Question 2:

```
#Inputs
#w:  w[1:d] is the normal vector of a hyperplane,
#    w[d+1] = -c is the negative offset parameter.
#n: sample size

#Outputs
#S: n by (d+1) sample matrix with last col 1
#y: vector of the associated class labels

fakedata <- function(w, n){

if(! require(MASS))
{
    install.packages("MASS")
}
if(! require(mvtnorm))
{
    install.packages("mvtnorm")
}

require(MASS)
require(mvtnorm)

# obtain dimension
d <- length(w) - 1

# compute the offset vector and a Basis consisting of w and its nullspace
offset <- -w[length(w)] * w[1:d] / sum(w[1:d]^2)
Basis <- cbind(Null(w[1:d]), w[1:d])

# Create samples, correct for offset, and extend
# rmvnorm(n,mean,sigme) ~ generate n samples from N(0,I) distribution
S <- rmvnorm(n, mean=rep(0,d),sigma = diag(1,d)) %*%  t(Basis)
S <- S + matrix(rep(offset,n),n,d,byrow=T)
S <- cbind(S,1)

# compute the class assignments
y <- as.vector(sign(S %*% w))

# add corrective factors to points that lie on the hyperplane.
S[y==0,1:d] <- S[y==0,1:d] + runif(1,-0.5,0.5)*10^(-4)
y = as.vector(sign(S %*% w))
return(list(S=S, y=y))

} # end function fakedata
```

part 1):

```r
classify <- function(S,z) {

  fx <- sign(S %*% z)

  return(fx)
  }
```

part 2):

```r
perceptrain <- function(S,y){

     #random generate z
     z <- c(rnorm(3))

     #compute classifer
     fx_s <- sign(S %*% z)

     #check f(x) = yi

     check <- (fx_s == y)

     n <- sum(check == "FALSE")

     #add learning rate
     k <- 1

     List <- list()

      while(n > 0){

       Z_history <- z

       Gp <- matrix(c(S[check == "FALSE"]), byrow = F, ncol = dim(S)[2])

       #compute z(k+1)
       z <- c(Z_history - (1/k) * (-y[check == "FALSE"] %*% Gp))

       #re-compute fx and check
       fx_s <- sign(S %*% z)
       check <- (fx_s == y)
       n <- sum(check == "FALSE")

       #store the Z_history to a matrix form
       List[[k]] <- z

       Matrix <- do.call(rbind, List)

       k <- k + 1
```

```r
    }

  return(list(z = z, Z_history = Matrix))

     }
```

part 3):

```r
z <- c(rnorm(6))
S <- fakedata(z,25)
```

```
## Loading required package: MASS
## Loading required package: mvtnorm
```

```r
y <- S$y
S <- S$S
fx <- classify(S,z)

sum(fx == y)
```

```
## [1] 25
```

The output is the total number of correct classifiers

part 4):

```r
set.seed(5)
z <- c(rnorm(3))
w <- z
S <- fakedata(w,100)
y <- S$y
data <- data.frame(S)
S <- S$S

A <- perceptrain(S,y)
A
```

```
## $z
## [1] -13.34726  31.86092 -24.80498
##
## $Z_history
##             [,1]     [,2]       [,3]
##  [1,]  11.34082 14.81394 -22.607883
##  [2,] -18.17590 40.23286  -7.607883
##  [3,] -17.03389 37.25658 -14.274550
##  [4,] -15.65079 35.52521 -18.274550
##  [5,] -14.55392 34.57908 -20.474550
##  [6,] -14.14240 34.03938 -21.474550
##  [7,] -13.89586 33.62132 -22.188835
```

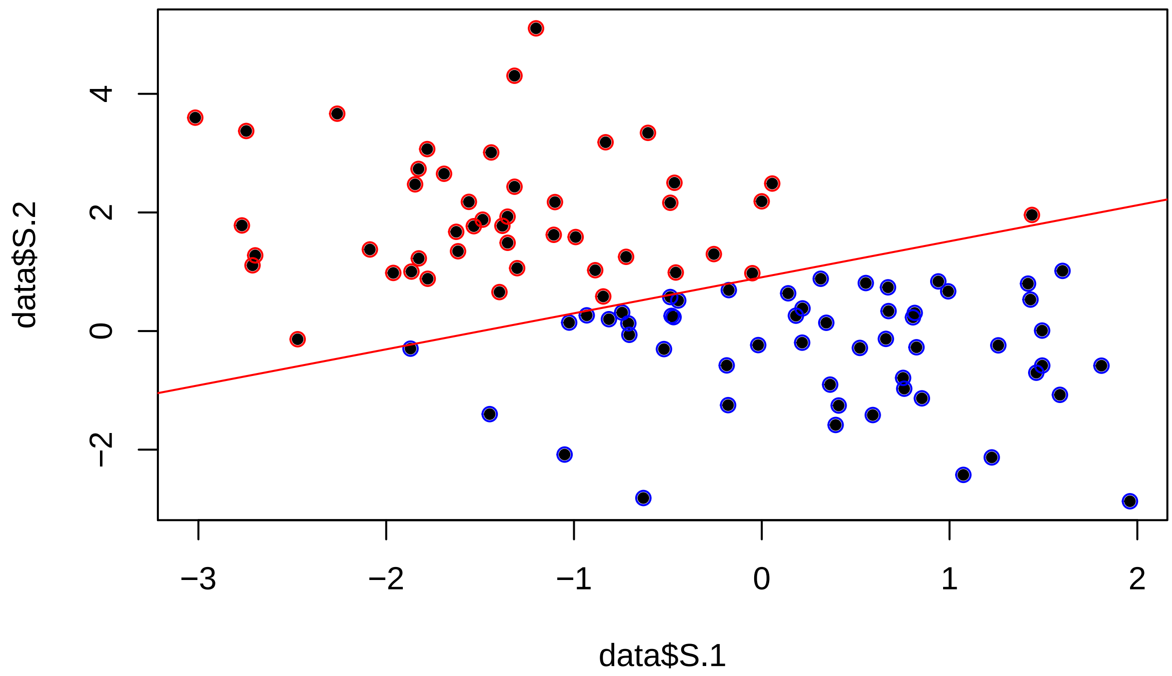
```
##  [8,] -13.79666 33.28859 -22.688835
##  [9,] -13.70849 32.99282 -23.133280
## [10,] -13.67360 32.77817 -23.433280
## [11,] -13.64189 32.58303 -23.706007
## [12,] -13.61282 32.40415 -23.956007
## [13,] -13.58599 32.23903 -24.186776
## [14,] -13.53864 32.14872 -24.329634
## [15,] -13.49444 32.06444 -24.462967
## [16,] -13.45301 31.98542 -24.587967
## [17,] -13.42435 31.95167 -24.646790
## [18,] -13.39727 31.91980 -24.702346
## [19,] -13.37163 31.88961 -24.754977
## [20,] -13.34726 31.86092 -24.804977
```

```
hist <- matrix(unlist(A[2]), byrow = F, ncol = length(z))
hist
```

```
##            [,1]     [,2]       [,3]
##  [1,]  11.34082 14.81394 -22.607883
##  [2,] -18.17590 40.23286  -7.607883
##  [3,] -17.03389 37.25658 -14.274550
##  [4,] -15.65079 35.52521 -18.274550
##  [5,] -14.55392 34.57908 -20.474550
##  [6,] -14.14240 34.03938 -21.474550
##  [7,] -13.89586 33.62132 -22.188835
##  [8,] -13.79666 33.28859 -22.688835
##  [9,] -13.70849 32.99282 -23.133280
## [10,] -13.67360 32.77817 -23.433280
## [11,] -13.64189 32.58303 -23.706007
## [12,] -13.61282 32.40415 -23.956007
## [13,] -13.58599 32.23903 -24.186776
## [14,] -13.53864 32.14872 -24.329634
## [15,] -13.49444 32.06444 -24.462967
## [16,] -13.45301 31.98542 -24.587967
## [17,] -13.42435 31.95167 -24.646790
## [18,] -13.39727 31.91980 -24.702346
## [19,] -13.37163 31.88961 -24.754977
## [20,] -13.34726 31.86092 -24.804977
```

```
plot(data$S.1,data$S.2, pch = 20)
points(subset(data,y== 1),col="red")
points(subset(data,y== -1),col="blue")

#final classifier
intercept <- - z[3]/z[2]
slope <- - z[1]/z[2]
abline(intercept, slope, col = "red")
```



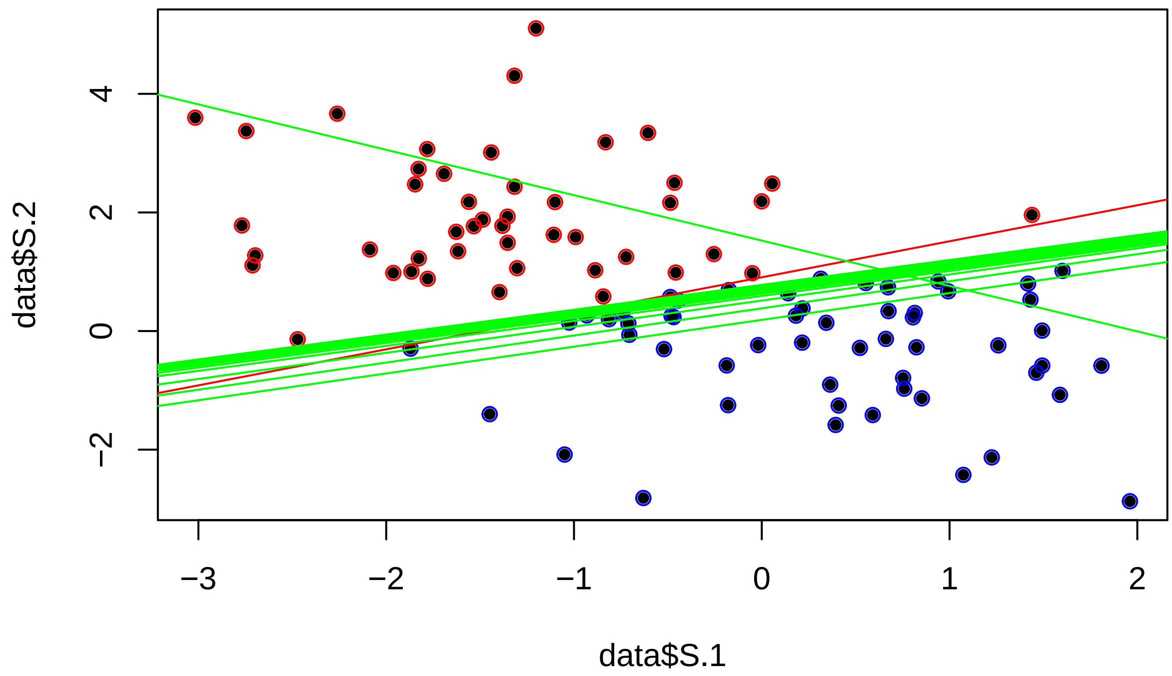

```r
#ploting the preious step
plot(data$S.1,data$S.2, pch = 20)
points(subset(data,y== 1),col="red")
points(subset(data,y== -1),col="blue")

intercept <- - z[3]/z[2]
slope <- - z[1]/z[2]
abline(intercept, slope, col = "red")

intercept_hist <- c(rep(1,dim(hist)[1]))
slope_hist <- c(rep(1,dim(hist)[1]))
for(i in 1:dim(hist)[1]){

  intercept_hist[i] <- - hist[i,3]/hist[i,2]
  slope_hist[i] <- - hist[i,1]/hist[i,2]
  abline(intercept_hist[i], slope_hist[i], col = "green")

  }
```



Red line is final classifier with 0 error. Green line is preious step.

Question 3

a1):

```
set.seed(121)
library(e1071)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
label <- read.table("~/Desktop/uspscl.txt")
data <-read.table("~/Desktop/uspsdata.txt")

#split dataset
i <- sample(1:200,160)
train_data <- data[i,]
test_data <- data[-i,]
train_label<- label[i,]
test_label<- label[-i,]

#
```

```r
liner <- function(cost){

  rate <- c()

for (i in 1:length(cost)){

  svm.model <- svm(train_data, train_label, cost = cost[i], kernel="linear", cross = 5)
  svm.pred  <- predict(svm.model, test_data)
  label.pred <- sign(svm.pred)
  error <- sum(label.pred != test_label)
  rate[i] <- error/length(test_data)

 }
   return(rate)
}

cost <- seq(0.005, 0.9, length = 20)
rate <- liner(cost)
rate
```

```
##  [1] 0.01171875 0.01171875 0.01562500 0.01562500 0.01562500 0.01562500
##  [7] 0.01562500 0.01562500 0.01562500 0.01562500 0.01953125 0.01953125
## [13] 0.01953125 0.01953125 0.01953125 0.01953125 0.01953125 0.01953125
## [19] 0.01953125 0.01953125
```

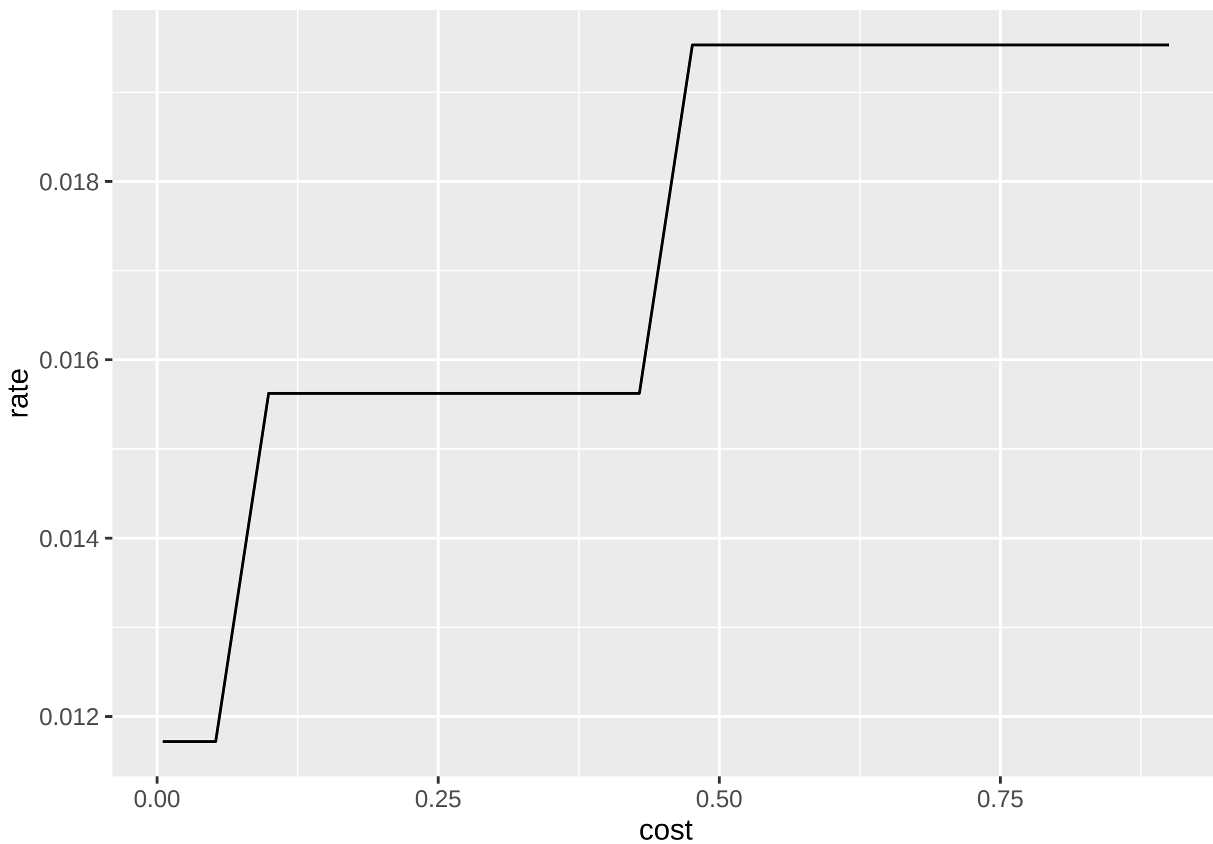
```r
which.min(rate)
```

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

```r
cost[which.min(rate)]
```

```
## [1] 0.005
```

```r
ggplot() + geom_line(aes(cost,rate))
```



a2):

```
radial <- function(cost,gamma){

  rate <- matrix(1, nrow = 5, ncol = 5, byrow = F)

for (i in 1:length(gamma)){

 for(j in 1:length(cost)){
  svm.model <- svm(train_data, train_label, cost = cost[j],  kernel="radial", cross = 5, gamma = gamma[
  svm.pred  <- predict(svm.model, test_data)
  label.pred <- sign(svm.pred)
  error <- sum(label.pred != test_label)
  rate[i,j] <- error/length(test_data)
  }

}

   return(rate)
}

cost <- seq(0.0005,1, length = 5)
gamma <- sort((10^-(1:5)),decreasing = F)
```

```r
table <- radial(cost,gamma)
table
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]
## [1,] 0.08984375 0.08984375 0.08984375 0.08984375 0.08984375
## [2,] 0.08984375 0.08984375 0.00390625 0.00781250 0.00781250
## [3,] 0.08984375 0.00390625 0.00390625 0.00390625 0.00390625
## [4,] 0.08984375 0.01171875 0.00390625 0.00390625 0.00390625
## [5,] 0.08984375 0.08984375 0.08984375 0.08984375 0.07031250
```

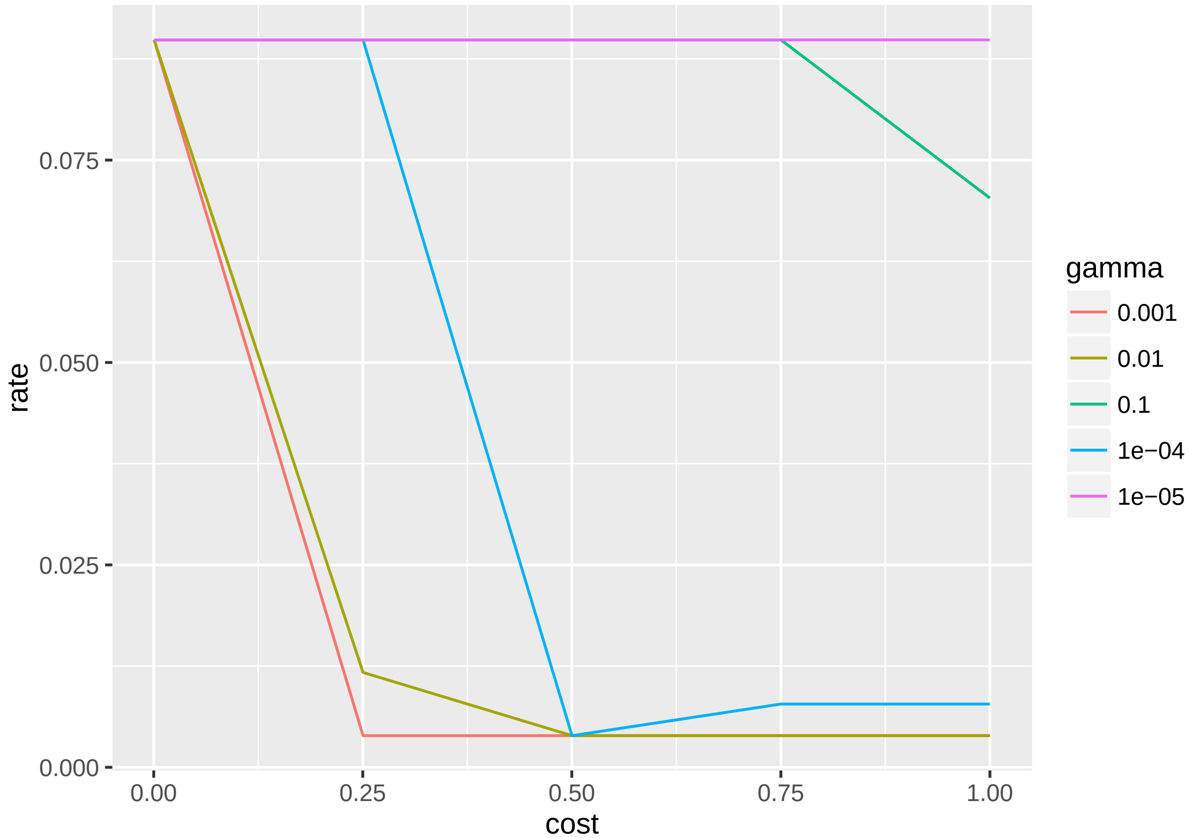
```r
rate <- c(table[,1],table[,2],table[,3],table[,4],table[,5])
rate <- as.numeric(rate)
gamma <- c(rep(as.character(gamma),5))
gamma
```

```
##  [1] "1e-05" "1e-04" "0.001" "0.01"  "0.1"   "1e-05" "1e-04" "0.001"
##  [9] "0.01"  "0.1"   "1e-05" "1e-04" "0.001" "0.01"  "0.1"   "1e-05"
## [17] "1e-04" "0.001" "0.01"  "0.1"   "1e-05" "1e-04" "0.001" "0.01"
## [25] "0.1"
```

```r
cost <- c(rep((cost)[1],5),rep((cost)[2],5),rep((cost)[3],5),rep((cost)[4],5),rep((cost)[5],5))
table <- data.frame(cbind(rate,gamma,cost))
table$rate <- as.numeric(as.character(table$rate))
table$cost <- as.numeric(as.character(table$cost))
table
```

```
##          rate gamma     cost
## 1  0.08984375 1e-05 0.000500
## 2  0.08984375 1e-04 0.000500
## 3  0.08984375 0.001 0.000500
## 4  0.08984375  0.01 0.000500
## 5  0.08984375   0.1 0.000500
## 6  0.08984375 1e-05 0.250375
## 7  0.08984375 1e-04 0.250375
## 8  0.00390625 0.001 0.250375
## 9  0.01171875  0.01 0.250375
## 10 0.08984375   0.1 0.250375
## 11 0.08984375 1e-05 0.500250
## 12 0.00390625 1e-04 0.500250
## 13 0.00390625 0.001 0.500250
## 14 0.00390625  0.01 0.500250
## 15 0.08984375   0.1 0.500250
## 16 0.08984375 1e-05 0.750125
## 17 0.00781250 1e-04 0.750125
## 18 0.00390625 0.001 0.750125
## 19 0.00390625  0.01 0.750125
## 20 0.08984375   0.1 0.750125
## 21 0.08984375 1e-05 1.000000
## 22 0.00781250 1e-04 1.000000
## 23 0.00390625 0.001 1.000000
## 24 0.00390625  0.01 1.000000
## 25 0.07031250   0.1 1.000000
```

```r
ggplot(table,aes(cost,rate)) + geom_line(aes(color = gamma))
```




```r
which.min(table$rate)
```

```
## [1] 8
```

part b):

```r
l.svm <- svm(train_data, train_label, cost = 0.005, kernel="linear", cross = 5, gamma = 0)
svm.pred  <- predict(l.svm, test_data)
label.pred <- sign(svm.pred)
error.l <- sum(label.pred != test_label)
error.l
```

```
## [1] 3
```

```r
r.svm <- svm(train_data, train_label, cost = 0.25375,  kernel="radial", cross = 5, gamma = 0.001)
svm.pred  <- predict(r.svm, test_data)
label.pred <- sign(svm.pred)
error.r <- sum(label.pred != test_label)
error.r
```

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