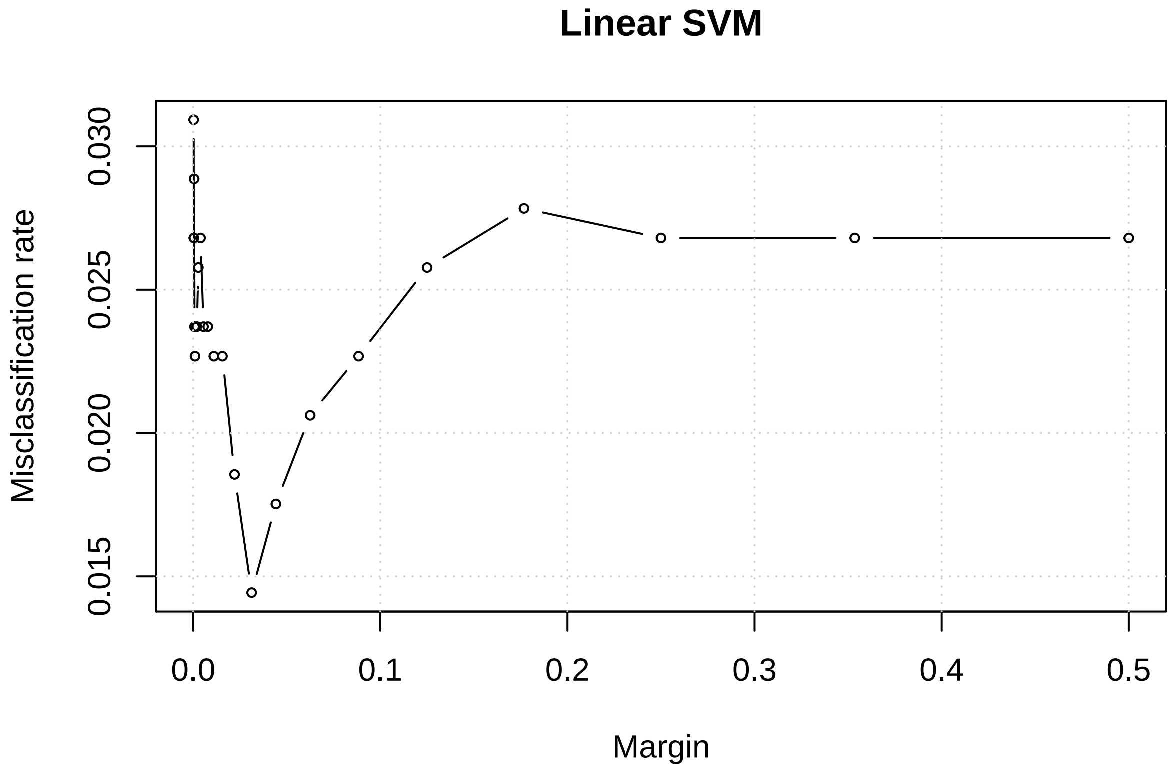
```r
rm(list = ls())
cat("\014")
```

```r
library(e1071)
library(ggplot2)
```

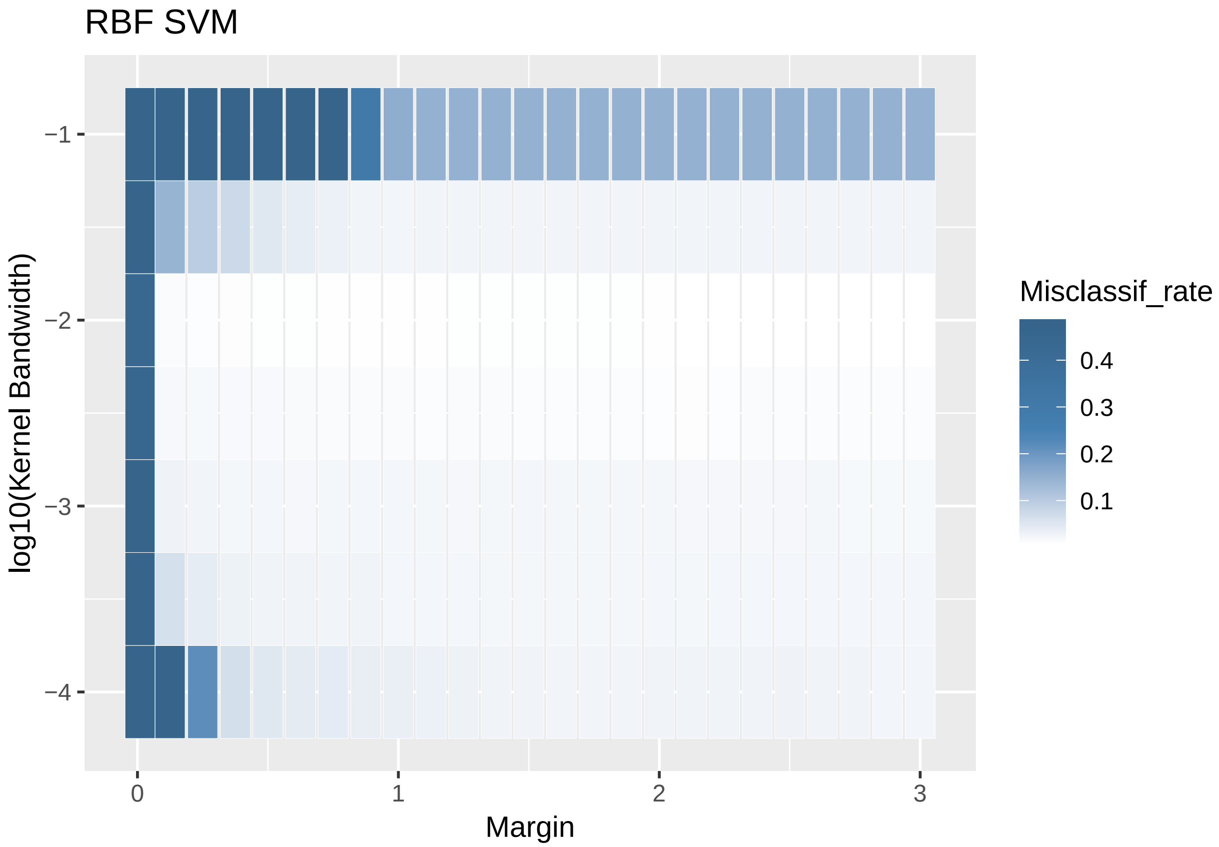
```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from 
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```r
set.seed(10)

z_main<-function(gamma,cost,x_train,y_train,kernel){
  set.seed(10)
  nfolds<-10
  fold<-sample(rep(1:nfolds,nrow(x_train)/nfolds))
  m_error<-matrix(NA,length(cost)*length(gamma),3)
  cont_cont<-1
  for (cont_g in 1:length(gamma)){
    for (cont_c in 1:length(cost)){
      err_train<-0
      for (cont_f in 1:nfolds){
        i_test<-which(fold %in% cont_f)
        svm_fit<-svm(x_train[-i_test,],y_train[-i_test],type="C",scale=F,
                     kernel=kernel,cost=cost[cont_c],gamma=gamma[cont_g])
        svm_pred<-predict(svm_fit,x_train[i_test,])
        err_train<-err_train+(sum(svm_pred!=y_train[i_test])/length(i_test))
      }
      m_error[cont_cont,]<-c(err_train/nfolds,cost[cont_c],gamma[cont_g])
      cont_cont<-cont_cont+1
    }
  }
  return(m_error)
}


x1<-read.csv("train.5.txt",header=FALSE)
x2<-read.csv("train.6.txt",header=FALSE)

x<-rbind(x1,x2)
y<-c(rep(1,nrow(x1)),rep(-1,nrow(x2)))

test_index <- sort(sample(1:nrow(x),nrow(x)/5))
x_test<-x[test_index,]
x_train<-x[-test_index,]
y_test<-as.factor(y[test_index])
y_train<-as.factor(y[-test_index])

gamma<-1
cost<-2^(seq(-12,-1,0.5))
kernel<-'linear'
z_error_linear<-z_main(gamma,cost,x_train,y_train,kernel)
```

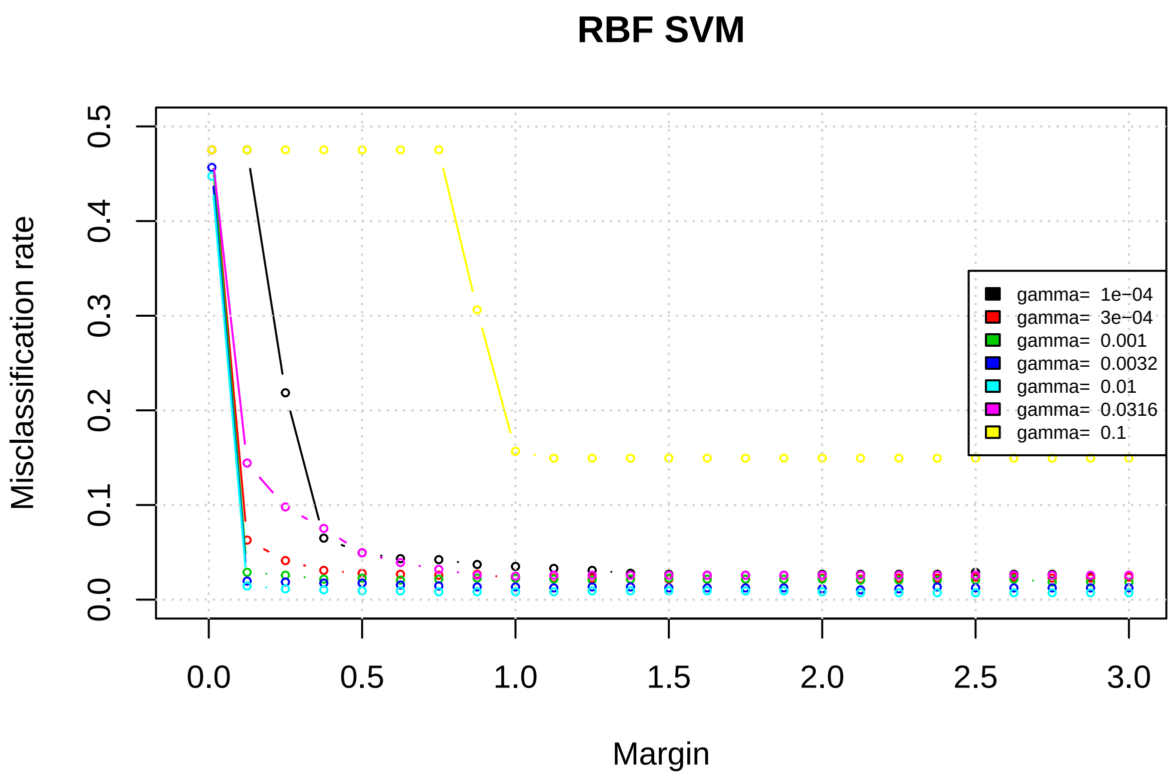
```
plot(z_error_linear[,2],z_error_linear[,1],type='b',
     xlab="Margin",ylab="Misclassification rate",
     main="Linear SVM",cex=0.6);grid(NULL)
```



```
gamma<-10^seq(-4,-1,0.5)
cost<-c(0.01,seq(0.125,3,0.125))
kernel<-'radial'
z_error_RBF<-z_main(gamma,cost,x_train,y_train,kernel)

Misclassif_rate<-z_error_RBF[,1]

ggplot()+
    geom_tile(mapping=aes(x=z_error_RBF[,2],y=log10(z_error_RBF[,3]),fill=Misclassif_rate),colour="whit
    scale_fill_gradient2(low = ("white"), mid = "steelblue",
                        high = ("steelblue4"), midpoint = 0.5*(min(Misclassif_rate)+max(Misclassif_rate
                        space = "Lab",na.value = "grey50", guide = "colourbar")+
  labs(x="Margin",y="log10(Kernel Bandwidth)",z="")+
    ggtitle("RBF SVM")
```



```
plot(0,0,type='n',xlab="Margin",ylab="Misclassification rate",
     main="RBF SVM",xlim=c(-0.05,3),ylim=c(0,0.5))
for (cont_gamma in 1:length(gamma)){
  this_gamma<-gamma[cont_gamma]
  uh_x<-z_error_RBF[which(z_error_RBF[,3]==this_gamma),2]
  uh_y<-z_error_RBF[which(z_error_RBF[,3]==this_gamma),1]
  lines(uh_x,uh_y,type='b',col=cont_gamma,main="RBF SVM",cex=0.5)
  grid(NULL)
}
legend("right",legend=paste("gamma= ",round(gamma,4)),fill=factor(gamma),cex=0.6)
```



```
min_gamma_linear<-z_error_linear[which.min(z_error_linear[,1]),3]
min_cost_linear<-z_error_linear[which.min(z_error_linear[,1]),2]
kernel<-"linear"
error_linear<-z_main(min_gamma_linear,min_cost_linear,x_test,y_test,kernel)
"Linear Kernel"
```

```
## [1] "Linear Kernel"
```

```
c(paste("Gamma: ",round(error_linear[3],4)))
```

```
## [1] "Gamma:  1"
```

```
c(paste("Margin: ",round(error_linear[2],4)))
```

```
## [1] "Margin:  0.0312"
```

```
c(paste("Misclassification rate: ",round(error_linear[1],4)))
```

```
## [1] "Misclassification rate:  0.0333"
```

```
min_gamma_RBF<-z_error_RBF[which.min(z_error_RBF[,1]),3]
min_cost_RBF<-z_error_RBF[which.min(z_error_RBF[,1]),2]
kernel<-"radial"
error_RBF<-z_main(min_gamma_RBF,min_cost_RBF,x_test,y_test,kernel)
"Radial Kernel"
```

```
## [1] "Radial Kernel"
```

```
c(paste("Gamma: ",round(error_RBF[3],4)))
```

```
## [1] "Gamma:  0.01"
```

```
c(paste("Margin: ",round(error_RBF[2],4)))
```

```
## [1] "Margin:  2.125"
```

```
c(paste("Misclassification rate: ",round(error_RBF[1],4)))
```

```
## [1] "Misclassification rate:  0.0167"
```