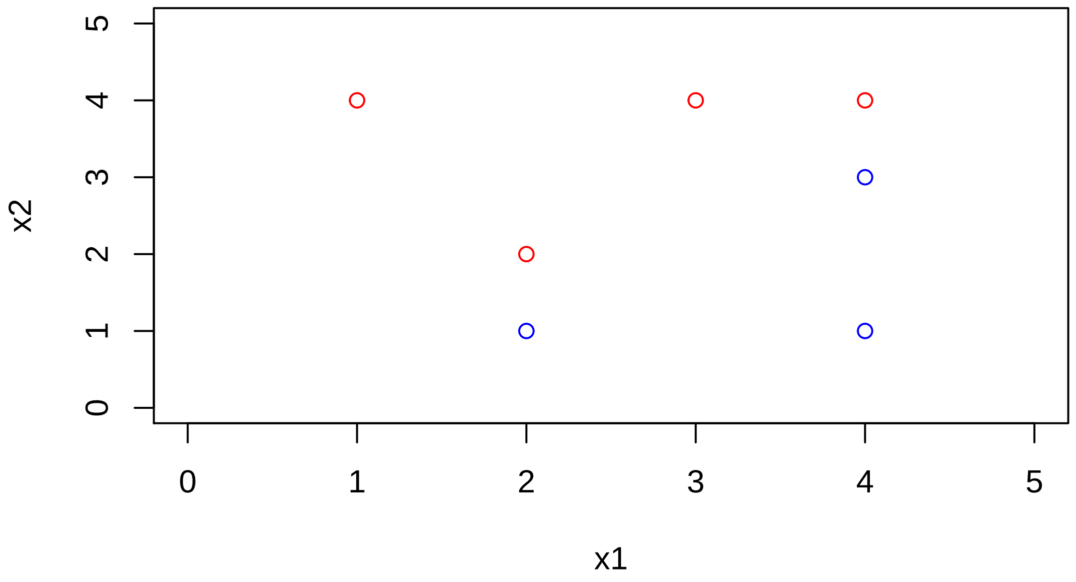
STATS415hw10

Yunguo Cai. Lab Section: 003. uniqname: cyunguo. UMID: 38349078

1. (a)

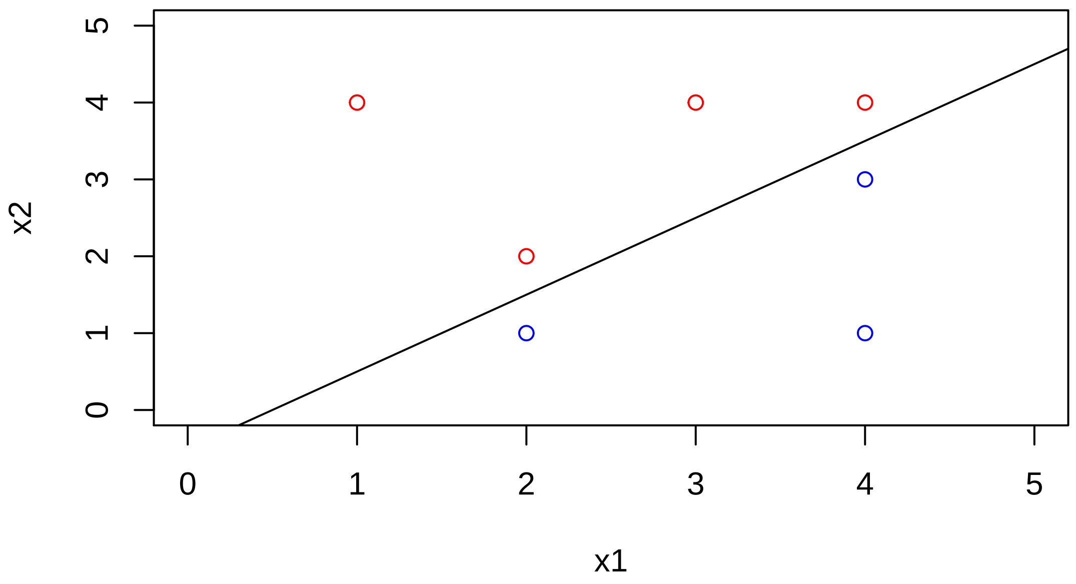
```r
x1 = c(3,2,4,1,2,4,4)
x2 = c(4,2,4,4,1,3,1)
y = c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1,x2,col=y,xlim=c(0,5),ylim=c(0,5))
```



(b) From the plot, the optimal separating hyperplane has to be between the observations (2,1) and (2,2), and between the observations (4,3) and (4,4). So the line of form 9.1 passes through the points (2,1.5) and (4,3.5). The equation is

$$-0.5 + X_1 - X_2 = 0$$

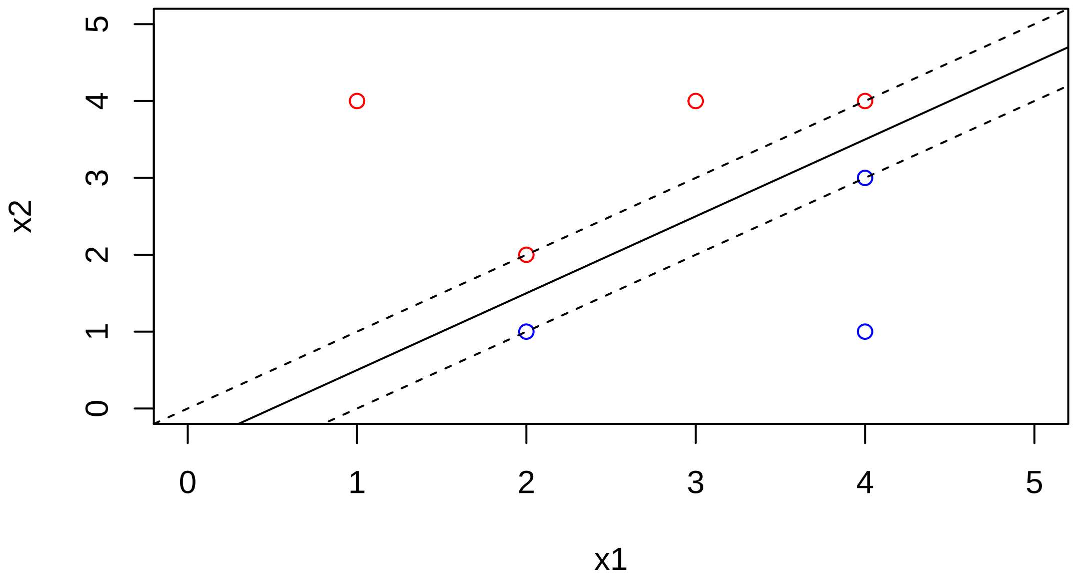
```r
x1 = c(3,2,4,1,2,4,4)
x2 = c(4,2,4,4,1,3,1)
y = c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1,x2,col=y,xlim=c(0,5),ylim=c(0,5))
abline(-0.5,1)
```



(c) The classification rule is Classify to Red if $-X_1 + X_2 + 0.5 > 0$, and classify to Blue otherwise. $\beta_0 = 0.5, \beta_1 = -1, \beta_2 = 1$

(d) The margin here is 1/4.

```r
x1 = c(3,2,4,1,2,4,4)
x2 = c(4,2,4,4,1,3,1)
y = c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1,x2,col=y,xlim=c(0,5),ylim=c(0,5))
abline(-0.5,1)
abline(-1, 1, lty = 2)
abline(0, 1, lty = 2)
```



(e) The support vectors for the maximal margin classifier are the points (2,1),(2,2),(4,3),(4,4).

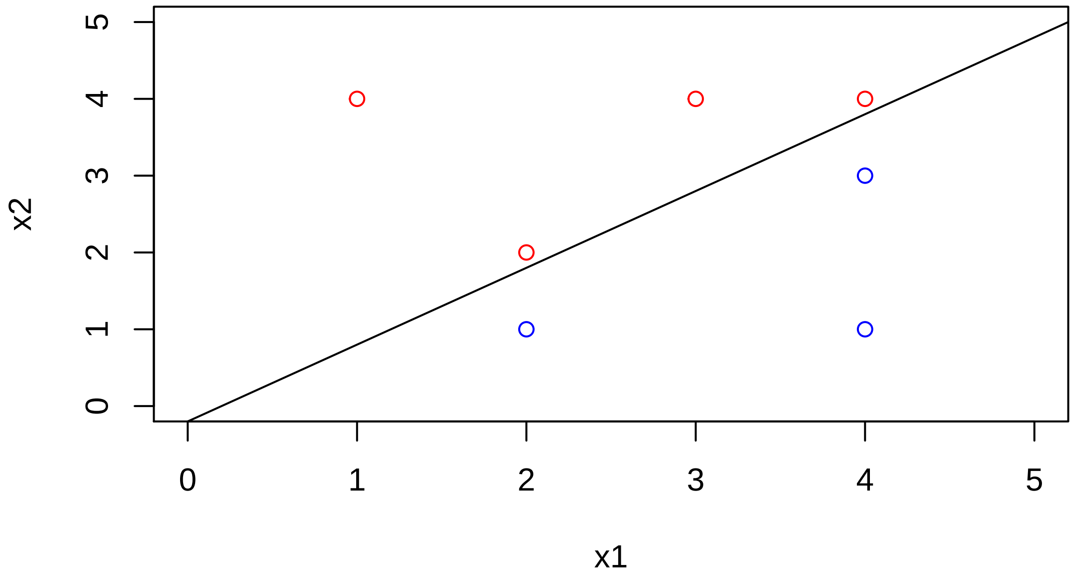
(f) From the plot, it shows that if we moved the observation (4,1) , we would not change the maximal margin hyperplane because it is not a support vector.

(g) The equation for the hyperplane is

$$X_1 - X_2 - 0.3 = 0$$

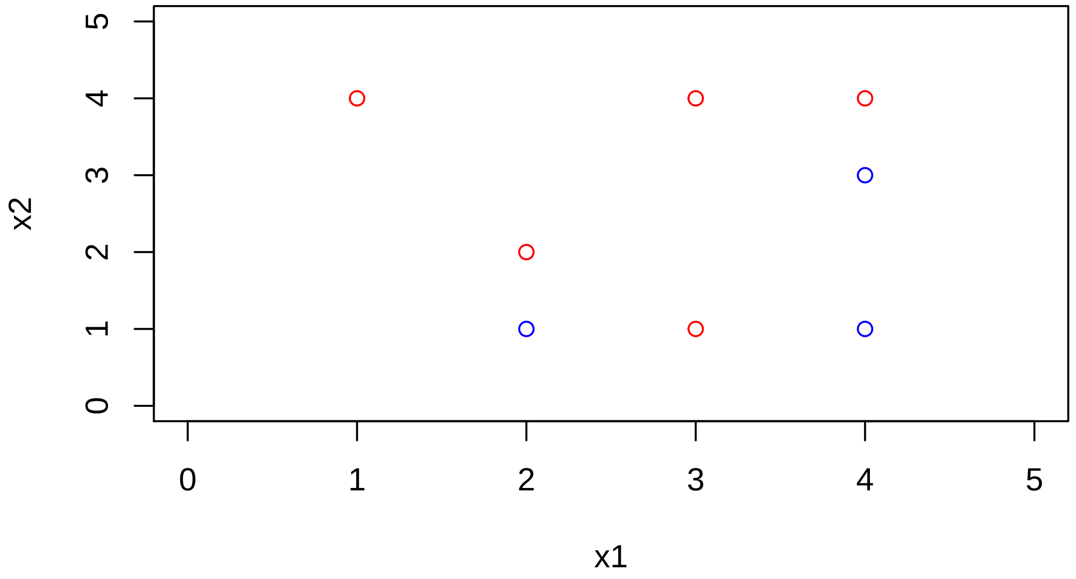
.

```r
x1 = c(3,2,4,1,2,4,4)
x2 = c(4,2,4,4,1,3,1)
y = c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1, x2, col = y, xlim = c(0, 5), ylim = c(0, 5))
abline(-0.2, 1)
```



(h)

```r
x1 = c(3,2,4,1,2,4,4)
x2 = c(4,2,4,4,1,3,1)
y = c("red", "red", "red", "red", "blue", "blue", "blue")
plot(x1, x2, col = y, xlim = c(0, 5), ylim = c(0, 5))
points(3, 1, col = c("red"))
```



2.

```r
library(MASS)
attach(crabs)
set.seed(45678)
blueMale = which(sp == "B" & sex == "M")
orangeMale = which(sp == "O" & sex == "M")
blueFemale = which(sp == "B" & sex == "F")
orangeFemale = which(sp == "O" & sex == "F")
train_id = c(sample(blueMale, size = trunc(0.80 * length(blueMale))),
sample(orangeMale, size = trunc(0.80 * length(orangeMale))),
sample(blueFemale, size = trunc(0.80 * length(blueFemale))),
sample(orangeFemale, size = trunc(0.80 * length(orangeFemale))))
crabs_train = crabs[train_id, ]
crabs_test = crabs[-train_id,]
```

(a)

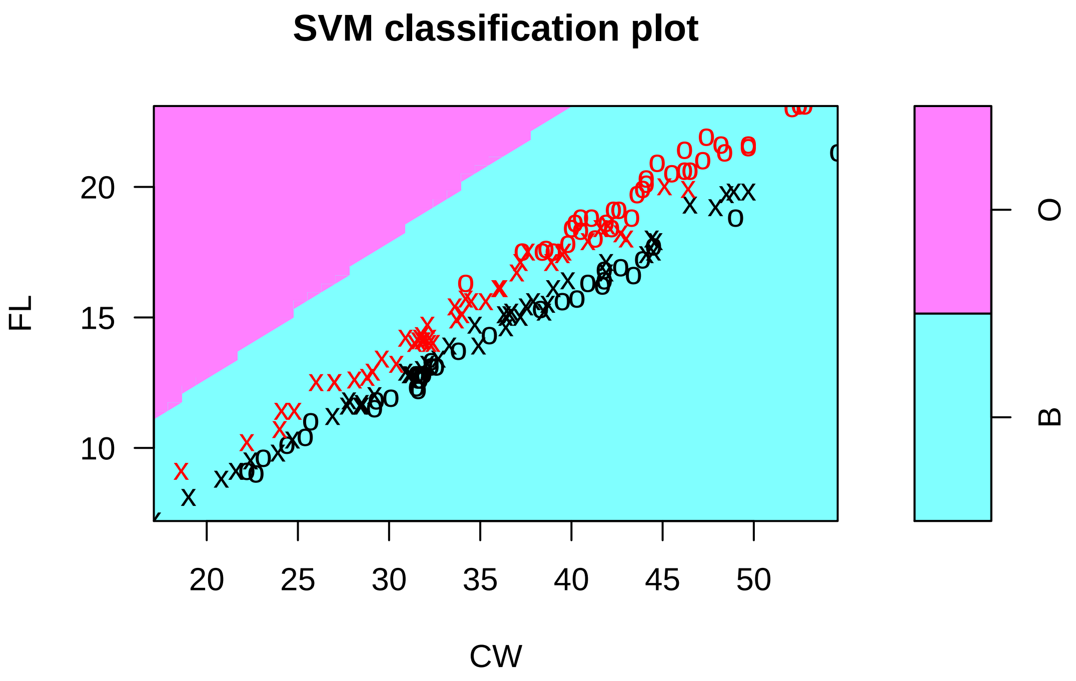
```r
library(e1071)
set.seed(45678)
tune.out <- tune(svm, sp ~ FL + RW + CL + CW + BD, data = crabs_train, kernel = "linear",
                 ranges = list(cost = c(0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 10)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.3
##
## - best performance: 0
##
## - Detailed performance results:
##    cost   error dispersion
## 1  0.01 0.35000 0.10704360
## 2  0.10 0.10625 0.10643366
## 3  0.20 0.03125 0.06073908
## 4  0.30 0.00000 0.00000000
## 5  0.40 0.00000 0.00000000
## 6  0.50 0.00000 0.00000000
## 7  1.00 0.00000 0.00000000
## 8 10.00 0.00000 0.00000000
```

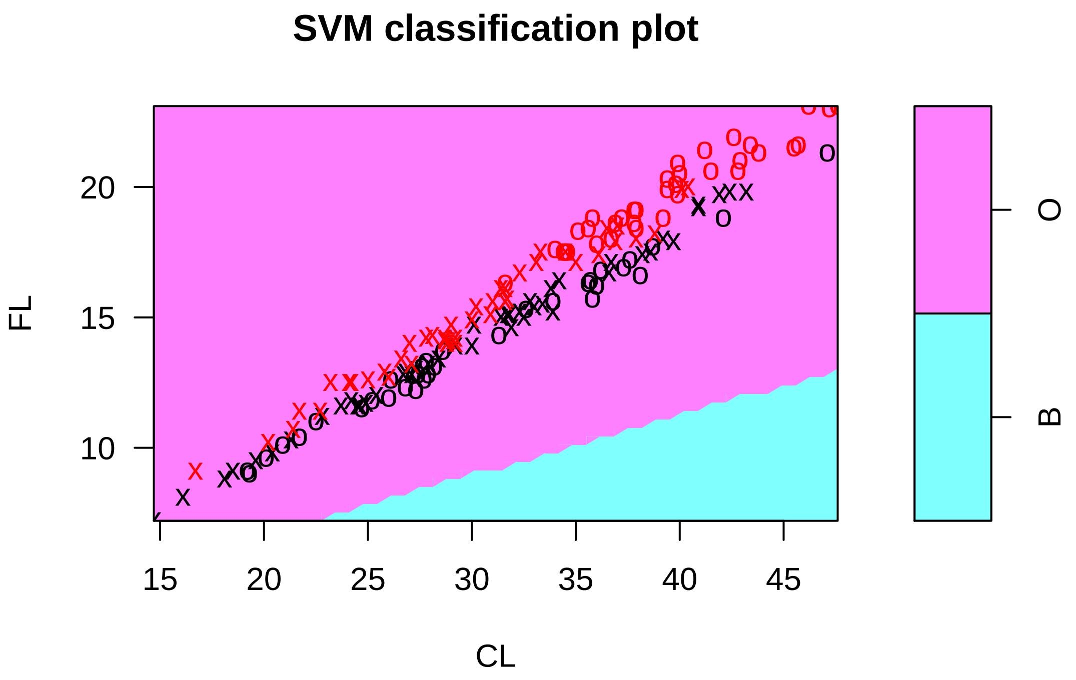
```r
bestmod <- tune.out$best.model
ypred <- predict(bestmod, crabs_test)
table(predict = ypred, truth = crabs_test$sp)
```

```
##        truth
## predict  B  O
##       B 20  1
##       O  0 19
```

```r
plot(bestmod, crabs_train, FL~CW)
```



```
plot(bestmod, crabs_train, FL~CL)
```



The cross-validation errors associated with different values of cost are shown in the previous output. The model with a cost 0.3 has the lowest cv error (which is 0). The test error is also small, 0.025. There's no appropriate pair of preditors that could show the boundary clearly. This might be caused by the boundary hyperplane not parallel to any subplane made up by any pair of preditors. Plotting in higher dimension(more than 2D) might help. Plotting with the first two principal components might also help.

(b)

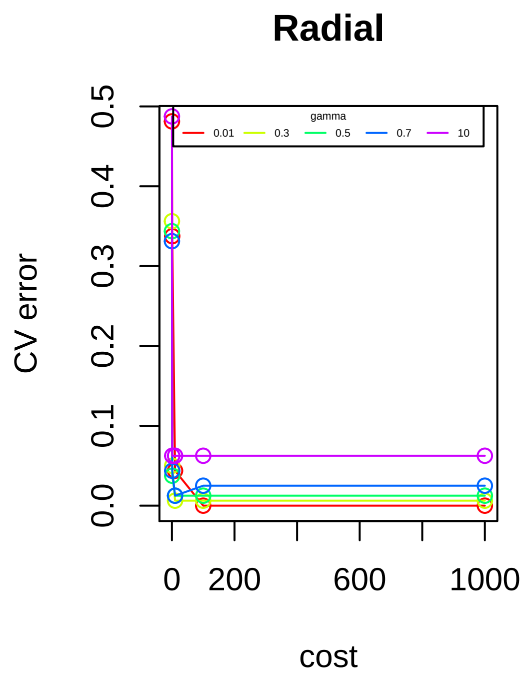
```
bestmod.rad <- tune.out.rad$best.model
#summary(bestmod.rad)
table(true = crabs_test$sp, pred = predict(bestmod.rad, newdata = crabs_test))
```

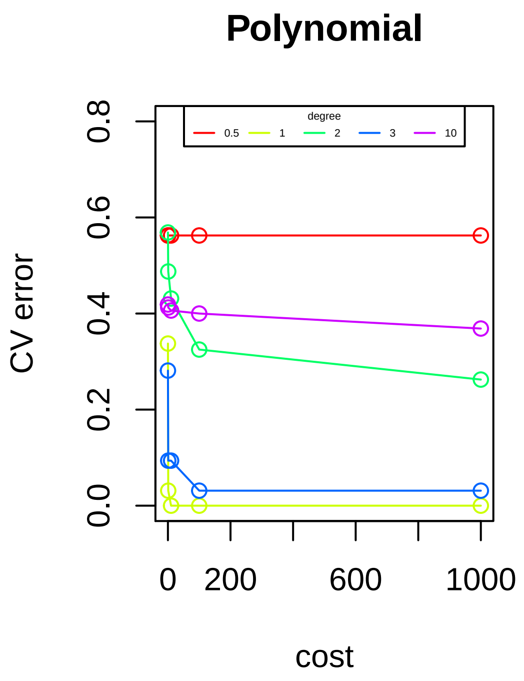
```
##     pred
## true  B  O
##    B 20  0
##    O  0 20
```

```
bestmod.poly <- tune.out.poly$best.model
#summary(bestmod.poly)
table(true = crabs_test$sp, pred = predict(bestmod.poly, newdata = crabs_test))
```

```
##     pred
## true  B  O
##    B 20  0
##    O  0 20
```

```
par(mfrow = c(1,2))
with(tune.out.rad$performances, {
plot(error[gamma == 0.01] ~ cost[gamma == 0.01], main = "Radial",
     type = "o", col = rainbow(5)[1], ylab = "CV error", xlab = "cost")
lines(error[gamma == 0.3] ~ cost[gamma == 0.3], type = "o", col = rainbow(5)[2])
lines(error[gamma == 0.5] ~ cost[gamma == 0.5], type = "o", col = rainbow(5)[3])
lines(error[gamma == 0.7] ~ cost[gamma == 0.7], type = "o", col = rainbow(5)[4])
lines(error[gamma == 10] ~ cost[gamma == 10], type = "o", col = rainbow(5)[5])
})
legend("top", horiz = T, legend = c(0.01, 0.3, 0.5, 0.7, 10), col = rainbow(5),
       lty = 1, cex = .35, title = "gamma")
with(tune.out.poly$performances, {
plot(error[degree == 0.5] ~ cost[degree == 0.5], ylim = c(0,0.8),
     main = "Polynomial",type = "o", col = rainbow(5)[1], ylab = "CV error", xlab = "cost")
lines(error[degree == 1] ~ cost[degree == 1],
type = "o", col = rainbow(5)[2])
lines(error[degree == 2] ~ cost[degree == 2],
type = "o", col = rainbow(5)[3])
lines(error[degree == 3] ~ cost[degree == 3],
type = "o", col = rainbow(5)[4])
lines(error[degree == 10] ~ cost[degree == 10],
type = "o", col = rainbow(5)[5])
})
legend("top", horiz = T, legend = c(0.5, 1, 2, 3, 10), col = rainbow(5),
       lty = 1, cex = .35, title = "degree")
```





The cross-validation errors associated with different values of parameters are shown in the previous output. For radial kernal, the model with cost = 100 and gamma = 0.01 gives the lowest cv error (which is 0). Its test error is also 0. For polynomial kernal, the model with cost = 10 and degree = 1 gives the lowest cv error (which is 0). Its test error is also 0. As in part (a), there's no appropriate pair of preditors that could show the boundary clearly.