sepal.Length  0.27041649  0.09461336  0.17093294  0.03918506  
## Sepal.Width   0.09461336  0.11774261  0.05637096  0.03337776  
## Petal.Length  0.17093294  0.05637096  0.18896710  0.04353603  
## Petal.Width   0.03918506  0.03337776  0.04353603  0.04273636  
## attr(),"means")  
##           Sepal.Length Sepal.Width Petal.Length Petal.Width  
## setosa        5.006       3.428       1.462       0.246  
## versicolor    5.936       2.770       4.260       1.326  
## virginica     6.588       2.974       5.552       2.026  
##  
## $distance  
##      [,1]      [,2]      [,3]  
## [1,] 0.0000 88.06690 175.79702  
## [2,] 88.0669 0.00000 16.85705  
## [3,] 175.7970 16.85705 0.00000
```

```
sqrt(Mahala$distance)
```

```
##      [,1]      [,2]      [,3]  
## [1,] 0.000000 9.384397 13.258847  
## [2,] 9.384397 0.000000 4.105733  
## [3,] 13.258847 4.105733 0.000000
```

1. viii) b) WHICH TWO SPECIES SEEM THE MOST SIMILAR

Versicolor and virginica. The mahalanobis distance between them is very low, and this can be verified again with a visual test when looking at the marginal plots. A clustering algorithm would probably have the hardest time identifying these 2 species as separate clusters.

1. viii) c)

Petal length and petal width both seem to discriminate best between the three species. If we take a look at the draftsman plot, we can see that the data can almost be clustered just based on one of these variables.

2. i) HOTELLING T TEST

We get an extremely significant p value (very close to 0) and so we reject the null hypothesis (that there is no difference in means between versicolor and virginica at a 0.05 significance level).

```
library(Hotelling)

## Warning: package 'Hotelling' was built under R version 4.1.1

## Loading required package: corpcor

## 
## Attaching package: 'corpcor'

## The following object is masked from 'package:matlib':
##   mpower

h = hotelling.test(iris[51:100,1:4], iris[101:150,1:4])
h

## Test stat: 355.47
## Numerator df: 4
## Denominator df: 95
## P-value: 0
```

2. ii)

```
# for (i in 31:50){
#   versi = iris[51:100,1:4][sample(nrow(iris[51:100,1:4]), i), ]
#   virgi = iris[101:150,1:4][sample(nrow(iris[101:150,1:4]), i), ]
#   hot = hotelling.test(versi,virgi)
#   append(x = p, values = hot$pval)
#   append(x = h, values = i)
# }
```

3. i) DERIVATION OF THE EM ALGORITHM

Attached at the end of this PDF

3. ii) GMM WITH SPHERICAL COVARIANCE MATRICES VS K MEANS

This is not true because first of all, K-means is merely an algorithm that partitions clusters in the feature space, whereas a Gaussian mixture model is a data model. The EM-algorithm is simply a means to estimate the parameters of the model.

Although they are two fundamentally different things, we can use the EM-algorithm to mimic K means, but spherical covariance matrices is not enough. This is because the update equations of the component means are different. K means hard assigns an observation to an estimated component mean at each iteration, then updates the new estimated component means by taking the mean of the observations assigned to a particular component. The EM algorithm assigns each observation a probability that they belong to a particular component. For the EM algorithm, the updated estimated component mean in a successive iteration depends not only on the current estimated component mean, but also on the current estimated component proportion and covariance matrix. Secondly, the K-Means algorithm will always perfectly partition the clusters in the feature space whereas the EM algorithm is used to

To make the EM algorithm behave similarly to the K means algorithm whilst retaining data generating capabilities, besides having spherical covariance matrices with the same number of components, we require the following modifications:

- The EM algorithm is run with a common (spherical) covariance matrix over all the components.
- Do hard allocation instead of soft allocation. That is, let h_{\max} denote the component with the highest posterior probability that observation i has been drawn from

component h. Re-assign this posterior probability for h_{\max} to 1, and re-assign it to 0 for all the other components. This mimics how K means assigns every observation to an estimated component mean.

-Once the algorithm has been halted, we can use the estimated parameters to fit the Gaussian mixture model and assign observations to clusters based on the most probable component.

-If we further add the constraint that the component proportions must be equal, then the update equation for the component means of our modified EM-Algorithm would behave identically to K-means. I think it might still possible to get something similar to K means without this constraint if we have good initialization of our component means, and the component proportions are initialized as equal.

3. iii) CHOOSING THE NUMBER OF CLUSTERS FOR K MEANS AND MIXTURE MODELS

I had no time left to write this down in pseudocode, but I will state the relevant methods. For K means, we can use the gap statistic. For Mixture models, the bayesian information criterion (BIC) can be used.

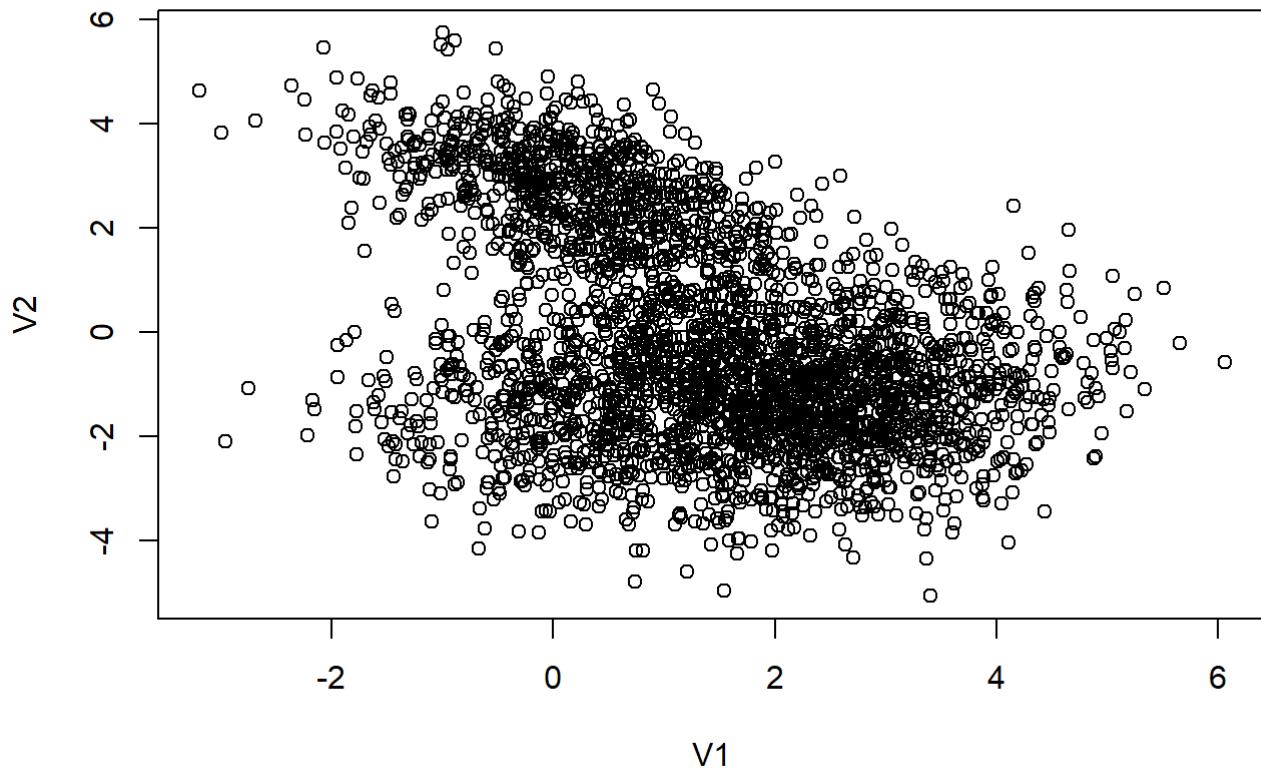
3. iv) HOW MANY CLUSTERS ARE PRESENT IN THE ARTIFICIAL DATASET?

Looking at the following plot, just by eye it looks like there are two clusters. The following density plots also confirm that there is a bimodality when you look at the marginal density V2. In the following sections I justify this by considering the BIC.

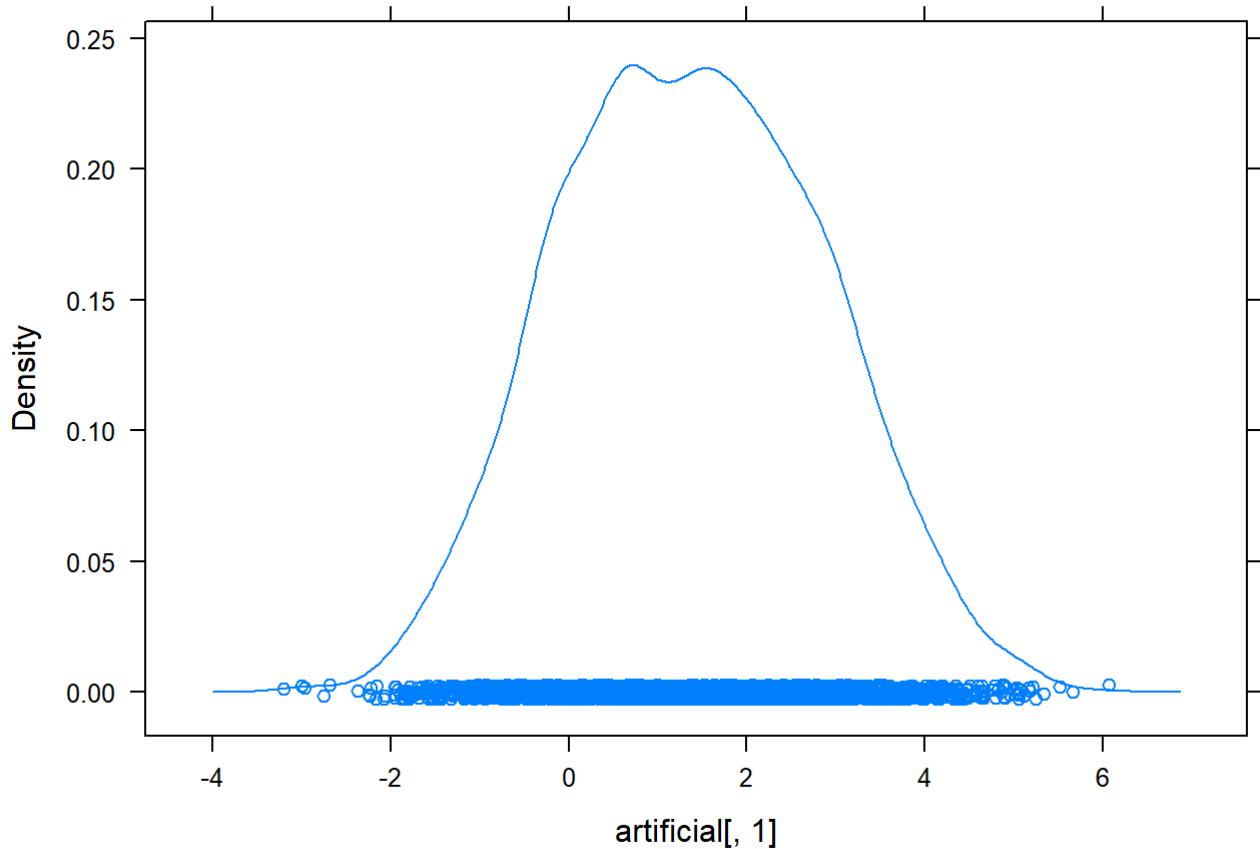
```
summary(artificial)
```

```
##          V1            V2
##  Min.   :-3.1969   Min.   :-5.0525
##  1st Qu.: 0.2946   1st Qu.:-1.6722
##  Median : 1.3657   Median :-0.6436
##  Mean   : 1.3802   Mean   :-0.1351
##  3rd Qu.: 2.4450   3rd Qu.: 1.4525
##  Max.   : 6.0645   Max.   : 5.7482
```

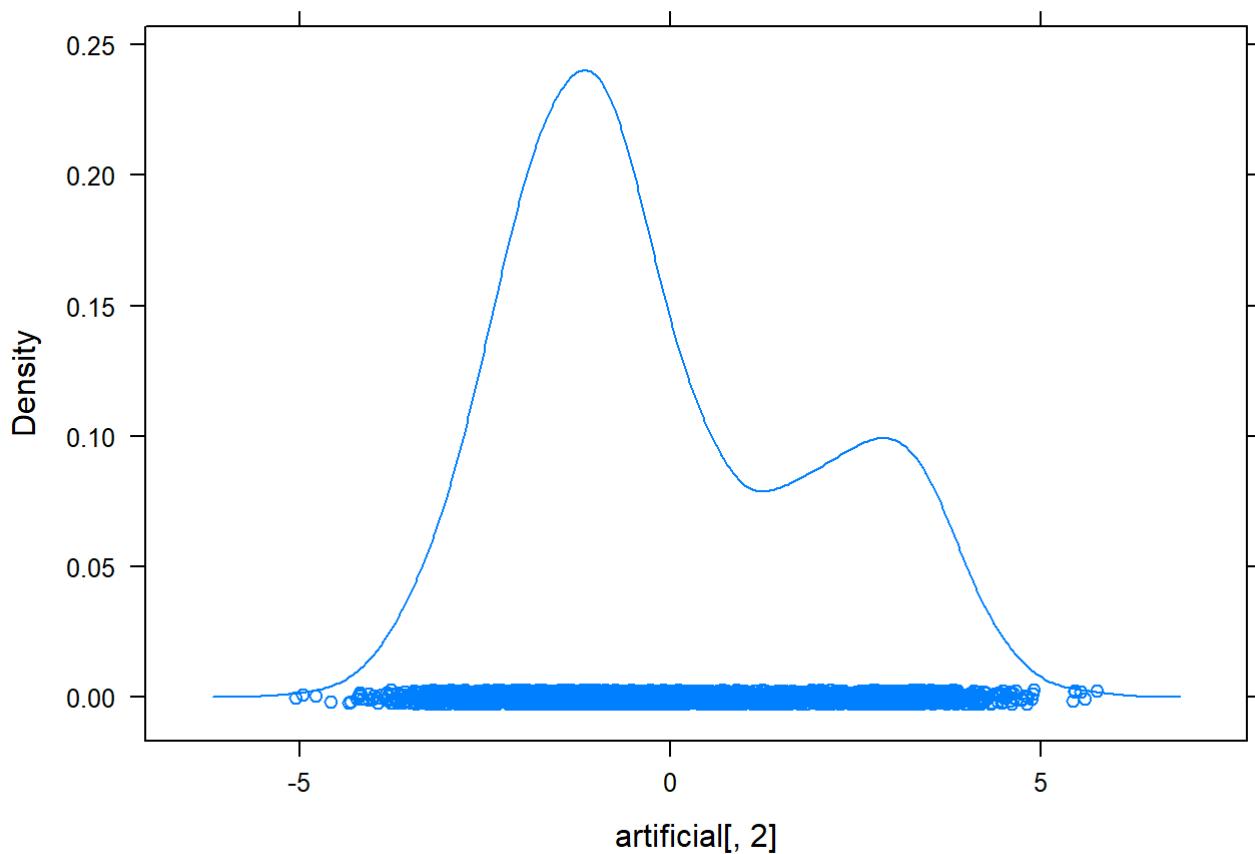
```
plot(artificial)
```



```
densityplot(artificial[,1])
```



```
densityplot(artificial[,2])
```



3. v) APPLYING K MEANS AND MIXTURE MODEL (EM) ALGORITHMS TO THE IRIS AND ARTIFICIAL DATASET

For the iris data set, based on the gap statistic, 3 or 4 seems like a reasonable choice. Looking at the K means performance for $k = 2$, $k = 3$, and $k = 4$, it seems to do a good job of clustering. The $k = 3$ case looks very similar to our actual classes, and even though for instance $k = 2$ is wrong, the separation looks very reasonable.

Now looking at the BIC, it recommends 2 or 3 clusters as the best choices. Again this is very reasonable. The EM algorithm also does a good job of clustering here as it has managed to cluster the 3 species.

Moving on to the artificial dataset, without any further information I would assume that it has 2 clusters just looking at it visually. The gap statistic and BIC also agrees with this and recommends $k = 2$. However, we see that here, the EM algorithm clusters look

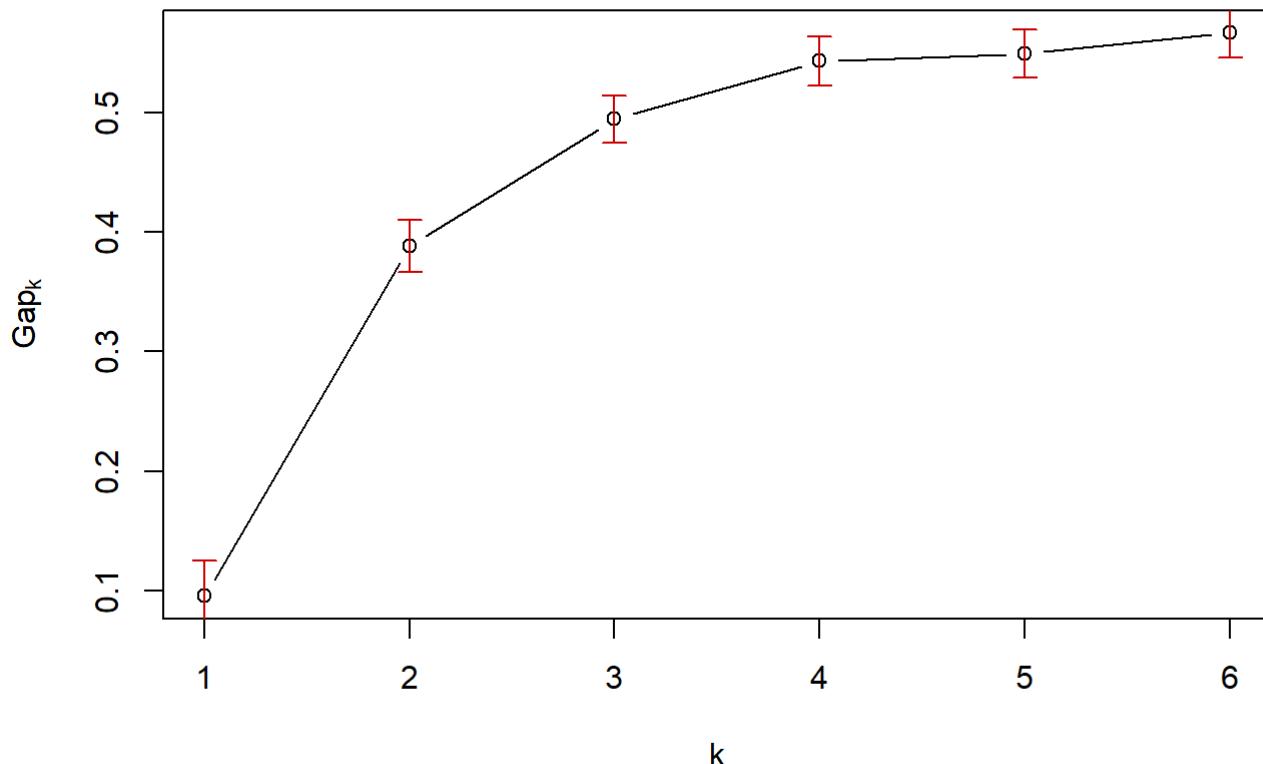
much more realistic than the K means clustering. This is because K means linearly partitions the observations in the feature space. This was not clear in the iris dataset because we were looking at marginal plots. But now that 2 dimensions allows us to see the whole feature space, we can see how K means clearly partitions the data, allowing no overlaps between clusters. Since the EM algorithm is in this case being used to find estimates for a gaussian mixture model, the clusters look much more realistic.

K MEANS ON THE IRIS DATASET

```
set.seed(5)
# Modified code from the tutorial
library(cluster)
gapstatres <- clusGap(iris[,1:4], FUN = kmeans, nstart = 20, K.max = 6, B = 100) # use 100 bootstrap samples, maximum clusters = 4,
gapstatres
```

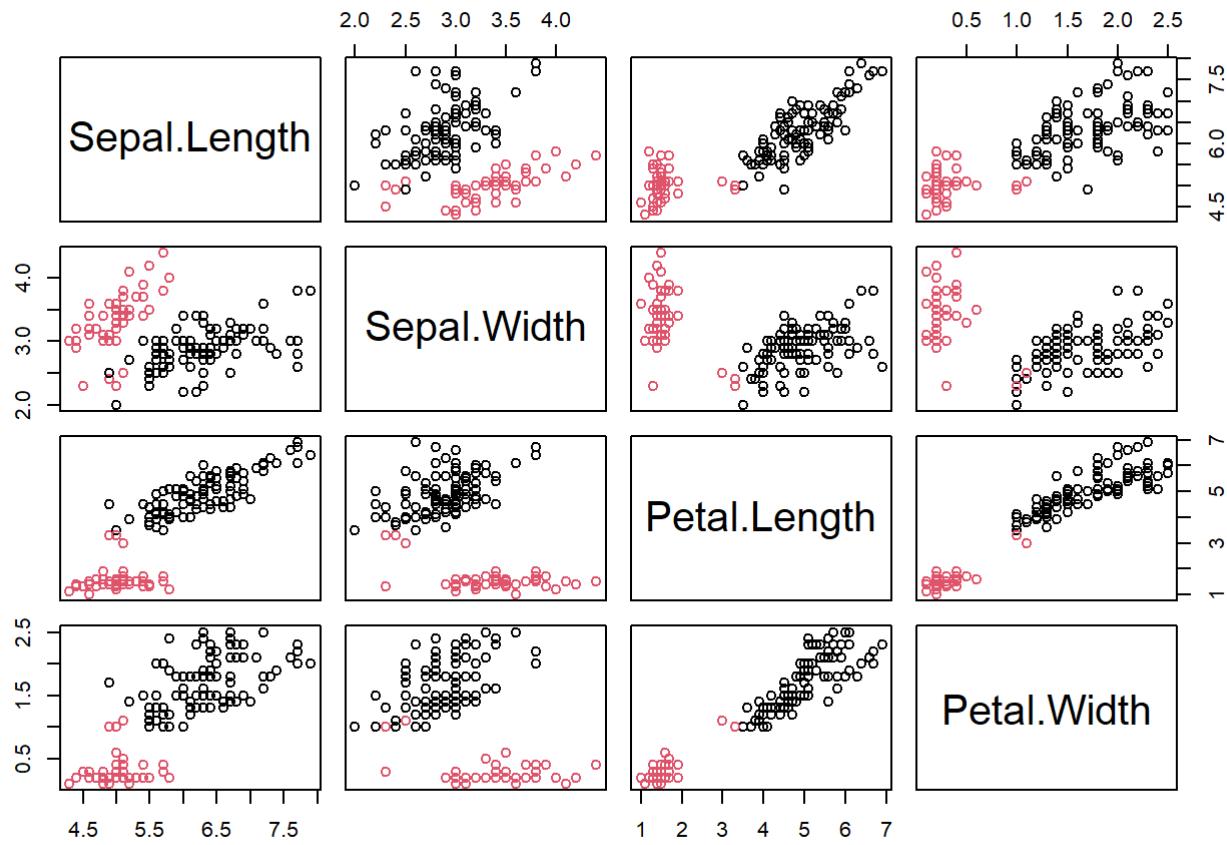
```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = iris[, 1:4], FUNcluster = kmeans, K.max = 6, B = 100,      nstart = 20)
## B=100 simulated reference sets, k = 1..6; spaceH0="scaledPCA"
## --> Number of clusters (method 'firstSEmax', SE.factor=1): 5
##          logW    E.logW      gap     SE.sim
## [1,] 4.551642 4.647258 0.09561611 0.02919563
## [2,] 3.808397 4.196600 0.38820265 0.02179286
## [3,] 3.519407 4.013754 0.49434748 0.01943919
## [4,] 3.368124 3.911116 0.54299206 0.02033707
## [5,] 3.277033 3.826240 0.54920784 0.02031318
## [6,] 3.183405 3.749877 0.56647151 0.02051491
```

```
plot(gapstatres, main = "Gap statistic with 100 bootstrap samples and 20 restarts per k")
```

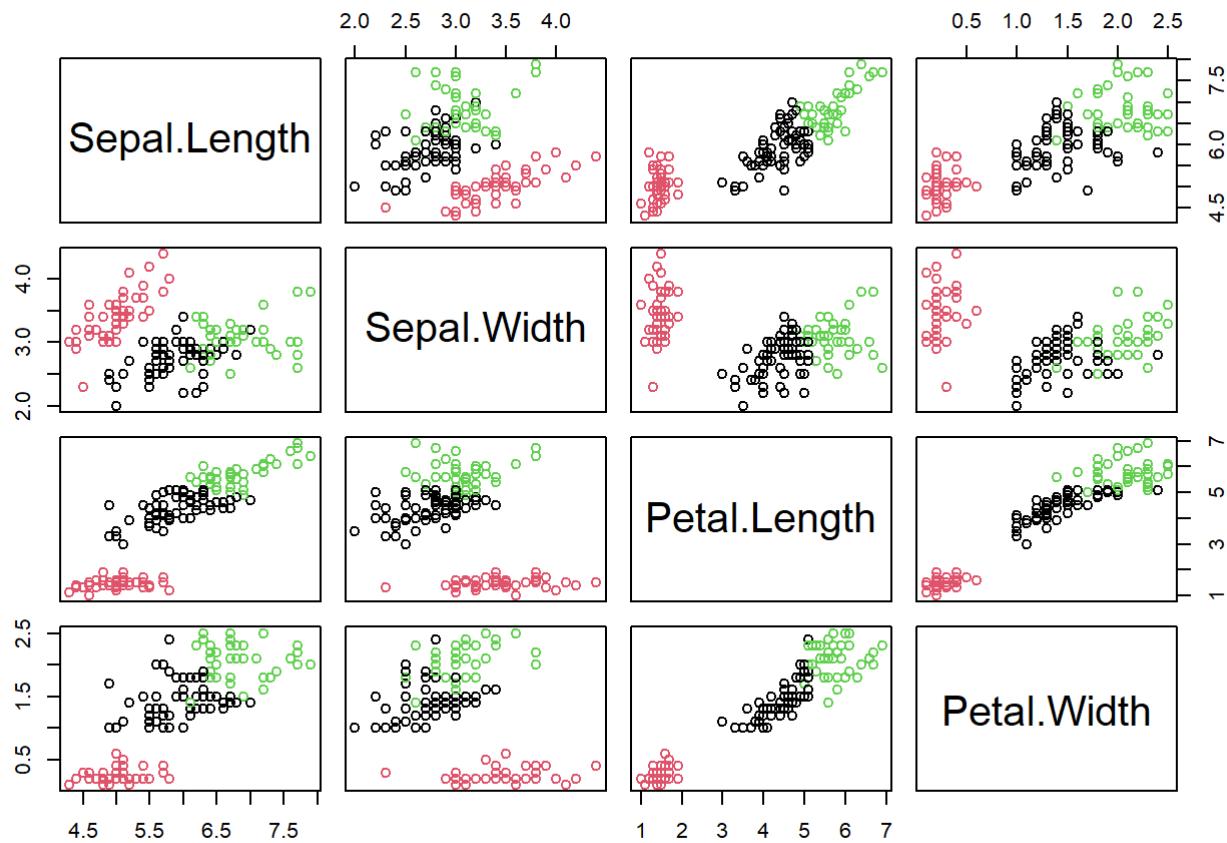
Gap statistic with 100 bootstrap samples and 20 restarts per k

```
set.seed(5)
kres2 <- kmeans(iris[,1:4],centers=2)
set.seed(5)
kres3 <- kmeans(iris[,1:4],centers=3)
set.seed(5)
kres4 <- kmeans(iris[,1:4],centers=4)

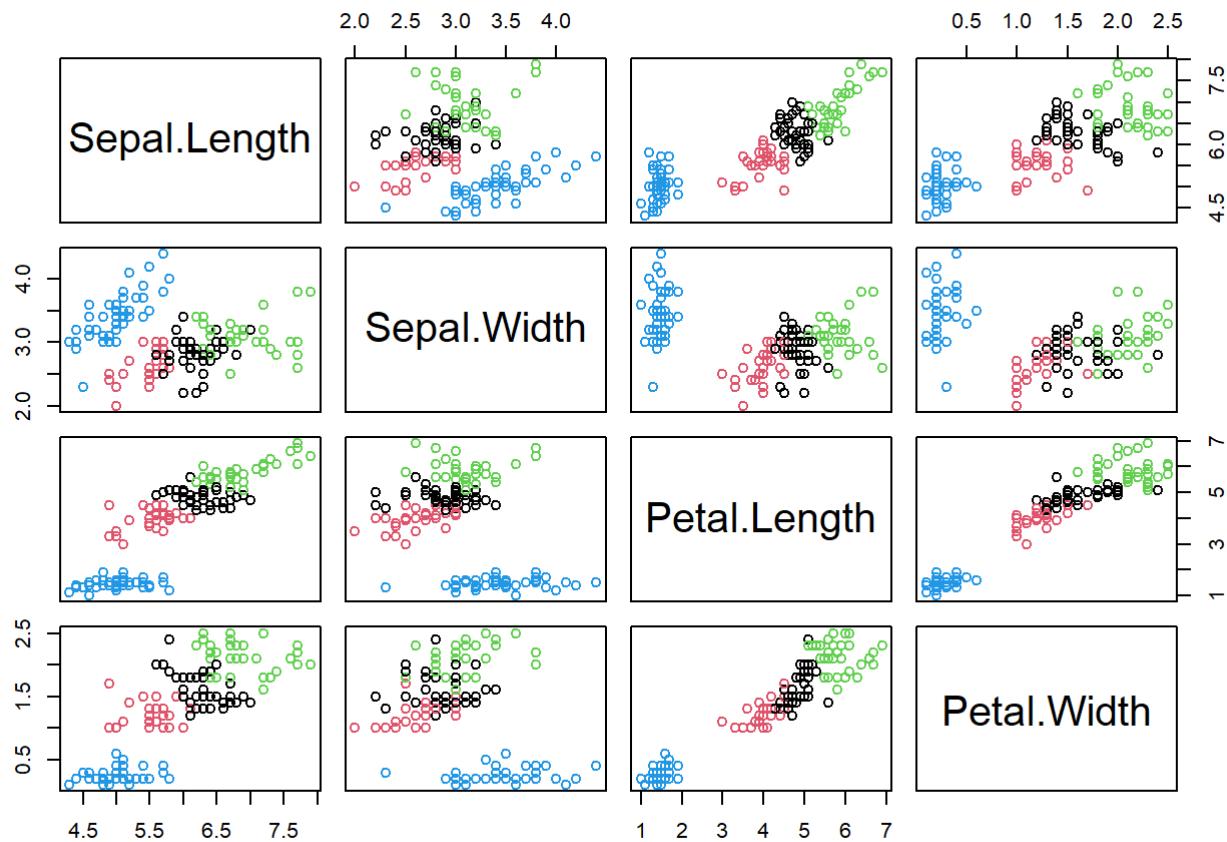
pairs(iris[,1:4], col=c(1:2)[kres2$cluster])
```



```
pairs(iris[,1:4], col=c(1:3)[kres3$cluster])
```



```
pairs(iris[,1:4], col=c(1:4)[kres4$cluster])
```



EM ALGORITHM ON THE IRIS DATASET

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.1.1
```

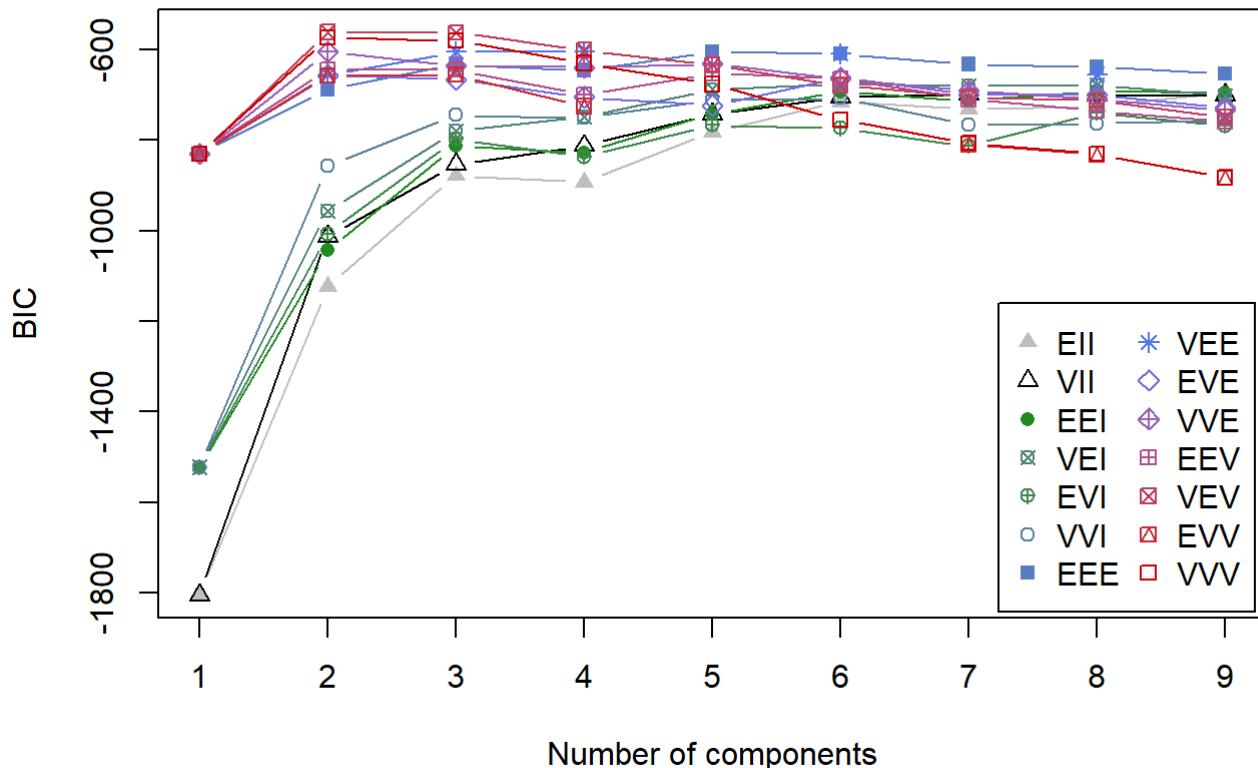
```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:psych':
##     sim
```

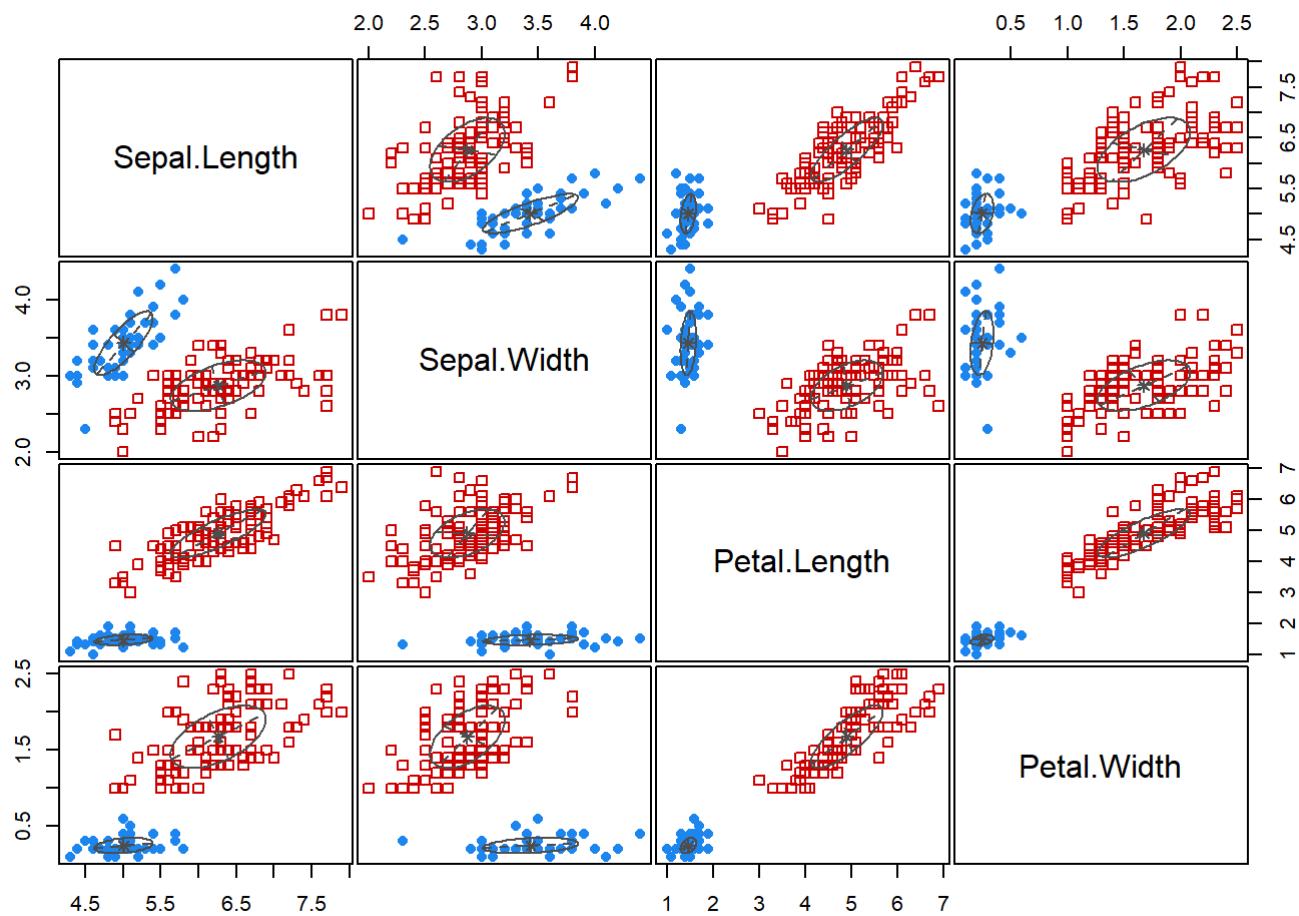
```
## The following object is masked from 'package:MVN':
##     mvn
```

```
set.seed(5)
BIC_iris = mclustBIC(iris[,1:4])
plot(BIC_iris)
```

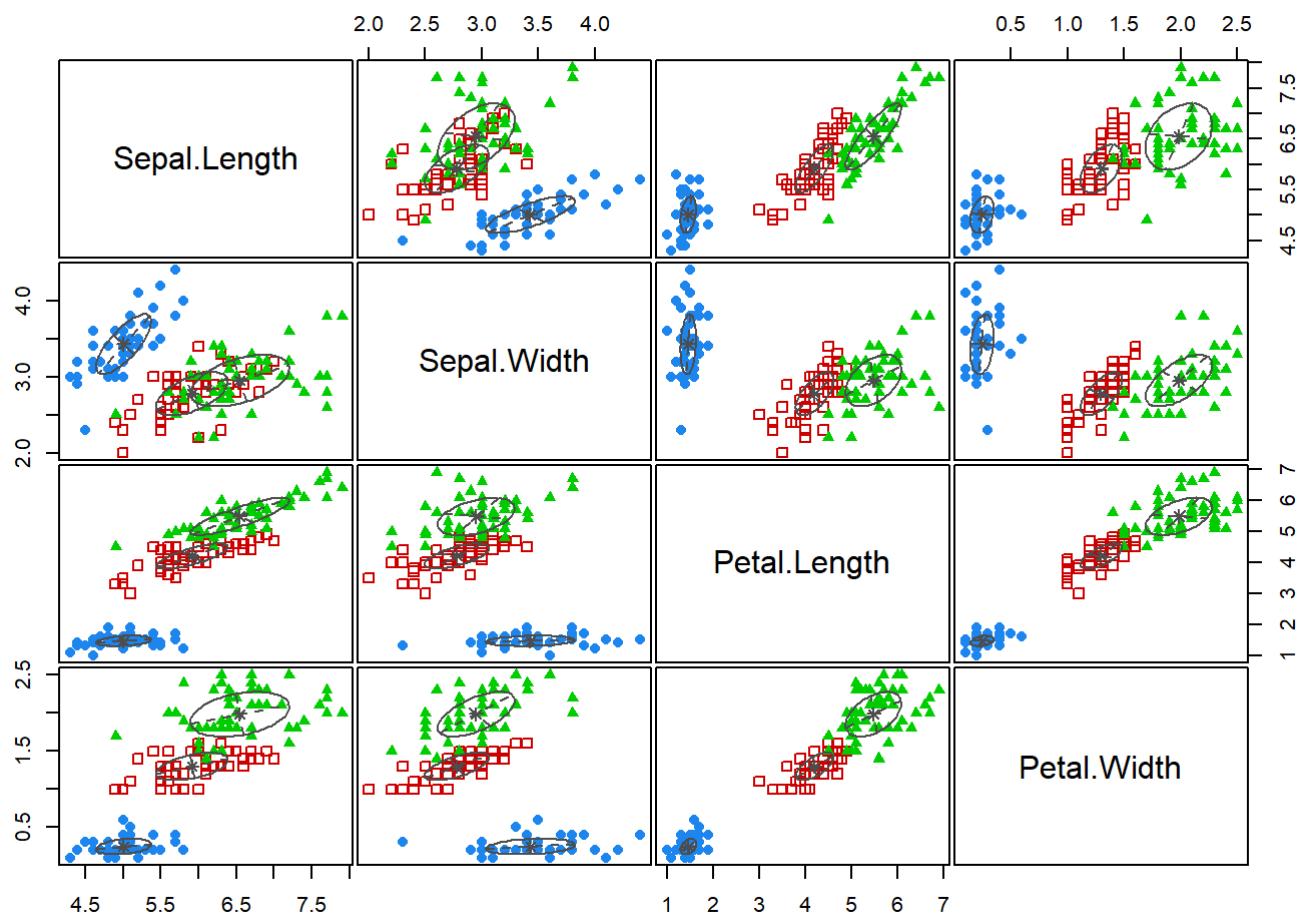


```
set.seed(5)
EM_iris2 = Mclust(iris[,1:4], G = 2)
set.seed(5)
EM_iris3 = Mclust(iris[,1:4], G = 3)
set.seed(5)
EM_iris4 = Mclust(iris[,1:4], G = 4)

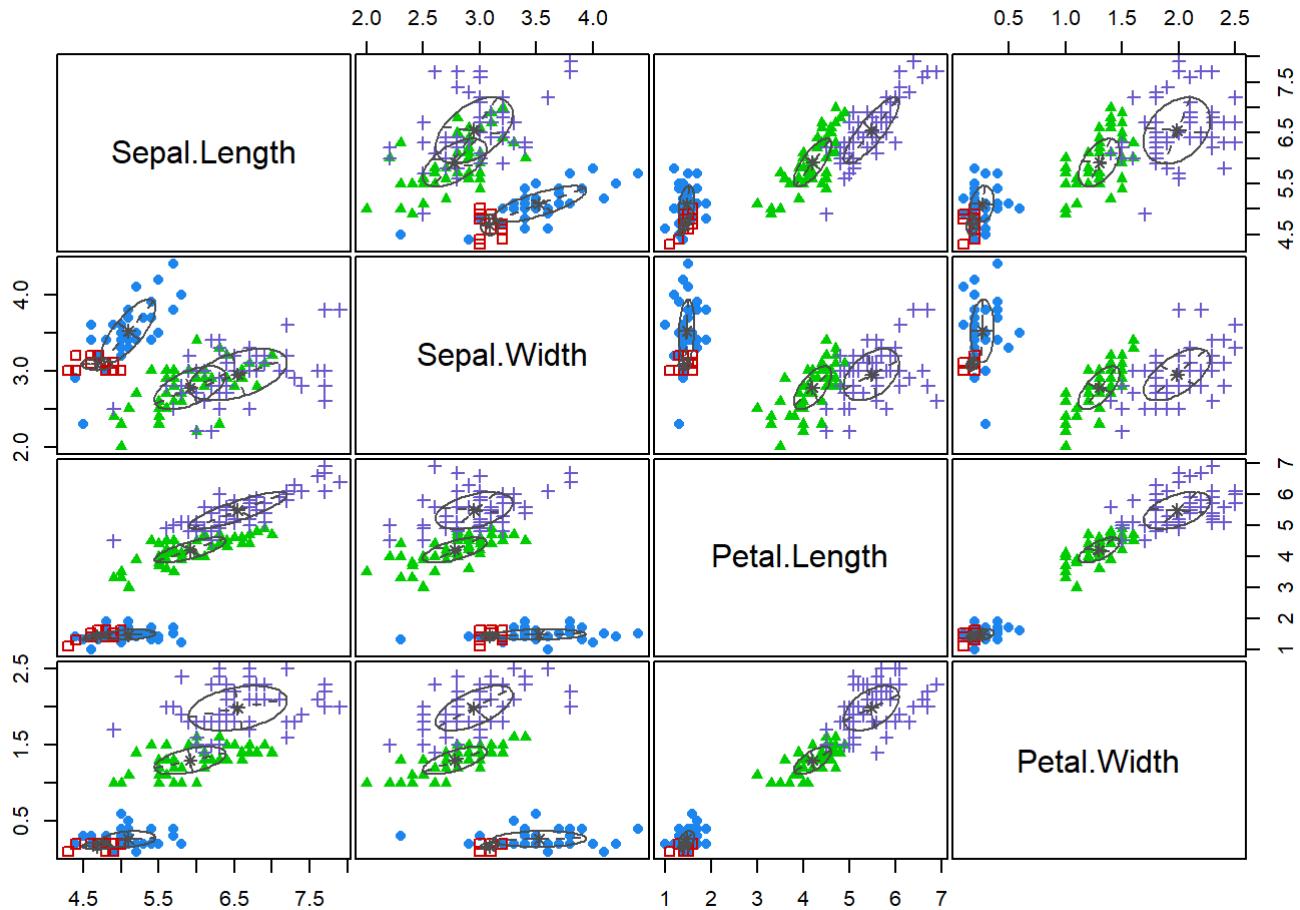
plot(EM_iris2, what = "classification")
```



```
plot(EM_iris3, what = "classification")
```



```
plot(EM_iris4, what = "classification")
```

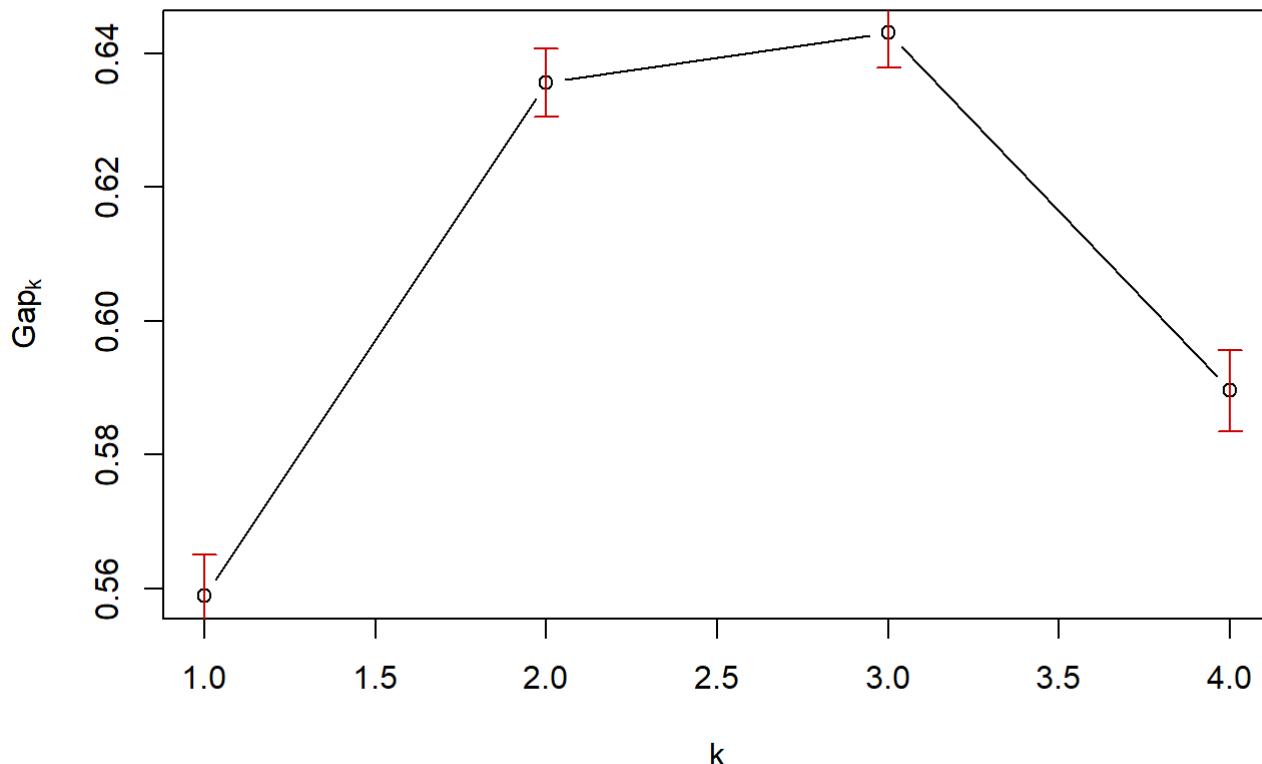


K MEANS ON THE ARTIFICIAL DATASET

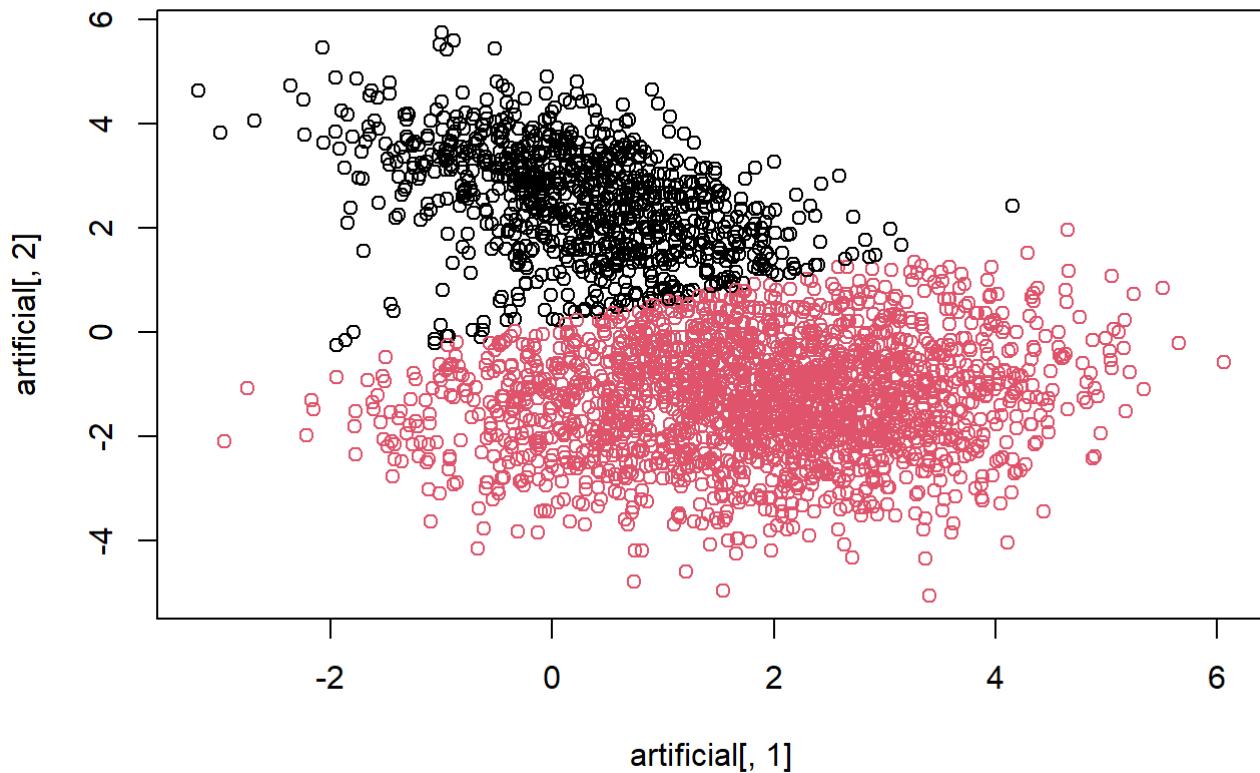
```
set.seed(5)
# Modified code from the tutorial
gapstatres <- clusGap(artificial, FUN = kmeans, nstart = 20, K.max = 4, B = 100) # use 100 bootstrap samples, maximum clusters = 4,
gapstatres
```

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = artificial, FUNcluster = kmeans, K.max = 4, B = 100,      nstart = 20)
## B=100 simulated reference sets, k = 1..4; spaceH0="scaledPCA"
## --> Number of clusters (method 'firstSEmax', SE.factor=1): 3
##          logW    E.logW      gap     SE.sim
## [1,] 7.755788 8.314640 0.5588526 0.006221050
## [2,] 7.347401 7.982994 0.6355925 0.005154825
## [3,] 7.154973 7.797983 0.6430096 0.005194937
## [4,] 7.028296 7.617836 0.5895399 0.006021900
```

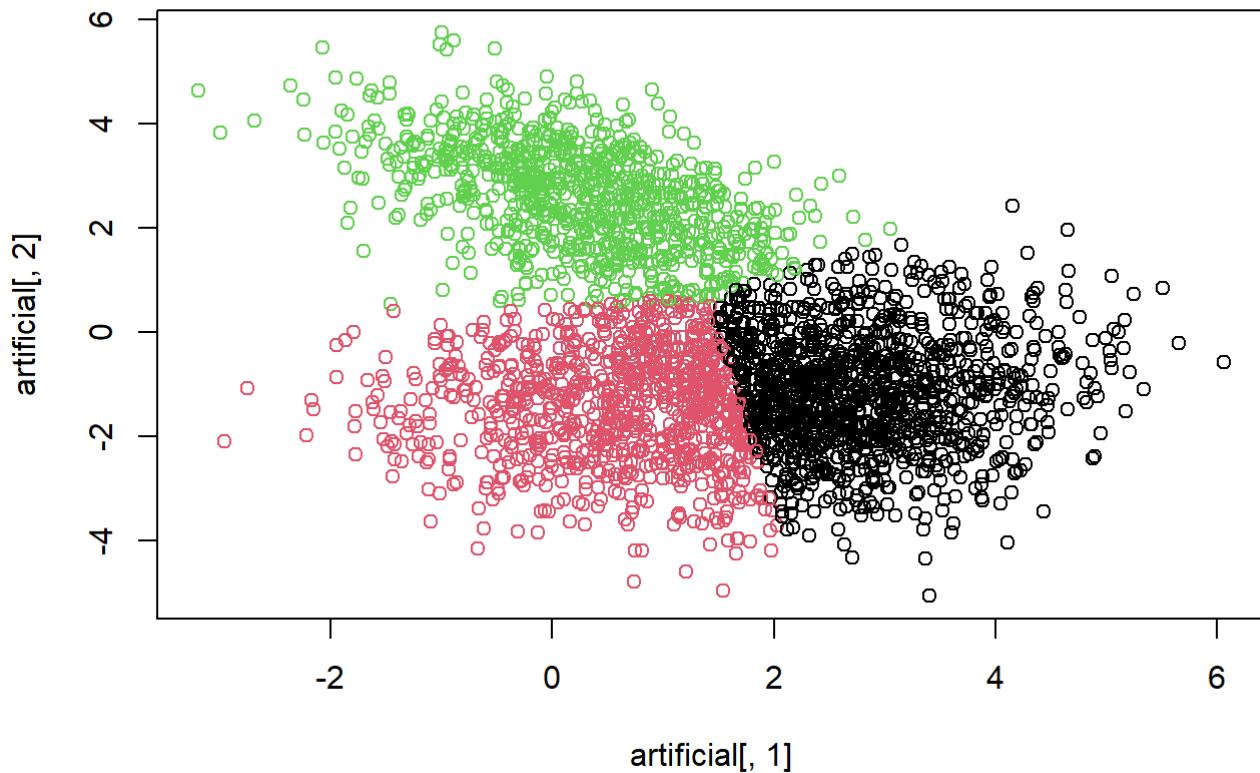
```
plot(gapstatres, main = "Gap statistic with 100 bootstrap samples and 20 restarts per k")
```

Gap statistic with 100 bootstrap samples and 20 restarts per k

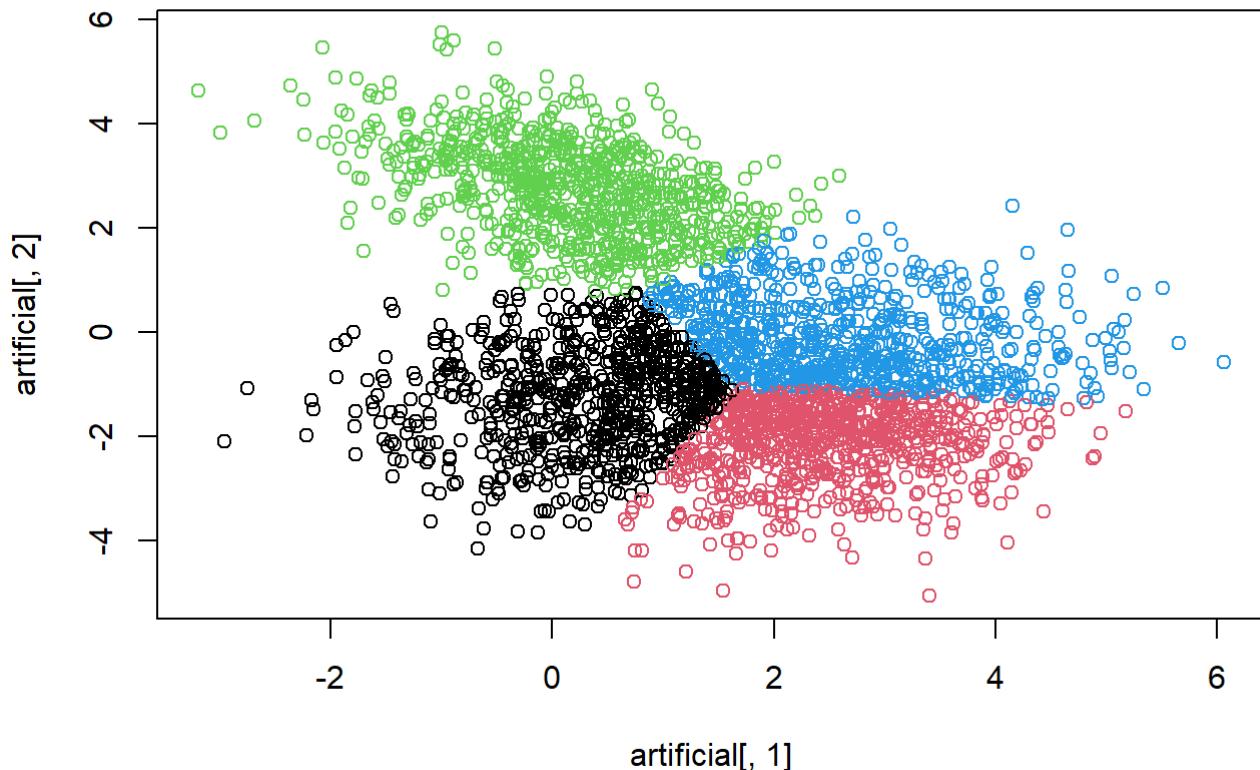
```
set.seed(5)
km_art2 <- kmeans(artificial, centers=2)
plot(artificial[,1], artificial[,2], col=km_art2$cluster)
```



```
set.seed(5)
km_art3 <- kmeans(artificial, centers=3)
plot(artificial[,1],artificial[,2],col=km_art3$cluster)
```

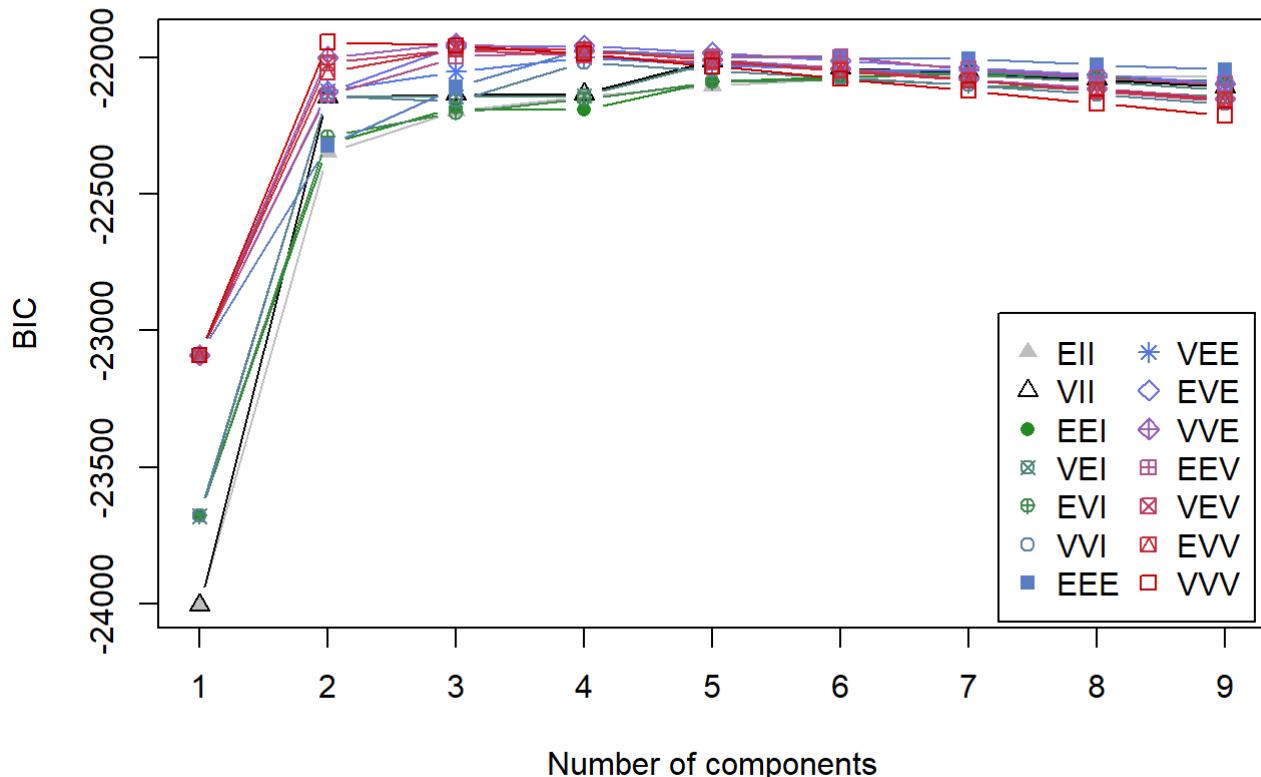


```
set.seed(5)
km_art4 <- kmeans(artificial, centers=4)
plot(artificial[,1],artificial[,2],col=km_art4$cluster)
```

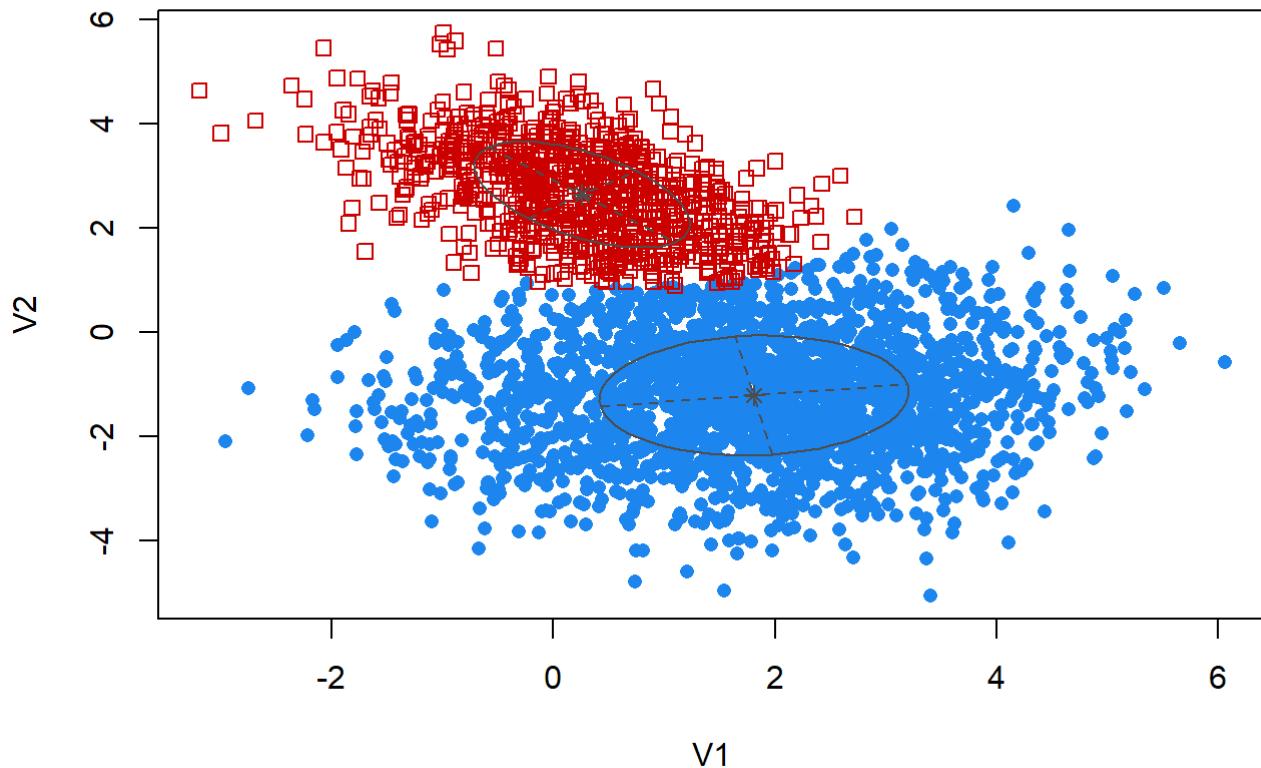


EM ALGORITHM ON THE ARTIFICIAL DATASET

```
set.seed(5)
BIC = mclustBIC(artificial)
plot(BIC)
```



```
set.seed(5)
EM_art = Mclust(artificial, x=BIC)
plot(EM_art, what = "classification")
```



```
EM_art$parameters
```

```

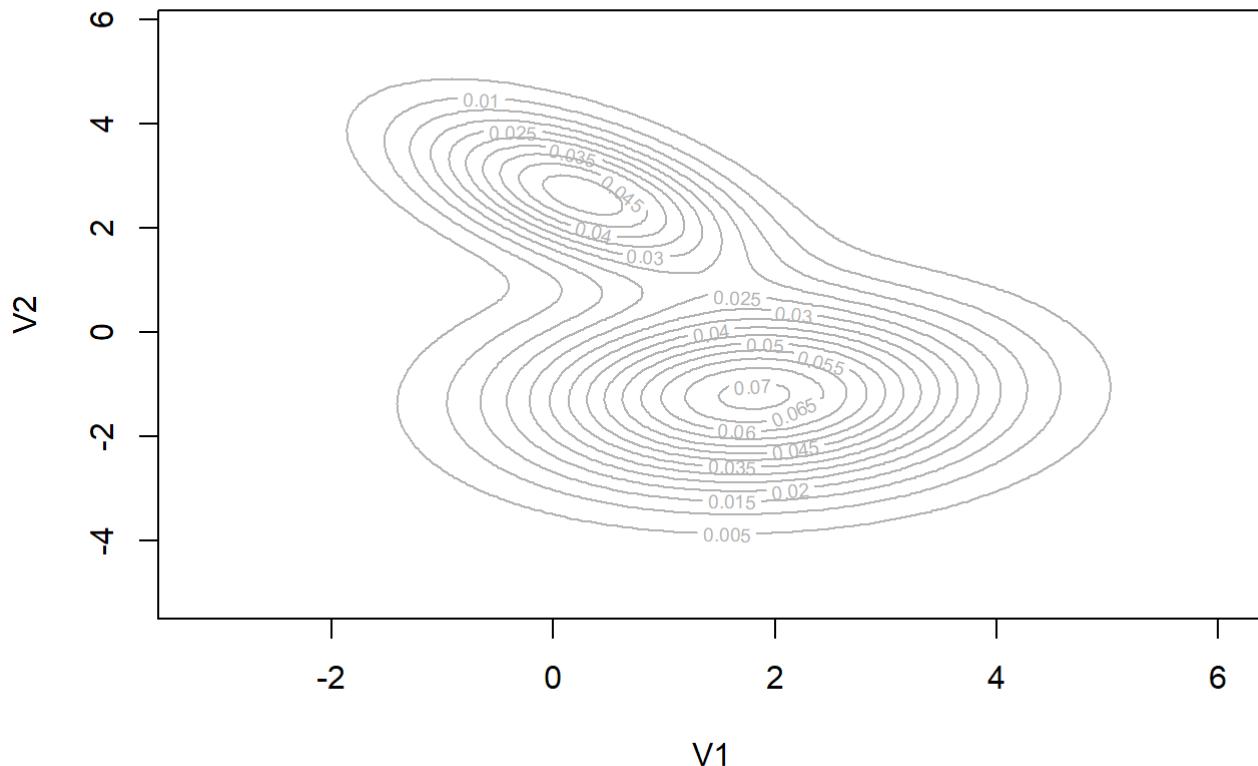
## $pro
## [1] 0.7226437 0.2773563
##
## $mean
##      [,1]      [,2]
## V1  1.810845 0.2580788
## V2 -1.204362 2.6509137
##
## $variance
## $variance$modelName
## [1] "VVV"
##
## $variance$d
## [1] 2
##
## $variance$G
## [1] 2
##
## $variance$sigma
## , , 1
##
##      V1      V2
## V1 1.9455884 0.0962579
## V2 0.0962579 1.3221111
##
## , , 2
##
##      V1      V2
## V1 0.9497321 -0.544177
## V2 -0.5441770  1.032094
##
## $variance$cholsigma
## , , 1
##
##      V1      V2
## V1 1.394844 0.06900982
## V2 0.000000 1.14775812
##
## , , 2
##
##      V1      V2
## V1 0.974542 -0.5583925
## V2 0.000000 -0.8486999

```

3. vi) PARAMETER ESTIMATES AND 95% CONFIDENCE INTERVALS

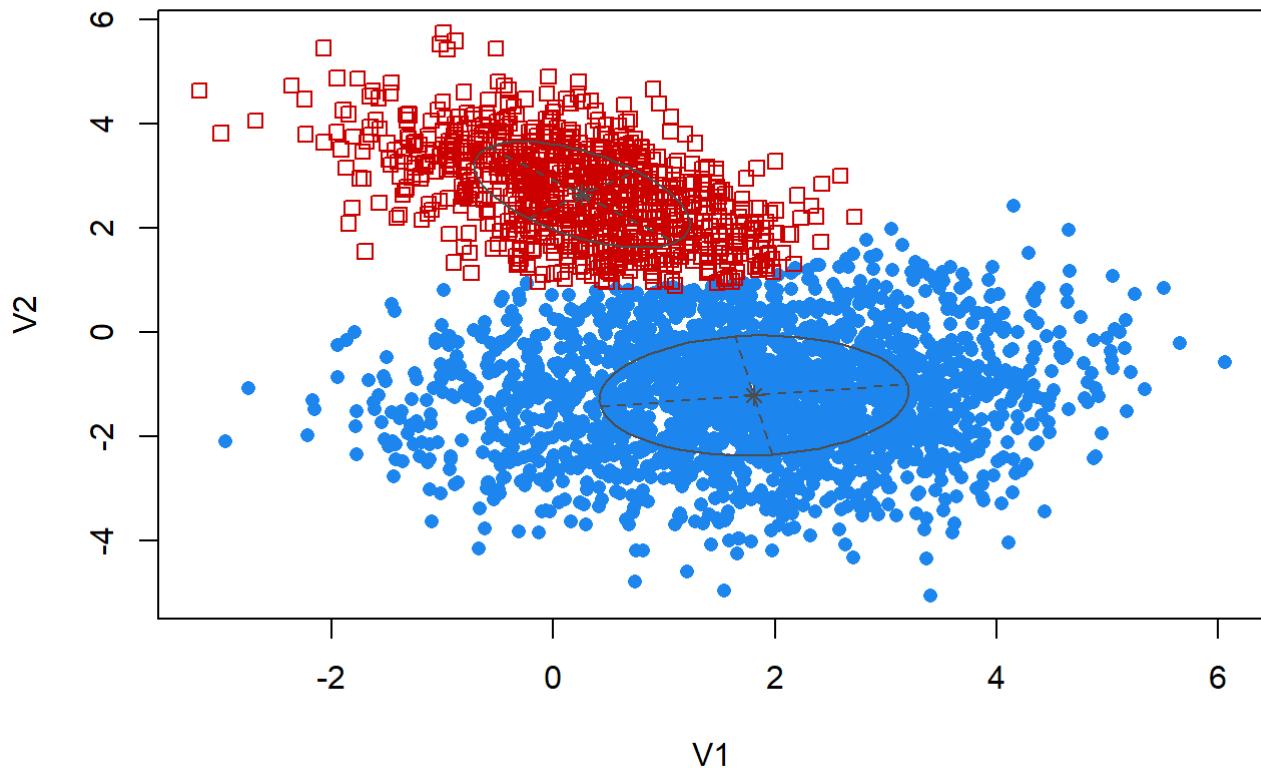
3. vii) a) CONTOUR PLOT OF THE FITTED MIXTURE DISTRIBUTION ON THE ARTIFICIAL DATA

```
plot(EM_art, what = "density")
```



3. vii) b) CONTOUR PLOTS OF THE COMPONENTS OF THE FITTED MIXTURE DISTRIBUTION ON THE ARTIFICIAL DATA

```
plot(EM_art, what = "classification")
```



Name: Chee Kitt Win

Student Number: 45589140

STAT3006 Assignment 2

Question 3 part i)

Notation

Following the notation in the notes, let $\tilde{x}_i = \begin{pmatrix} \tilde{x}_{i1} \\ \vdots \\ \tilde{x}_{ip} \end{pmatrix}$ denote the i th observation of our sample of n p -dimensional continuous data which we assume to be drawn i.i.d from a Gaussian mixture model with fixed number of components, H , component means $\tilde{\mu}_h$, and common covariance matrix Σ .

Let $X_{\text{All}} = \begin{pmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_n^T \end{pmatrix}$ which is an n by p matrix of all the observed data.

Let $Z = \begin{pmatrix} z_1^T \\ \vdots \\ z_n^T \end{pmatrix}$ which is an n by H matrix of 0's and 1's where the z_i 's are vectors of length H such that if \tilde{x}_i belongs to the h th component, then 1 is placed in the h th position of z_i . The other elements of z_i are all 0.

Note: X_{All} and Z will be treated as matrices, not vectors of vectors. Therefore, z_{ih} for instance, refers to the element in the i th row and h th column of matrix Z .

As given in the notes (the derivation has been done, so I will simply state it),

the complete data log likelihood of our Gaussian mixture model is:

$$\textcircled{1} \quad l_c(\Psi; X_{\text{All}}, Z) = \sum_{i=1}^n \sum_{h=1}^H z_{ih} \left(\ln \tilde{\pi}_h - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\tilde{x}_i - \tilde{\mu}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{\mu}_h) \right)$$

where Ψ is simply a vector of all the parameters we are trying to estimate:

- The component means, $\tilde{\mu}_h = \begin{pmatrix} \tilde{\mu}_{h1} \\ \vdots \\ \tilde{\mu}_{hp} \end{pmatrix}$ for $h = 1, 2, \dots, H$
- The unique elements (variances and covariances) of the covariance matrix, Σ
- The probabilities of the components, $\tilde{\pi}_h$ for $h = 1, 2, \dots, H-1$ (Since $\tilde{\pi}_H$ is simply $1 - \sum_{h=1}^{H-1} \tilde{\pi}_h$)

If we let $f_h(\tilde{x}_i; \tilde{\mu}_h, \Sigma) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (\tilde{x}_i - \tilde{\mu}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{\mu}_h) \right)$ which is the multivariate normal distribution corresponding to the h th component of our mixture model, then we can also write

$$l_c(\Psi; X_{\text{All}}, Z) = \sum_{i=1}^n \sum_{h=1}^H z_{ih} (\ln \tilde{\pi}_h + \ln f_h(\tilde{x}_i; \tilde{\mu}_h, \Sigma)) \quad \textcircled{2}$$

(More explicitly, let $\Psi = (\tilde{\mu}_1^T, \dots, \tilde{\mu}_H^T, \tilde{\Sigma}^T, \tilde{\pi}_1, \dots, \tilde{\pi}_{H-1})^T$ where $\tilde{\Sigma}$ is a vector containing the unique elements of Σ)

Deriving the E and M steps of the E-M Algorithm

We wish to maximize $\ell_c(\Psi; X_{\text{All}}, Z)$ using the E-M algorithm. We will first define the Q function for the E-step, then derive the update equations for M_h, J_h and Σ for the M-step by maximizing the Q function.

Defining the Q function for the E step

Let $\Psi^{(k)} = (\tilde{M}_1^{(k)}, \dots, \tilde{M}_n^{(k)}, \tilde{\Sigma}^{(k)}, J_1^{(k)}, \dots, J_{H-1}^{(k)})$ where $\tilde{\Sigma}^{(k)}$ is a vector containing the unique elements of $\Sigma^{(k)}$. $\Psi^{(k)}$ is a vector of the parameters obtained in the kth iteration of the E-M algorithm and hence they are all denoted with a superscript (k) to distinguish them from the parameters to be estimated in the current step.

We define the Q function to be

$$\begin{aligned} Q(\Psi; \Psi^{(k)}) &= E_{Z; \Psi^{(k)}} \{ \ell_c(\Psi) | X_{\text{All}} \} \\ &= \sum_{i=1}^n \sum_{h=1}^H E_{Z; \Psi^{(k)}} (z_{ih} | \tilde{x}_i) (\ln J_h + \ln f_h(\tilde{x}_i; \tilde{M}_h, \Sigma)) \quad \textcircled{3} \end{aligned}$$

Note that we previously defined z_{ih} to be the element of Z corresponding to the i th row and h th column, and it is a binary variable

$$\begin{aligned} E_{Z; \Psi^{(k)}} (z_{ih} | \tilde{x}_i) &= 0 \cdot P(z_{ih}=0 | \tilde{x}_i) + 1 \cdot P(z_{ih}=1 | \tilde{x}_i) \\ &= \frac{P(z_{ih}=1, \tilde{x}_i)}{P(\tilde{x}_i)} \\ &= \frac{\underbrace{J_h^{(k)} f_h(\tilde{x}_i; \tilde{M}_h^{(k)}, \Sigma^{(k)})}_{\sum_{m=1}^H J_m^{(k)} f_m(\tilde{x}_i; \tilde{M}_m^{(k)}, \Sigma^{(k)})}}{\sum_{m=1}^H J_m^{(k)} f_m(\tilde{x}_i; \tilde{M}_m^{(k)}, \Sigma^{(k)})} \quad \textcircled{4} \end{aligned}$$

These probabilities are based on $\Psi^{(k)}$

For convenience, let $T_h(\tilde{x}_i; \Psi^{(k)})$ denote the expression on the R.H.S of $\textcircled{4}$

Continuing off from $\textcircled{3}$, we have

$$Q(\Psi; \Psi^{(k)}) = \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(k)}) (\ln J_h + \ln f_h(\tilde{x}_i; \tilde{M}_h, \Sigma)) \quad \textcircled{5}$$

Deriving the update equations for the M step

Component probabilities, π_{θ_h}

Our approach will be to maximize the Q function using lagrange multipliers.

Let $Q(\Psi; \Psi^{(k)})$ be the objective function and $g(\Psi; \Psi^{(k)}) = \sum_{m=1}^H \pi_m - 1$ be the constraint function.

$$\text{Let } L(\Psi; \Psi^{(k)}) = Q(\Psi; \Psi^{(k)}) - \lambda g(\Psi; \Psi^{(k)})$$

We set $\nabla L(\Psi; \Psi^{(k)}) = 0$, and focus on the derivative with respect to π_{θ_h}

$$\frac{\partial}{\partial \pi_{\theta_h}} Q(\Psi; \Psi^{(k)}) = \lambda \frac{\partial}{\partial \pi_{\theta_h}} g(\Psi; \Psi^{(k)})$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(x_i; \Psi^{(k)}) \frac{\partial}{\partial \pi_{\theta_h}} (\ln \pi_{\theta_h} + \ln f_n(x_i; \mu_h, \Sigma)) = \lambda \quad (1)$$

$$\Rightarrow \sum_{i=1}^n T_h(x_i; \Psi^{(k)}) \left(\frac{1}{\pi_{\theta_h}} \right) = \lambda$$

$$\Rightarrow \sum_{i=1}^n T_h(x_i; \Psi^{(k)}) = \pi_{\theta_h} \lambda \quad (6)$$

(6) is the equation for the hth component. There are H-1 other such components.
Adding the L.H.S and R.H.S of these H equations yields

$$\sum_{h=1}^H \sum_{i=1}^n T_h(x_i; \Psi^{(k)}) = \lambda \sum_{h=1}^H \pi_{\theta_h}$$

$$\Rightarrow n = \lambda \quad (7) \quad \left(\begin{array}{l} \text{Note that } T_h(x_i; \Psi^{(k)}) \text{ is the probability that a particular} \\ \text{observation belongs to a particular component. Clearly, summing} \\ \text{over the components and all the observations yields } n \end{array} \right)$$

Subbing (7) into (6), we get

$$\pi_{\theta_h}^{(k+1)} = \frac{1}{n} \sum_{i=1}^n T_h(x_i; \Psi^{(k)})$$

Component means, \tilde{M}_h

Our approach here is to set $\frac{\partial}{\partial \tilde{M}_h} Q(\Psi; \Psi^{(0)}) = 0$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \tilde{M}_h} (\ln J_{fh} + \ln f_h(\tilde{x}_i; \tilde{M}_h, \Sigma)) = 0$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \tilde{M}_h} \left(-\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) \right) = 0$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \tilde{M}_h} \left(-\frac{1}{2} (\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) \right) = 0$$

From result 86 in the Matrix Cookbook,

$$\text{If } \Sigma^{-1} \text{ is symmetric, then } \frac{\partial}{\partial \tilde{M}_h} ((\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h)) = -2\Sigma^{-1}(\tilde{x}_i - \tilde{M}_h)$$

Since Σ is symmetric, its inverse is also symmetric, therefore we have

$$\Rightarrow \sum_{i=1}^n T_h(\tilde{x}_i; \Psi^{(0)}) (\Sigma^{-1}(\tilde{x}_i - \tilde{M}_h)) = 0$$

$$\Rightarrow \sum_{i=1}^n T_h(\tilde{x}_i; \Psi^{(0)}) \Sigma^{-1} \tilde{x}_i = \sum_{i=1}^n T_h(\tilde{x}_i; \Psi^{(0)}) \tilde{M}_h \Sigma^{-1}$$

$$\Rightarrow \tilde{M}_h^{(0+)} = \frac{\sum_{i=1}^n T_h(\tilde{x}_i; \Psi^{(0)}) \tilde{x}_i}{\sum_{i=1}^n T_h(\tilde{x}_i; \Psi^{(0)})}$$

Common covariance matrix, Σ

Here, we set $\frac{\partial}{\partial \Sigma} Q(\Psi; \Psi^{(0)}) = 0$.

Note: For derivational convenience, we take the partial w.r.t Σ^{-1} instead of Σ .

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \Sigma} (\ln J_{fh} + \ln f_h(\tilde{x}_i; \tilde{M}_h, \Sigma)) = 0$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \Sigma} \left(-\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) \right) = 0$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \Sigma^{-1}} \left(\frac{1}{2} \ln |\Sigma^{-1}| - \frac{1}{2} (\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) \right) = 0 \quad \begin{matrix} \text{(From first year} \\ \text{linear algebra,} \\ |\Sigma^{-1}| = |\Sigma^{-1}| \end{matrix}$$

Note that if k is a scalar, then $\text{tr}(k)$, i.e. the trace of k , is equal to k .

Also, from result 16 in the Matrix Cookbook, $\text{tr}(ABC) = \text{tr}(BCA)$. Since

$(\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h)$ is a scalar, it is equal to

$$\text{tr}((\tilde{x}_i - \tilde{M}_h)^T \Sigma^{-1} (\tilde{x}_i - \tilde{M}_h)) = \text{tr}(\Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) (\tilde{x}_i - \tilde{M}_h)^T). \text{ This gives}$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H T_h(\tilde{x}_i; \Psi^{(0)}) \frac{\partial}{\partial \Sigma^{-1}} \left(\frac{1}{2} \ln |\Sigma^{-1}| - \frac{1}{2} \text{tr}(\Sigma^{-1} (\tilde{x}_i - \tilde{M}_h) (\tilde{x}_i - \tilde{M}_h)^T) \right) = 0 \quad (8)$$

From result 139 in the Matrix Cookbook, $\frac{\partial \text{Tr}(AX)}{\partial X} = A + A^T - (A \circ I) = 2A - (A \circ I)$ if A is symmetric, where \circ means element-wise multiplication, and I is the identity matrix.

Let $A = (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T$. This is a symmetric p by p matrix. This gives

$$(9) \quad \frac{\partial}{\partial \Sigma^{-1}} \left(\text{tr}(\Sigma^{-1}(\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T) \right) = 2(\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T - (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T \circ I$$

From result 141 in the Matrix Cookbook, $\frac{\partial \ln|X|}{\partial X} = 2X^{-1} - (X^{-1} \circ I)$ if X is symmetric. Σ^{-1} is symmetric, therefore $\frac{\partial}{\partial \Sigma^{-1}} (\ln |\Sigma^{-1}|) = 2\Sigma - (\Sigma \circ I)$ (10)

Subbing (9) and (10) into (8), we have

$$\sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) \left[\left(\Sigma - \frac{1}{2}(\Sigma \circ I) \right) - \left((\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T - \frac{1}{2}(\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T \circ I \right) \right] = 0$$

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) \left(\Sigma - (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T \right) = \frac{1}{2} \sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) \left(\Sigma \circ I - (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T \circ I \right)$$

Note that the above is of the form $M = \frac{1}{2}M \circ I$ where M is a matrix. It then follows that $M = 0$ (The zero matrix). This is because the elements of the L.H.S will always be twice the value of the elements on the diagonal of the R.H.S unless $M = 0$. This gives

$$\Rightarrow \sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) \Sigma = \sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T$$

$$\Rightarrow \sum_{i=1}^{(k+1)} = \frac{\sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T}{\sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)})}$$

$$\sum_{i=1}^{(k+1)} = \frac{\sum_{i=1}^n \sum_{h=1}^H r_h(\tilde{x}_i; \bar{\Psi}^{(k)}) (\tilde{x}_i - \tilde{\mu}_h)(\tilde{x}_i - \tilde{\mu}_h)^T}{n} \rightarrow \hat{X}$$