# HW10

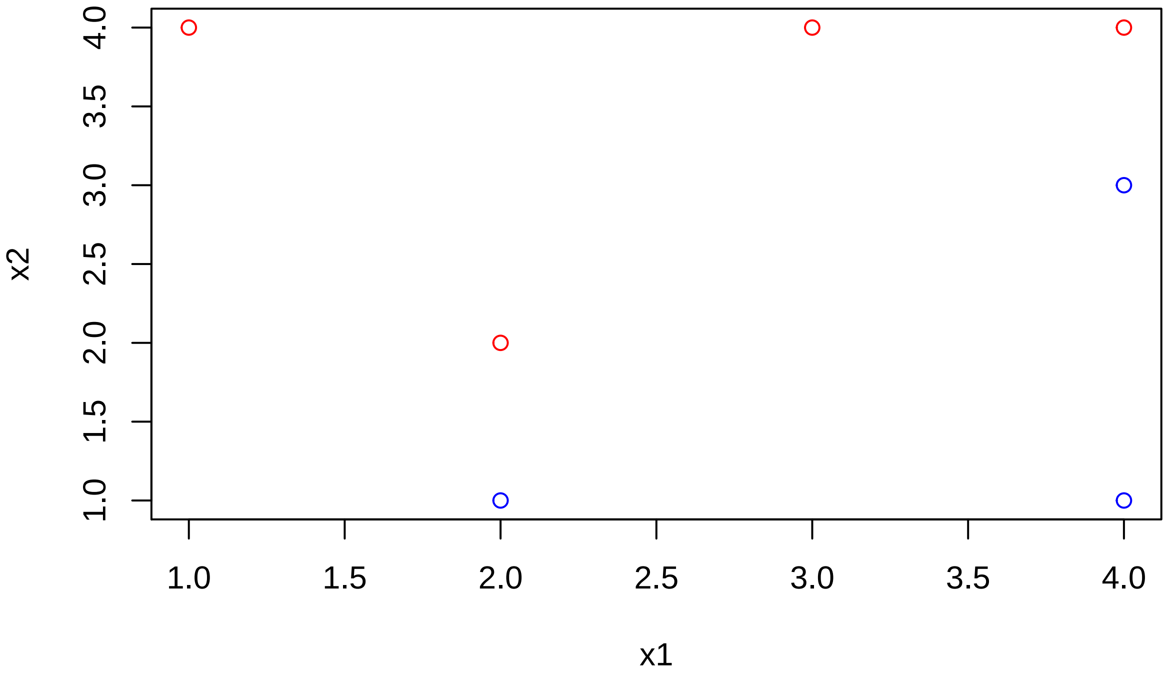
*Yigao Li*

*April 29, 2018*

## 9.7 - 3

### (a)

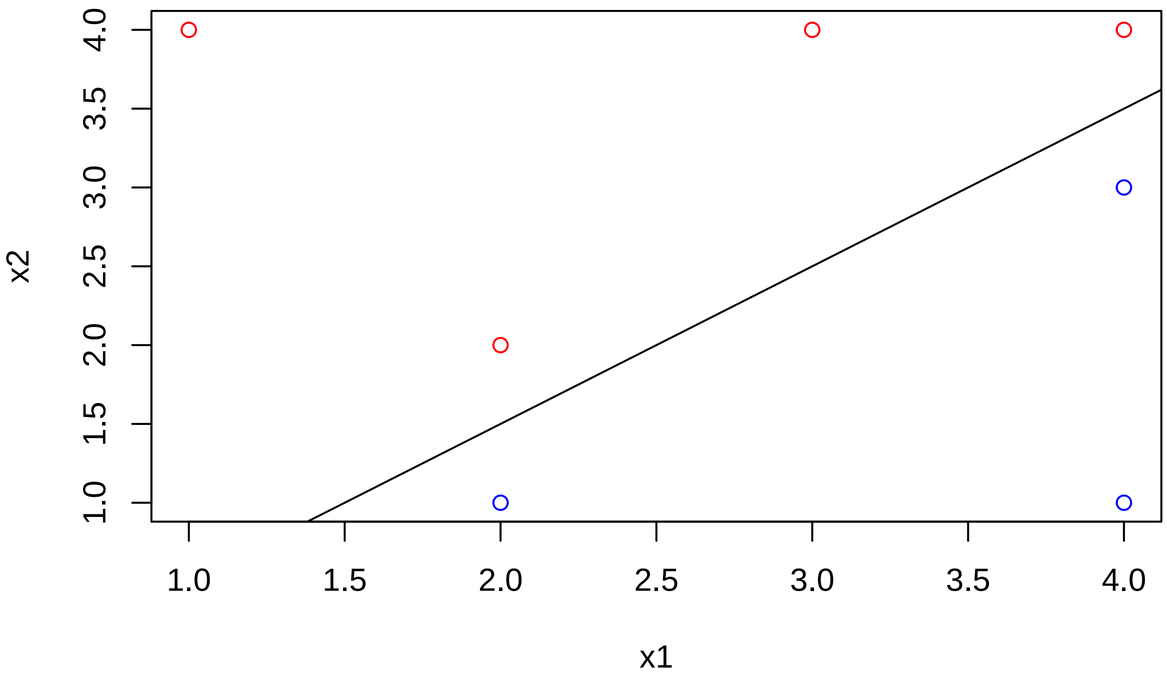
```
x1 <- c(3,2,4,1,2,4,4)
x2 <- c(4,2,4,4,1,3,1)
c <- c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1, x2, col = c)
```



### (b)

The optimal separating hyperplane must pass point $(2, 1.5)$ and $(4, 3.5)$. The equation for this hyperplane is $-0.5 + x_1 - x_2 = 0$.

```
plot(x1, x2, col = c)
abline(-0.5, 1)
```

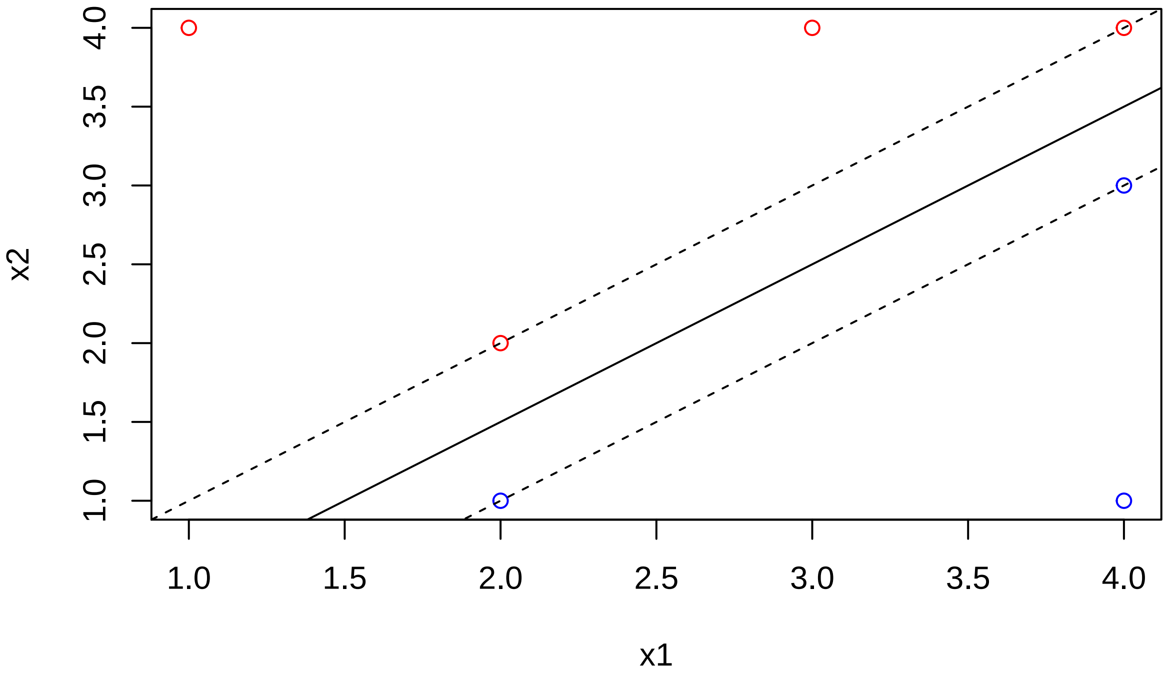


### (c)

Classify to *Red* if $-0.5 + x_1 - x_2 < 0$, and classify to *Blue* otherwise.

### (d)

```r
plot(x1, x2, col = c)
abline(-0.5, 1)
abline(0, 1, lty = 2)
abline(-1, 1, lty = 2)
```



### (e)

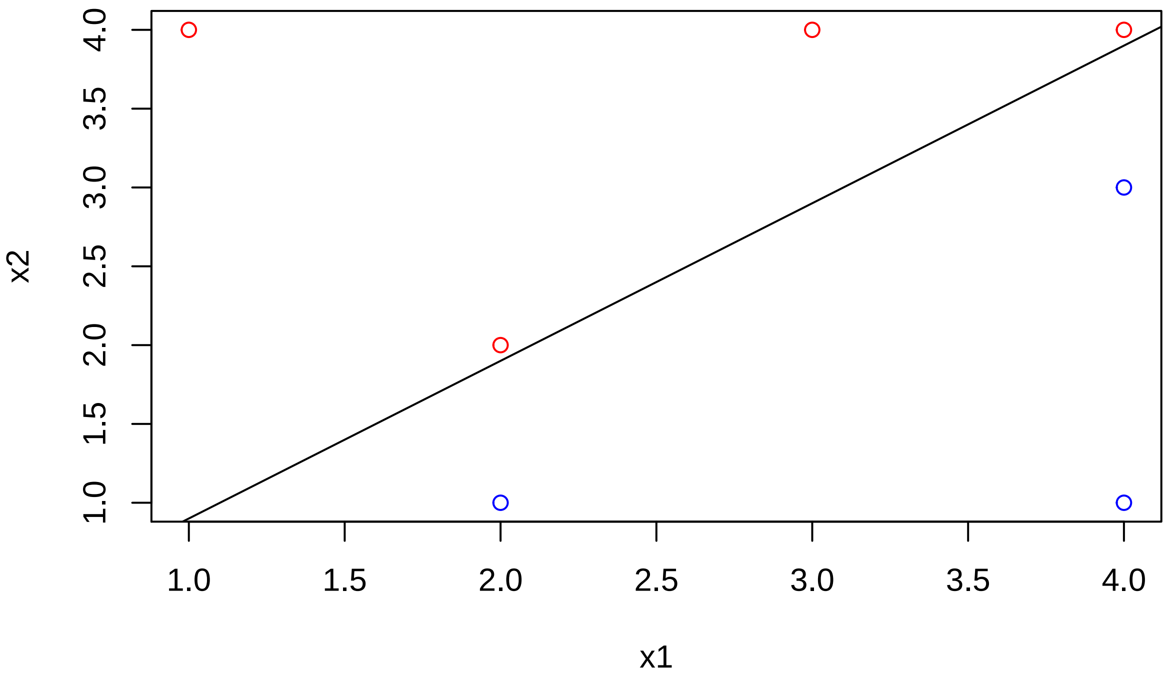
Support vectors are $(2, 2)$, $(4, 4)$, $(2, 1)$ and $(4, 3)$.

### (f)

7th observation point is $(4, 1)$, which is away from blue margin. So, slight movement of this point does not change maximal margin hyperplane.

### (g)

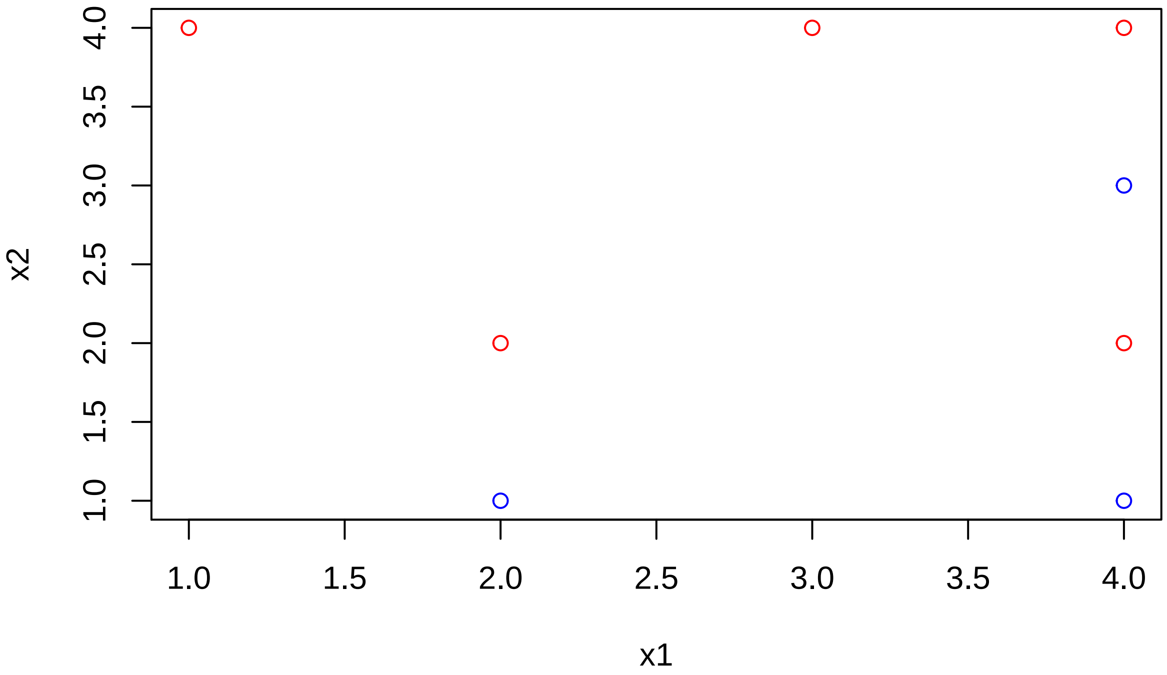
```
plot(x1, x2, col = c)
abline(-0.1, 1)
```



The equation for this hyperplane is $-0.1 + x_1 - x_2 = 0$.

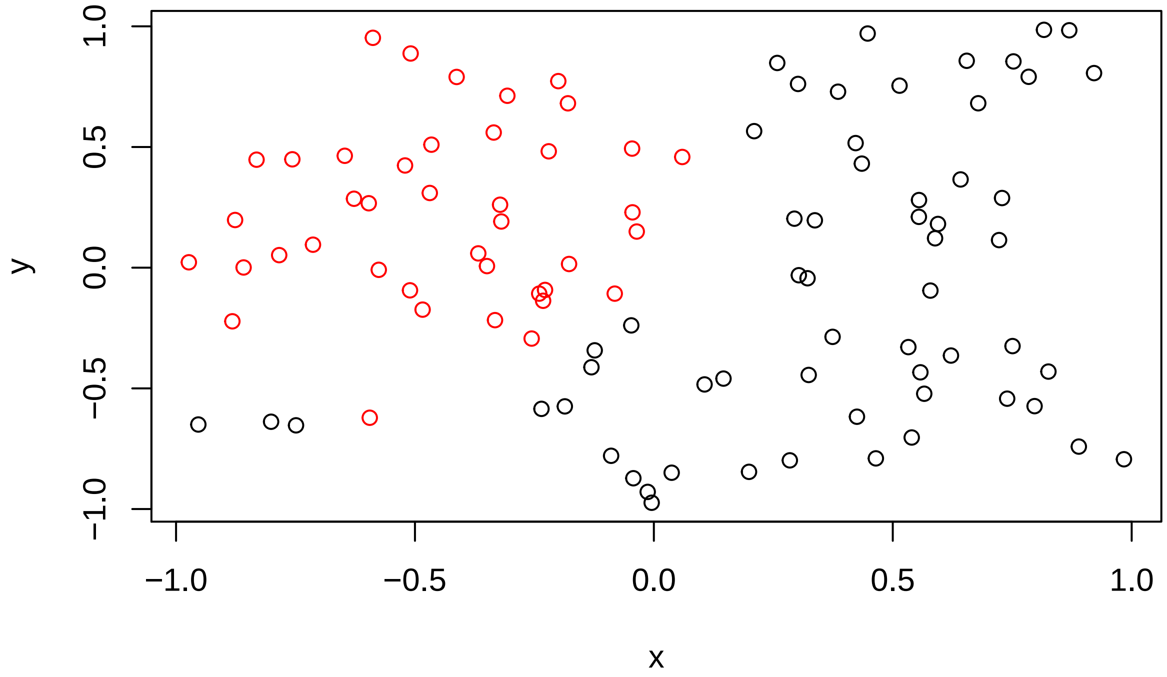
### (h)

```r
plot(x1, x2, col = c)
points(4,2,col = "red")
```



## 9.7 - 4

```
set.seed(1)
x <- runif(100, min = -1, max = 1)
y <- runif(100, min = -1, max = 1)
z <- as.numeric(y > 3*x^2+3*x)
plot(x, y, col = z+1)
```



```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.4.4
```

```r
library(scatterplot3d)
```

```
## Warning: package 'scatterplot3d' was built under R version 3.4.4
```

```r
set.seed(5322)
mydf <- data.frame(x,y,z)
train <- sample(100, 59)
test <- -train
traindf <- mydf[train,]
testdf <- mydf[test,]
svm.1 <- svm(factor(z) ~ ., data = traindf, scale = FALSE, kernel = "polynomial", d = 1, cost = 2)
trainpred <- predict(svm.1, traindf)
traintable <- table(traindf$z, trainpred)
traintable
```

```
##    trainpred
##      0  1
##   0 33  2
##   1  0 24
```

```r
(traintable[2]+traintable[3])/59
```

```
## [1] 0.03389831
```

```
testpred <- predict(svm.1, testdf)
testtable <- table(testdf$z, testpred)
testtable
```

```
##    testpred
##      0  1
##   0 23  1
##   1  1 16
```

```
(testtable[2]+testtable[3])/41
```

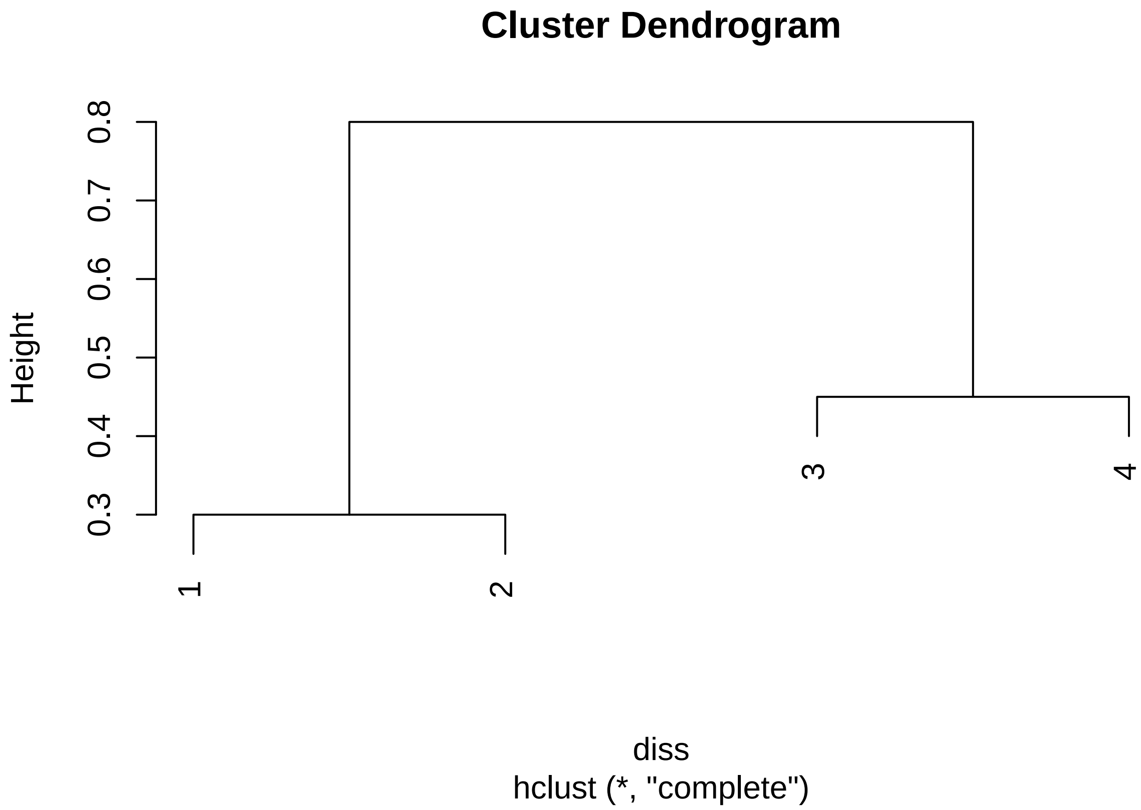
```
## [1] 0.04878049
```

The best SVM option is actually the linear separation with cost 2. Evne though we separate the points with quadratic function, There are only 3 points at bottom left can be error points after classification. Training error rate is 3.39% and test error rate is 4.88%.

# 10.7 - 2

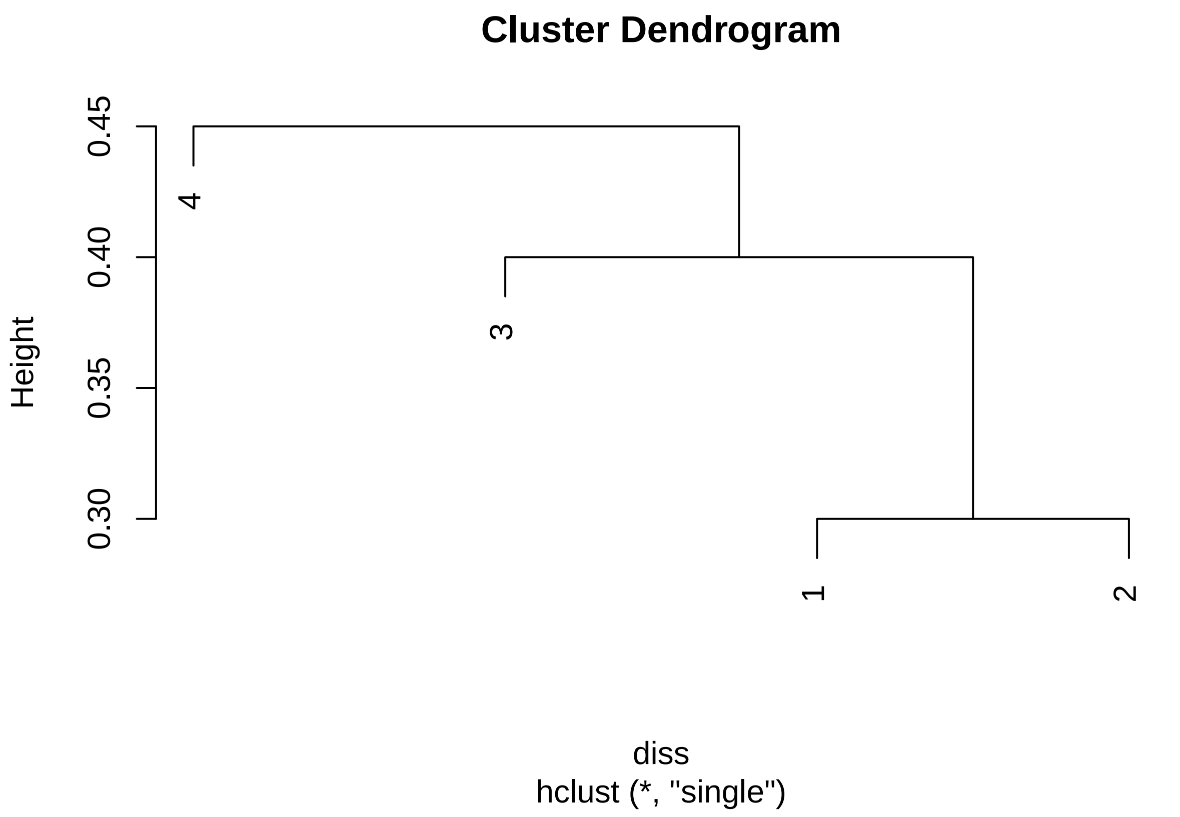
## (a)

```
diss = as.dist(matrix(c(0, 0.3, 0.4, 0.7,
                        0.3, 0, 0.5, 0.8,
                        0.4, 0.5, 0, 0.45,
                        0.7, 0.8, 0.45, 0), nrow=4))
plot(hclust(diss, method="complete"))
```



(b)

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



### (c)

Observation 1 and 2 are in a cluster and observation 3 and 4 are in the other cluster.

### (d)

Observation 1, 2 and 3 are in a cluster and observation 4 is in the other cluster.

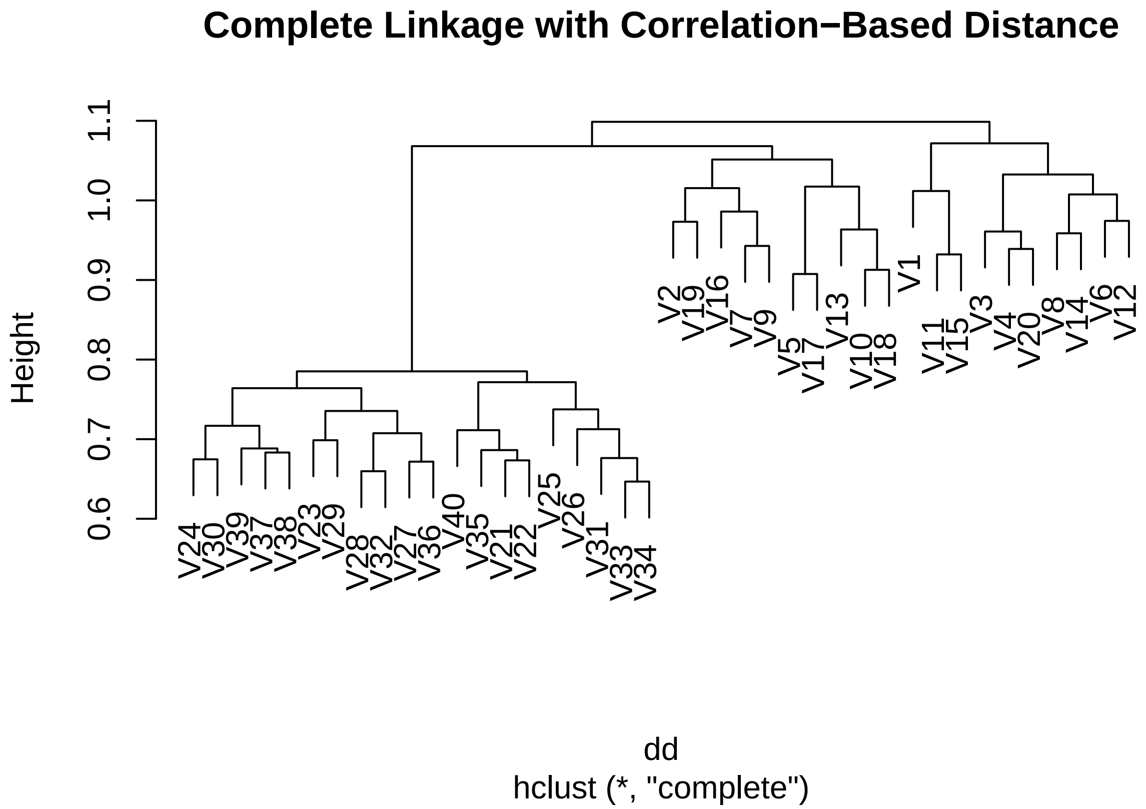
## 10.7 - 11

### (a)

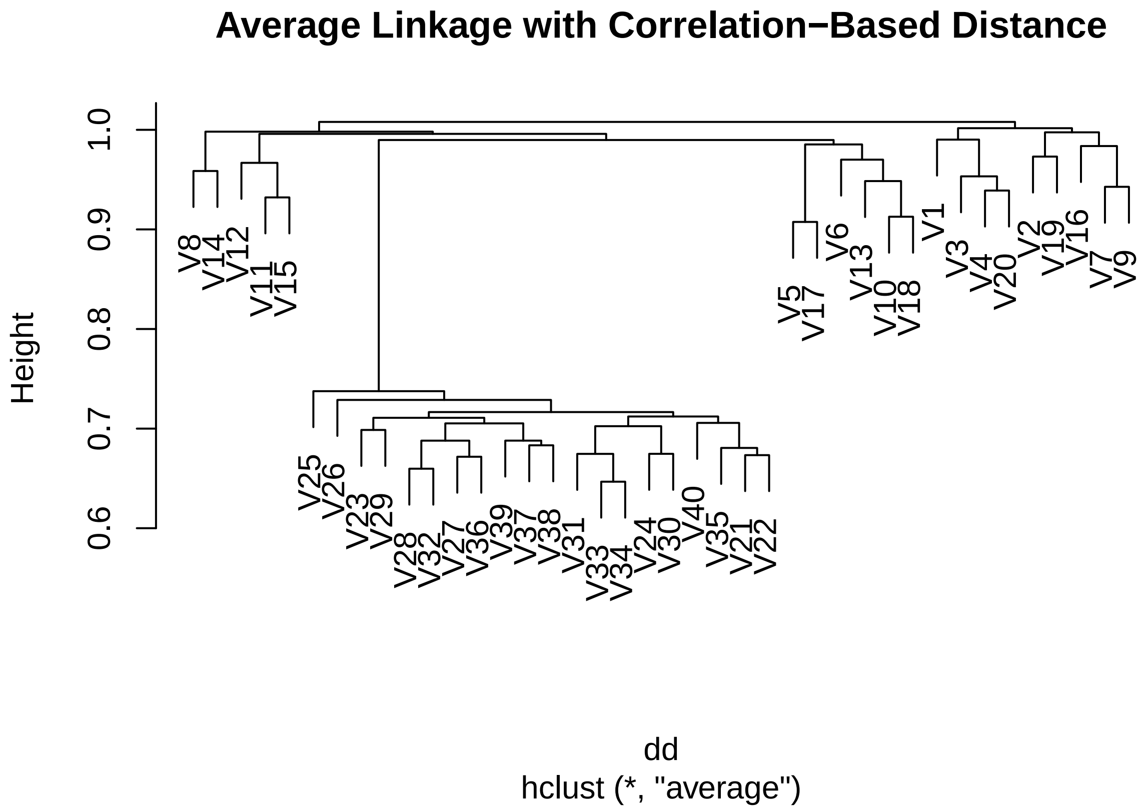
```
ch10ex11 <- read.csv("http://www-bcf.usc.edu/~gareth/ISL/Ch10Ex11.csv", header = FALSE)
```

### (b)

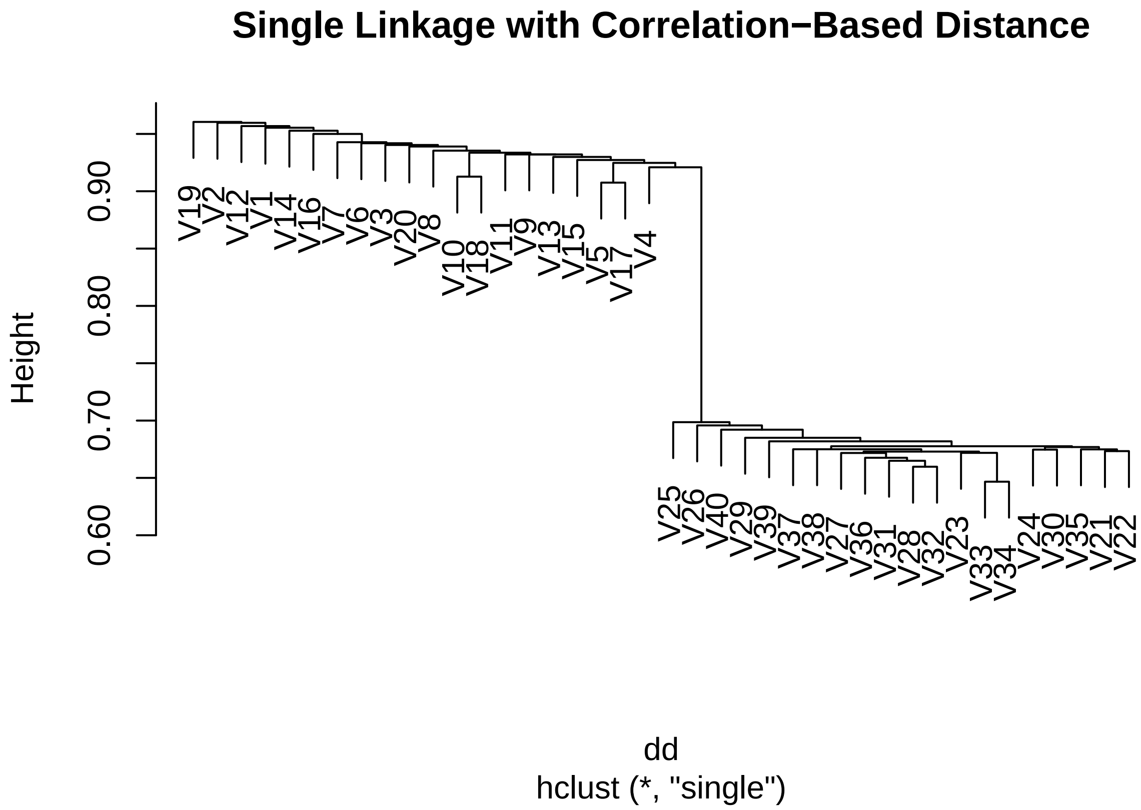
```
dd <- as.dist(1-cor(ch10ex11))
hc.complete <- hclust(dd, method = "complete")
hc.average <- hclust(dd, method = "average")
hc.single <- hclust(dd, method = "single")
plot(hc.complete, main = "Complete Linkage with Correlation-Based Distance")
```



```
plot(hc.average, main = "Average Linkage with Correlation-Based Distance")
```



```
plot(hc.single, main = "Single Linkage with Correlation-Based Distance")
```



Yes, complete and average linkage can separate the samples into 2 groups with a gene.

### (c)

```
index <- seq(1,20)
ch10ex11.1 <- ch10ex11[,index]
ch10ex11.2 <- ch10ex11[,-index]
mean.1 <- apply(ch10ex11.1, 1, mean)
mean.2 <- apply(ch10ex11.2, 1, mean)
abs.mean <- abs(mean.1 - mean.2)
which.max(abs.mean)
```

```
## [1] 600
```

```
n <- length(abs.mean)
which(abs.mean == sort(abs.mean,partial=n-1)[n-1])
```

```
## [1] 584
```

```
which(abs.mean == sort(abs.mean,partial=n-1)[n-2])
```

```
## [1] 513
```

```
which(abs.mean == sort(abs.mean,partial=n-1)[n-3])
```

```
## [1] 562
```

```r
which(abs.mean == sort(abs.mean,partial=n-1)[n-4])
```

```
## [1] 549
```

To find which genes differ the most across the two groups, we look at their difference in means. The top 5 genes are Gene 593, 562, 568, 576, 502.