In [8]:

```
library(readr)
```

In [9]:

```
irisR <- read_csv("C:/Users/admin/Desktop/iris.csv")
```

```
Parsed with column specification:
cols(
  Sepal.Length = col_double(),
  Sepal.Width = col_double(),
  Petal.Length = col_double(),
  Petal.Width = col_double(),
  Species = col_character()
)
```

In [10]:

```
library(e1071)
```

In [11]:

```
library(GGally)
```

In [12]:

```
library(ggplot2)
```

In [13]:

```
str(iris)
```

```
'data.frame':	150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

In [14]:

```
head(iris,5)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |

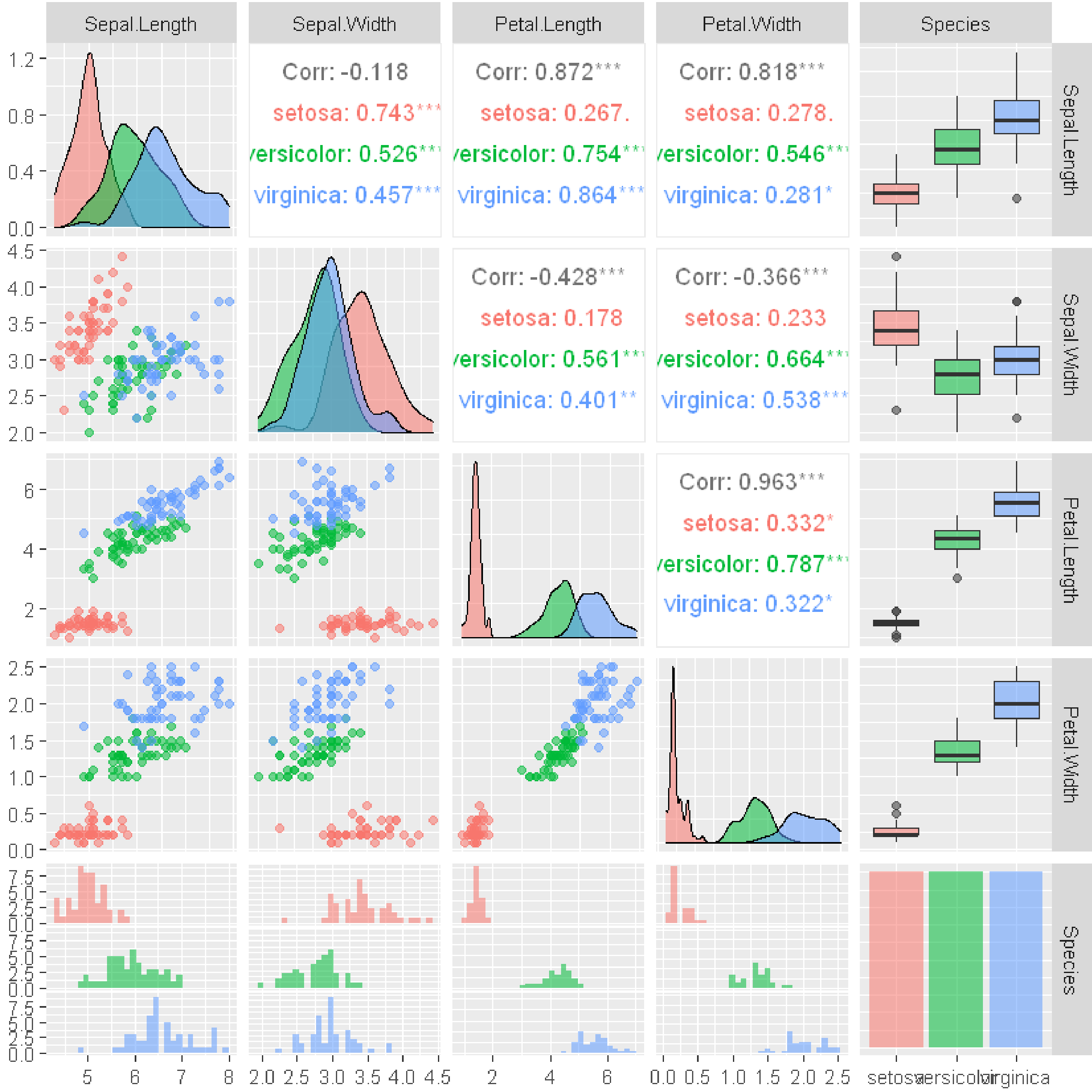
In [15]:

```
# Create SVM Model

#RADIAL

svm_model <- svm(Species ~ ., data=iris,
                kernel="radial") #linear/polynomial/sigmoid
```

In [16]:

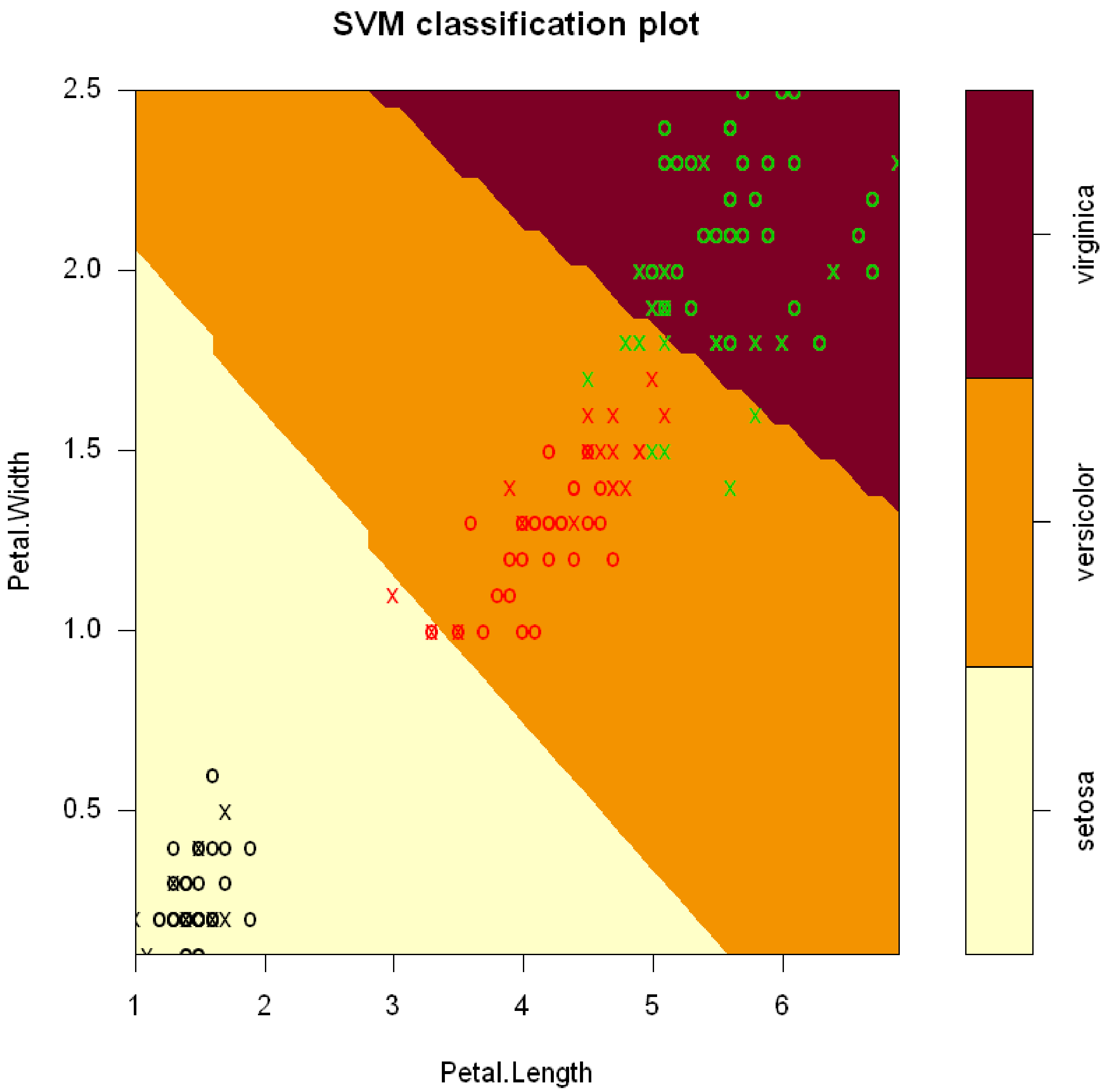
```
ggpairs(iris, ggplot2::aes(colour = Species, alpha = 0.4))
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.




In [17]:

```
plot(svm_model, data=iris,
     Petal.Width~Petal.Length,
     slice = list(Sepal.Width=3, Sepal.Length=4)

     )
```




In [18]:

```
#Predict each Species
#Confusion matrix and missclasscation Error

pred = predict(svm_model,iris)
tab = table(Predicted=pred, Actual = iris$Species)
tab
```

```
            Actual
Predicted    setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48
```

In [19]:

```
1-sum(diag(tab)/sum(tab)) #Missclasification error
```

0.0266666666666666

In [27]:

```
sum(diag(tab)/sum(tab))  #Accuracy
```

0.966666666666667

In [22]:

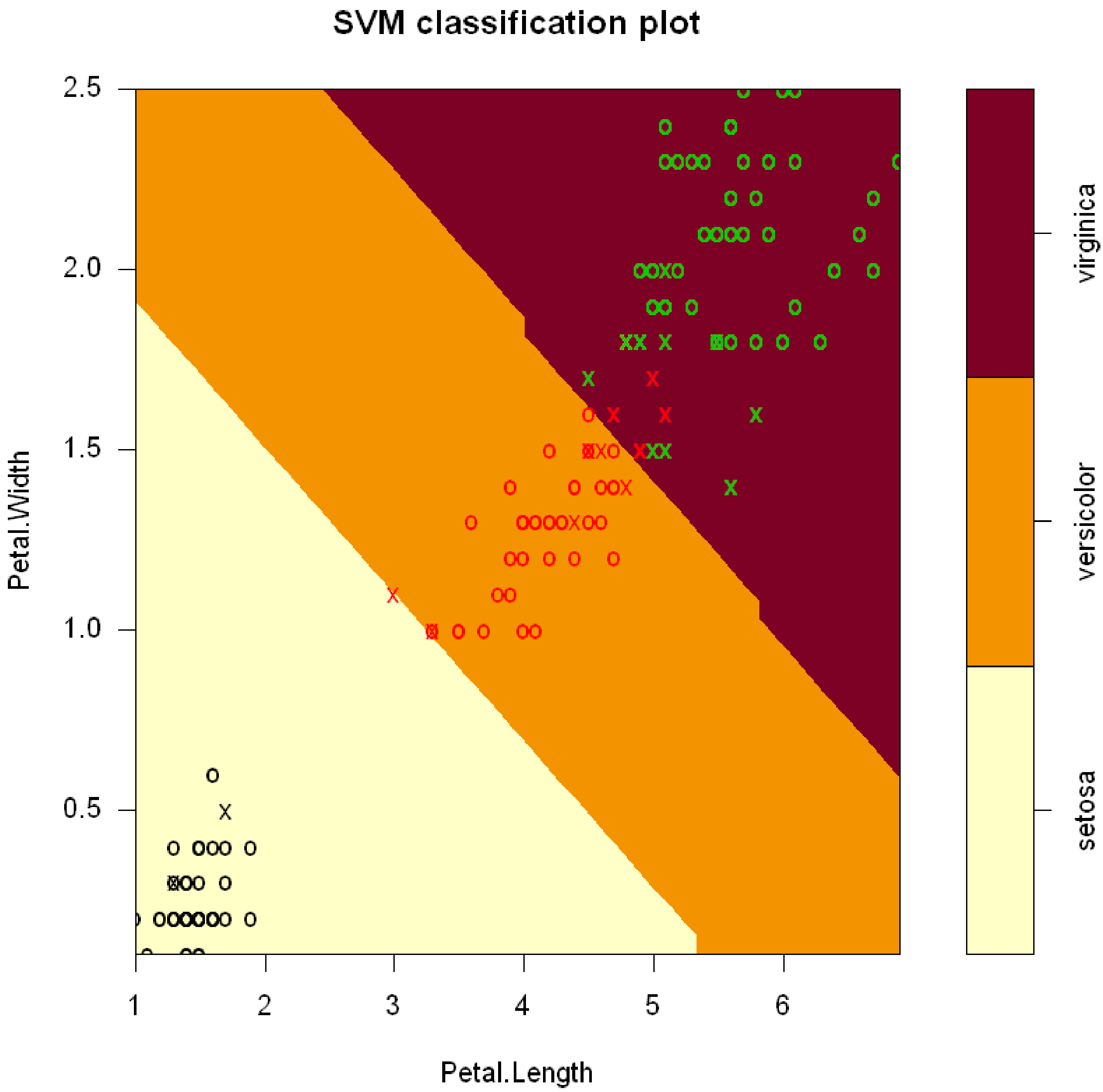
```
#LINEAR
```

In [23]:

```
svm_model <- svm(Species ~ ., data=iris,
            kernel="linear") #linear/polynomial/sigmoid
```

In [24]:

```
plot(svm_model, data=iris,
    Petal.Width~Petal.Length,
    slice = list(Sepal.Width=3, Sepal.Length=4)

    )
```




In [25]:

```
#Predict each Species
#Confusion matrix and missclassification Error and Accuracy

pred = predict(svm_model,iris)
tab = table(Predicted=pred, Actual = iris$Species)
tab
```

```
          Actual
Predicted    setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         46         1
  virginica       0          4        49
```

In [26]:

```
1-sum(diag(tab)/sum(tab))
```

0.0333333333333334

In [28]:

```
sum(diag(tab)/sum(tab))
```

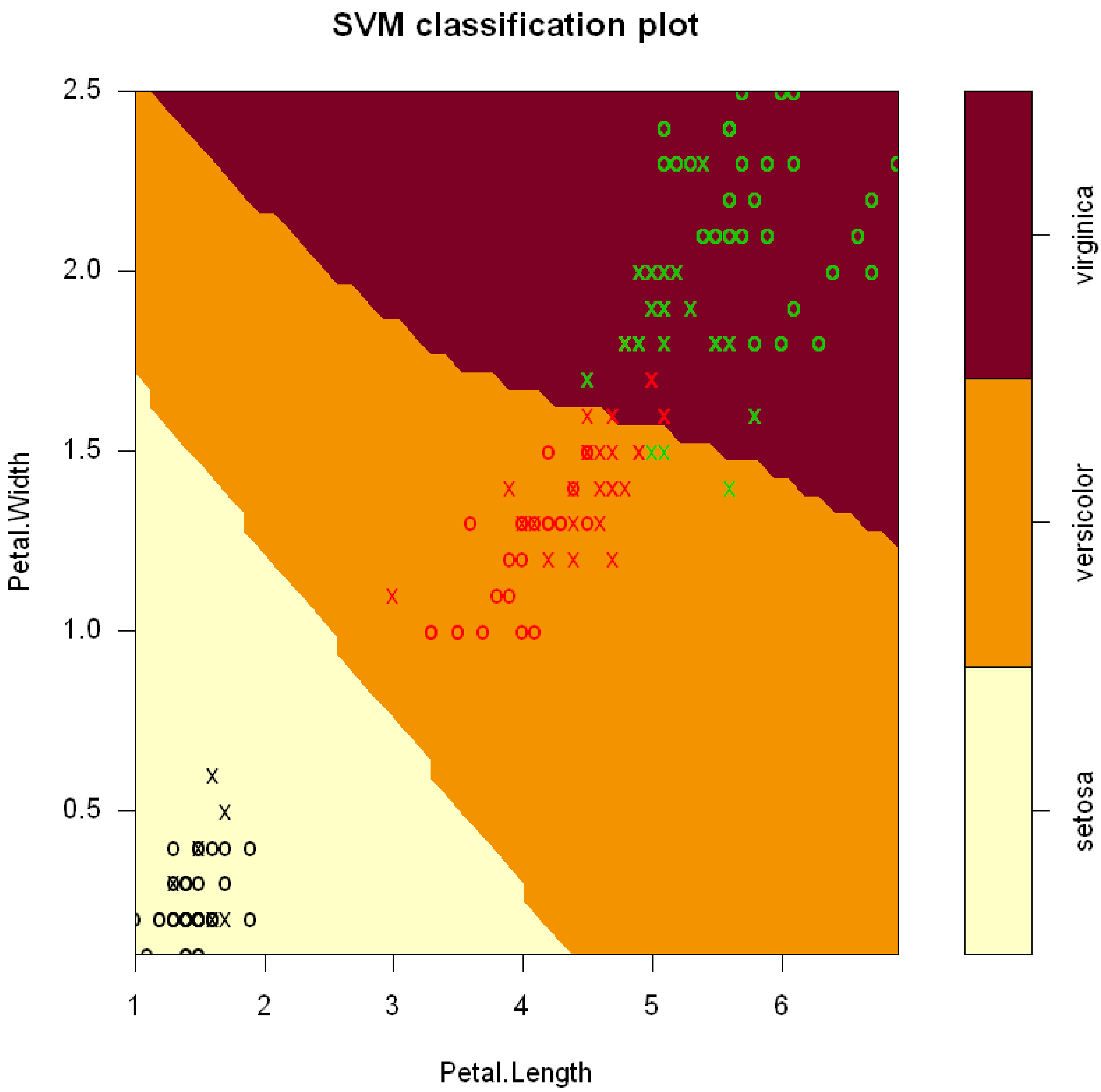
0.966666666666667

In [29]:

```
#POLYNOMIAL

svm_model <- svm(Species ~ ., data=iris,
              kernel="poly") #linear/polynomial/sigmoid
```

In [30]:

```
plot(svm_model, data=iris,
    Petal.Width~Petal.Length,
    slice = list(Sepal.Width=3, Sepal.Length=4)

    )
```




In [31]:

```
#Predict each Species
#Confusion matrix and missclassification Error and Accuracy

pred = predict(svm_model,iris)
tab = table(Predicted=pred, Actual = iris$Species)
tab
```

| | Actual | | |
|---|---|---|---|
| Predicted | setosa | versicolor | virginica |
| setosa | 50 | 0 | 0 |
| versicolor | 0 | 50 | 7 |
| virginica | 0 | 0 | 43 |

In [32]:

```
1-sum(diag(tab)/sum(tab))
```

0.0466666666666666

In [33]:

```
sum(diag(tab)/sum(tab))
```

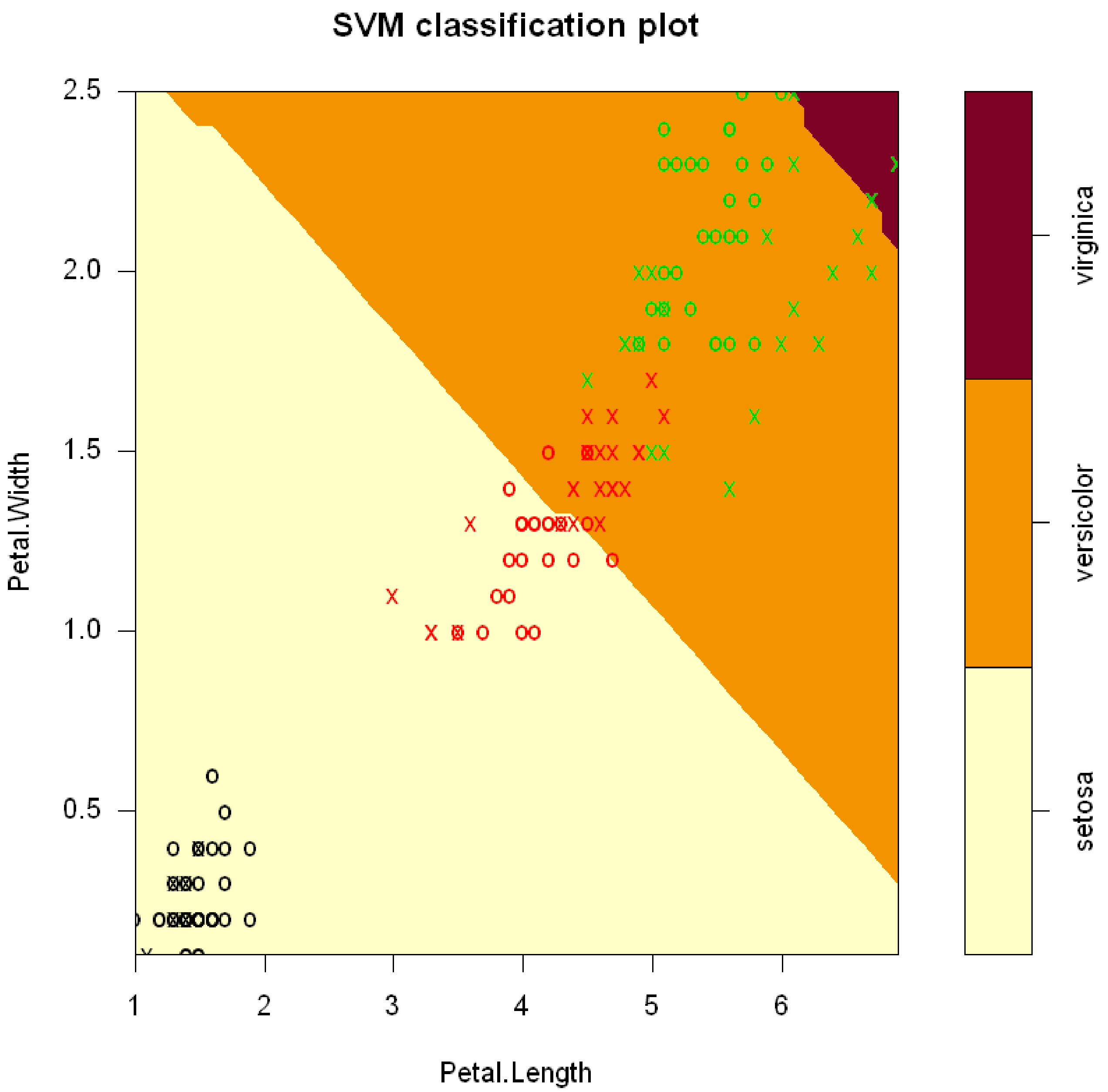
0.953333333333333

In [34]:

```
#SIGMOID

svm_model <- svm(Species ~ ., data=iris,
                kernel="sigmoid") #linear/polynomial/sigmoid
```

In [35]:

```
plot(svm_model, data=iris,
    Petal.Width~Petal.Length,
    slice = list(Sepal.Width=3, Sepal.Length=4)

    )
```




In [36]:

```
#Predict each Species
#Confusion matrix and missclassification Error and Accuracy

pred = predict(svm_model,iris)
tab = table(Predicted=pred, Actual = iris$Species)
tab
```

```
            Actual
Predicted    setosa versicolor virginica
  setosa         49          0         0
  versicolor      1         41         7
  virginica       0          9        43
```

In [37]:

```
1-sum(diag(tab)/sum(tab))
```

0.113333333333333

In [38]:

```
sum(diag(tab)/sum(tab))
```

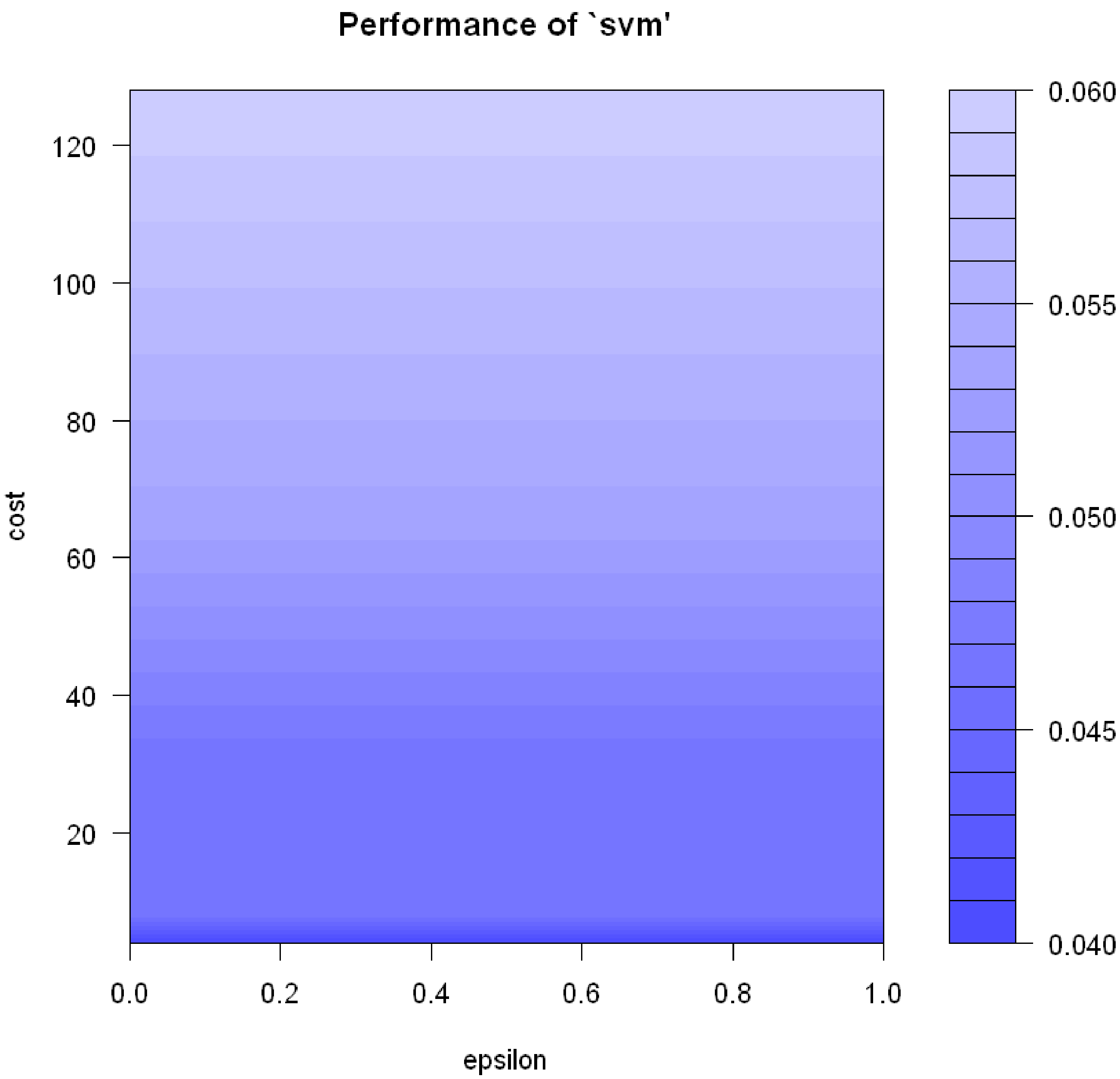
0.886666666666667

In [39]:

```
#Parameter Tunning
```

In [40]:

```
set.seed(123)
tmodel=tune(svm,Species~., data=iris,
    ranges=list(epsilon= seq(0,1,0.1), cost = 2^(2:7)))
plot(tmodel)
```




In [41]:

```
summary(tmodel)
```

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 epsilon cost
       0    4

- best performance: 0.04

- Detailed performance results:
   epsilon cost      error dispersion
1      0.0    4 0.04000000 0.04661373
2      0.1    4 0.04000000 0.04661373
3      0.2    4 0.04000000 0.04661373
4      0.3    4 0.04000000 0.04661373
5      0.4    4 0.04000000 0.04661373
6      0.5    4 0.04000000 0.04661373
7      0.6    4 0.04000000 0.04661373
8      0.7    4 0.04000000 0.04661373
9      0.8    4 0.04000000 0.04661373
10     0.9    4 0.04000000 0.04661373
11     1.0    4 0.04000000 0.04661373
12     0.0    8 0.04666667 0.06324555
13     0.1    8 0.04666667 0.06324555
14     0.2    8 0.04666667 0.06324555
15     0.3    8 0.04666667 0.06324555
16     0.4    8 0.04666667 0.06324555
17     0.5    8 0.04666667 0.06324555
18     0.6    8 0.04666667 0.06324555
19     0.7    8 0.04666667 0.06324555
20     0.8    8 0.04666667 0.06324555
21     0.9    8 0.04666667 0.06324555
22     1.0    8 0.04666667 0.06324555
23     0.0   16 0.04666667 0.04499657
24     0.1   16 0.04666667 0.04499657
25     0.2   16 0.04666667 0.04499657
26     0.3   16 0.04666667 0.04499657
27     0.4   16 0.04666667 0.04499657
28     0.5   16 0.04666667 0.04499657
29     0.6   16 0.04666667 0.04499657
30     0.7   16 0.04666667 0.04499657
31     0.8   16 0.04666667 0.04499657
32     0.9   16 0.04666667 0.04499657
33     1.0   16 0.04666667 0.04499657
34     0.0   32 0.04666667 0.04499657
35     0.1   32 0.04666667 0.04499657
36     0.2   32 0.04666667 0.04499657
37     0.3   32 0.04666667 0.04499657
38     0.4   32 0.04666667 0.04499657
39     0.5   32 0.04666667 0.04499657
40     0.6   32 0.04666667 0.04499657
41     0.7   32 0.04666667 0.04499657
42     0.8   32 0.04666667 0.04499657
43     0.9   32 0.04666667 0.04499657
44     1.0   32 0.04666667 0.04499657
45     0.0   64 0.05333333 0.06126244
46     0.1   64 0.05333333 0.06126244
47     0.2   64 0.05333333 0.06126244
48     0.3   64 0.05333333 0.06126244
49     0.4   64 0.05333333 0.06126244
50     0.5   64 0.05333333 0.06126244
51     0.6   64 0.05333333 0.06126244
52     0.7   64 0.05333333 0.06126244
53     0.8   64 0.05333333 0.06126244
54     0.9   64 0.05333333 0.06126244
55     1.0   64 0.05333333 0.06126244
56     0.0  128 0.06000000 0.05837300
57     0.1  128 0.06000000 0.05837300
58     0.2  128 0.06000000 0.05837300
59     0.3  128 0.06000000 0.05837300
60     0.4  128 0.06000000 0.05837300
61     0.5  128 0.06000000 0.05837300
62     0.6  128 0.06000000 0.05837300
63     0.7  128 0.06000000 0.05837300
64     0.8  128 0.06000000 0.05837300
65     0.9  128 0.06000000 0.05837300
66     1.0  128 0.06000000 0.05837300
```

In [42]:

```
mymodel=tmodel$best.model
summary(mymodel)
```

```
Call:
best.tune(method = svm, train.x = Species ~ ., data = iris, ranges = list(epsilon = seq(0, 
    1, 0.1), cost = 2^(2:7)))


Parameters:
   SVM-Type:  C-classification 
 SVM-Kernel:  radial 
       cost:  4 

Number of Support Vectors:  37

 ( 6 17 14 )


Number of Classes:  3 

Levels: 
 setosa versicolor virginica
```

In [43]:

```
# Best model
```

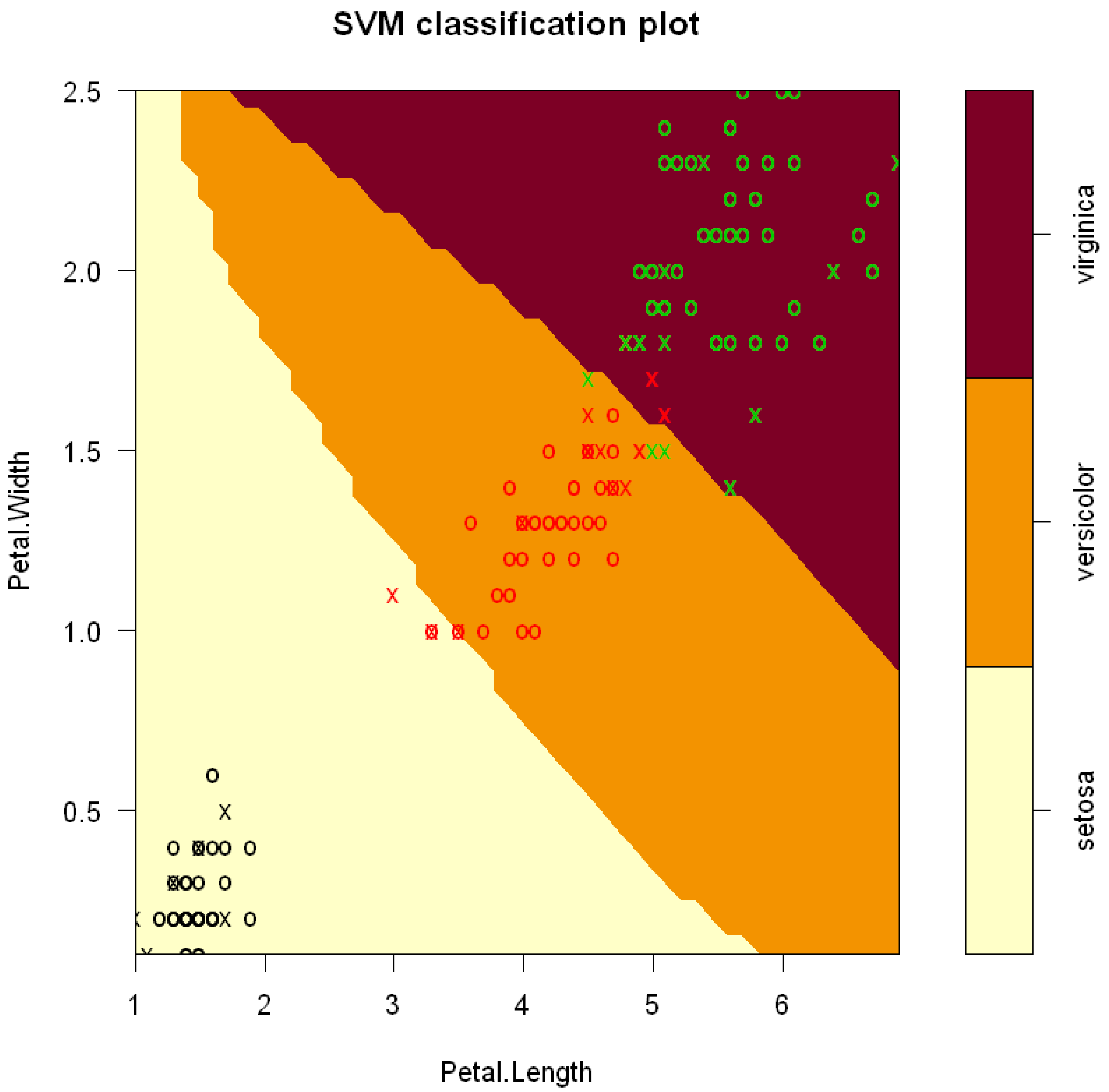
In [44]:

```
mymodel=tmodel$best.model
summary(mymodel)
```


```
Call:
best.tune(method = svm, train.x = Species ~ ., data = iris, ranges = list(epsilon = seq(0, 
    1, 0.1), cost = 2^(2:7)))


Parameters:
   SVM-Type:  C-classification 
 SVM-Kernel:  radial 
       cost:  4 

Number of Support Vectors:  37

 ( 6 17 14 )


Number of Classes:  3 

Levels: 
 setosa versicolor virginica
```


In [51]:

```
# RADIAL model was selected as the best

plot(mymodel, data=iris,
     Petal.Width~Petal.Length,
     slice = list(Sepal.Width=3, Sepal.Length=4)
)
```




In [52]:

*# Confusion matrix and missclassification rate and accuracy using best parameter*

In [53]:

```
pred1 = predict(mymodel,iris)
tab1 = table(Predicted=pred1, Actual = iris$Species)
tab1
```

```
           Actual
Predicted    setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         0
  virginica       0          2        50
```

In [54]:

```
1-sum(diag(tab1)/sum(tab1))
```

0.0133333333333334

In [55]:

```
sum(diag(tab1)/sum(tab1))
```

0.986666666666667

In [ ]: