```r
### Data ###################################################
##wine data
library(caret)
white.wine <- ("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv")
dat.white <- read.table(white.wine, header = T, sep = ";")
str(dat.white)
colnames(dat.white)
red.wine <- ("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv")
dat.red <- read.table(red.wine, header = T, sep = ";")
str(dat.red)
colnames(dat.red)
anyNA(dat.white);anyNA(dat.red)
set.seed(123)
dat.w <- createDataPartition( dat.white$quality , p = 0.75 , list = F )
train.w <- dat.white[ dat.w , ]
test.w <- dat.white[ -dat.w , ]
set.seed(123)
dat.r <- createDataPartition( dat.red$quality , p = 0.75 , list = F )
train.r <- dat.red[ dat.r , ]
test.r <- dat.red[ -dat.r , ]

###SVR.回帰
##ksvm
library(kernlab)
svm.w.d <- ksvm(quality ~ . ,data = train.w ,cross = 5)
print(svm.w.d)
#
#rbfdot(sigma = 1),polydot(degree = 1, scale = 1, offset = 1)
#tanhdot(scale = 1, offset = 1),vanilladot()
#laplacedot(sigma = 1),besseldot(sigma = 1, order = 1, degree = 1)
#anovadot(sigma = 1, degree = 1),splinedot()
svm.w <- ksvm(quality ~ . ,data = train.w ,
              type = "eps-bsvr" , #regression,C-svc;classification
              kernel = "rbfdot",
              kpar = list(sigma=0.2),
              C = 1, # margin
              cross = 3)
print(svm.w) #
##svm
library(e1071)
tun <- tune(svm, quality ~ . ,
            data = train.w ,
            ranges = list(gamma = 2^(-5:-2), #0.031-0.25
                          cost = 2^(-1:2)),  #0.5-4
            tunecontrol = tune.control(sampling = "fix"))
tun
svm.w2 <- svm(quality ~ . ,
              data = train.w ,gamma = 0.25 , cost = 2 )
summary(svm.w2)
##Fit
library(dplyr)
d.pre <- predict(svm.w , test.w ) #ksvm
round(cor(data.frame(test.w$quality,d.pre)),3)#観測変数と予測値のcor
d.pre2 <- predict(svm.w2 , test.w ) #svm
round(cor(data.frame(test.w$quality,d.pre2)),3)#観測変数と予測値のcor
sqrt( sum((test.w$quality - d.pre)^2) / 13483 )  #ksvm
sqrt( sum((test.w$quality - d.pre2)^2) / 13483 ) #svm

##SVM.分類
###data
library(kernlab);data(spam)
library(caret)
anyNA(spam)
set.seed(123)
dat.s <- createDataPartition( spam$type , p = 0.7 , list = F )
train.spam <- spam[ dat.s ,]
test.spam  <- spam[-dat.s ,]

#tune
library(e1071)
set.seed(123)
svm.d <- svm(type ~ ., data = train.spam )
print(svm.d)
library(e1071)
#gammarange = 10^(-2:-0.8)
#costrange = 2^(-0:3)
set.seed(123)
t <- tune.svm(type ~ ., data = train.spam ,
              gamma=seq(0.01,0.15,by=0.01), cost=seq(1,5,by=1),
              tunecontrol = tune.control(sampling="cross", cross = 3))
summary(t)
plot(t)

##SVM()
svm.tu <- svm(type ~ ., data = train.spam ,
              method="C-classification",kernel="radial")
svm.tu
estt <- xtabs( ~ type + svm.tu$fitted , data = train.spam )
estt
round((estt[1,1]+estt[2,2])/length(train.spam$type),3)*100
```

```r

#線形のSV分類
svm.spam <- ksvm(type ~ ., data = train.spam,
                 type="C-svc",        #分類
                 kernel="vanilladot")#線形データへ適用
svm.spam
s <- fitted(svm.spam)
ss <- xtabs( ~ type + s , data = train.spam )
ss
round((ss[1,1]+ss[2,2])/length(train.spam$type),4)*100

#非線形のSV分類:polydot(多項式カーネル)
svm.spam.2 <- ksvm(type ~ ., data = train.spam,
                   type="C-svc",
                   kernel="polydot",kpar = list(degree=2),#2乗の多項式
                   C=4,cross=5,scaled = T)
svm.spam.2
s2 <- fitted(svm.spam.2)
ss2 <- xtabs( ~ type + s2 , data = train.spam )
ss2
round((ss2[1,1]+ss2[2,2])/length(train.spam$type),4)*100

#非線形のSV分類:rbfdot(ガウスカーネル)
svm.spam.3 <- ksvm(type ~ ., data = train.spam,
                   type="C-svc",
                   kernel="rbfdot",kpar = list(sigma=0.028),#
                   C=4,cross=5,scaled = T)
svm.spam.3
s3 <- fitted(svm.spam.3)
ss3 <- xtabs( ~ type + s3 , data = train.spam )
ss3
round((ss3[1,1]+ss3[2,2])/length(train.spam$type),4)*100

#
library(kernlab)
fit.svm <- ksvm(type ~ ., data = train.spam)#kernelとkparは自動設定
fit.svm
fit.svm2 <- ksvm(type~., data=train.spam,kernel="rbfdot",
                 kpar=list(sigma=0.05),C=10,cross=5,scaled = T)
fit.svm2

###Test.data
test.pre <- predict(svm.spam.3 , test.spam) #svm.spam.3のtest.data fit
table(test.pre , test.spam$type)
library(caret)
confusionMatrix(test.pre , test.spam$type )

###多値分類
library(kernlab)
library(mlbench)
test.su <- ("https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits//optdigits.tes")
test.su <- read.table(test.su  , sep = ",")
train.su <- ("https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits//optdigits.tra")
train.su <- read.table(train.su , sep = ",")
str(train.su)
colnames(train.su)
str(test.su)
colnames(test.su)
table(test.su$V65)
table(train.su$V65)
#y:factor変換 confusion matrix出力のため(factor変換:解析には不要)
train.su$V65 <- factor(train.su$V65)
str(train.su)
test.su$V65 <- factor(test.su$V65)
str(test.su)
colnames(train.su)
colnames(test.su)
table(test.su$V65)
table(train.su$V65)
names(test.su)[65]<-c("class")
names(train.su)[65]<-c("class")
##ksvm
##vanilladot
su.out.l <- ksvm( class~., data = train.su ,
                  scaled = F , kernel = "vanilladot" )
su.out.l
prd.test.l <- predict( su.out.l , test.su)
table.l <- table(prd.test.l , test.su$class)
round(sum(diag(table.l)) /sum(table.l),4)*100
library(caret)
confusionMatrix(prd.test.l , test.su$class)

##polydot
su.out.p <- ksvm( class~., data = train.su  ,
                  scaled = F , kernel = "polydot" )
su.out.p
prd.test.p <- predict( su.out.p , test.su)
table.p <- table(prd.test.p , test.su$class)
round(sum(diag(table.p)) /sum(table.p),4)*100
library(caret)
```

```r
confusionMatrix(prd.test.p , test.su$class )

##rbfdot
su.out.r <- ksvm( class~., data = train.su  ,
                  scaled = F , kernel = "rbfdot" )
su.out.r
prd.test.r <- predict( su.out.r , test.su)
table.r <- table(prd.test.r , test.su$class)
round(sum(diag(table.r)) /sum(table.r),4)*100
library(caret)
confusionMatrix(prd.test.r , test.su$class )

##精度比較
round(sum(diag(table.l)) /sum(table.l),4)*100 #線形
round(sum(diag(table.p)) /sum(table.p),4)*100 #多項式
round(sum(diag(table.r)) /sum(table.r),4)*100 #ガウジアン

```