```
# Rudy Meza
# Machine Learning in R
# Michael Chang
# Spring
# '***Lab and Quiz***'
# '********************Machine Learning*********************'

# 9.6.1 Support Vector Classifier
set.seed(1)
x <- matrix(rnorm(20 * 2), ncol = 2)
y <- c(rep(-1, 10), rep(1, 10))
x[y == 1, ] <- x[y ==1, ] + 1
plot(x, col = (3 - y))
dat <- data.frame(x = x, y = as.factor(y))
library(e1071)
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 10, scale = FALSE)
plot(svmfit, dat)
svmfit$index
summary(svmfit)
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 0.1, scale = FALSE)
plot(svmfit, dat)
svmfit$index
set.seed(1)
tune.out <- tune(svm, y ~ ., data = dat, kernel = "linear", ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10,
100)))
summary(tune.out)

bestmod <- tune.out$best.model
summary(bestmod)

xtest <- matrix(rnorm(20 * 2), ncol = 2)
ytest <- sample(c(-1, 1), 20, rep = TRUE)
xtest[ytest == 1, ] <- xtest[ytest == 1, ] + 1
testdat <- data.frame(x = xtest, y = as.factor(ytest))
ypred <- predict(bestmod, testdat)
table(predict = ypred, truth = testdat$y)
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 0.01, scale = FALSE)
ypred <- predict(svmfit, testdat)
table(predict = ypred, truth = testdat$y)
x[y == 1, ] <- x[y == 1, ] + 0.5
plot(x, col = (y + 5)/2, pch = 19)
dat <- data.frame(x = x, y = as.factor(y))
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 1e+05)
summary(svmfit)

plot(svmfit, dat)

svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 1)
```

```
summary(svmfit)
plot(svmfit, dat)
# 9.6.2 Support Vector Machine
set.seed(1)
x <- matrix(rnorm(200 * 2), ncol = 2)
x[1:100, ] <- x[1:100, ] + 2
x[101:150, ] <- x[101:150, ] - 2
y <- c(rep(1, 150), rep(2, 50))
dat <- data.frame(x = x, y = as.factor(y))
plot(x, col = y)
train <- sample(200, 100)
svmfit <- svm(y ~ ., data = dat[train, ], kernel = "radial", gamma = 1, cost = 1)
plot(svmfit, dat[train, ])
summary(svmfit)
svmfit <- svm(y ~ ., data = dat[train, ], kernel = "radial", gamma = 1, cost = 1e+05)
plot(svmfit, dat[train, ])
set.seed(1)
tune.out <- tune(svm, y ~ ., data = dat[train, ], kernel = "radial", ranges = list(cost = c(0.1, 1, 10, 100,
1000), gamma = c(0.5, 1, 2, 3, 4)))
summary(tune.out)
table(true = dat[-train, "y"], pred = predict(tune.out$best.model, newdata = dat[-train, ]))

# 9.6.3 ROC Curves

library(ROCR)

rocplot <- function(pred, truth, ...) {
   predob <- prediction(pred, truth)
   perf <- performance(predob, "tpr", "fpr")
   plot(perf, ...)
}

svmfit.opt <- svm(y ~ ., data = dat[train, ], kernel = "radial", gamma = 2, cost = 1, decision.values = TRUE)
fitted <- attributes(predict(svmfit.opt, dat[train, ], decision.values = TRUE))$decision.values

par(mfrow = c(1, 2))
rocplot(fitted, dat[train, "y"], main = "Training Data")

svmfit.flex <- svm(y ~ ., data = dat[train, ], kernel = "radial", gamma = 50, cost = 1, decision.values =
TRUE)
fitted <- attributes(predict(svmfit.flex, dat[train, ], decision.values = T))$decision.values
rocplot(fitted, dat[train, "y"], add = T, col = "red")

fitted <- attributes(predict(svmfit.opt, dat[-train, ], decision.values = T))$decision.values
rocplot(fitted, dat[-train, "y"], main = "Test Data")
fitted <- attributes(predict(svmfit.flex, dat[-train, ], decision.values = T))$decision.values
rocplot(fitted, dat[-train, "y"], add = T, col = "red")
```

```
# 9.6.4 SVM with Multiple Classes

set.seed(1)
x <- rbind(x, matrix(rnorm(50 * 2), ncol = 2))
y <- c(y, rep(0, 50))
x[y == 0, 2] <- x[y == 0, 2] + 2
dat <- data.frame(x = x, y = as.factor(y))
par(mfrow = c(1, 1))
plot(x, col = (y + 1))

svmfit <- svm(y ~ ., data = dat, kernel = "radial", cost = 10, gamma = 1)
plot(svmfit, dat)

# 9.6.5 Application to Gene Expression Data

library(ISLR)
names(Khan)

dim(Khan$xtrain)

dim(Khan$xtest)


length(Khan$ytrain)

length(Khan$ytest)
table(Khan$ytrain)

table(Khan$ytest)

dat <- data.frame(x = Khan$xtrain, y = as.factor(Khan$ytrain))
out <- svm(y ~ ., data = dat, kernel = "linear", cost = 10)
summary(out)

table(out$fitted, dat$y)

dat.te <- data.frame(x = Khan$xtest, y = as.factor(Khan$ytest))
pred.te <- predict(out, newdata = dat.te)
table(pred.te, dat.te$y)

  # Quiz

# Question 1

set.seed(1)
x <- matrix(rnorm(20 * 2), ncol = 2)
y <- c(rep(-1, 10), rep(1, 10))
x[y == 1, ] <- x[y == 1, ] + 1
```

```
dat <- data.frame(x = x, y = as.factor(y))
library(e1071)
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 1, scale = FALSE)
svmfit$index

# Question 2
set.seed (1)
tune.out=tune(svm ,y~.,data=dat ,kernel ="linear",
ranges =list(cost=c(0.001 , 0.01, 0.1, 1,5,10,100) ))
summary(tune.out)

# Question 3
# 2

# Question 4
set.seed (1)
train=sample (200 ,100)
svmfit1 =svm(y~.,data=dat[train,], kernel ="radial", gamma=20,
cost=1)

svmfit2=svm(y~.,data=dat[train,], kernel ="radial", gamma=10, cost=1)
svmfit3=svm(y~.,data=dat[train,], kernel ="radial", gamma=2, cost=1)
svmfit4=svm(y~.,data=dat[train,], kernel ="radial", gamma=.1, cost=1)
par(mfrow=c(2,2))
 plot(svmfit1,dat[train,])
plot(svmfit2,dat[train,])
plot(svmfit3,dat[train,])
plot(svmfit4,dat[train,])

# Question 5
set.seed(1)
train=sample (200 ,100)
tune.out <- tune(svm, y ~ ., data = dat[train, ], kernel = "radial", ranges = list(cost = c(0.1,0.5), gamma =
c(0.5, 1, 2, 3, 4,0.1,1.5,5)))
summary(tune.out)
# Question 6


set.seed(1)
x=matrix (rnorm (200*2) , ncol =2)
x[1:100 ,]=x[1:100 ,]+2
x[101:150 ,]= x[101:150 ,] -2
y=c(rep (1 ,150) ,rep (2 ,50) )
dat=data.frame(x=x,y=as.factor (y))
train=sample (200 ,100)
svmfitopt=svm(y~.,data=dat[train,], kernel ="radial", gamma=0.5,
cost=100)
```

```
fitted1<- attributes(predict(svmfitopt, dat[train, ], decision.values = TRUE))$decision.values

svmfitopt2=svm(y~.,data=dat[train,], kernel ="radial", gamma=2,
cost=0.5)
fitted2<- attributes(predict(svmfitopt2, dat[train, ], decision.values = TRUE))$decision.values


par(mfrow = c(1, 2))
rocplot(fitted1, dat[train,"y"], main = "Training Data")
rocplot(fitted2, dat[train,"y"], main = "Training Data")

# Question 7

set.seed(1)
x <- rbind(x, matrix(rnorm(50 * 2), ncol = 2))
y <- c(y, rep(0, 50))
x[y == 0, 2] <- x[y == 0, 2] + 2
dat <- data.frame(x = x, y = as.factor(y))
par(mfrow = c(1, 1))
plot(x, col = (y + 1))

svmfit1 <- svm(y ~ ., data = dat, kernel = "radial", cost = 10, gamma = 5)
svmfit2 <- svm(y ~ ., data = dat, kernel = "radial", cost = 10, gamma = 10)
svmfit3 <- svm(y ~ ., data = dat, kernel = "radial", cost = 10, gamma = 20)
svmfit4 <- svm(y ~ ., data = dat, kernel = "radial", cost = 10, gamma = 50)

par(mfrow = c(2, 2))
plot(svmfit1, dat)
plot(svmfit2, dat)
plot(svmfit3, dat)
plot(svmfit4, dat)
```