Homework 4

Joshua Oswari - A14751270

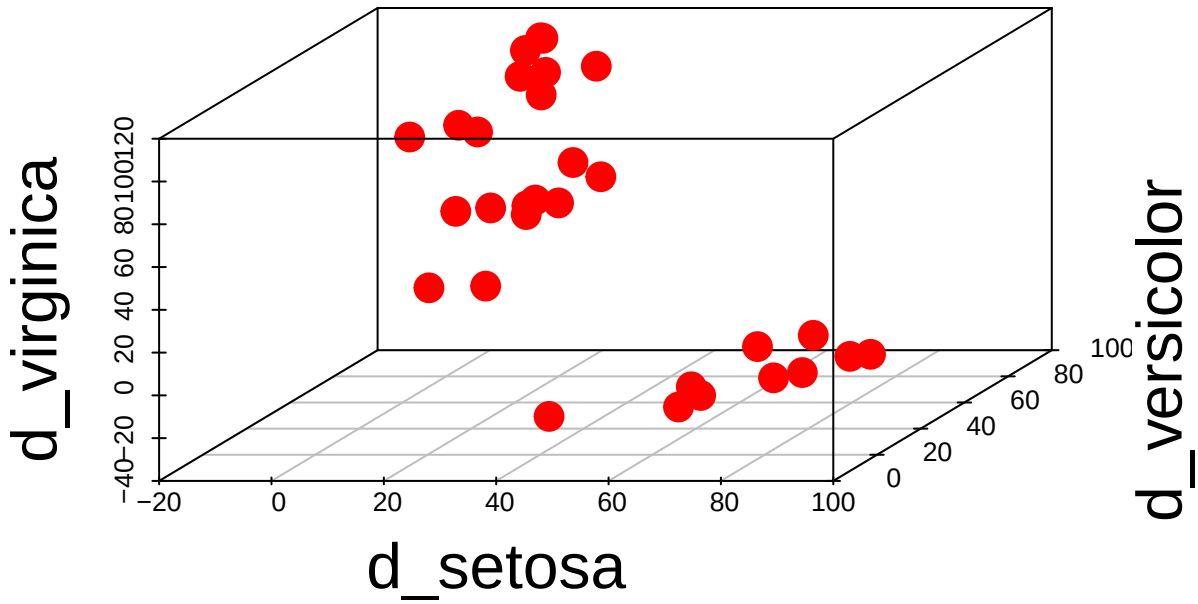
5/1/2019

Problem 1

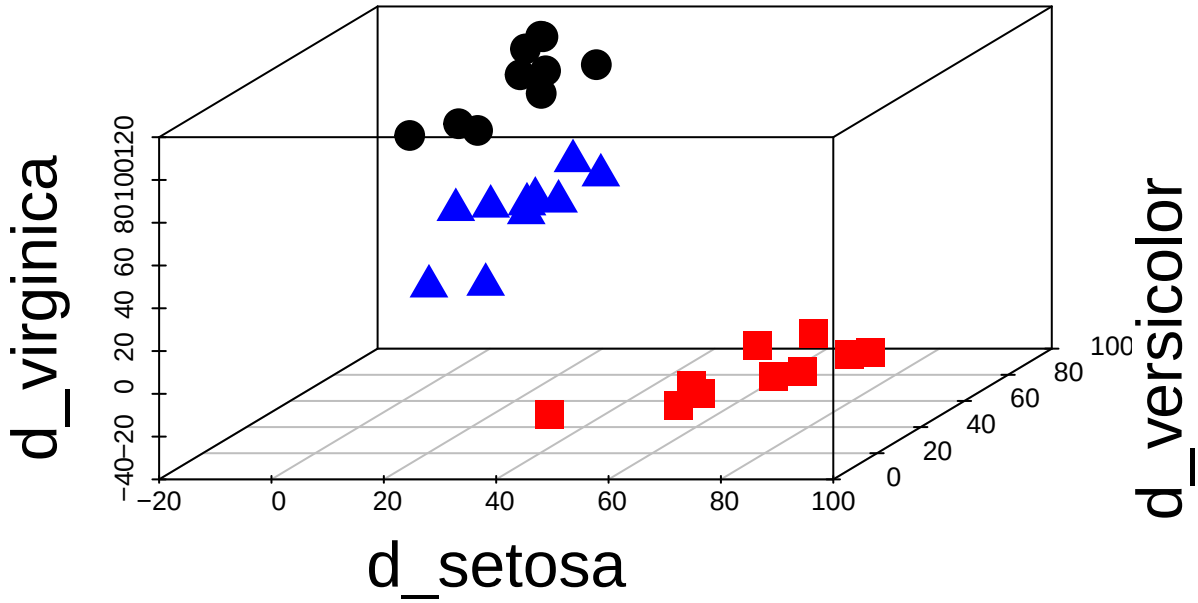
============================== Part 1 ==============================

```
data = read.csv(file = "~/Documents/Math189/iris.csv", header = TRUE, fill = TRUE)

iris = data[,2:6]

#Divide data into train and test
train=iris[c(1:40,51:90, 101:140),]
test1=iris[c(41:50,91:100, 141:150),]

#Sample size
n_setosa=40
n_versicolor=40
n_virginica=40

############## Choose Prior ##############
#Prior=relative sample size in train data

#1st prior
p_setosa1=0.8
p_versicolor1=0.1
p_virginica1=0.1

############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train[1:40,1:4])
Mean_versicolor=colMeans(train[41:80,1:4])
Mean_virginica=colMeans(train[81:120,1:4])

############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train[1:40,1:4])
S_versicolor=cov(train[41:80,1:4])
S_virginica=cov(train[81:120,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n

S_inv=solve(S_pooled)

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa + log(p_setosa1)
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor + log(p_versicolor1)
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica + log(p_virginica1)
```

```
############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica

############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test1)){
  #Read an observation in test data
  x=t(test1[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test1$prediction=prediction

#3D scatter plot

library("scatterplot3d")
col_vec=c(rep("red", 10), rep("blue", 10), rep("black", 10))
pch_vec=c(rep(15, 10), rep(17, 10), rep(19, 10))

scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color="red", pch=19, angle = 55 , cex.symbols=2, cex.lab=2
)
```



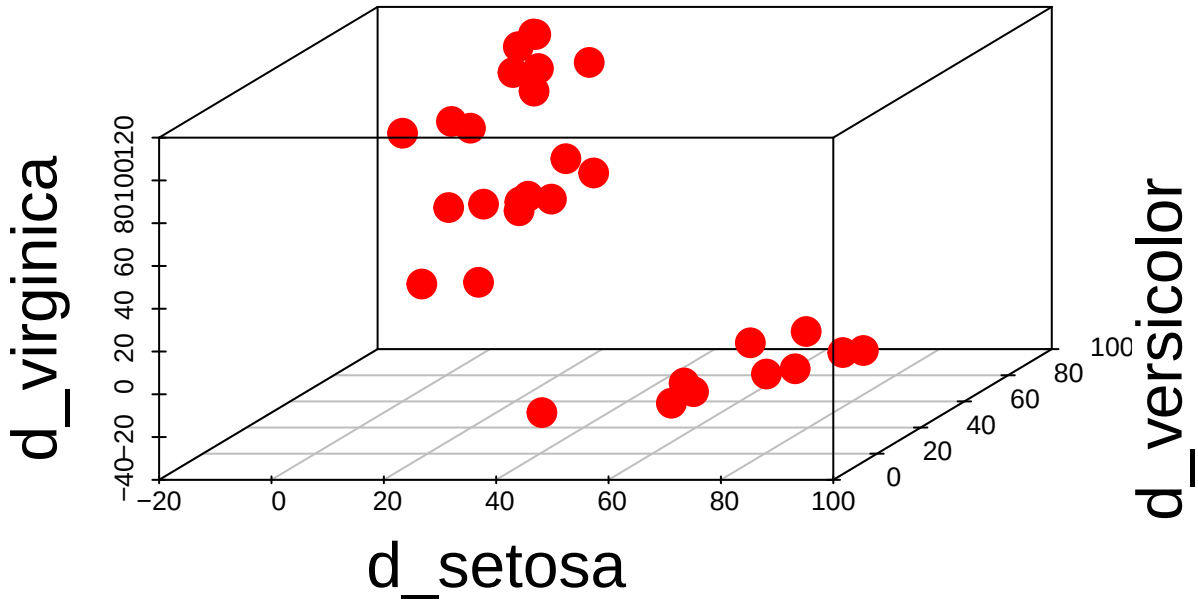

```
scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color=col_vec, pch=pch_vec, angle = 55 , cex.symbols=2, cex.lab=2
)
```



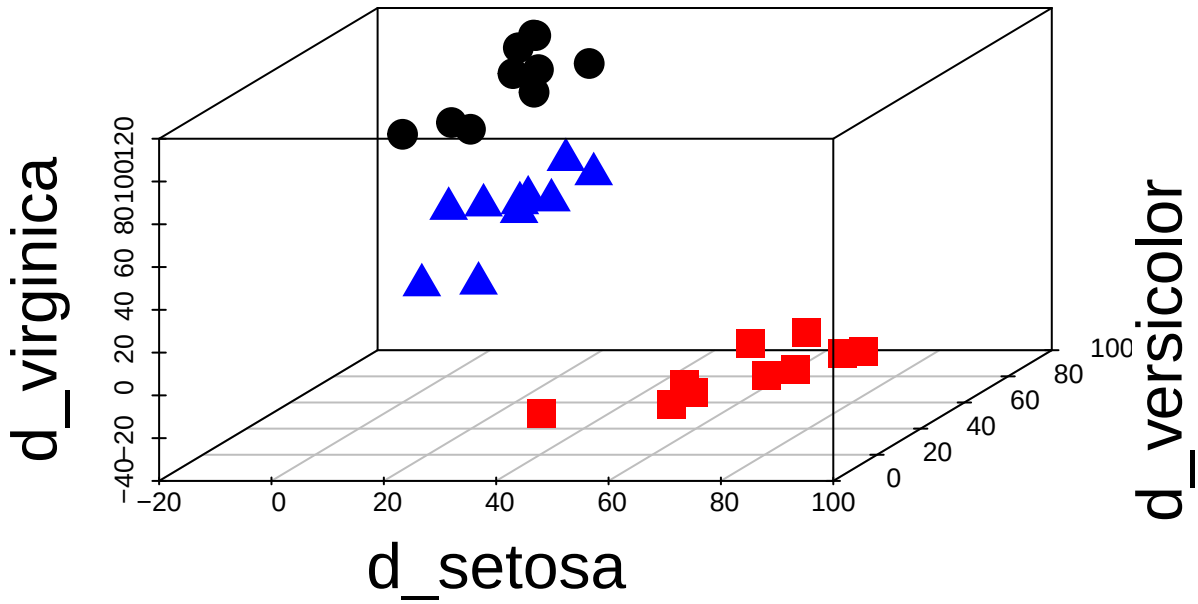

```
#---------------------- 2nd Prior -----------------#


#Divide data into train and test
train=iris[c(1:40,51:90, 101:140),]
test2=iris[c(41:50,91:100, 141:150),]

############### Choose Prior ###############
#Prior=relative sample size in train data

#2nd prior
```

```r
p_setosa2=0.1
p_versicolor2=0.8
p_virginica2=0.1

############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train[1:40,1:4])
Mean_versicolor=colMeans(train[41:80,1:4])
Mean_virginica=colMeans(train[81:120,1:4])

############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train[1:40,1:4])
S_versicolor=cov(train[41:80,1:4])
S_virginica=cov(train[81:120,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n

S_inv=solve(S_pooled)

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa + log(p_setosa2)
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor + log(p_versicolor2)
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica + log(p_virginica2)

############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica

############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test2)){
  #Read an observation in test data
  x=t(test2[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
```

```
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test2$prediction=prediction

#3D scatter plot

library("scatterplot3d")
col_vec=c(rep("red", 10), rep("blue", 10), rep("black", 10))
pch_vec=c(rep(15, 10), rep(17, 10), rep(19, 10))

scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color="red", pch=19, angle = 55 , cex.symbols=2, cex.lab=2
)
```

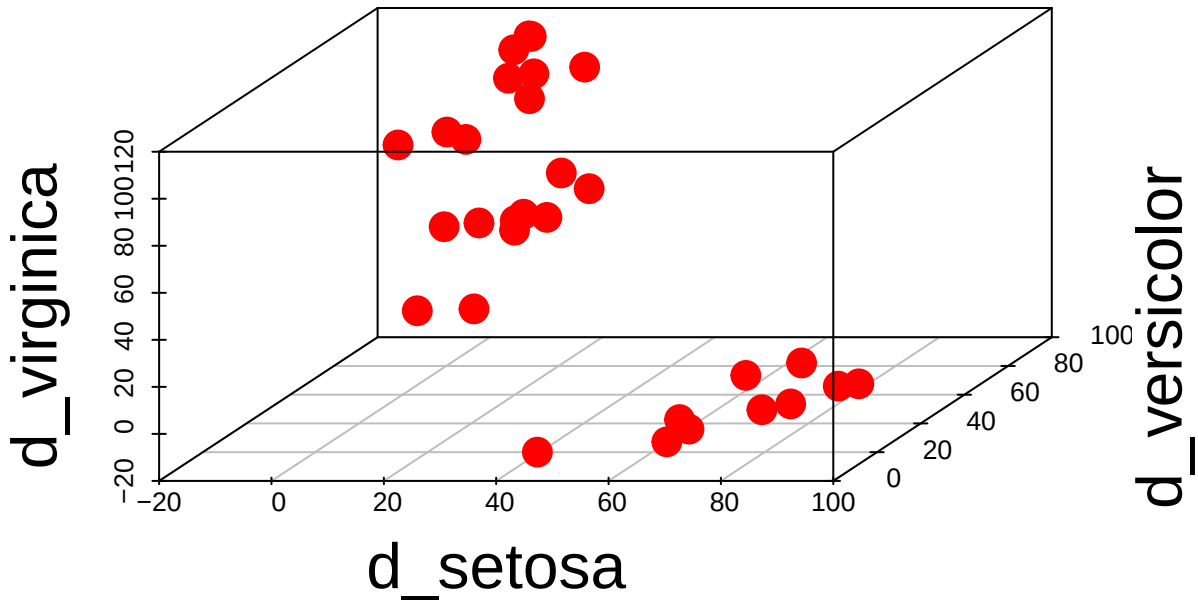



```
scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color=col_vec, pch=pch_vec, angle = 55 , cex.symbols=2, cex.lab=2
)
```

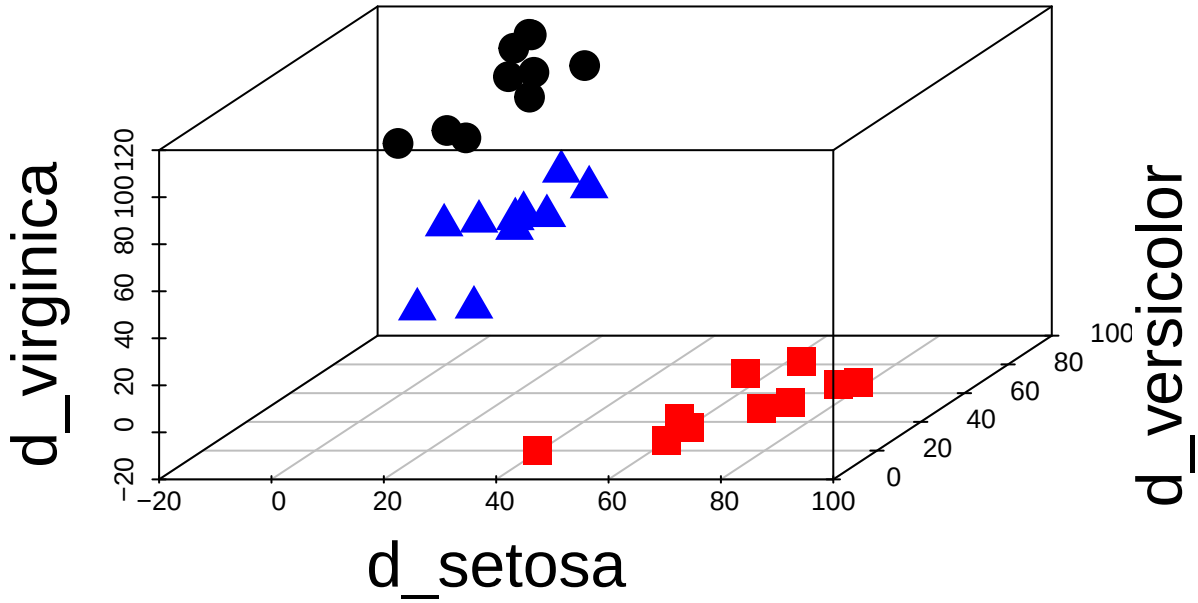



```
#----------------------- 3rd Prior -----------------#
#Divide data into train and test
train=iris[c(1:40,51:90, 101:140),]
test3=iris[c(41:50,91:100, 141:150),]

############## Choose Prior ##############
#Prior=relative sample size in train data

#3rd prior
p_setosa3=0.1
p_versicolor3=0.1
p_virginica3=0.8

############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train[1:40,1:4])
Mean_versicolor=colMeans(train[41:80,1:4])
Mean_virginica=colMeans(train[81:120,1:4])

############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train[1:40,1:4])
S_versicolor=cov(train[41:80,1:4])
S_virginica=cov(train[81:120,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n_

S_inv=solve(S_pooled)

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa + log(p_setosa3)
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor + log(p_versicolor3)
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica + log(p_virginica3)
```

```
############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica

############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test3)){
  #Read an observation in test data
  x=t(test3[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test3$prediction=prediction

#3D scatter plot

library("scatterplot3d")
col_vec=c(rep("red", 10), rep("blue", 10), rep("black", 10))
pch_vec=c(rep(15, 10), rep(17, 10), rep(19, 10))

scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color="red", pch=19, angle = 55 , cex.symbols=2, cex.lab=2
)
```




```
scatterplot3d(x = d_setosa_vec, y = d_versicolor_vec, z=d_virginica_vec,
              xlab = "d_setosa", ylab = "d_versicolor", zlab="d_virginica",
              color=col_vec, pch=pch_vec, angle = 55 , cex.symbols=2, cex.lab=2
)
```




```
# Report:
# No, the LDA Method is not sensitive to the choices of prior
# between all those three tests, we get the same answer between
# the species and predicition.
```

========== Part 2 ==========

```
############## For size 90 ##############
#Divide data into train and test size 90
train1=iris[c(1:30,51:80, 101:130),]
test1=iris[c(31:50,81:100, 131:150),]
```

```r
#Sample size
n_setosa=30
n_versicolor=30
n_virginica=30

############## Choose Prior ##############
#Prior=relative sample size in train data
p_setosa=n_setosa/90
p_versicolor=n_versicolor/90
p_virginica=n_virginica/90

############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train1[1:30,1:4])
Mean_versicolor=colMeans(train1[31:60,1:4])
Mean_virginica=colMeans(train1[61:90,1:4])

############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train1[1:30,1:4])
S_versicolor=cov(train1[31:60,1:4])
S_virginica=cov(train1[61:90,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n_

S_inv=solve(S_pooled)

#Simple way
#S_pooled=(S_setosa+S_versicolor+S_virginica)/3

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica


############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica


############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test1)){
  #Read an observation in test data
```

```
  x=t(test1[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test1$prediction=prediction

#result from test1: there are 2 wrong in the prediction

#-------------------------------------------------------------#

############### For size 60 ###############
#Divide data into train and test size 60
train2=iris[c(1:20,51:70, 101:120),]
test2=iris[c(21:50,71:100, 121:150),]

#Sample size
n_setosa=20
n_versicolor=20
n_virginica=20

############### Choose Prior ###############
#Prior=relative sample size in train data
p_setosa=n_setosa/60
p_versicolor=n_versicolor/60
p_virginica=n_virginica/60

############### Calculate sample mean vectors ###############
Mean_setosa=colMeans(train2[1:20,1:4])
Mean_versicolor=colMeans(train2[21:40,1:4])
Mean_virginica=colMeans(train2[41:60,1:4])

############### Calculate pooled variance-covariance matrix ###############
#Sample variance-covariance matrix for each species
S_setosa=cov(train2[1:20,1:4])
S_versicolor=cov(train2[21:40,1:4])
S_virginica=cov(train2[41:60,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n
```

```r
S_inv=solve(S_pooled)

#Simple way
#S_pooled=(S_setosa+S_versicolor+S_virginica)/3

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica


############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica


############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test2)){
  #Read an observation in test data
  x=t(test2[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test2$prediction=prediction

#result from test2: there are 3 wrong in the prediction

#-------------------------------------------------------------#

############## For size 30 ##############
#Divide data into train and test size 30
```

```r
train3=iris[c(1:10,51:60, 101:110),]
test3=iris[c(11:50,61:100, 111:150),]

#Sample size
n_setosa=10
n_versicolor=10
n_virginica=10

############## Choose Prior ##############
#Prior=relative sample size in train data
p_setosa=n_setosa/30
p_versicolor=n_versicolor/30
p_virginica=n_virginica/30

############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train3[1:10,1:4])
Mean_versicolor=colMeans(train3[11:20,1:4])
Mean_virginica=colMeans(train3[21:30,1:4])

############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train3[1:10,1:4])
S_versicolor=cov(train3[11:20,1:4])
S_virginica=cov(train3[21:30,1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n

S_inv=solve(S_pooled)

#Simple way
#S_pooled=(S_setosa+S_versicolor+S_virginica)/3

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica


############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica


############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")
```

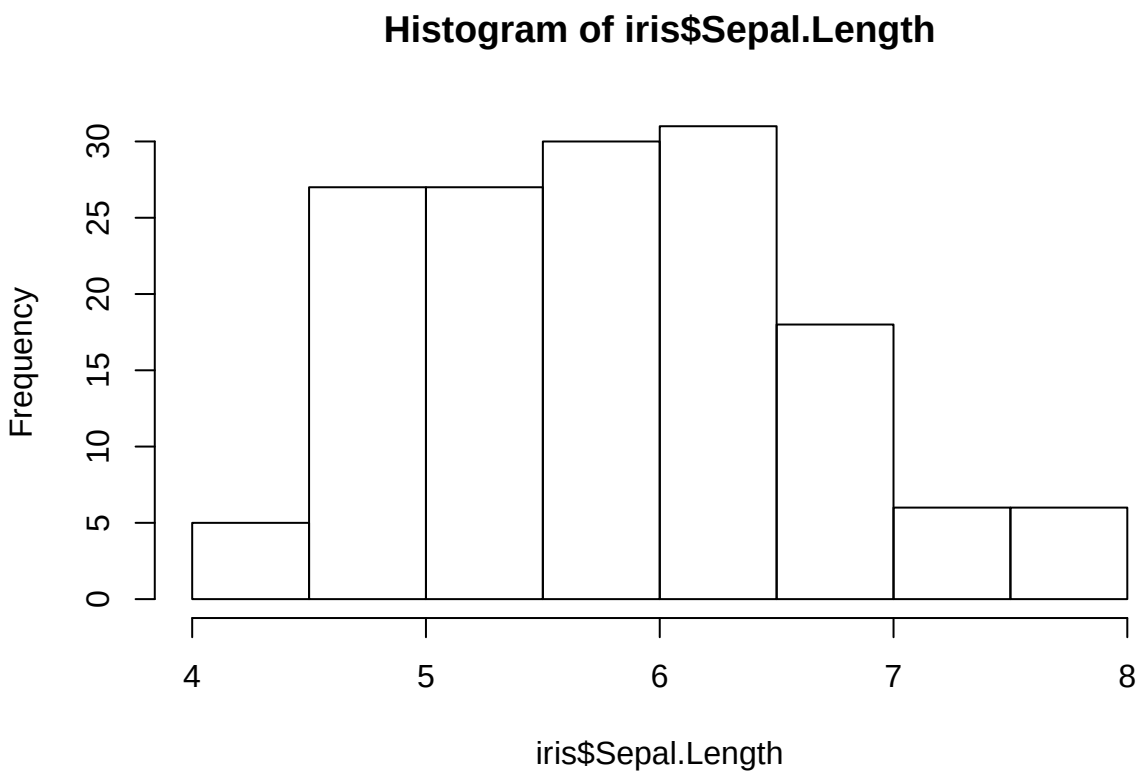
```r
for(i in 1:nrow(test3)){
  #Read an observation in test data
  x=t(test3[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#Combine the predicted results to the test dataset.
test3$prediction=prediction

#result from test3: there are >3 wrong in the prediction

# In conclusion the smaller the sample size it goes, then
# the more fail prediction it will get. Vice versa, the larger
# the sample size we chose then the more accurate the prediction we
# will get. Here, we can see from sample size 90 , 60, 30.
# sample size 60 failed 2 predictions, and then it goes
# exponentially the more failure prediction we got.
```

========== Part 3 ==========

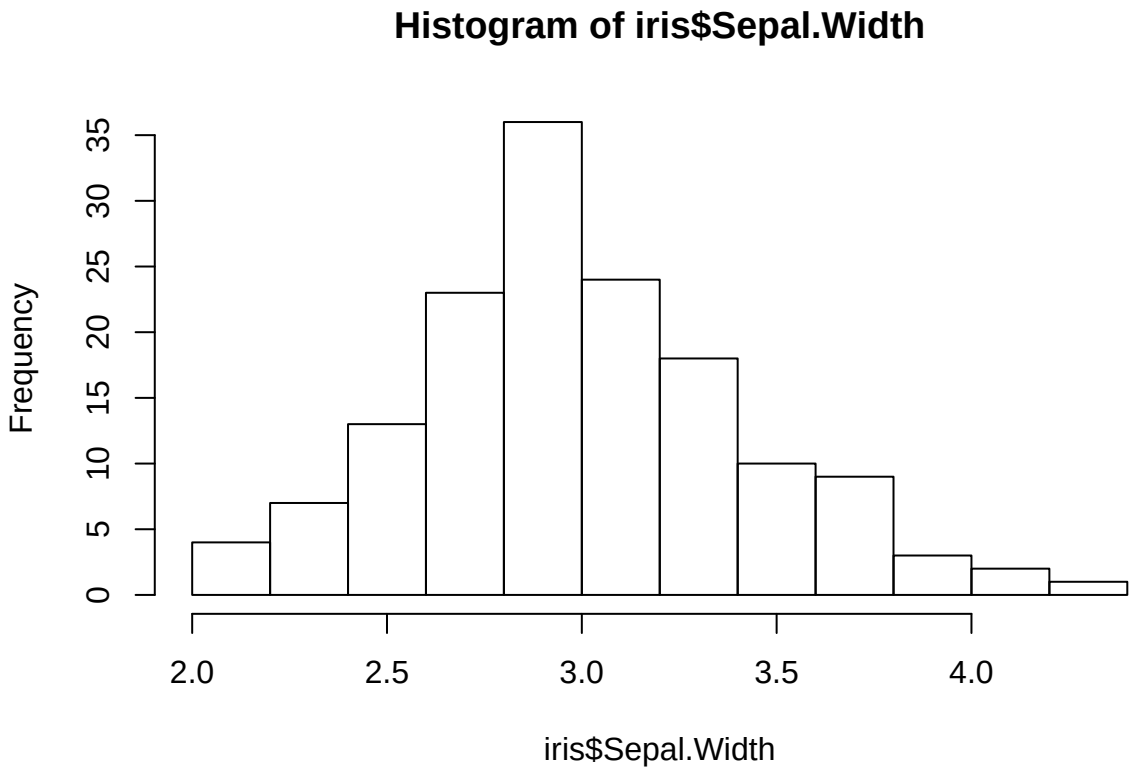
```r
############## Load Data ##############
#train 50
#test 100
trainsample = sample(1:150, 50, replace=FALSE)
testsample = sample(1:150, 100, replace= FALSE)
b = 100


for(i in 1:b){
#Divide data into train and test
train = iris[trainsample[b],]
test = train[testsample[b],]
#Sample size
n_setosa=trainsample[b]
n_versicolor=trainsample[b]
n_virginica=trainsample[b]

############## Choose Prior ##############
#Prior=relative sample size in train data
p_setosa=trainsample[1]/50
p_versicolor=trainsample[1]/50
p_virginica=trainsample[1]/50
```

```
############## Calculate sample mean vectors ##############
Mean_setosa=colMeans(train[trainsample[b],1:4])
Mean_versicolor=colMeans(train[trainsample[b],1:4])
Mean_virginica=colMeans(train[trainsample[b],1:4])



############## Calculate pooled variance-covariance matrix ##############
#Sample variance-covariance matrix for each species
S_setosa=cov(train[trainsample[b],1:4])
S_versicolor=cov(train[trainsample[b],1:4])
S_virginica=cov(train[trainsample[b],1:4])

#Complete fomula
S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-1)*S_virginica)/(n_setosa+n

S_inv=solve(S_pooled)

#Simple way
#S_pooled=(S_setosa+S_versicolor+S_virginica)/3

############## Calculate alpha_i ##############

alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa + log(p_setosa)
alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor + log(p_versicolor)
alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica + log(p_virginica)


############## Calculate beta_i ##############

beta_setosa=S_inv %*% Mean_setosa
beta_versicolor=S_inv %*% Mean_versicolor
beta_virginica=S_inv %*% Mean_virginica


############## Classification ##############
prediction=c()
d_setosa_vec=c()
d_versicolor_vec=c()
d_virginica_vec=c()
label=c("setosa", "versicolor", "virginica")

for(i in 1:nrow(test)){
  #Read an observation in test data
  x=t(test[i,1:4])

  #Calculate linear discriminant functions for each species
  d_setosa=alpha_setosa+ t(beta_setosa) %*% x
  d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
  d_virginica=alpha_virginica+ t(beta_virginica) %*% x

  #Classify the observation to the species with highest function value
  d_vec=c(d_setosa, d_versicolor, d_virginica)
```

```
  prediction=append(prediction, label[which.max( d_vec )])

  d_setosa_vec=append(d_setosa_vec, d_setosa)
  d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
  d_virginica_vec=append(d_virginica_vec, d_virginica)
}

#test$prediction=prediction
}

#plot a hist
x <- hist(iris$Sepal.Length)
```




```
y = hist(iris$Sepal.Width)
```


Histogram of iris$Sepal.Width
Frequency
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
iris$Sepal.Width