Stat 407 Lab 10 Classification Trees Fall 2012
Due: Wednesday, Oct 31 2012 in class Hand in one solution per group.

Purpose: This lab is about supervised classification using trees. We'll look at Dr Cook's music data and classify music tracks as either rock or classical based on variables calculated on the first 40 seconds of sound.

There are 54 data points (tracks), 30 are Rock tracks and 24 are Classical tracks. (There are also 3 tracks from Enya, and 5 unknown tracks.) The variables are:

LVar, LAve, LMax — the average, variance, maximum of the frequencies of the left channels

LFEner — an indicator of the amplitude or loudness of the sound.

LFreq — Average of the top 15 frequencies as computed by the `periodogram` function.

1. Compute a tree classifier to separate Rock from Classical tracks. Report the rule.
2. Plot the tree.
3. Use a scatterplot to display the variables used by the tree, and draw the splits on the plot.
4. Report the error rate. Is this an underestimate of the error likely with new cases?
5. Using your rule classify the five unknown tracks as Rock or Classical. How do these differ from the classifications by LDA?
6. Force the tree to make more splits, creating a better fit for this data, using the `minsplit` option. How does it differ from the simpler tree. Compute the error. Why is the tree what we might call overfitted?

Hints:

- Read data

```
music<-read.csv("data/music-plusnew-sub.csv",row.names=1)
```

- Subset data

```
music.sub<-subset(music,Type=="Classical"|Type=="Rock",
    select=Type:LFreq)
type_to_num<-c("Classical"=1,"Rock"=2)
music.sub[,1]<-factor(music.sub[,1],exclude="New wave")
music.class<-type_to_num[music.sub[,1]]
```

- Fit tree

```
library(rpart)
music.rp<-rpart(Type˜., data=music.sub)
plot(music.rp)
text(music.rp)
```

- Plot tree boundaries

```
# Using ggplot2
p<-qplot(LFreq,LFEner,data=music.sub,facets=.~Type)
x<-data.frame(x=c(175.5881,175.5881),y=c(88,115))
p<-p+geom_line(data=x,aes(x=x,y=y),colour="red")
x<-data.frame(x=c(175.5881,850),y=c(103.2803,103.2803))
p<-p+geom_line(data=x,aes(x=x,y=y),colour="red")
x<-data.frame(x=c(100,600,600),y=c(90,90,112),lb=c("Rock","Classical","Rock"))
p<-p+geom_text(data=x,aes(x=x,y=y,label=lb),colour="red",size=3)

# Or using base graphics
par(mar=c(4,4,1,1))
plot(music.sub$LFreq,music.sub$LFEner,xlim=c(40,900),ylim=c(80,115),
     pch=as.numeric(music.sub$Type),xlab="LFreq",ylab="LFEner")
abline(v=175.5881,col="red")
lines(c(175.5881,900),c(103.2803,103.2803),col="red")
text(c(100,600,600),c(90,90,112),c("Rock","Classical","Rock"),col="red")
```

- Misclassification table, error rate and predictions

```
table(music.sub[,1],predict(music.rp,music.sub,type="class"))
cbind(music.sub[,1],predict(music.rp,music.sub,type="class"))
predict(music.rp,music[58:62,], type="class")
```

- Plot the new data

```
# ggplot2
x<-data.frame(LFreq=music$LFreq[58:62],LFEner=music$LFEner[58:62],
              lb=c("1","2","3","4","5"))
p+geom_text(data=x,aes(x=LFreq,y=LFEner,label=lb),colour=I("blue"),size=3)

# base graphics
points(music$LFreq[58:62],music$LFEner[58:62],
      pch=16,col="orange")
```

- Force the algorithm to do more splits

```
music.rp2<-rpart(Type~., data=music.sub, minsplit=2)
plot(music.rp2)
text(music.rp2)
table(music.sub[,1],predict(music.rp2,music.sub,type="class"))
predict(music.rp2,music[58:62,], type="class")
```