Problem Set 5

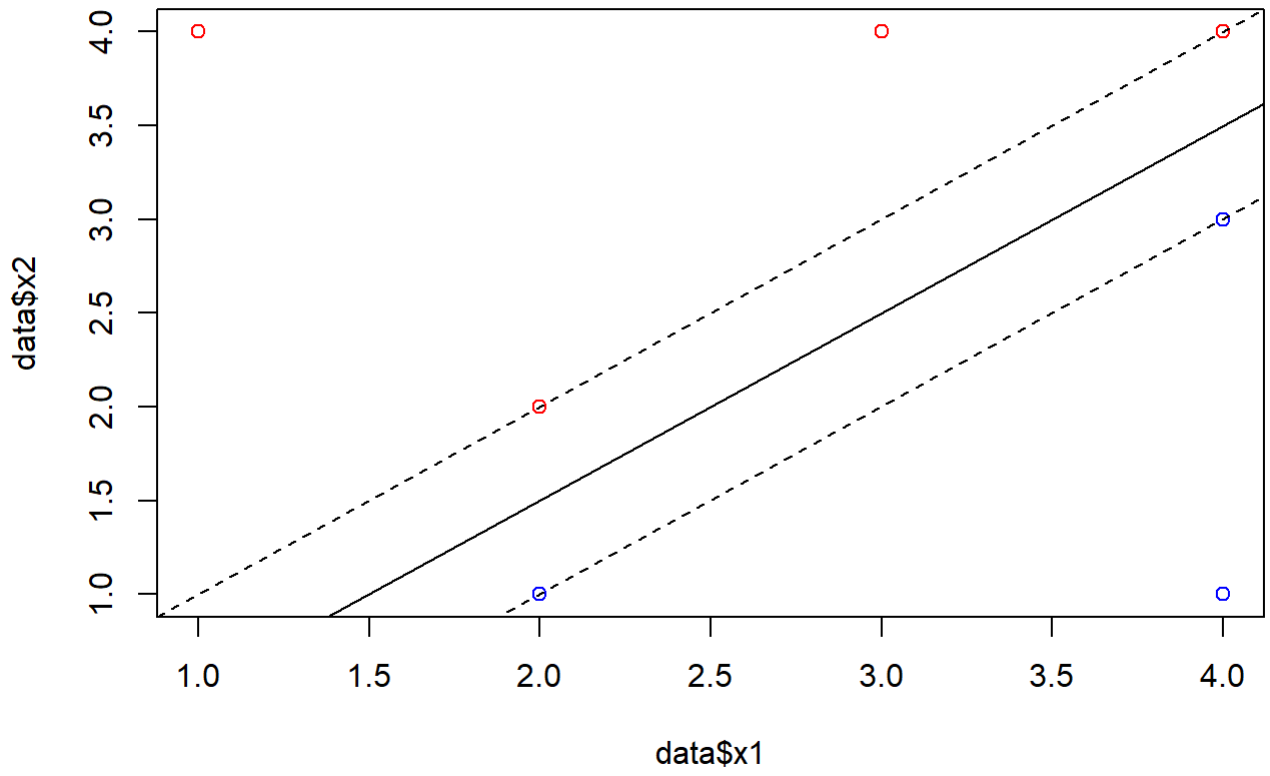
Tom Hanna

11/26/2020

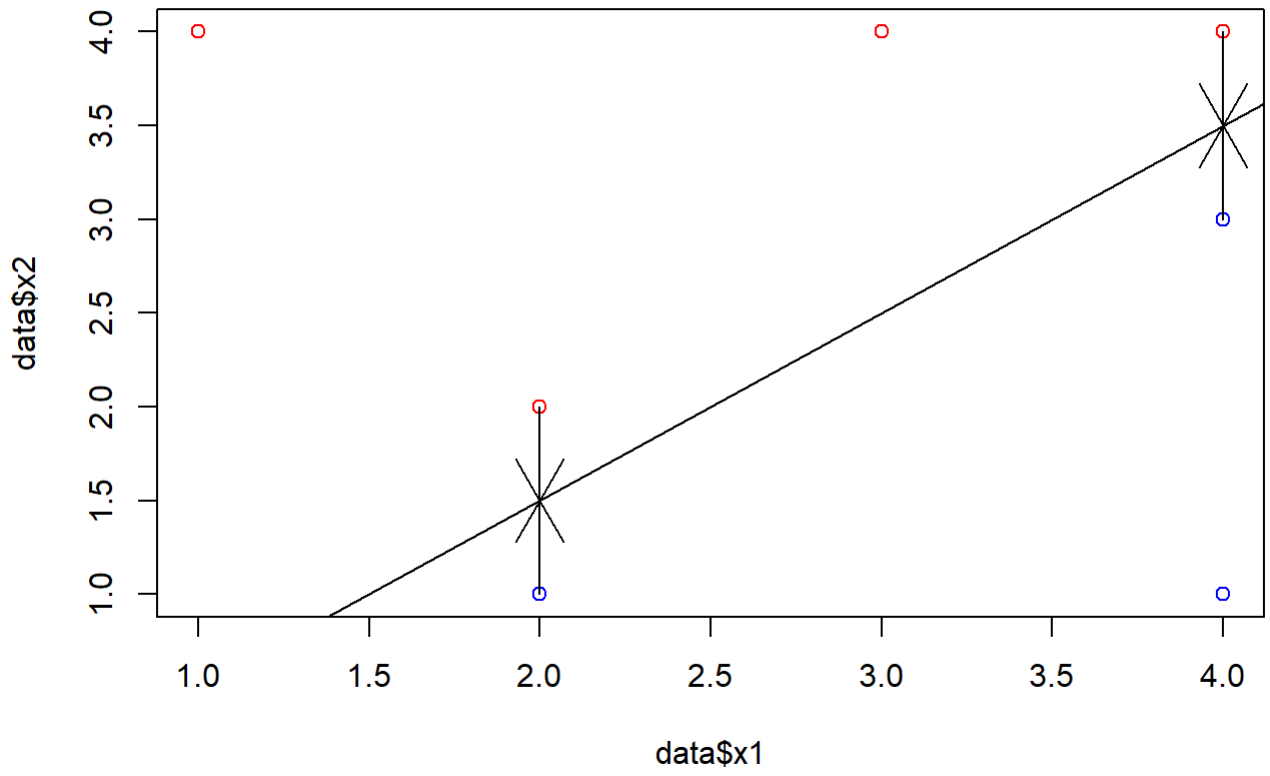
```
gc()
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 398332 21.3     818767 43.8   638648 34.2
## Vcells 732168  5.6    8388608 64.0  1632921 12.5
```

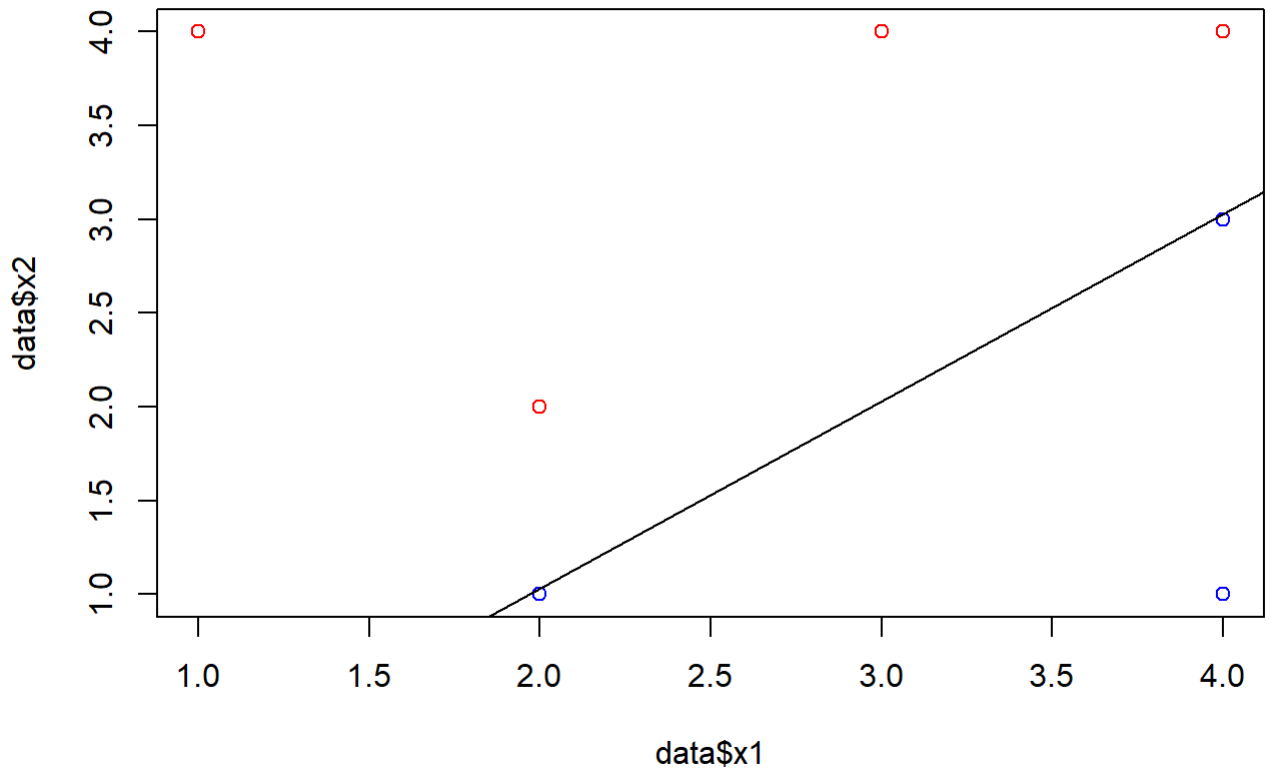
```
rm(list=ls())
options(scipen = 999)

setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 5")

##Chapter 9

#3

#a

set.seed(75)
x1 = c(3, 2, 4, 1, 2, 4, 4)
x2 = c(4, 2, 4, 4, 1, 3, 1)
colors = c("red","red","red","red","blue","blue","blue")


data <- as.data.frame(cbind(x1,x2,colors))

plot(data$x1,data$x2, col = colors)

#b

#β0 + β1X1 + β2X2 = 0 (from 9.1)

#The equation is -.5 + X1 - X2 = 0


abline(-0.5, 1)

#c

#Classify as Red if -.5 + X1 - X2 < 0
#Classify as Blue if -.5 + X1 - X2 > 0

#d

abline(-1, 1, lty = 2)
abline(0, 1, lty = 2)
```



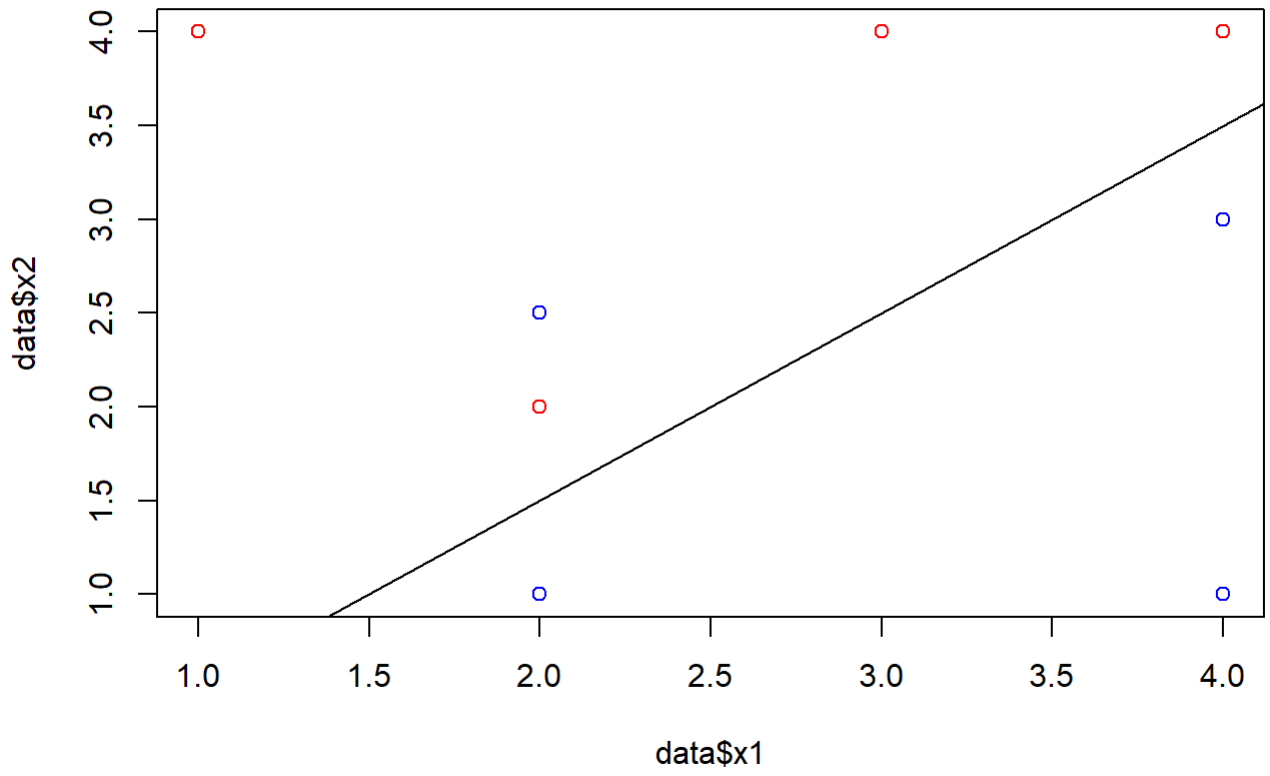

```
#e

plot(data$x1, data$x2, col = colors)
abline(-0.5, 1)
arrows(2, 1, 2, 1.5)
arrows(2, 2, 2, 1.5)
arrows(4, 4, 4, 3.5)
arrows(4, 3, 4, 3.5)
```




```
#f - The 7th observation is well into the blue territory, well outside the margin.

#g -

plot(data$x1,data$x2, col = colors)
abline(-.97,1)
```




```
# equation for the hyperplane is -0.97 + x1 + x2 = 0

plot(data$x1,data$x2, col = colors)
abline(-0.5,1)
points(c(2),c(2.5), col = c("blue"))
```




```
#7

#a

library(ISLR)

data.auto <- Auto
View(data.auto)
mpg.median = median(data.auto$mpg)

data.auto$high.mpg = ifelse(data.auto$mpg > mpg.median, 1, 0)
data.auto$mileage = as.factor(data.auto$high.mpg)
summary(data.auto$mileage)
```

```
##   0   1
## 196 196
```

```
data.auto <- subset(data.auto, select = -c(high.mpg) )

library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

```
svmfita =svm(data.auto$mileage ~., data=data.auto , kernel ="linear", cost=10, scale=FALSE)
summary(svmfita)
```

```
##
## Call:
## svm(formula = data.auto$mileage ~ ., data = data.auto, kernel = "linear",
##     cost = 10, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  13
##
##  ( 6 7 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
svmfita$index
```

```
##  [1]  16 110 193 281 359 384  49  82 101 167 169 176 297
```

```
set.seed(1)
tune.out=tune(svm,mileage~.,data=data.auto,kernel="linear", ranges =
                  list(cost=c(0.001,0.01,0.1,1,5,10,100)))

summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     1
##
## - best performance: 0.01025641
##
## - Detailed performance results:
##      cost      error dispersion
## 1   0.001 0.09442308 0.04519425
## 2   0.010 0.07653846 0.03617137
## 3   0.100 0.04596154 0.03378238
## 4   1.000 0.01025641 0.01792836
## 5   5.000 0.02051282 0.02648194
## 6  10.000 0.02051282 0.02648194
## 7 100.000 0.03076923 0.03151981
```

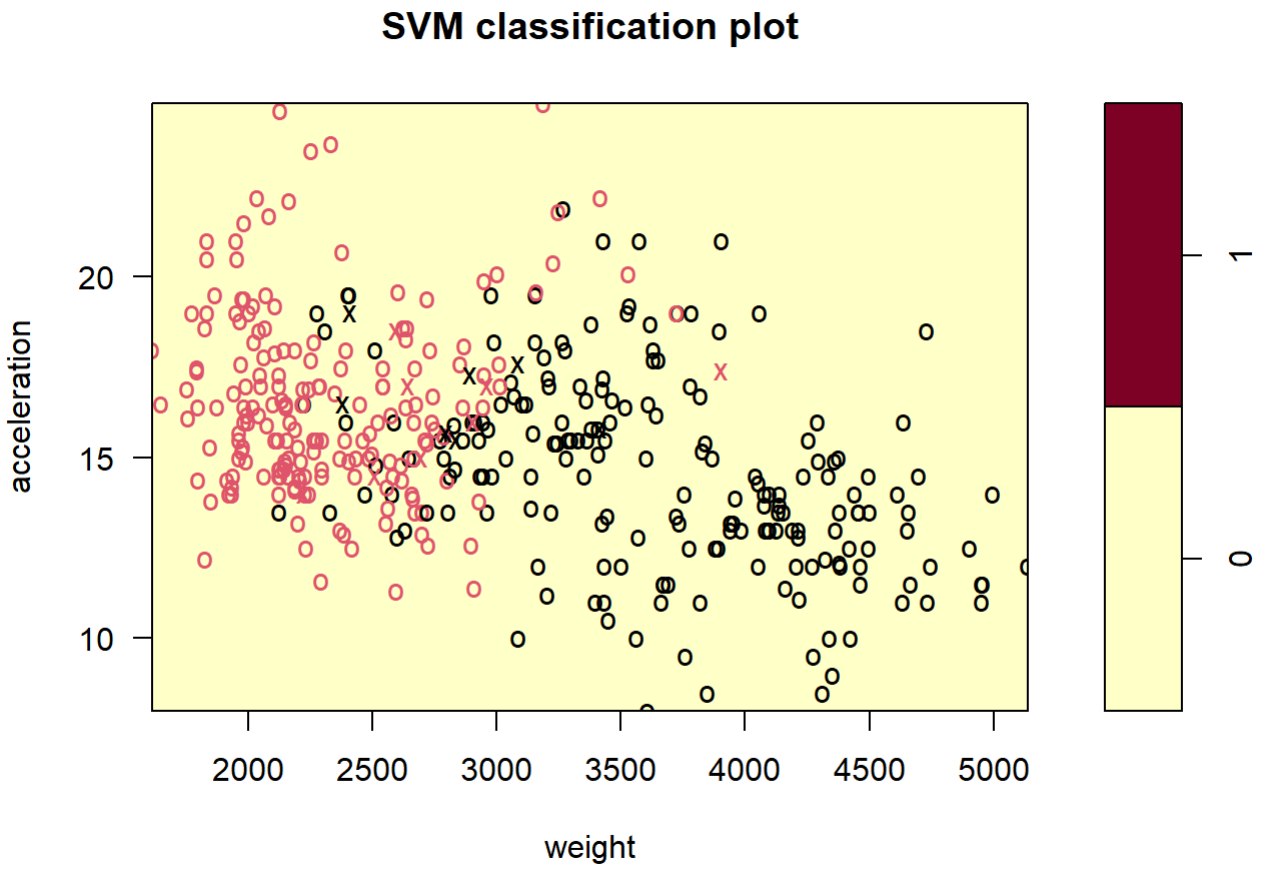
```
#cross validation is minimized at cost 1 with error = 0.01025641.

#c

set.seed(1)
tune.outp=tune(svm,mileage~.,data=data.auto,kernel="polynomial", ranges =
                  list(cost=c(0.001,0.01,0.1,1,5,10,100), degree = c(2, 3, 4)))

summary(tune.outp)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost degree
##   100      2
##
## - best performance: 0.3013462
##
## - Detailed performance results:
##       cost degree     error dispersion
## 1    0.001      2 0.5511538 0.04366593
## 2    0.010      2 0.5511538 0.04366593
## 3    0.100      2 0.5511538 0.04366593
## 4    1.000      2 0.5511538 0.04366593
## 5    5.000      2 0.5511538 0.04366593
## 6   10.000      2 0.5130128 0.08963366
## 7  100.000      2 0.3013462 0.09961961
## 8    0.001      3 0.5511538 0.04366593
## 9    0.010      3 0.5511538 0.04366593
## 10   0.100      3 0.5511538 0.04366593
## 11   1.000      3 0.5511538 0.04366593
## 12   5.000      3 0.5511538 0.04366593
## 13  10.000      3 0.5511538 0.04366593
## 14 100.000      3 0.3446154 0.09821588
## 15   0.001      4 0.5511538 0.04366593
## 16   0.010      4 0.5511538 0.04366593
## 17   0.100      4 0.5511538 0.04366593
## 18   1.000      4 0.5511538 0.04366593
## 19   5.000      4 0.5511538 0.04366593
## 20  10.000      4 0.5511538 0.04366593
## 21 100.000      4 0.5511538 0.04366593
```

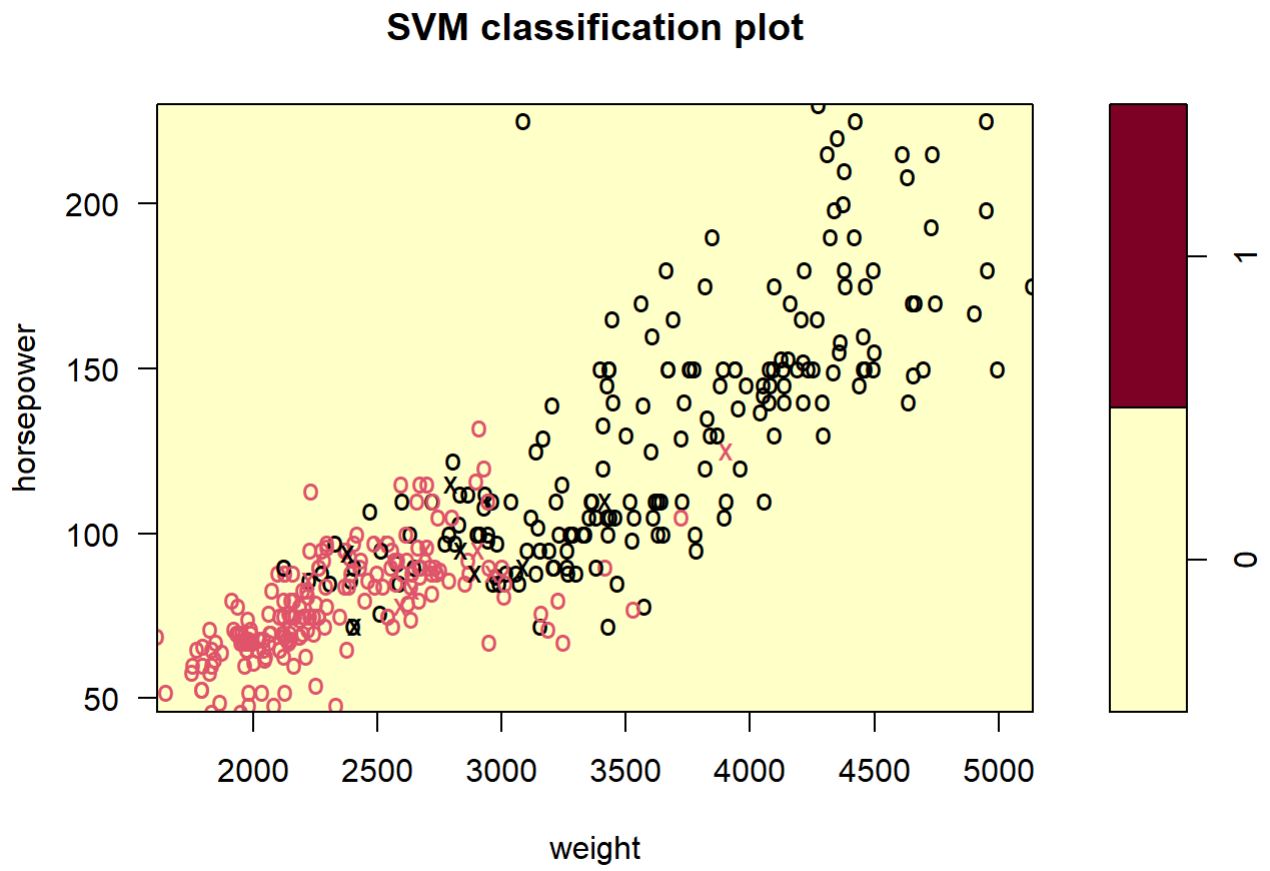
```
#error is minimized at cost = 100 and degree = 2 with a error = 0.3013462

set.seed(1)
tune.outr=tune(svm,mileage~.,data=data.auto,kernel="radial", ranges =
                  list(cost=c(0.001,0.01,0.1,1,5,10,100), gamma = c(0.01, 0.1, 1, 5, 10)))

summary(tune.outr)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##   100  0.01
##
## - best performance: 0.01282051
##
## - Detailed performance results:
##       cost gamma      error dispersion
## 1    0.001  0.01 0.55115385 0.04366593
## 2    0.010  0.01 0.55115385 0.04366593
## 3    0.100  0.01 0.08929487 0.04382379
## 4    1.000  0.01 0.07403846 0.03522110
## 5    5.000  0.01 0.04852564 0.03303346
## 6   10.000  0.01 0.02557692 0.02093679
## 7  100.000  0.01 0.01282051 0.01813094
## 8    0.001  0.10 0.55115385 0.04366593
## 9    0.010  0.10 0.21711538 0.09865227
## 10   0.100  0.10 0.07903846 0.03874545
## 11   1.000  0.10 0.05371795 0.03525162
## 12   5.000  0.10 0.02820513 0.03299190
## 13  10.000  0.10 0.03076923 0.03375798
## 14 100.000  0.10 0.03583333 0.02759051
## 15   0.001  1.00 0.55115385 0.04366593
## 16   0.010  1.00 0.55115385 0.04366593
## 17   0.100  1.00 0.55115385 0.04366593
## 18   1.000  1.00 0.06384615 0.04375618
## 19   5.000  1.00 0.05884615 0.04020934
## 20  10.000  1.00 0.05884615 0.04020934
## 21 100.000  1.00 0.05884615 0.04020934
## 22   0.001  5.00 0.55115385 0.04366593
## 23   0.010  5.00 0.55115385 0.04366593
## 24   0.100  5.00 0.55115385 0.04366593
## 25   1.000  5.00 0.49493590 0.04724924
## 26   5.000  5.00 0.48217949 0.05470903
## 27  10.000  5.00 0.48217949 0.05470903
## 28 100.000  5.00 0.48217949 0.05470903
## 29   0.001 10.00 0.55115385 0.04366593
## 30   0.010 10.00 0.55115385 0.04366593
## 31   0.100 10.00 0.55115385 0.04366593
## 32   1.000 10.00 0.51794872 0.05063697
## 33   5.000 10.00 0.51794872 0.04917316
## 34  10.000 10.00 0.51794872 0.04917316
## 35 100.000 10.00 0.51794872 0.04917316
```

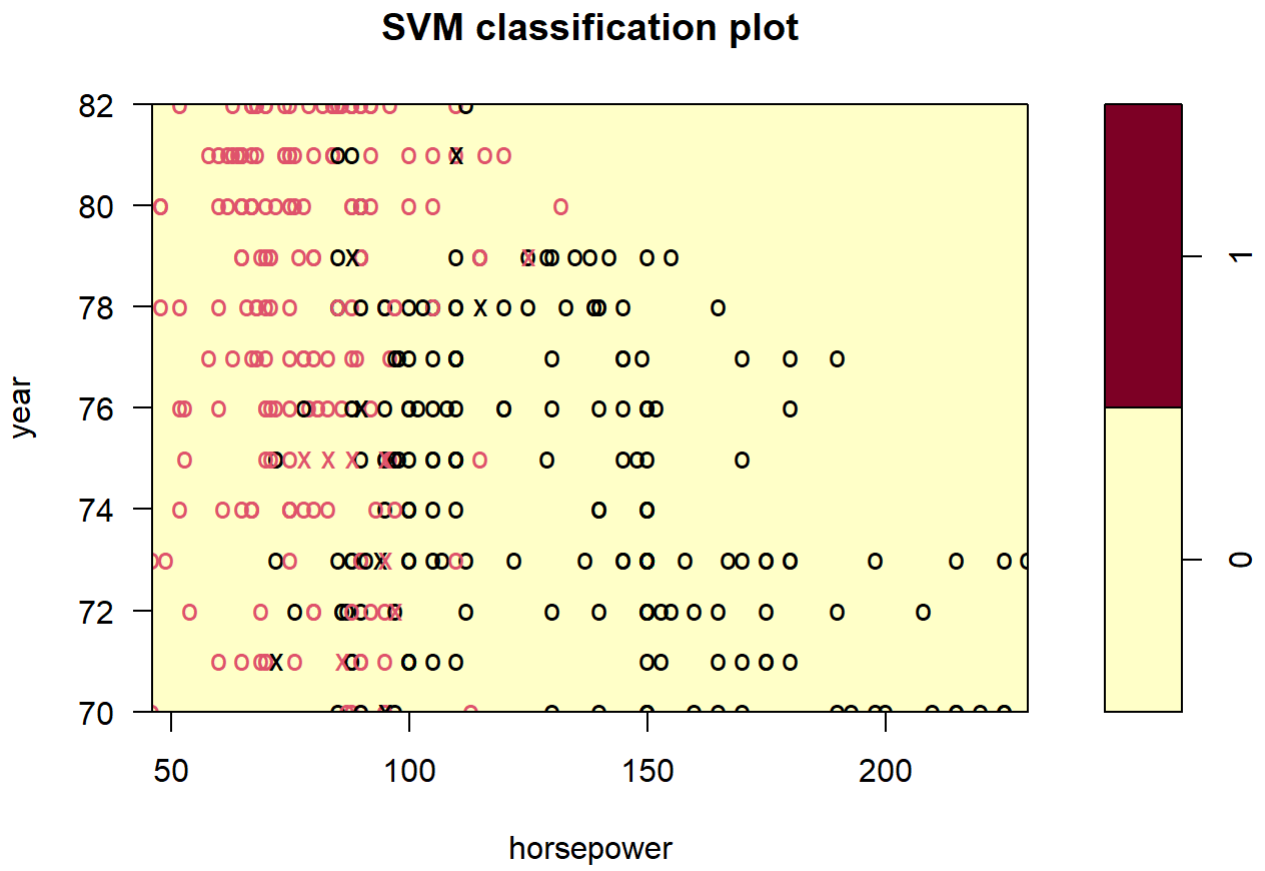
```
#error is minimized at cost = 100 and gamma = 0.01 with error = 0.01282051

#d

svm.linear <- svm(mileage ~., data=data.auto , kernel ="linear", cost=1, scale=FALSE)

plot(svm.linear,data = data.auto, acceleration ~ weight)
```



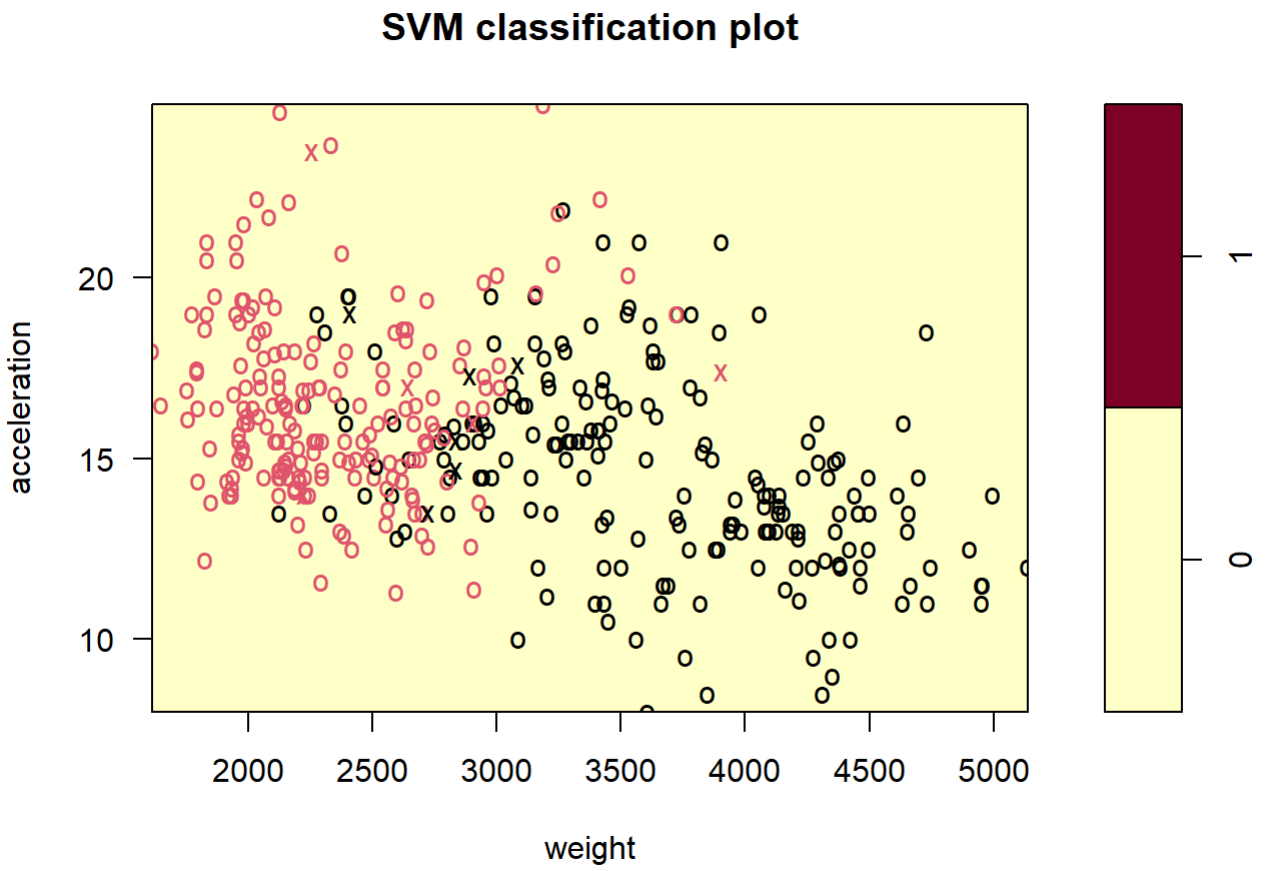

```
plot(svm.linear, data = data.auto, horsepower ~ weight)
```



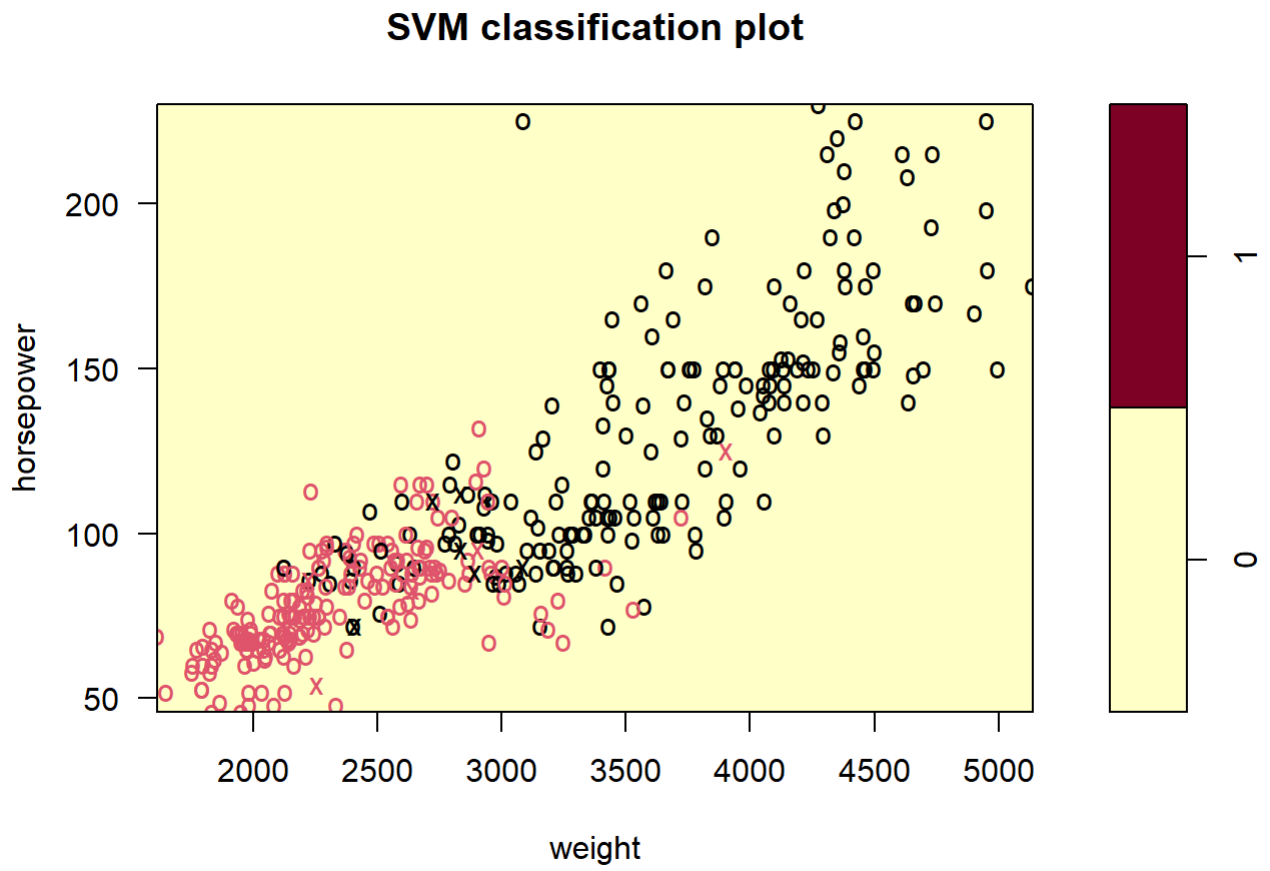

```
plot(svm.linear, data = data.auto, year ~ horsepower)
```



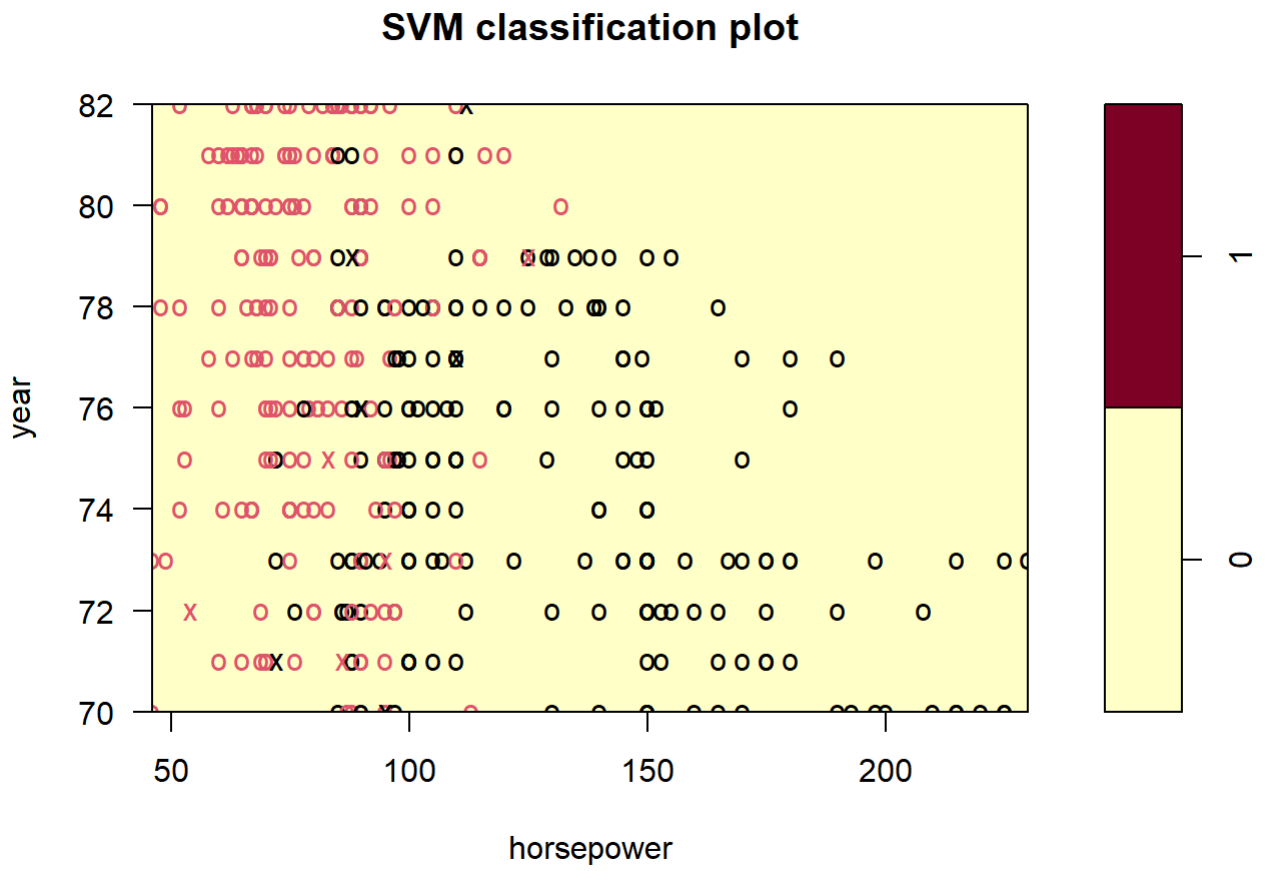

```
svm.polynomial <- svm(mileage ~., data=data.auto , kernel ="polynomial", cost=100, degree = 2, s
cale=FALSE)

plot(svm.polynomial,data = data.auto, acceleration ~ weight)
```



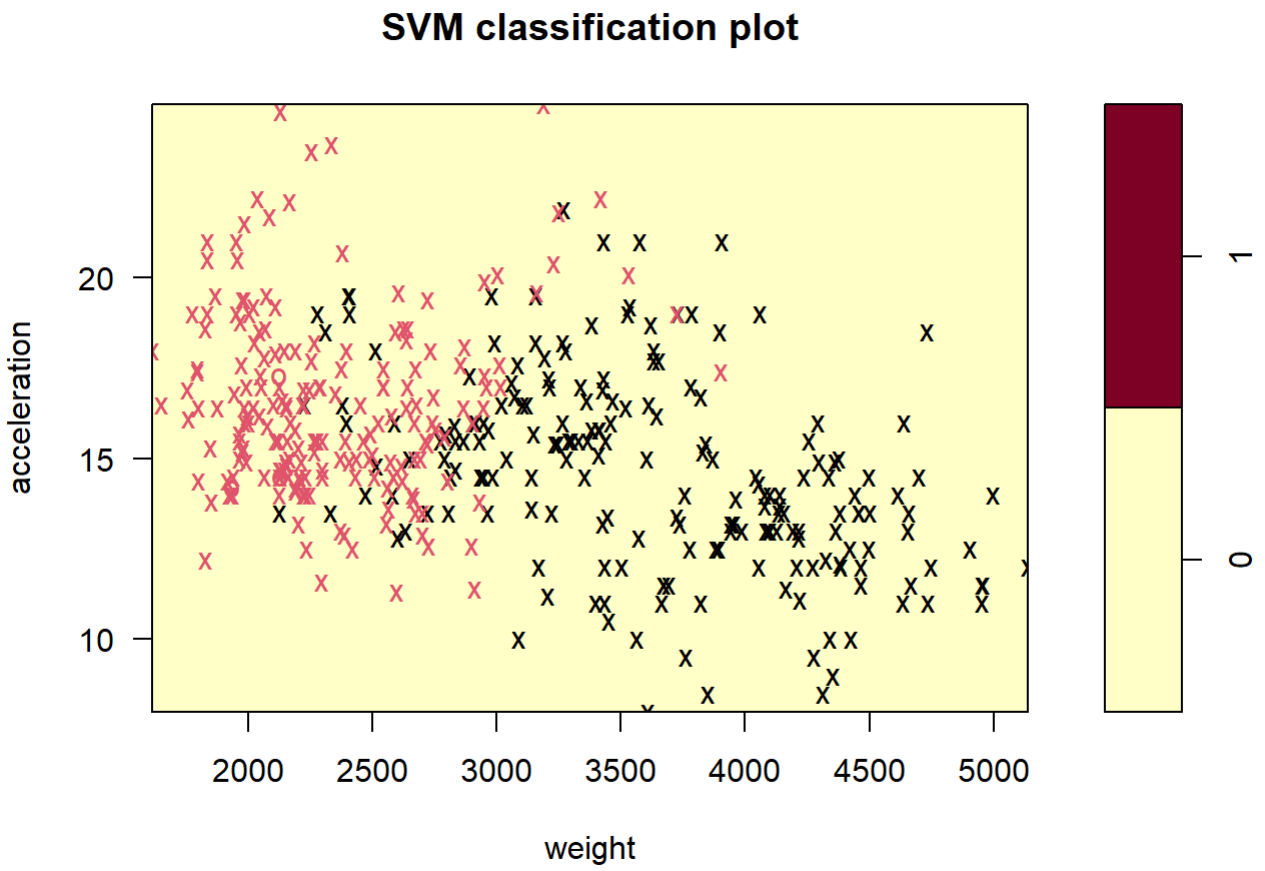

```
plot(svm.polynomial, data = data.auto, horsepower ~ weight)
```



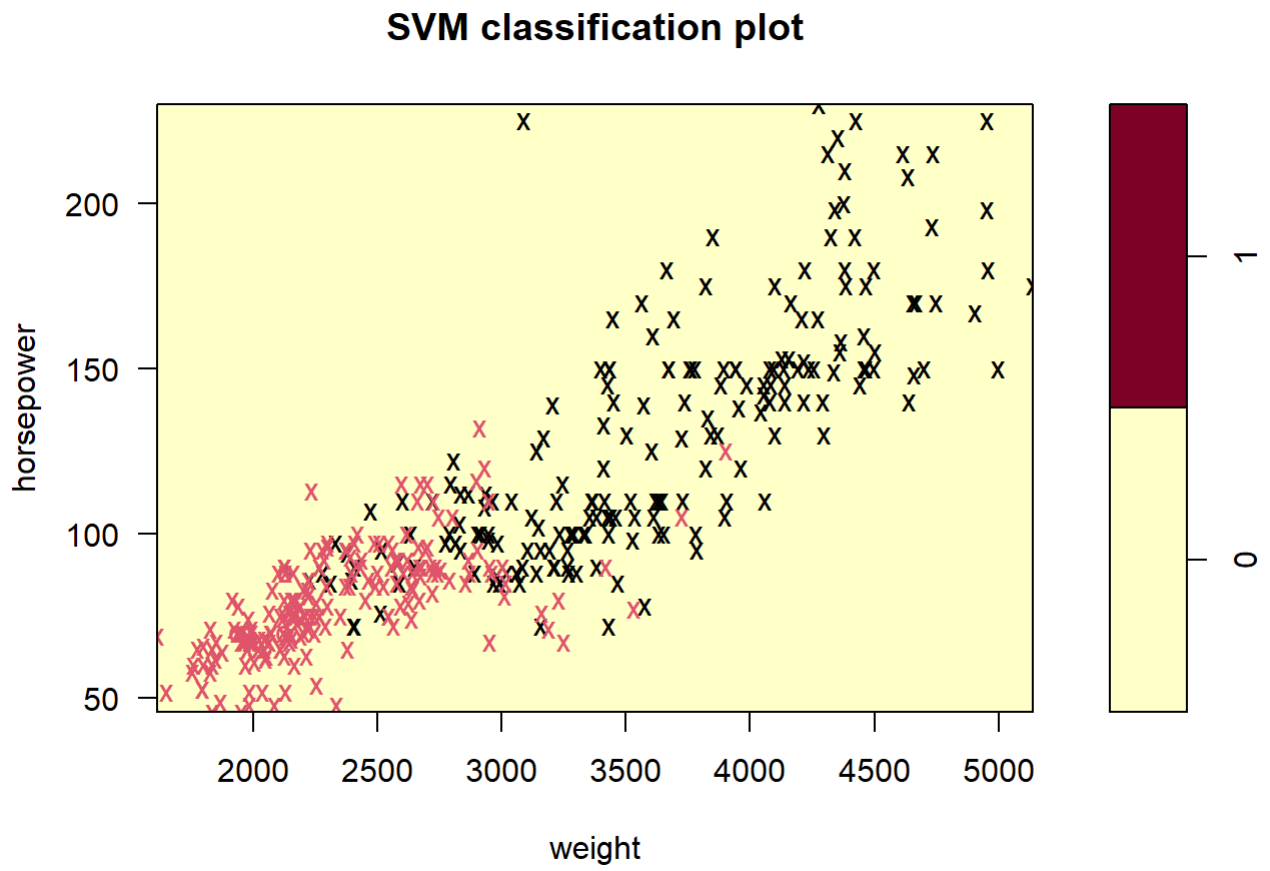

```
plot(svm.polynomial, data = data.auto, year ~ horsepower)
```



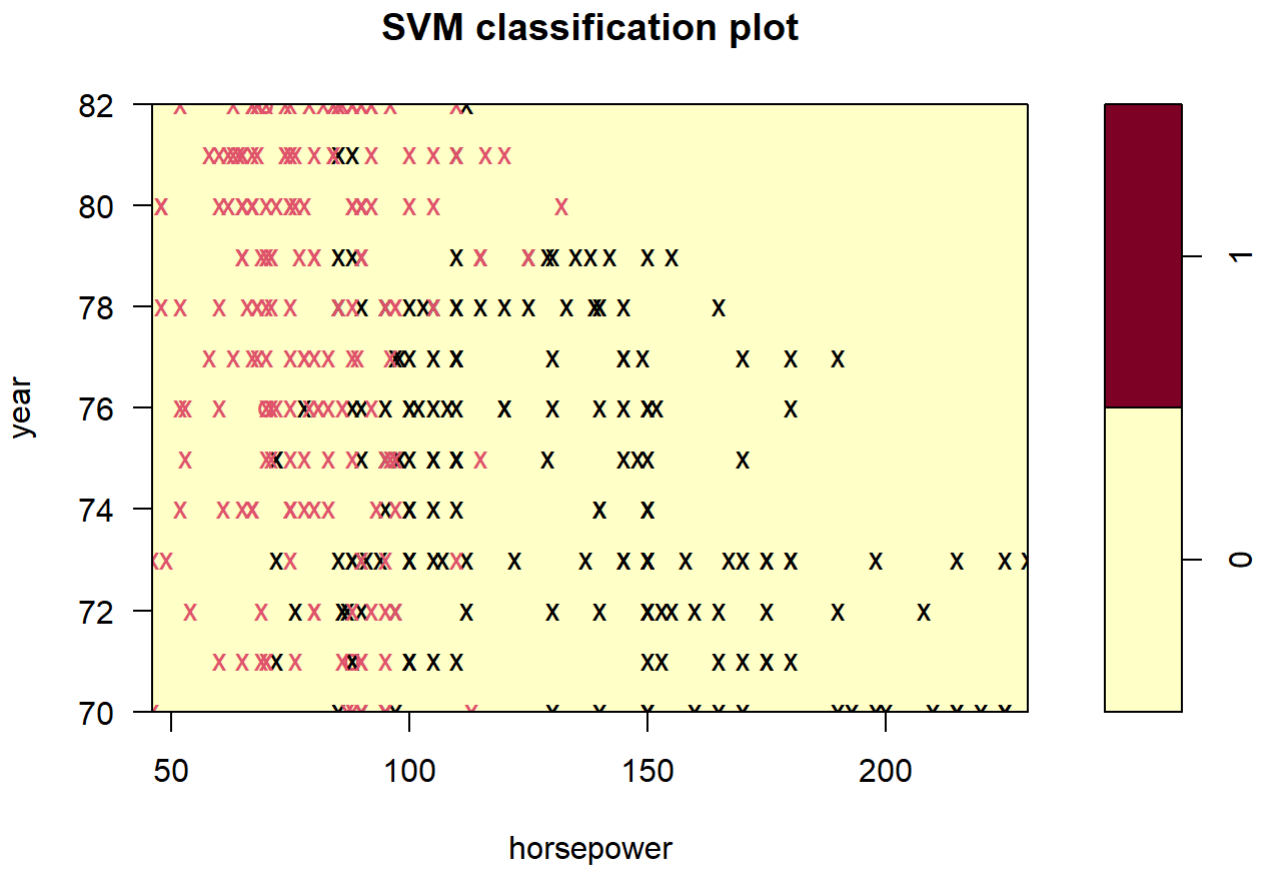

```
svm.radial <- svm(mileage ~., data=data.auto , kernel ="radial", cost=100, gamma = .01, scale=FA
LSE)

plot(svm.radial,data = data.auto, acceleration ~ weight)
```



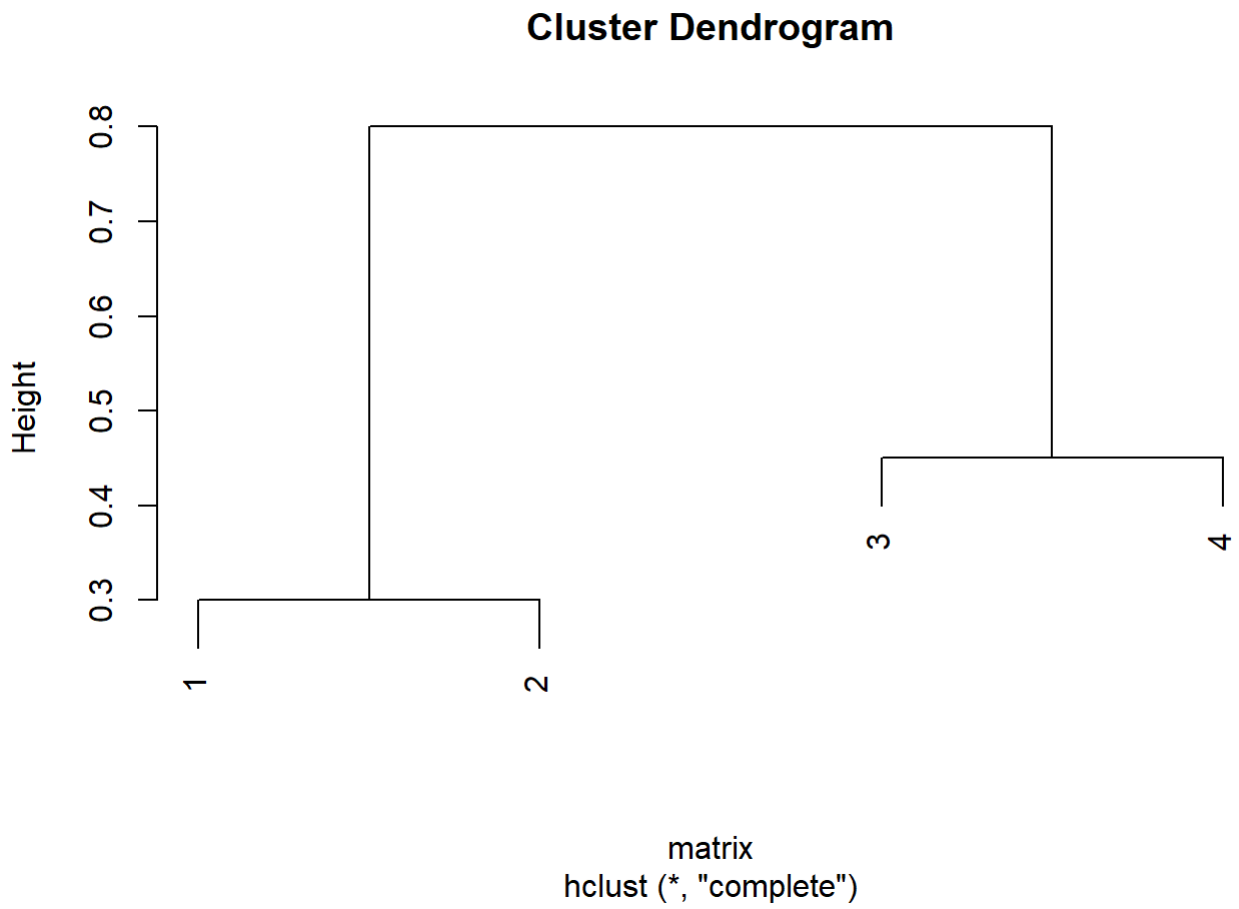

```
plot(svm.radial, data = data.auto, horsepower ~ weight)
```



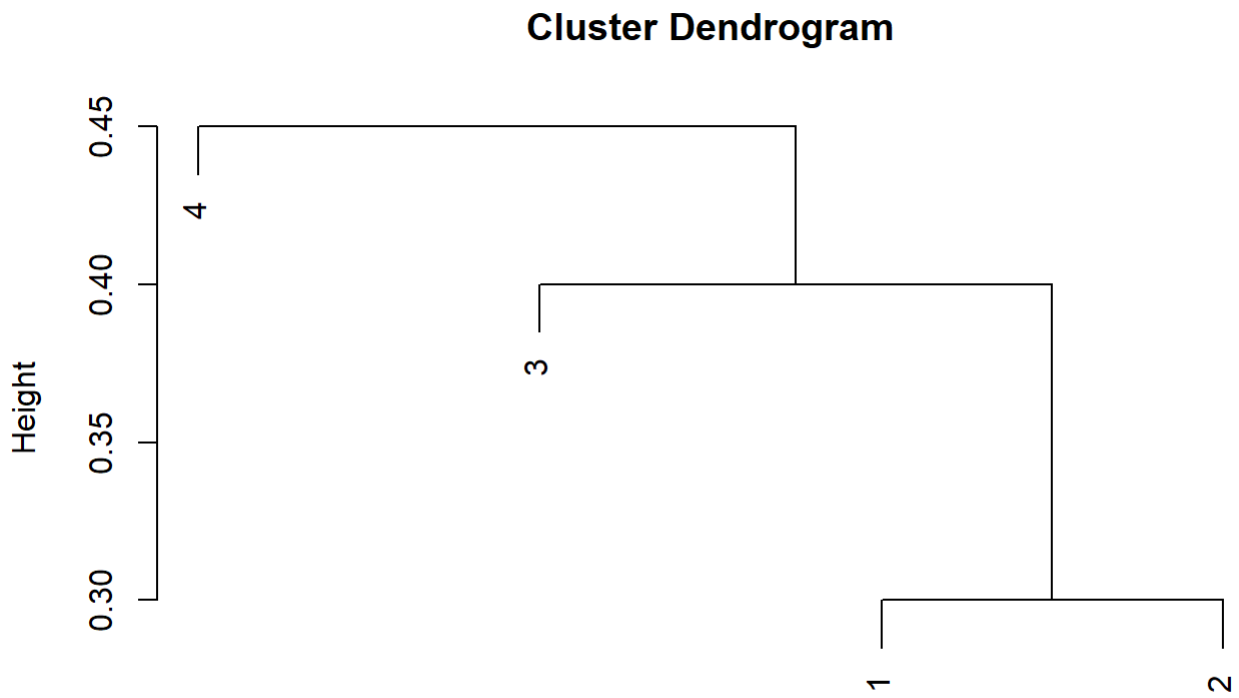

```
plot(svm.radial, data = data.auto, year ~ horsepower)
```



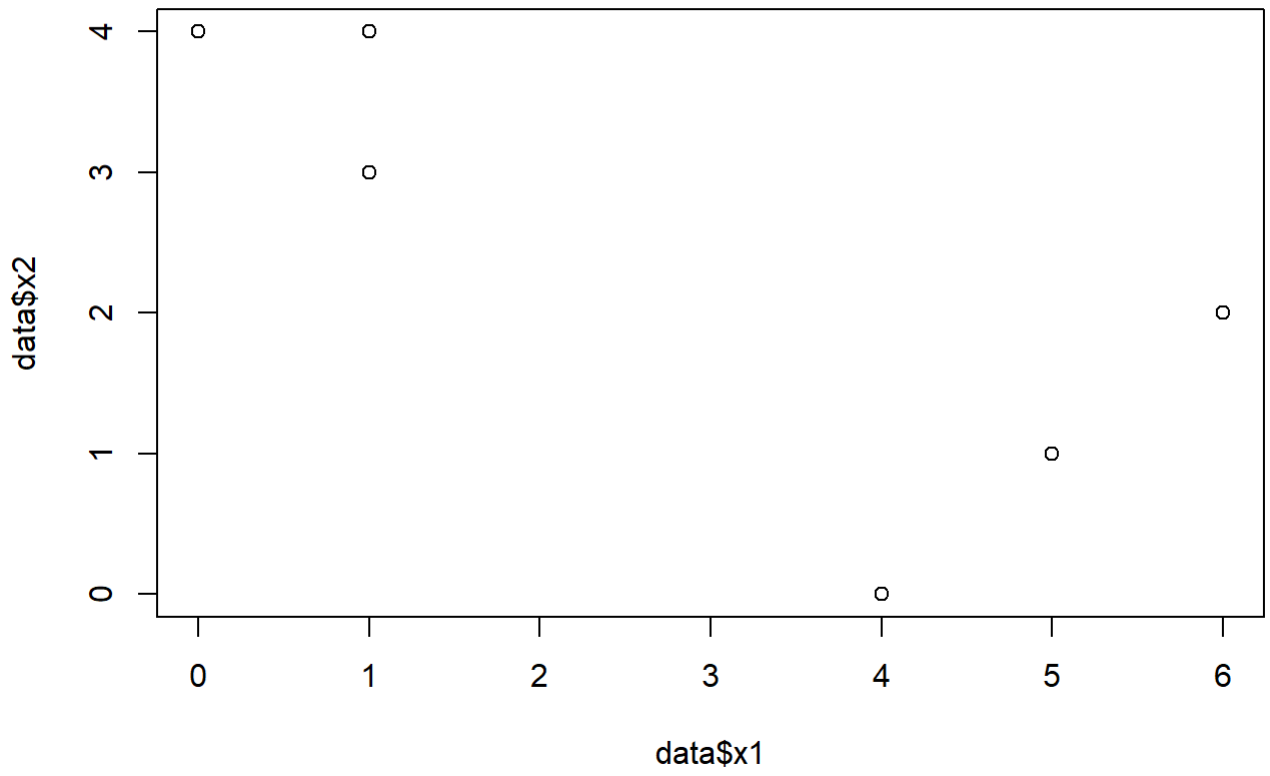

```
rm(list=ls())
options(scipen = 999)

setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 5")



#Chapter 10

#2

matrix = as.dist(matrix(c(0, 0.3, 0.4, 0.7,
                      0.3, 0, 0.5, 0.8,
                      0.4, 0.5, 0.0, 0.45,
                      0.7, 0.8, 0.45, 0.0), nrow=4))

#a

plot(hclust(matrix, method="complete"))
```




```
#b

plot(hclust(matrix, method="single"))
```




```
#c

#cluster a - 1,2   cluster b - 3,4

#d

#cluster a - 1,2,3 cluster b - 4

#3

x1 <- c(1,1,0,5,6,4)
x2 <- c(4,3,4,1,2,0)

data <- as.data.frame(cbind(x1,x2))
View(data)

#a

plot(data$x1,data$x2)
```




```
#b

data$labels = sample(2, 6, replace=T)
data$labels
```

```
## [1] 1 2 1 2 2 1
```

```
#c

cluster.one <- data[which (data$labels==1), ]
cluster.two <- data[which (data$labels!=1), ]

centroid.one <- c(mean(cluster.one$x1), mean(cluster.one$x2))
centroid.one
```

```
## [1] 1.666667 2.666667
```

```
centroid.two = c(mean(cluster.two$x1), mean(cluster.two$x2))
centroid.two
```

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

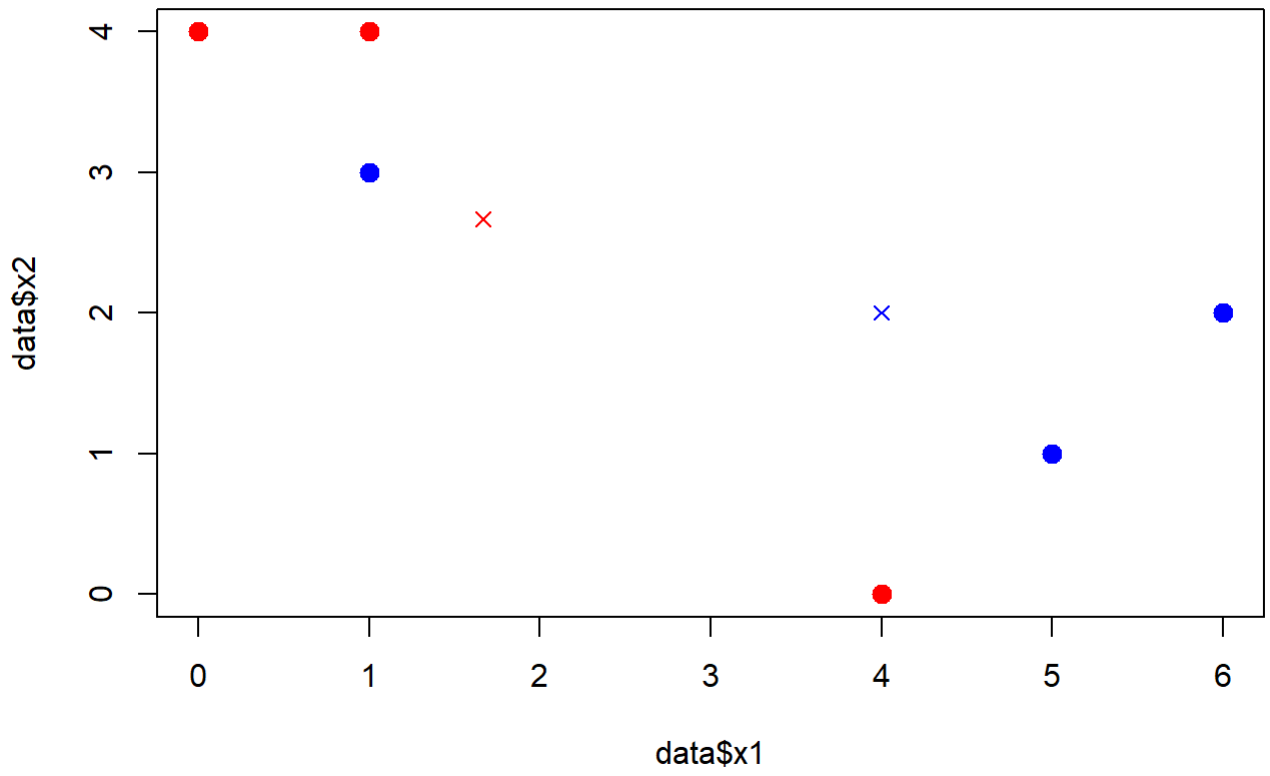
```
#d

distance <- function (x, y){
    return(sqrt((x[1] - y[1])^2 + (x[2] - y[2])^2))
}


distance(data, centroid.one)
```

```
##         x1
## 1 1.490712
## 2 0.745356
## 3 2.134375
## 4 3.726780
## 5 4.384315
## 6 3.543382
```

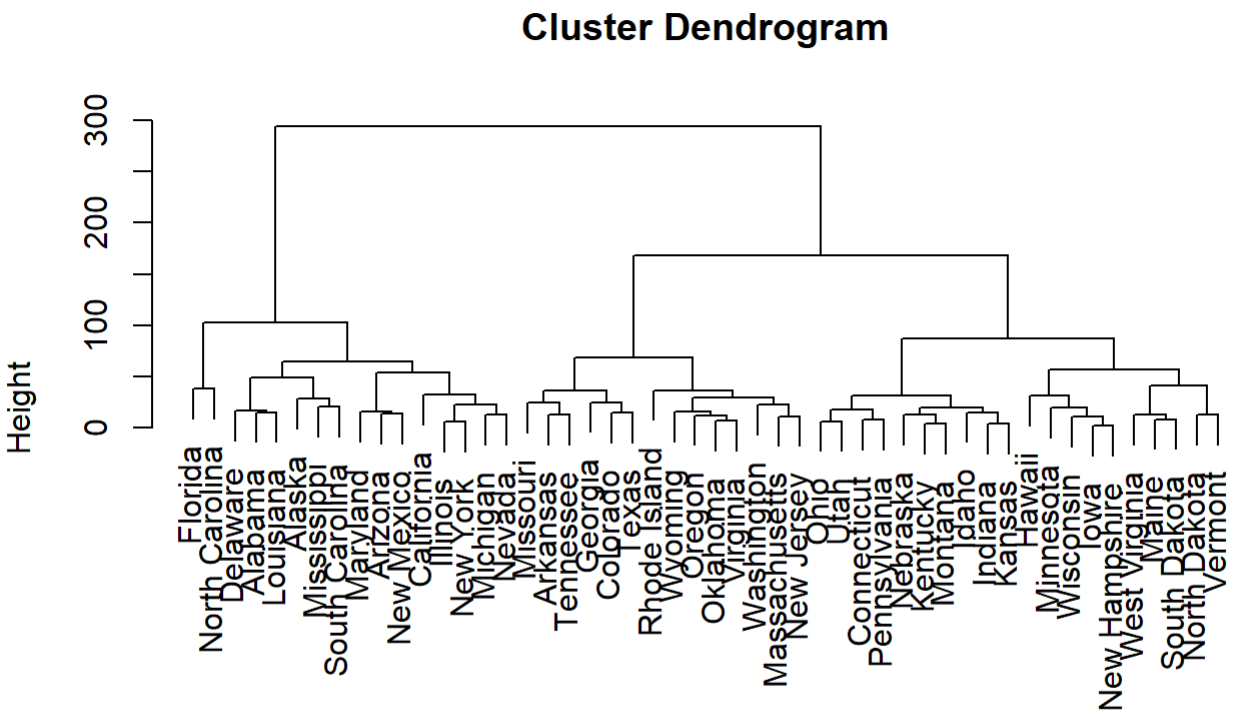
```
distance(data, centroid.two)
```

```
##         x1
## 1 3.605551
## 2 3.162278
## 3 4.472136
## 4 1.414214
## 5 2.000000
## 6 2.000000
```

```
#1 is closer to centroid.one
#2 is closer to centroid.one
#3 is closer to centroid.one
#4 is closer to centroid.two
#5 is closer to centroid.two
#6 is closer to centroid.two



#f

data$color <- ifelse(data$labels == 1,"Red","Blue")

plot(data$x1, data$x2, col = data$color, pch = 20, cex = 2)
points(centroid.one[1], centroid.one[2], col = "Red", pch = 4)
points(centroid.two[1], centroid.two[2], col = "Blue", pch = 4)
```




```
#9

library(ISLR)
set.seed(75)

data.arrests <- USArrests

#a

cluster1 <- hclust(dist(data.arrests), method="complete")
plot(cluster1)
```



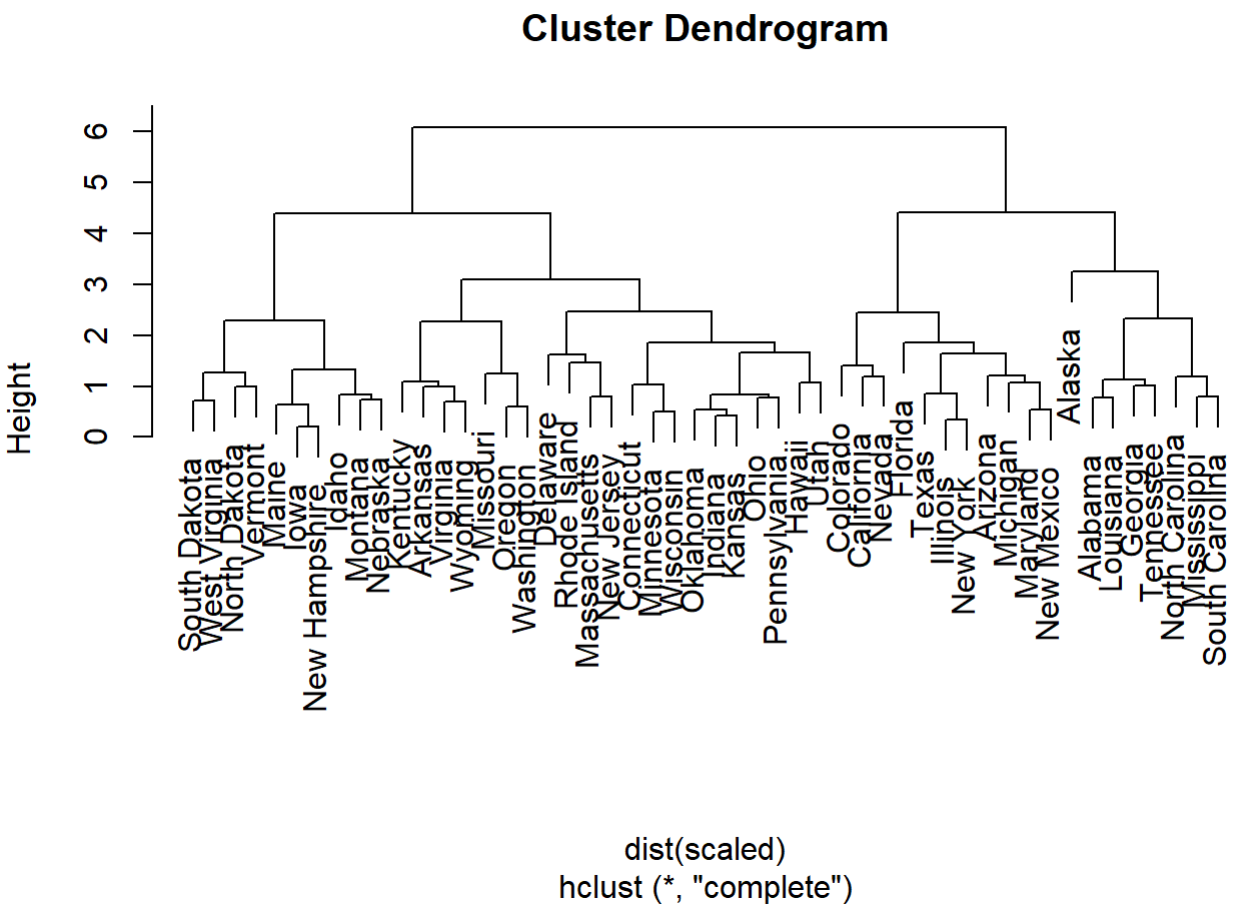

```
#b

cutree(cluster1, 3)
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              2              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              1              1              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              1              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              1              3              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              1              3              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              1              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              3              2              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              2              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              3              3              2
```

```
table(cutree(cluster1, 3))
```

```
##
##  1  2  3
## 16 14 20
```

```
#c

scaled <- scale(data.arrests)
cluster2 <- hclust(dist(scaled), method = "complete")
plot(cluster2)
```




```
cutree(cluster2, 3)
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              2              3              2
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              3              2              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              2              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              2
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              3              2              3              1              3
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              2              3              3
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              2              2              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              3              3              3              3              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              1              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              3              3              3              3              3
```

```
table(cutree(cluster2,3))
```

```
##
##  1  2  3
##  8 11 31
```

```
#d

#It's difficult to interpret much difference from the plot. The table of groupings with the cuts
#into 3 distinct clusters do show a difference, with more in the first cluster without scaling a
nd
#more in the third cluster with scaling. I would scale the data because the units, and standard
#deviations, for the different variables are not comparable. Making them more comparable should
 at least
#improve intertepration and may eliminate some bias.
```