```
#################################
##第十回「k-近傍法」
#################################

library(FNN)
library(kknn)

#################
## k-NN密度推定コード

#Z is query point matrix, X is data.
knn.density <- function(Z,X,k=10){
    dd <- dim(X)
    n <- dd[1]
    d <- dd[2]
    tmp <- matrix(0,length(Z[,1]),d)
    tmp <- knnx.dist(X,Z,k=k)
    res <- k/n/(pi^(d/2)/gamma(d/2+1)*tmp[,k]^d)
}

#################################
## 世田谷区の中古マンション価格データ
## 価格 (price) と面積 (area) の分布を求める.
## 二次元データの密度関数

#データの読み込み
x <- read.csv("setagaya_manshion.csv",header=T)
ind = grep("１",x[[4]])
Y = 0*c(1:length(x[[1]]))
Y[ind] = 1
Y = factor(Y)
sman = data.frame(price=x[[1]],walk=x[[3]],area=x[[5]],str=ifelse(x[[6]]=="ＲＣ"
,1,0),kenpei=x[[7]],youseki=x[[8]],tikunen=x[[9]],kyuko=x[[10]]) #構造はダミー変

plot(sman$price,sman$area,xlab="price",ylab="area")

#訓練データの作成
train = matrix(0,length(Y),2)
train[,1] = sman$price/sd(sman$price) #データのスケールを揃える
train[,2] = sman$area/sd(sman$area) #データのスケールを揃える
colnames(train) <- c("price","area")
sman.sd = c(sd(sman$price),sd(sman$area))

#プロット用グリッドデータの作成
dif = max(train[,1])-min(train[,1])
x1 = seq(min(train[,1])-dif*0.05,max(train[,1])+dif*0.05,len=100)
dif = max(train[,2])-min(train[,2])
x2 = seq(min(train[,2])-dif*0.05,max(train[,2])+dif*0.05,len=100)
xgrid = expand.grid(x1,x2)
prob <- knn.density(xgrid,train,k=30)
prob_mat <- matrix(prob, length(x1), length(x2))

#ヒートマップのプロット
image(x = x1*sman.sd[1], y = x2*sman.sd[2], z = prob_mat,xlab = "price",
ylab = "area",main="k=30")
par(new=F)
points(train[,1]*sman.sd[1],train[,2]*sman.sd[2])
contour(x1*sman.sd[1],x2*sman.sd[2], prob_mat, add=TRUE)

#俯瞰図のプロット
persp(x = x1*sman.sd[1],
  y = x2*sman.sd[2],
  z = prob_mat,
  xlab = "price",
  ylab = "area",
  zlab = "estimated density",
  theta = 45, axes = TRUE, box = TRUE)

####判別################
#部屋数1とそれ以外を判別

res = knn(train, xgrid, Y, k = 10, prob=TRUE)
```

```
prob <- attr(res, "prob")
prob <- ifelse(res=="1", prob, 1-prob)

plot(train[,1]*sman.sd[1],train[,2]*sman.sd[2], col=ifelse(Y==1, "coral", "cornf
lowerblue"),main="",xlab="price",ylab="area")
contour(x = x1*sman.sd[1], y = x2*sman.sd[2], z = matrix(prob,length(x1),length(
x2)),levels=0.5, add=TRUE)
points(x = xgrid[,1]*sman.sd[1], y = xgrid[,2]*sman.sd[2], pch=".", cex=1.2, col
=ifelse(prob>=0.5, "coral", "cornflowerblue"))
box()


##############################################
# MNIST手書き文字認識
# k-NN判別

data.train_origin <- read.csv("train_data.csv",header=FALSE) #訓練データの読み込
み
data.test <- read.csv("test_data.csv",header=FALSE) #テストデータの読み込み

train_size <- dim(data.train_origin)
dimx <- train_size[2] - 1
dimy <- train_size[2]
res = knn(data.train_origin[,1:dimx], data.test[,1:dimx], data.train_origin[,dim
y], k = 10, prob=FALSE) #計算には数分かかる
clerr = mean(res!=data.test[,dimy]) # 0.0335, 96.65%の確率で正解.
table(res,data.test[,dimy])

# 間違った例の表示
show_digit <- function(arr784, col=gray(12:1/12), ...) {
  x <- expand.grid(1:28,1:28)
  image(matrix(arr784, nrow=28)[,28:1], col=col, ...)
}
misclss_ind <- sample(which((res!=data.test[,dimy])),3)
show_digit(as.numeric(data.test[misclss_ind[1],1:dimx]))
show_digit(as.numeric(data.test[misclss_ind[2],1:dimx]))
show_digit(as.numeric(data.test[misclss_ind[3],1:dimx]))

##############
## k-NNの図
library(ElemStatLearn)
require(class)

data(mixture.example)
x <- mixture.example$x
g <- mixture.example$y
xnew <- mixture.example$xnew
mod15 <- knn(x, xnew, g, k=15, prob=TRUE)
prob <- attr(mod15, "prob")
prob <- ifelse(mod15=="1", prob, 1-prob)
px1 <- mixture.example$px1
px2 <- mixture.example$px2
prob15 <- matrix(prob, length(px1), length(px2))

par(mar=rep(2,4))
contour(px1, px2, prob15, levels=0.5, labels="", xlab="", ylab="", main=
        "15-nearest neighbour", axes=FALSE)
points(x, col=ifelse(g==1, "coral", "cornflowerblue"))
gd <- expand.grid(x=px1, y=px2)
points(gd, pch=".", cex=1.2, col=ifelse(prob15>0.5, "coral", "cornflowerblue"))
box()
```