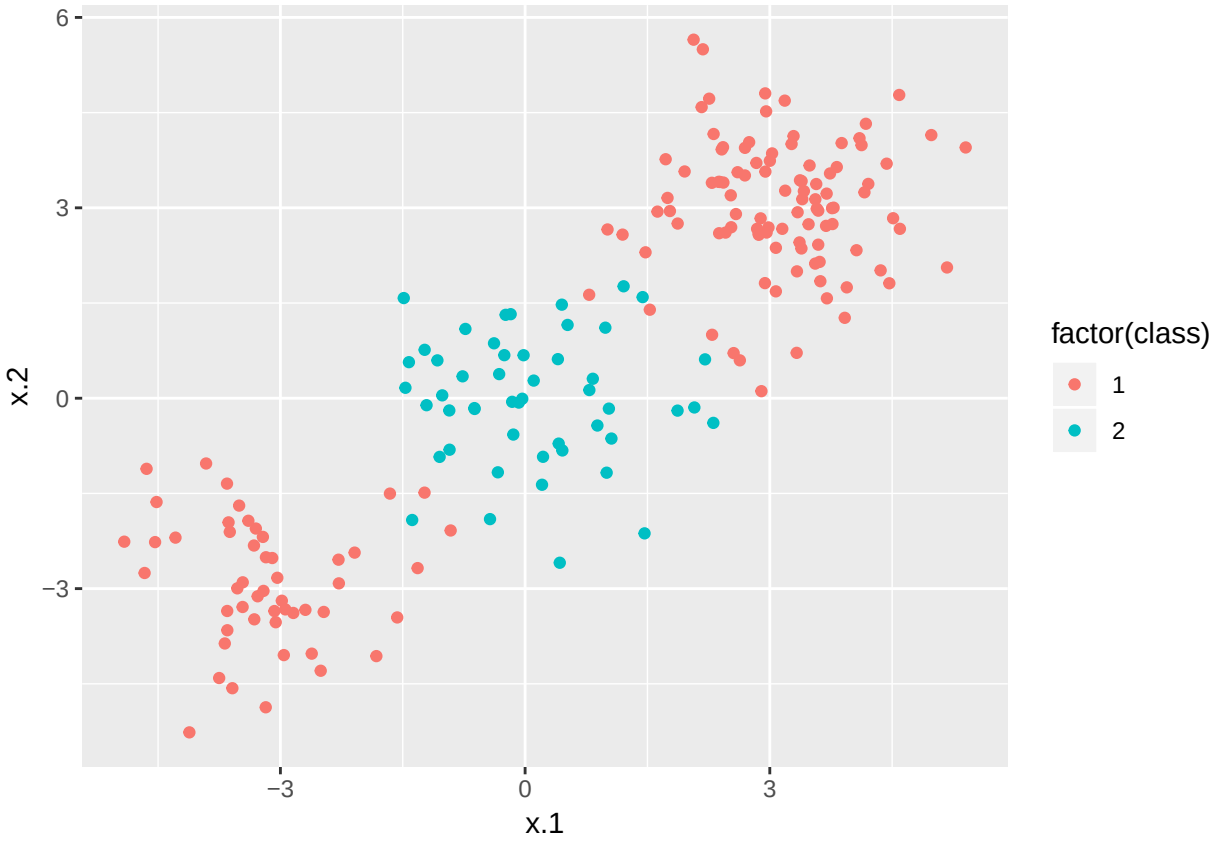
q25

Julia Lee

12/8/2018

```
set.seed(1)
x = matrix(rnorm(200*2), ncol = 2)
x[1:100,] = x[1:100,]+3
x[101:150,] = x[101:150,]-3
class = c(rep(1,150),rep(2,50))
nonlinear_data = data.frame(x = x, class = as.factor(class))
ggplot(nonlinear_data, aes(x.1, x.2, colour = factor(class))) +
geom_point()
```




```
nonlinear_train = nonlinear_data %>%
sample_frac(0.5)
nonlinear_test = nonlinear_data %>%
setdiff(nonlinear_train)
```

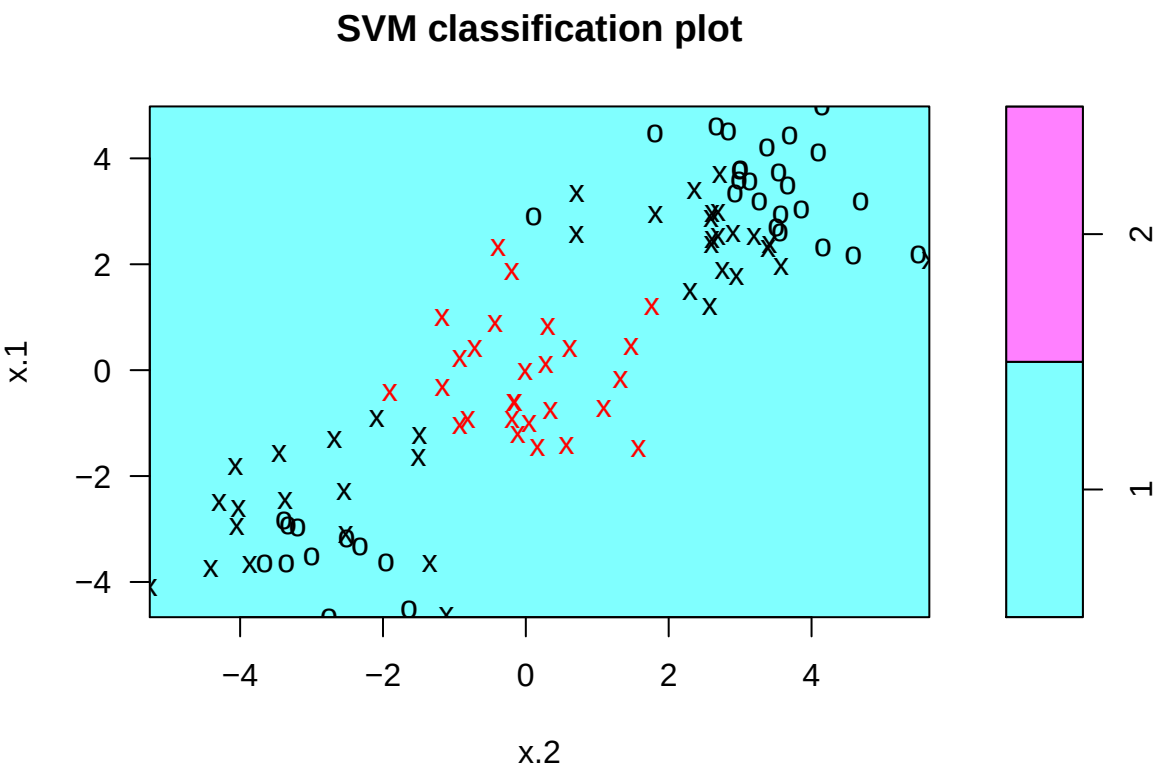
As we can see there is a visible but non-linear separation between the two classes.

Polynomial kernel

```
library(e1071)
set.seed(1)
# perform cross-validation using tune() to select the best choice of degree and cost for an SVM with a p
tune_out = tune(svm, class~., data = nonlinear_train, kernel = "polynomial")
ranges = list(cost = c(0.1,1,10,100,1000), degree = c(2,3,4,5,6),gamma = c(0.5,1))
bestmod = tune_out$best.model
summary(bestmod)
```

```
##
## Call:
## best.tune(method = svm, train.x = class ~ ., data = nonlinear_train,
##     kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##       gamma:  0.5
##      coef.0:  0
##
## Number of Support Vectors:  65
##
##  ( 27 38 )
##
##
## Number of Classes:  2
##
## Levels:
##  1 2
```

```
plot(bestmod,nonlinear_train)
```




```
table(true = nonlinear_test$class, pred = predict(tune_out$best.model, newdata = nonlinear_test))
```

```
##     pred
## true  1  2
##    1 77  0
##    2 23  0
```

The best polynominal kernel SVM has a cost = 1 and degree = 3 and gamma = 0.5 77% of the test observations are correctly classified by this SVM.

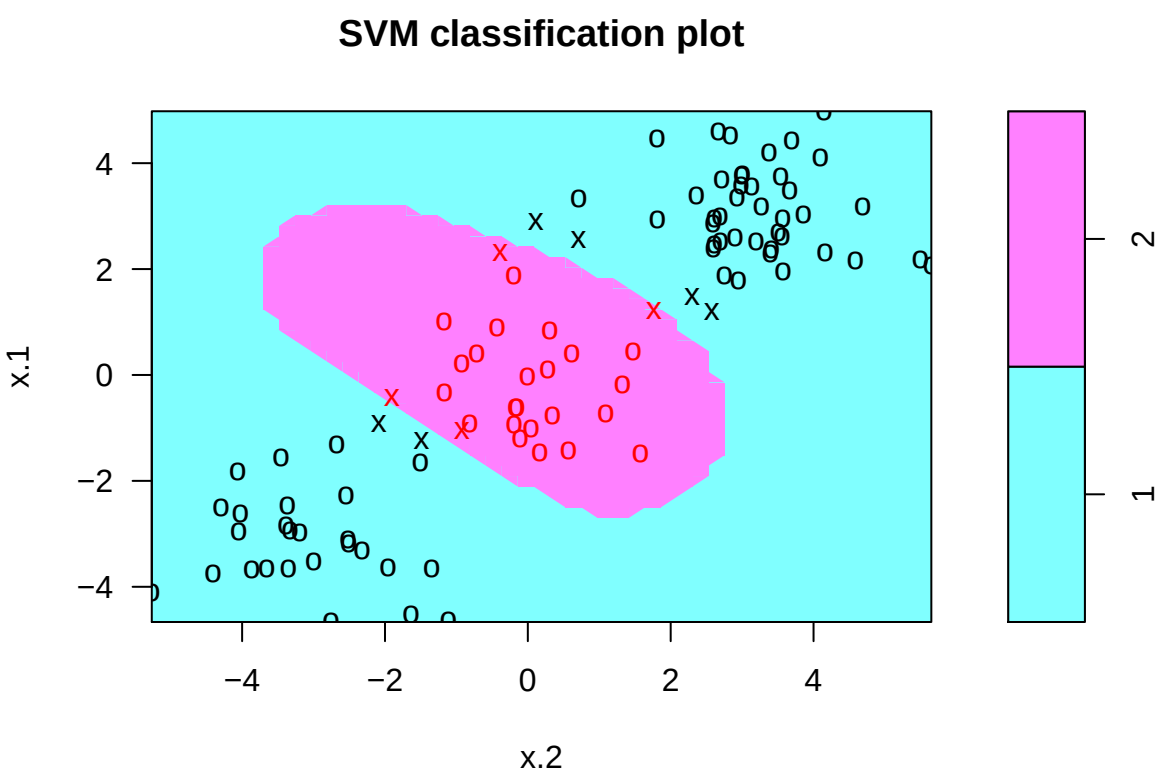
radial kernel

```
library(e1071)
set.seed(1)
# perform cross-validation using tune() to select the best choice of  and cost for an SVM with a radial
tune_out = tune(svm, class~., data = nonlinear_train, kernel = "radial",
ranges = list(cost = c(0.1,1,10,100,1000), gamma = c(0.5,1,2,3,4)))
bestmod = tune_out$best.model
summary(bestmod)
```

```
##
## Call:
## best.tune(method = svm, train.x = class ~ ., data = nonlinear_train,
##     ranges = list(cost = c(0.1, 1, 10, 100, 1000), gamma = c(0.5,
##         1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
```

```
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  1
##
## Number of Support Vectors:  10
##
##  ( 4 6 )
##
##
## Number of Classes:  2
##
## Levels:
##  1 2
```

```
plot(bestmod,nonlinear_train)
```




```
table(true = nonlinear_test$class, pred = predict(tune_out$best.model, newdata = nonlinear_test))
```

```
##     pred
## true  1  2
##    1 75  2
##    2  1 22
```

The best radial kernel SVM has a cost of 10 and gamma = 1 99% of the test observations are correctly classified by this SVM.

As we can see that radial kernel technique performs best on the test data.