Assignment 3 Solutions

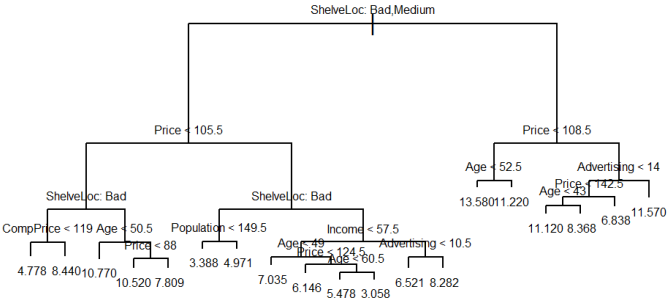
Question 1 (10 marks)

(a) **(3 marks: 0.5 for fitting the model, 0.5 for plot, 1 for errors and 1 for a comment)**

```
Train = read.csv("carseatTraining.csv",
                 header = TRUE)
Test = read.csv("carseatTesting.csv",
                header = TRUE)
tree.Carseats = tree(Sales~.,
                     Train)
summary(tree.Carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Age"         "Population"
## [6] "Income"      "Advertising"
## Number of terminal nodes:  19
## Residual mean deviance:  2.373 = 586.2 / 247
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.36500 -1.07100  0.05704  0.00000  1.03500  3.78900
```

```
plot(tree.Carseats)
text(tree.Carseats,
    pretty=0,
    cex = 0.5)
```




```
yhat.train = predict(tree.Carseats,
                     newdata=Train)
TrainError = mean((yhat.train-Train$Sales)^2)
TrainError
```

```
## [1] 2.203892
```

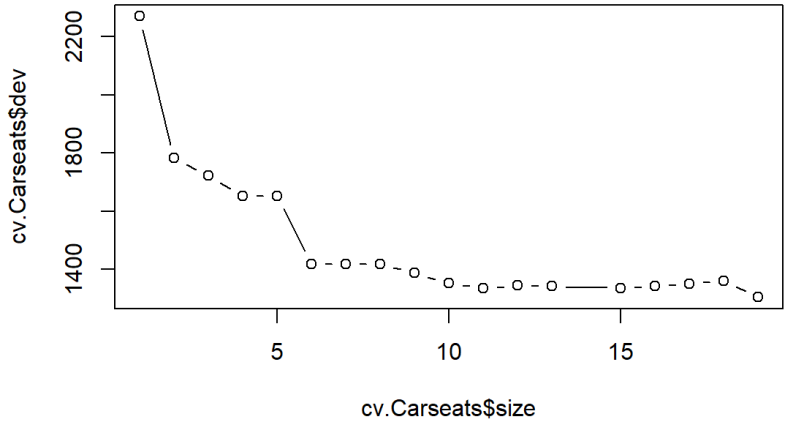
```
yhat.test = predict(tree.Carseats,
                    newdata=Test)
TestError = mean((yhat.test-Test$Sales)^2)
TestError
```

```
## [1] 4.844453
```

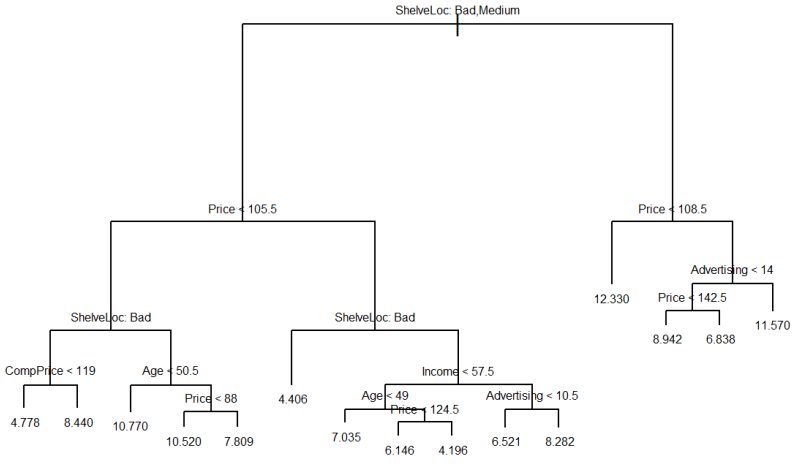
Possible Comments: The tree has 23 terminal nodes; Price, Age, Advertising and CompPrice are the most important predictors; splits near the top of tree reduce the RSS more; the training error is much less than the testing error, maybe we are overfitting; etc. **If they make a reasonable effort to comment about the tree give them the mark, but not if incorrect statements are made.**

(b) **(1 marks: 0.5 for pruning and 0.5 comment (the plots are not required))**

```
set.seed(15)
cv.Carseats = cv.tree(tree.Carseats)
plot(cv.Carseats$size,
    cv.Carseats$dev,
    type='b')
```




```
prune.Carseats = prune.tree(tree.Carseats,
                    best=15)
plot(prune.Carseats)
text(prune.Carseats,
    pretty=0,
    cex=0.5)
```



```
yhat.test = predict(prune.Carseats,
                    newdata=Test)
TestError=mean((yhat.test-Test$Sales)^2)
TestError
```

```
## [1] 4.561307
```

Pruning has been effective here because it decreased the test error. **Give the student the mark if their pruning logic and conclusion is correct.**

(c) **(2.5 marks: 1 for fitting the models, 0.5 for errors and 1 for a comment)**

```
set.seed(15)
bag.Carseats=randomForest(Sales~.,
                          data=Train,
                          mtry=9,
                          ntree=1000)
yhat.train=predict(bag.Carseats,
                   newdata=Train)
TrainError=mean((yhat.train-Train$Sales)^2)
TrainError
```

```
## [1] 0.4733006
```

```
yhat.test=predict(bag.Carseats,
                  newdata=Test)
TestError=mean((yhat.test-Test$Sales)^2)
TestError
```

```
## [1] 2.49507
```

```
rf.Carseats=randomForest(Sales~.,
                         data=Train,
                         ntree=1000)
yhat.train=predict(rf.Carseats,
                   newdata=Train)
TrainError=mean((yhat.train-Train$Sales)^2)
TrainError
```

```
## [1] 0.6208255
```

```
yhat.test=predict(rf.Carseats,
                  newdata=Test)
TestError=mean((yhat.test-Test$Sales)^2)
TestError
```

```
## [1] 2.690345
```

Bagging obtained slightly better results here so decorrelating the trees (using default settings) was not an effective strategy. **Give the student the mark if their conclusion is correct.**

(d) **(2.5 marks: 0.5 for fitting the model, 1 for experimenting, 0.5 for errors and 0.5 for choosing the best model)**

```
set.seed(15)
boost.Carseats=gbm(Sales~.,
                   data=Train,
                   distribution="gaussian",
                   n.trees=1000,
                   interaction.depth=1,
                   shrinkage=0.01)
summary(boost.Carseats,plotit=FALSE)
```

	var <fctr>	rel.inf <dbl>
Price	Price	34.95254306
ShelveLoc	ShelveLoc	31.67687873
CompPrice	CompPrice	10.48632141
Age	Age	10.00134960
Advertising	Advertising	8.44236318
Income	Income	4.04027384
Population	Population	0.23116423
Education	Education	0.13311211
US	US	0.03599384
Urban	Urban	0.00000000

1-10 of 10 rows

```
yhat.train=predict(boost.Carseats,
                   newdata=Train,
                   n.trees=1000)
TrainError=mean((yhat.train-Train$Sales)^2)
TrainError
```

```
## [1] 1.560761
```

```
yhat.test=predict(boost.Carseats,
                  newdata=Test,
                  n.trees=1000)
TestError=mean((yhat.test-Test$Sales)^2)
TestError
```

```
## [1] 1.955621
```

The boosted model should be the best for this data and they must show some evidence of trying different parameter values. I have only shown the stump here.

(e) **(1 mark: 0.5 for best model and 0.5 for important predictors)**

The best model is the boosted regression tree because it has the lowest testing error. The most important predictors were Price, CompPrice, Advertising and Age. **Give the mark if the student identifies the top 2, 4 or make reasonable comments.**

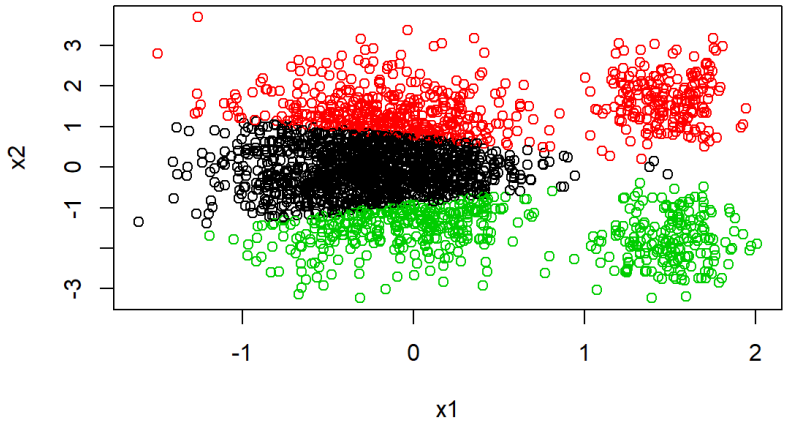
Question 2 **(4 marks: 2 marks for correct labelling (1.5 for mostly correct etc.), 1 for confidence and 1 for lift (deduct 0.5 if no comments or incorrect comments are made).**

(Solutions on last page)

Question 3 (6 marks)

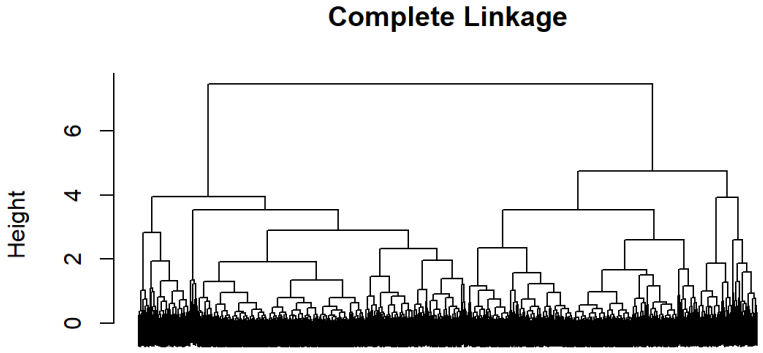
(a) (1 mark: 0.5 for k-means (must use nstart > 1) and 0.5 for plot)

```
data <- read.csv("A3data2.csv",
                 header = TRUE)

kc = kmeans(data[,1:2],
            3,
            iter.max = 100,
            nstart=500)
plot(data[,1:2],
     col=kc$cluster)
```

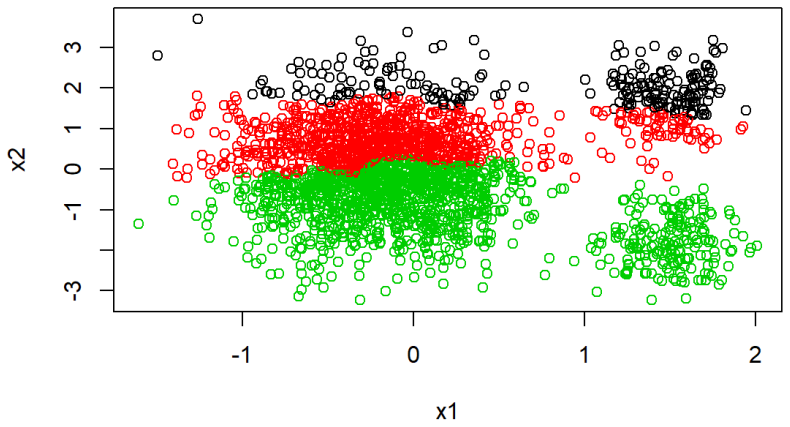


(b) (1.5 marks: 0.5 for complete linkage, 0.5 single linkage and 0.5 for dendrograms)

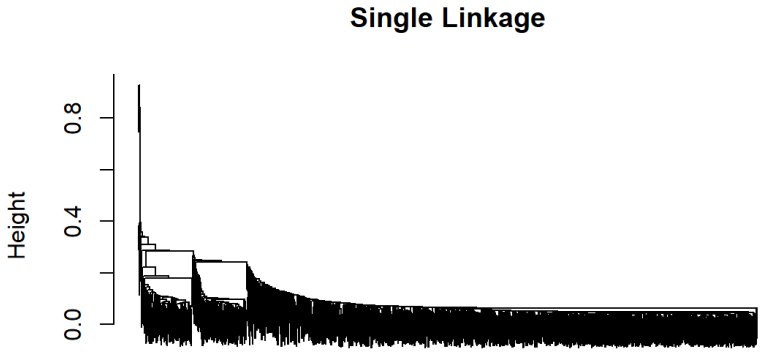
```
hc.complete=hclust(dist(data[,1:2]),
                   method="complete")
plot(hc.complete,
    main="Complete Linkage",
    xlab = "",
    sub="",
    labels=FALSE)
```



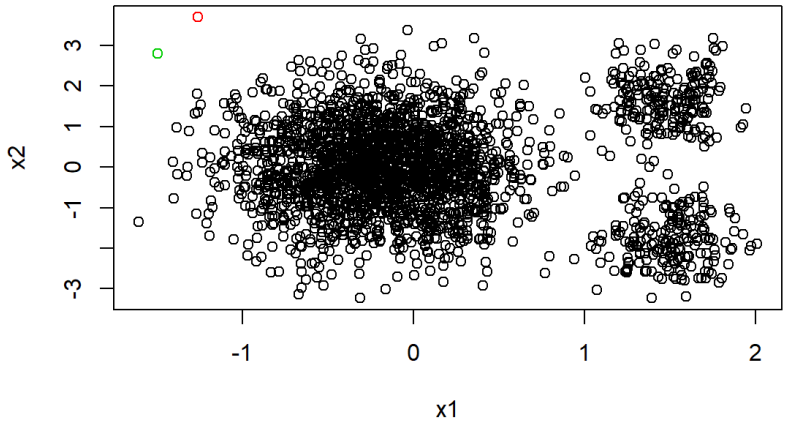

```
plot(data[,1:2],
    col=cutree(hc.complete, 3))
```




```
hc.single=hclust(dist(data[,1:2]),
                 method="single")

plot(hc.single,
     main="Single Linkage",
     xlab = "",
     sub="",
     labels=FALSE)
```




```
plot(data[,1:2],
     col=cutree(hc.single, 3))
```

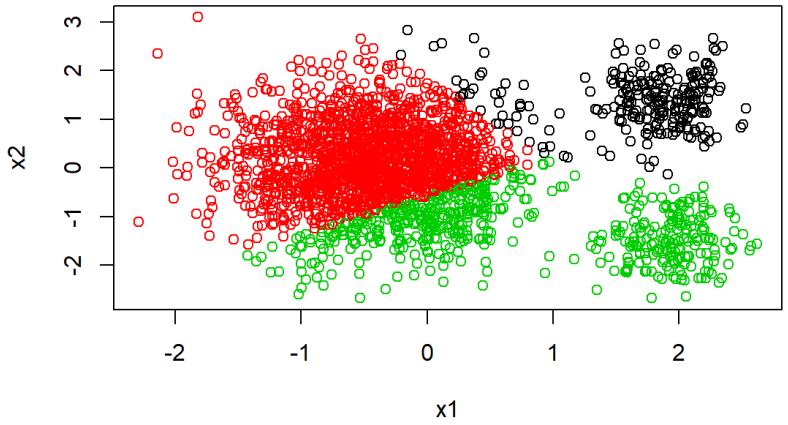


(c) (1 mark: for comments)

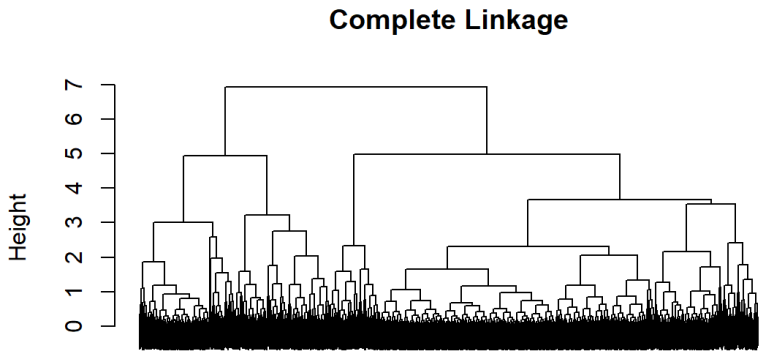
The variables x_1 and x_2 are on different scales. Hence, the separation between the 'natural clusters' in the x_1 dimension is smaller than it appears in the plot. The greatest direction of spread is in the x_2 dimension. Hence, distance-based clustering algorithms will tend merge points with respect to x_1 rather x_2 which gives the banded structure. Single linkage was also affected by two 'outliers'. None of the algorithms have performed well on this data. **The point of this question was to illustrate that applying clustering algorithms to unscaled data can give poor results on simple datasets. Give the mark if a reasonable effort was made to comment on the clusterings and the conclusion was nothing worked well.**

(d) (2.5 marks: 0.5 for complete linkage, 0.5 single linkage, 0.5 for dendrograms and 1 for comments)

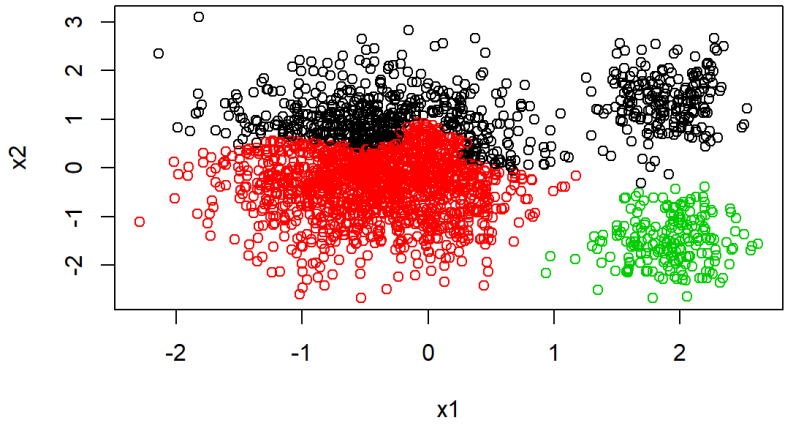
```
data <- read.csv("A3data2.csv",
                 header = TRUE)
data[,c(1,2)] <- scale(data[,c(1,2)],
                       center = TRUE,
                       scale = TRUE)

kc = kmeans(data[,1:2],
            3,
            iter.max = 100,
            nstart=500)
plot(data[,1:2],
     col=kc$cluster)
```



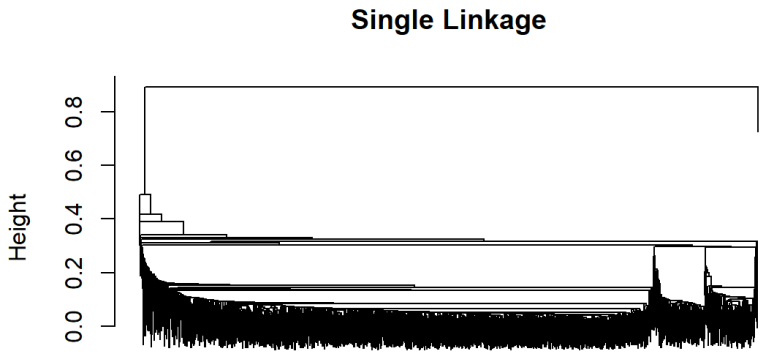
```
hc.complete=hclust(dist(data[,1:2]),
                   method="complete")
plot(hc.complete,
    main="Complete Linkage",
    xlab = "",
    sub="",
    labels=FALSE)
```



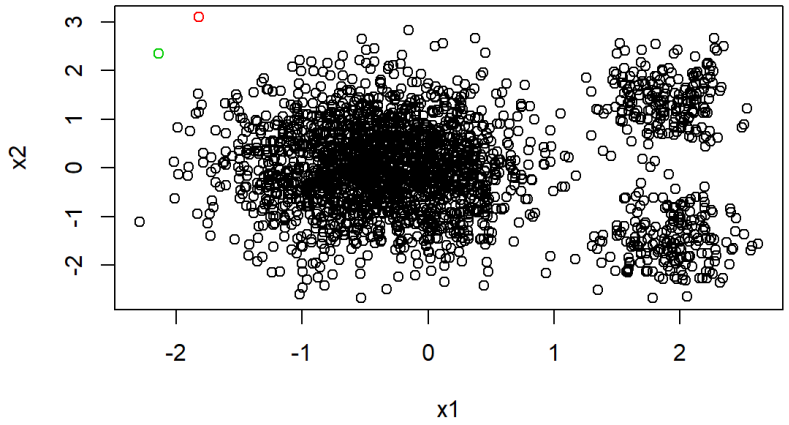

```
plot(data[,1:2],
    col=cutree(hc.complete, 3))
```




```
hc.single=hclust(dist(data[,1:2]),
                 method="single")

plot(hc.single,
    main="Single Linkage",
    xlab = "",
    sub="",
    labels=FALSE)
```




```
plot(data[,1:2],
    col=cutree(hc.single, 3))
```

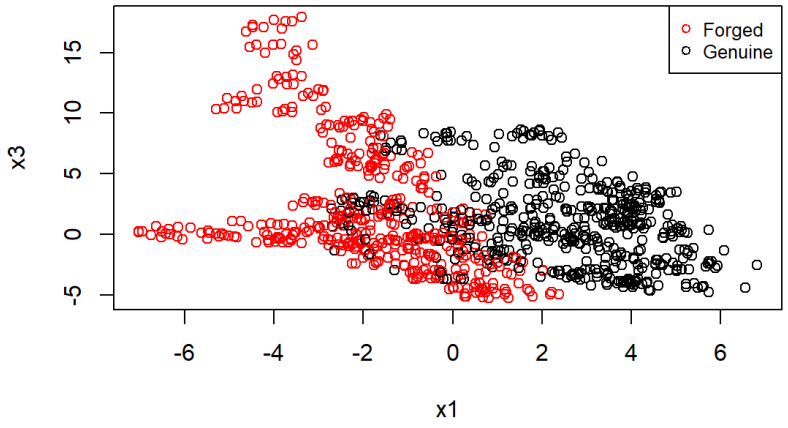


Overall, scaling has not helped that much. The clusters are globular in shape, but the densities and sizes are different. k-means was not able to find the clusters because splitting the large cluster reduces the objective more than resolving the 3 clusters. Single linkage was influenced by two outliers, giving a poor 3 cluster clustering. The three clusters were visible in the dendrogram for single linkage lower down. Complete linkage also struggled with this data, splitting the large cluster into three groups. None of the algorithms have performed well on this data. **If they look at the dendrograms and remove outliers before defining the clusters, give the marks but it is not expected. The point of this question was to illustrate that applying clustering algorithms without careful thought can give poor results on simple datasets. Give the mark if a reasonable effort was made to comment on the clusterings and the conclusion was nothing worked well.**

Question 4 (4 marks)

(a) (0.5 marks for comment and reason)

```
Train = read.csv("BankTrain.csv",
                 header = TRUE)
Train$y=as.factor(Train$y)
Test = read.csv("BankTest.csv",
                header = TRUE)
Test$y=as.factor(Test$y)
plot(Train$x1,
     Train$x3,
     col = Train$y,
     xlab="x1",
     ylab="x3")
legend("topright",
       legend = c("Forged", "Genuine"),
       col = c("red","black"),
       pch=21,
       cex = 0.8,
       text.col = "black",
       horiz = FALSE)
```



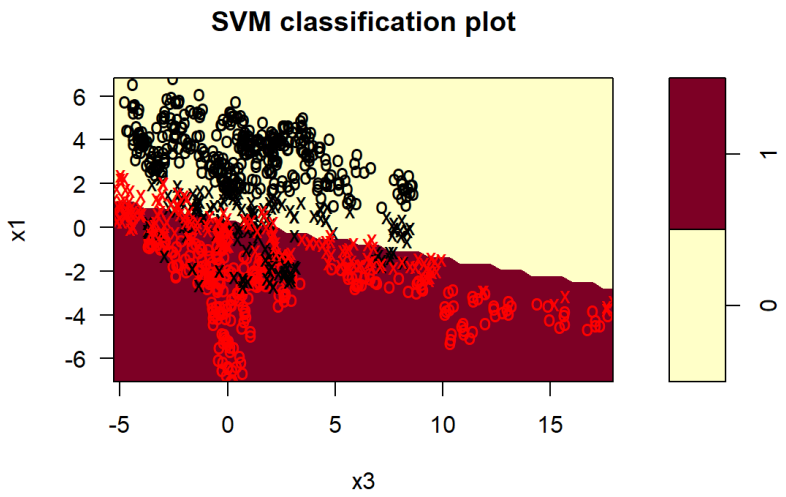
It is clear from the plot that a separating hyperplane does not exist.

(b) (1.5 marks: 0.5 for fitting, 0.5 for plot and 0.5 for comment)

```
library(e1071)
set.seed(15)
tune.out = tune(svm,
                y~x1+x3,
                data=Train,
                kernel="linear",
ranges=list(cost=c(0.01,0.1,1,10,100,1000)))

bestmodel = tune.out$best.model
ypred = predict(bestmodel,
                Test)
table(predict=ypred,
      truth=Test$y)
```

```
##         truth
## predict   0   1
##       0 197  11
##       1  39 165
```

```
plot(bestmodel,
     Train[,c(1,3,5)])
```



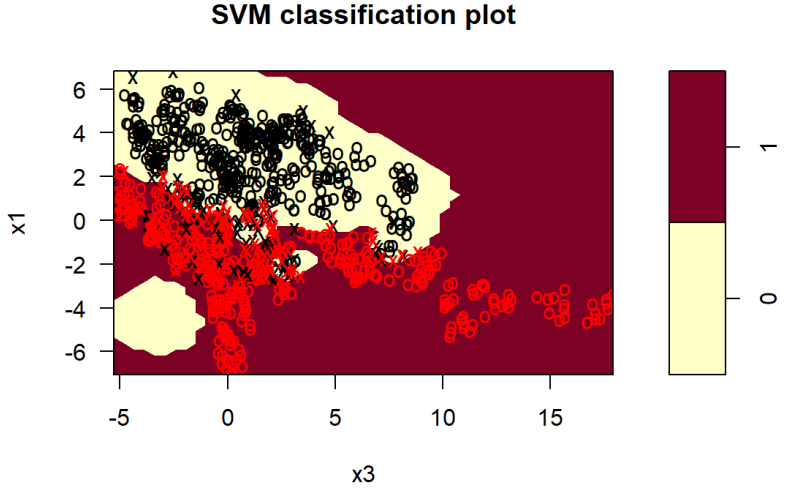
Possible Comments: The testing error was 0.12, sensitivity (0.94), specificity (0.83), number of support vectors (bias/variance) etc. **If they have made a reasonable effort to make useful comments give them the mark, but not if incorrect statements are made.**

(c) (2 marks: 1 for fit, 0.5 for plot and 0.5 for comment)

```
set.seed(15)
tune.out = tune(svm,
                y~x1+x3,
                data=Train,
                kernel="radial",
                ranges=list(cost=c(0.01,0.1,1,10,100,1000),
                gamma=c(0.5,1,2,3,4)))

bestmodel = tune.out$best.model
ypred = predict(bestmodel,
                Test)
table(predict=ypred,
    truth=Test$y)
```

```
##        truth
## predict   0   1
##       0 212  13
##       1  24 163
```

```
plot(bestmodel,
    Train[,c(1,3,5)])
```



Possible Comments: The testing error was 0.08 which is less the SVC model, sensitivity (0.94), specificity (0.90), number of support vectors (bias/variance) etc. The more complex model (SVM) was able to better capture the non-linear features in the data. **If they have made a reasonable effort to make useful comments give them the mark, but not if incorrect statements are made.**