

```
#### KNN
setwd("D:/Analytics/R/BA/Sec D")
mower.df <- read.csv("RidingMowers.csv")
mower.df<-RidingMowers
set.seed(111)
train.index <- sample(row.names(mower.df), 0.6*dim(mower.df)[1])
valid.index <- setdiff(row.names(mower.df), train.index)
train.df <- mower.df[train.index, ]
valid.df <- mower.df[valid.index, ]
## new household
new.df <- data.frame(Income = 60, Lot_Size = 20)

## scatter plot
plot(Lot_Size ~ Income, data=train.df, pch=ifelse(train.df$Ownership=="owner", 1, 3))
text(train.df$Income, train.df$Lot_Size, rownames(train.df), pos=4)
text(60, 20, "X")
legend("topright", c("owner", "non-owner", "newhousehold"), pch = c(1, 3, 4))

# initialize normalized training, validation data, complete data frames to originals
train.norm.df <- train.df
valid.norm.df <- valid.df
mower.norm.df <- mower.df
# use preProcess() from the caret package to normalize Income and Lot_Size.
library(caret)
norm.values <- preProcess(train.df[, 1:2], method=c("center", "scale"))
train.norm.df[, 1:2] <- predict(norm.values, train.df[, 1:2])
valid.norm.df[, 1:2] <- predict(norm.values, valid.df[, 1:2])
mower.norm.df[, 1:2] <- predict(norm.values, mower.df[, 1:2])
new.norm.df <- predict(norm.values, new.df)

# use knn() to compute knn.
# knn() is available in library FNN (provides a list of the nearest neighbors)
# and library class (allows a numerical output variable).
library(FNN)
nn <- knn(train = train.norm.df[, 1:2], test = new.norm.df,
          cl = train.norm.df[, 3], k = 3)

row.names(train.df)[attr(nn, "nn.index")]

# initialize a data frame with two columns: k, and accuracy.
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))

# compute knn for different k on validation.
for(i in 1:14) {
  knn.pred <- knn(train.norm.df[, 1:2], valid.norm.df[, 1:2],
                  cl = train.norm.df[, 3], k = i)
  accuracy.df[i, 2] <- sum(knn.pred==valid.norm.df$Ownership)/length(valid.norm.df
$Ownership)*100
}
accuracy.df

#### Table 7.4

knn.pred.new <- knn(mower.norm.df[, 1:2], new.norm.df,
                   cl = mower.norm.df[, 3], k = 4)
row.names(train.df)[attr(nn, "nn.index")]

```