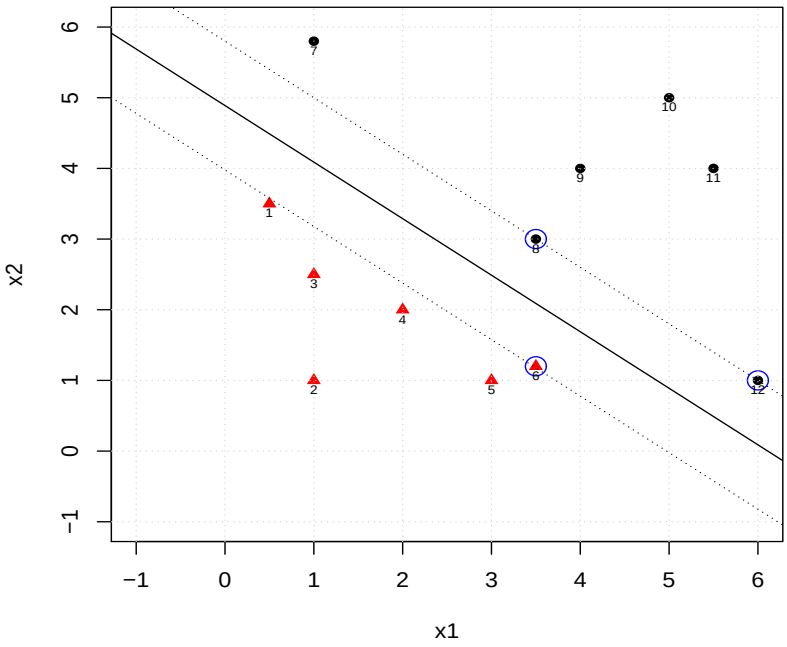
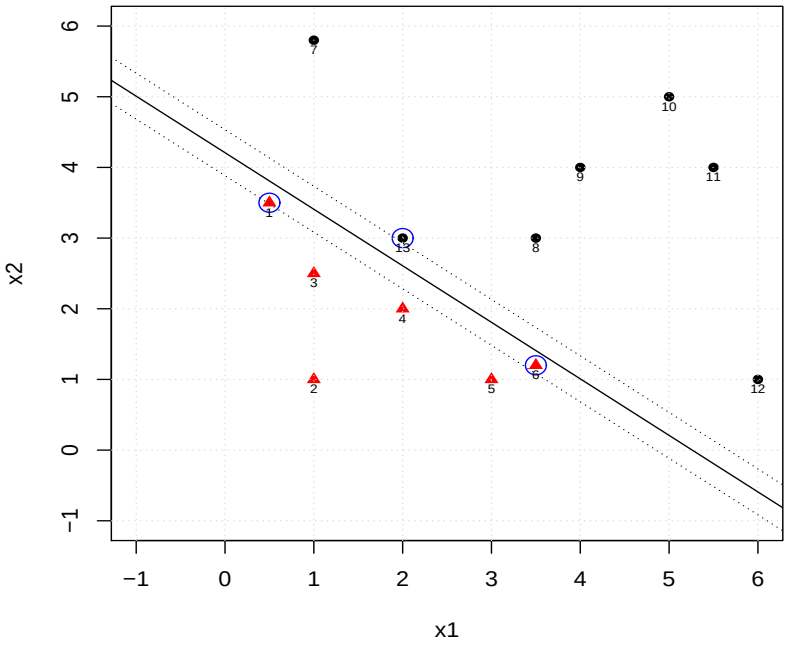
```
library(e1071)    # svm()
library(kernlab)  # ksvm()  another svm library

# create data set
x1  <- c(.5,1,1,2,3,3.5,     1,3.5,4,5,5.5,6)
x2  <- c(3.5,1,2.5,2,1,1.2,  5.8,3,4,5,4,1)
yval  = c(rep(+1,6),rep(-1,6))

y = factor(yval)
d1 <- data.frame(x1, x2, y)

#     x1  x2    y
# 1  0.5 3.5    1
# 2  1.0 1.0    1
# 3  1.0 2.5    1
# 4  2.0 2.0    1
# 5  3.0 1.0    1
# 6  3.5 1.2    1
# 7  1.0 5.8   -1
# 8  3.5 3.0   -1
# 9  4.0 4.0   -1
# 10 5.0 5.0   -1
# 11 5.5 4.0   -1
# 12 6.0 1.0   -1

# use y for symbol
color = 0.5*(yval+3)
symbol = color + 15
# data is separable
plot(d1[,-3],col=color, pch=symbol, xlim=c(-1,6), ylim=c(-1,6))
text(d1[,-3],labels=rownames(d1),offset=0.25,pos=1,cex=0.6)
grid()
```

```
# modeling
m1 = svm(y ~ .,d1,type='C-classification',kernel='linear',scale=F)
str(m1)
# m1$index row numbers of support vectors
# m1$SV    x-values of support vectors
# m1$coefs  one for each s.vector sum?
# m1$rho   negative intercept
# m1$decision.values  distance to boundary

m1$index
# [1]  6  8 12
m1$SV
#    x1  x2
#6  3.5 1.2
#8  3.5 3.0
#12 6.0 1.0
m1$coefs
#           [,1]
#[1,]  1.0000000
#[2,] -0.6487805
#[3,] -0.3512195

m1$rho
# -5.365853

b  = -m1$rho

w  = t(m1$coefs) %*% m1$SV
#             x1        x2
#[1,] -0.8780489 -1.097561

# intercept & slope
a = -b/w[1,2]             # 4.888889
slope = -w[1,1]/w[1,2]    # -0.8000001

# plotting
plot(d1[,-3],col=color, pch=symbol, xlim=c(-1,6), ylim=c(-1,6))
points(d1[m1$index,c(1,2)],col="blue",cex=2)  # support vectors
text(d1[,-3],labels=rownames(d1),offset=0.25,pos=1,cex=0.6)
grid()

abline(a,slope,lty=1)
upper = (-b-1)/w[1,2]
abline(upper,slope,lty=3)
lower = (-b+1)/w[1,2]
abline(lower,slope,lty=3)
```

```
# add point (2,3) row 13
#=========================================================

row13 = c(2,3,-1)
d2 = rbind(d1,row13)

# use y for symbol
color = 0.5*(yval+3)
color[13] = 4
symbol = color + 15
# data is separable
plot(d2[,-3],col=color, pch=symbol, xlim=c(-1,6), ylim=c(-1,6))
text(d2[,-3],labels=rownames(d2),offset=0.25,pos=1,cex=0.6)
grid()

# small margin, cost = 10
m2 = svm(y ~ .,d2,kernel='linear',scale=F,cost=100)
m2$index    # [1]  1  6 13   rows of support vectors

w  = t(m2$coefs) %*% m2$SV
b  = m2$rho

# plotting
plot(d2[,-3],col=color, pch=symbol, xlim=c(-1,6), ylim=c(-1,6))
points(d2[m2$index,c(1,2)],col="blue",cex=2)  # support vectors
text(d2[,-3],labels=rownames(d2),offset=0.25,pos=1,cex=0.6)
grid()

abline(a=b/w[1,2],     slope,lty=1)
abline(a=(b-1)/w[1,2], slope,lty=3)
abline(a=(b+1)/w[1,2], slope,lty=3)

# prediction
predict(m2,d1)
# 1  2  3  4  5  6  7  8  9 10 11 12
# 1  1  1  1  1  1 -1 -1 -1 -1 -1 -1
#Levels: -1 1

newval = data.frame(x1=2,x2=1)
predict(m2,newval)
#1
#1
#Levels: -1 1
```


x2
x1
−1
0
1
2
3
4
5
6
7
10
9
11
1
8
3
4
2
5
12


x2
x1
−1
0
1
2
3
4
5
6
7
10
9
11
1
13
8
3
4
2
5
12