These are my suggestions on how to analyze this data and organize the results. I found that although I took a long time to understand the nature of the missing values that I really didn’t need to do anything special with the missing values. I used trees, random forests and LDA to fit numerical classifiers, and LDA gave the best results. This matched with the plots of the data where differences between the types of tracks are in combinations of variables. Important components to include in the summary are: describe the classifier, write out the error rate, most important variables, classification of new tracks and other interesting aspects of the data. It's important to recognize the limitations of the data collection, but think of the data analysis as leading to questions that might be addressed in future experiments.

1 Data Description

This data was collected by Dr. Cook from her own CDs. Using a Mac she read the track into the music editing software Amadeus II, snipped and saved the first 40 seconds as a WAV file. (WAV is an audio format developed by Microsoft, commonly used on Windows but it is getting less popular.) These files were read into R using the package `tuneR`. This converts the audio file into numeric data. All of the CDs contained left and right channels, and variables were calculated on both channels. The resulting data has 57 rows (cases) and 72 columns (variables).

- LVar, LAve, LMax, RVar, RAve, RMax: average, variance, maximum of the frequencies of the left and right channels, respectively.
- LPer1-LPer15, LFreq1-LFreq15, RPer1-RPer15, RFreq1-RFreq15: height and frequency of the highest peak in the periodogram.
- LFEnner, RFEnner: an indicator of the amplitude or loudness of the sound.
- LFVar, RFVar: variance in the frequencies as computed by the `periodogram` function.

There are 30 tracks by Abba, the Beatles and the Eels, which would be considered to be Rock, and 24 tracks by Vivaldi, Mozart and Beethoven, considered to be Classical. The main question we want to answer is:

"Can Rock tracks be distinguished from Classical tracks using the given variables?"

Other questions of interest might be:

- How does Enya compare to Rock and Classical tracks?
- Are there differences from CD to CD?
- Are there differences between the tracks of different artists?
- Is any difference between Rock and Classical due to voice vs no voice?
## 2 Suggested approaches

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reason</th>
<th>Type of questions addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Restructuring</td>
<td>Summarize and possible impute missing values. Divide the data into training and test sets. Select the most important variables.</td>
<td></td>
</tr>
<tr>
<td>Summary statistics</td>
<td>Tabulate averages and standard deviations for the important variables, for each group.</td>
<td>“Is there a difference in average left channel variance in frequency for Rock and Classical tracks?”</td>
</tr>
<tr>
<td>Plots</td>
<td>Univariate plots and scatterplots of important variables.</td>
<td>“Are there differences between rock and classical tracks?”</td>
</tr>
<tr>
<td>Numerical classifiers</td>
<td>LDA, QDA, logistic regression, trees and random forests</td>
<td>“How do we predict a new track to be either Rock or Classical?”</td>
</tr>
</tbody>
</table>
3 Actual Results

3.1 Data restructuring

3.1.1 Missing Values

Table 1: Number of missings by variable.

<table>
<thead>
<tr>
<th>Number of missings</th>
<th>Number of Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1 contains a tabulation of the number of missings on each variable. The response variable, Type, has no missing values. Most of the predictor variables have no missing values. The missing values are concentrated in the Freq variables. Most of these Freq variables have 1 missing value (LFreq1-4, LFreq6-8, LFreq10-15, RFreq1-4, RFreq13), some have 2 (LFreq9, RFreq5-10, RFreq14-15), and a couple have 3 (RFreq11-12). LFreq5 has no missing values.

Tracks that have missings are

<table>
<thead>
<tr>
<th>Track</th>
<th>Num Missing</th>
<th>Variable(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Winner</td>
<td>1</td>
<td>LFreq</td>
</tr>
<tr>
<td>Cant.Buy.Me.Love</td>
<td>3</td>
<td>RFreq10-12</td>
</tr>
<tr>
<td>I.Feel.Fine</td>
<td>9</td>
<td>RFreq5-9, RFreq11-12, RFreq14-15</td>
</tr>
<tr>
<td>Beethoven 2</td>
<td>29</td>
<td>LFreq1-LFreq4, LFreq6-LFreq15, RFreq1-15</td>
</tr>
</tbody>
</table>

Maybe imputing the values for Beethoven 2 will be enough, and the other missings might be eliminated by choosing variables with no missings. Beethoven 2 is most similar to Vivaldi’s 2, 4, 8 tracks, on the variables LVar, LAve, LMax, RVa, RAve, RMax, and LFEner, RFEner, LFVar, RFVar. But on the one non-missing Freq value, LFreq5 it is very different from these tracks. It might not be so easy to impute the missings for this track.

We’ll use random forests to get some help in deciding important variables, and then decide what to do with the missing values. This is the top 10 variables according to MeanDecreaseAccuracy, and according to MeanDecreaseGini:
The most important variable is LVar, which is at the top of both lists. Other important variables appear to be RAve, RMax, RFreq13, LFEn, RFreq14, LFreq7. This would suggest we would want to consider using LVar, RAve, LMax, RMax, LFEn, RFEn, and several LFreq, RFreq variables (7,13,14). It seems a bit strange to take the 7,13,14th most high peaks in the periodogram. It may be easier to explain the results if these variables are not used at all. We’ll compare classifications with and without these variables.

Out of interest we examine the Freq variables more closely. We need to see if a track mostly has Freq values around a similar value. If so, then these variables might be summarized by an average value. Below are parallel coordinate plots of the LFreq and RFreq variables. In the LFreq variables the peaks of rock tracks are mostly at lower frequencies and the peaks of the classical are mostly at the higher frequencies. For the most part these tracks have similar frequencies for the peaks, seen by the mostly parallel lines. A few tracks have large differences in the frequencies of peaks: V1 and Dancing Queen. Similar observations can be made about the RFreq variables, although there are more tracks with varied frequencies: V1, Dancing Queen, SOS, I want to hold you hand, Can’t buy me love. It looks like taking an average of these LFreq and RFreq variables may be a reasonable way to reduce the number of variables and remove missing values.
There is one missing value left after doing this: B2 on RFreq. For this value we will substitute the LFreq value.

This leaves us with these 10 variables to use for the classification: LVar, RVar, LAve, RAve, LMax, RMax, LFEner, RFener, LFreq, RFreq.

### 3.2 Summary Statistics

There are 30 rock tracks (10 Abba, 10 Beatles, 10 Eels), and 24 classical tracks (10 Vivaldi, 6 Mozart 8 Beethoven). Table 2 contains the means and standard deviations of the important variables, broken out by Type of music and Artist.
<table>
<thead>
<tr>
<th>Type</th>
<th>LVar</th>
<th>RVar</th>
<th>LAve</th>
<th>RAve</th>
<th>LMax</th>
<th>RMax</th>
<th>LFEner</th>
<th>RFEn</th>
<th>LFreq</th>
<th>RFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>35.3*</td>
<td>30.7*</td>
<td>-27.2</td>
<td>-1.75</td>
<td>27562</td>
<td>27223</td>
<td>107</td>
<td>105</td>
<td>152</td>
<td>177</td>
</tr>
<tr>
<td>Classical</td>
<td>5.11*</td>
<td>5.21*</td>
<td>17.8</td>
<td>17.9</td>
<td>18241</td>
<td>16801</td>
<td>101</td>
<td>102</td>
<td>369</td>
<td>311</td>
</tr>
<tr>
<td>Abba</td>
<td>6.88*</td>
<td>7.37*</td>
<td>-80.2</td>
<td>-4.36</td>
<td>22839</td>
<td>23242</td>
<td>103</td>
<td>103</td>
<td>134</td>
<td>143</td>
</tr>
<tr>
<td>Beatles</td>
<td>48.0*</td>
<td>40.6*</td>
<td>-5.97</td>
<td>-5.99</td>
<td>28545</td>
<td>27931</td>
<td>110</td>
<td>110</td>
<td>140</td>
<td>208</td>
</tr>
<tr>
<td>Eels</td>
<td>51.1*</td>
<td>51.0*</td>
<td>4.59</td>
<td>5.10</td>
<td>31301</td>
<td>30496</td>
<td>108</td>
<td>107</td>
<td>181</td>
<td>179</td>
</tr>
<tr>
<td>Beethoven</td>
<td>7.61*</td>
<td>7.58*</td>
<td>-0.74</td>
<td>-0.07</td>
<td>21120</td>
<td>19632</td>
<td>101</td>
<td>101</td>
<td>350</td>
<td>286</td>
</tr>
<tr>
<td>Mozart</td>
<td>4.69*</td>
<td>5.61*</td>
<td>-5.94</td>
<td>-1.48</td>
<td>18875</td>
<td>18312</td>
<td>101</td>
<td>102</td>
<td>396</td>
<td>353</td>
</tr>
<tr>
<td>Vivaldi</td>
<td>3.35*</td>
<td>3.07*</td>
<td>46.9</td>
<td>51.9</td>
<td>15557</td>
<td>13709</td>
<td>102</td>
<td>102</td>
<td>367</td>
<td>306</td>
</tr>
<tr>
<td>Enya</td>
<td>50.3*</td>
<td>63.2*</td>
<td>-11.8</td>
<td>-56.3</td>
<td>16063</td>
<td>15921</td>
<td>103</td>
<td>104</td>
<td>95</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 2: Means (Standard deviations) of the variables by type of music, and artist. (* Raised by 10^6).

3.3 Plots

The plots below show the histograms of the selected variables. The variables with the biggest differences between rock and classical are LVar and RVar. LAve is only important to distinguish Abba tracks from the rest. LMax and RMax have a difference in distribution between the two classes: Rock tracks are more right-skewed, and classical are more uniformly distributed. LFEner and RFEn are surprising: although there appeared to be little differences between the means (Table 2) the rock tracks take noticeably larger values than the classical tracks. In LFreq and RFreq the rock tracks are more left-skewed than the classical tracks. It looks like further reducing the variables by half by considering only the left channel variables might be reasonable.
The scatterplot matrix below shows the left channel variables. Rock tracks are labeled with “+”, and classical tracks are labeled with “o”. The relationships between the variables is important: A combination of LFEnner and LFreq almost perfectly separates the two classes.
4 Classification

The data is broken into 2/3 training and 1/3 test sets based on stratified sampling by artist. There are 10 tracks from each of the Rock CDs, so 7 tracks from each of these are randomly sampled into the training set. There are 10 tracks from Vivaldi, 6 from Mozart and 8 from Beethoven CDs, which are respectively sampled at 7/10, 4/6, 6/8, into the training set. Break data into training and test. The tracks which are in my training set are 1,2,4,6,7,8,10,11,13,14,15,17,19,22,23,24,25,27,28,30,32 33,34,35,37,38,41,42,43,44,46,47,48,49,51,53,54. The tracks in the test set are 3,5,9,12,16,18,21,26,29,31,36,39,40,45,50,52.

Which classifier should we use? The variance differences between the groups in LVar and LAve would suggest that LDA might not work well. The separations appear to be in combinations of variables, which suggests trees may not work well.

Trees are simple so we’ll start with them. The results are summarized in Figure 1. The tree appears to fit the training data very well, although the second split is too close to the classical tracks. This is a curious choice of splits! Why didn’t the algorithm choose a split at LAve=-40? The misclassification table is:

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th></th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True/Pred</td>
<td>Class</td>
<td>Rock</td>
</tr>
<tr>
<td>Class</td>
<td>14</td>
<td>3</td>
<td>0.176</td>
</tr>
<tr>
<td>Rock</td>
<td>1</td>
<td>20</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Random forests do a little better with this data. The training error is 4/38=0.105, and the test error is 2/16=0.125.
Figure 1: Summary of the tree classifier.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th></th>
<th>Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True/Pred</td>
<td>Class</td>
<td>Rock</td>
<td>Marginal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td></td>
<td>14</td>
<td>3</td>
<td>0.176</td>
</tr>
<tr>
<td>Rock</td>
<td></td>
<td>1</td>
<td>20</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Linear discriminant analysis does extremely well with this data. There are 3 errors in the training data, and 0 errors in the test data. The two rock tracks that are misclassified are both Eels tracks, Restraining and Agony. The classical track that is misclassified is the 8th Beethoven track. Tracks that are close to the boundary are Beethoven 7, Vivaldi 9, Mozart 3, The Winner (Abba), Yesterday (Beatles), in the training data, and in the test data, Beethoven 4, Vivaldi 8, The Good Old Days (Eels), Eleanor Rigby (Beatles).

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th></th>
<th>Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True/Pred</td>
<td>Class</td>
<td>Rock</td>
<td>Marginal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td></td>
<td>16</td>
<td>1</td>
<td>0.059</td>
</tr>
<tr>
<td>Rock</td>
<td></td>
<td>2</td>
<td>19</td>
<td>0.095</td>
</tr>
</tbody>
</table>

LDA uses all 5 variables to build its rule. Here are the correlations between the predicted values and each variable:

<table>
<thead>
<tr>
<th>LVar</th>
<th>LMax</th>
<th>LFene</th>
<th>LFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.63</td>
<td>-0.55</td>
<td>0.60</td>
<td>0.76</td>
</tr>
<tr>
<td>-0.70</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The LDA classification rule would be:

Assign a new observation, $x_0$, to Rock if

$$a'x_0 - 15.9 > 0$$

else assign to Classical, where $a' = (1.87e-08 - 2.83e-02 3.94 - 05 1.47e-01 - 2.99e-03)$

We decided not to fit a quadratic discriminant analysis model because the LDA does very well and there is limited data for a more complex model.
5 What could I do better?

It would be important to go over this report, trim it down and re-do the plots to presentation quality.
6 Conclusions

- The final classifier is the one computed using LDA:

  Assign a new observation, $x_o$, to Rock if 
  $$a'x_o - 15.9 > 0$$
  else assign to Classical, where $a' = (1.87e-08 - 2.83e-02 3.94 - 05 1.47e-01 - 05 1.94e-03)$

- The training error is 8% and test error is 0. The Eels tracks Restraining and Agony and Beethoven’s 8th track are misclassified.

- The most important variables for the classification are LFEner, LFreq, LVar, LMax, LAve. Rock songs have generally higher LFEner and lower LFreq than classical tracks.

- There are numerous other interesting aspects of the data.
  - Enya tracks are similar to classical in the Ave, Var, Max variables, but more similar to Rock in the Freq variables. Using the LDA rule two are predicted to be classical (The Memory of Trees, Pax Decorum) and one rock (Anywhere Is), and the predicted values are close to the boundary.
  - When Abba tracks are very different from the others in that they have negative average values! This may be a CD effect. We might like to take a look at other Abba CDs to see if this persists.
  - These tracks are unusual: Saturday Morning (Eels), and Vivaldi 6. Saturday Morning has an unusual pattern in the time/frequency plot (appendix) in that it maxes out at the high and low values. This may be due to the axis limits in the plot function.

- This is a simple study. There is very little data and the tracks were chosen rather than randomly sampled from a larger population. We might use this study to propose hypotheses to test on a larger random sample of tracks. From this small study, it looks like it would be quite viable to develop a classifier for the type of music based on variables created on audio tracks. In a larger study we would want to test for CD effects, for different orchestra renditions of classical music, for other types of music such as country, and jazz.
Classifying new tracks

The 4 new tracks are classified as rock, classical, classical, classical, classical. The first track is strongly predicted to be rock, with predicted value 3.75. The next three tracks are close to the boundary but predicted to be classical. The last track is strongly predicted to be classical with a predicted value -3.20.

Examine plots of these new tracks. Track 1 is clearly an Abba song, because it has a low LAve value. Track 5 looks to be clearly a classical song with a very low value of LFEner. But it is an outlier in the plots of the data, that sometimes stays closer to the rock songs. The other three tracks (2,3,4) more consistently stay with the classical tracks.
8 References


Appendix

d.music<-read.csv("music.csv",row.names=1)

d.music<-d.music[,c(1:5,36:40,71:72)]
write.table(d.music,"music-sub.csv",append=T,quote=F,sep="",row.names=F,col.names=T)
write.table(d.music,"music-full.csv",append=T,quote=F,sep="",row.names=F,col.names=T)

summary(d.music)
summary(t(d.music))

# Random forests
library(randomForest)
music.rf <- randomForest(as.factor(d.music[1:54,2]) ~ .,
data=data.frame(d.music[1:54,-c(1,2)]),importance=TRUE,proximity=TRUE,mtry=3)
music.rf <- randomForest(as.factor(d.music[1:54,2]) ~ .,
data=data.frame(d.music[1:54,c(21:35,56:70)]),importance=TRUE,proximity=TRUE,mtry=6)
music.rf$importance[order(music.rf$importance[,4],decreasing=T),4:5]
music.rf$importance[order(music.rf$importance[,5],decreasing=T),4:5]

# It looks like averaging the frequency variable might be a reasonable
# approach to dealing with missing values, and reducing the number of
# variables. Mostly the tracks have similar values for the frequencies
# with highest peaks.
LFreq<-apply(d.music[,21:35],1,median,na.rm=T)
RFreq<-apply(d.music[,56:70],1,median,na.rm=T)
# There is one missing value left, B2 on RFreq. We're going to substitute
# the value for the LFreq.
RFreq[48]<-LFreq[48]

summary(d.music.sub[,1])
summary(d.music.sub[,2])

d.music<-cbind(d.music,LFreq,RFreq)
d.music.sub<-d.music[,c(1:5,37:40,72:74)]

apply(d.music.sub[d.music.sub[,2]=="Rock",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,2]=="Rock",-c(1,2)],2,sd)
apply(d.music.sub[d.music.sub[,2]=="Classical",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,2]=="Classical",-c(1,2)],2,sd)
apply(d.music.sub[d.music.sub[,1]=="Abba",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,1]=="Beatles",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,1]=="Eels",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,1]=="Beethoven",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,1]=="Mozart",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,1]=="Vivaldi",-c(1,2)],2,mean)
apply(d.music.sub[d.music.sub[,2]=="Enya",-c(1,2)],2,mean)

par(mfrow=c(2,2))
hist(d.music.sub[d.music.sub[,2]=="Rock",3],col=2,xlim=range(d.music.sub[,3]),
     xlab=names(d.music.sub)[3],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Rock",7],col=2,xlim=range(d.music.sub[,7]),
     xlab=names(d.music.sub)[7],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Classical",3],col=2,
     xlim=range(d.music.sub[,3]),
     xlab=names(d.music.sub)[3],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Classical",7],col=2,
     xlim=range(d.music.sub[,7]),
     xlab=names(d.music.sub)[7],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Rock",4],col=2,xlim=range(d.music.sub[,4]),
     xlab=names(d.music.sub)[4],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Rock",8],col=2,xlim=range(d.music.sub[,8]),
     xlab=names(d.music.sub)[8],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Classical",4],col=2,
     xlim=range(d.music.sub[,4]),
     xlab=names(d.music.sub)[4],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Classical",8],col=2,
     xlim=range(d.music.sub[,8]),
     xlab=names(d.music.sub)[8],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Rock",5],col=2,xlim=range(d.music.sub[,5]),
     xlab=names(d.music.sub)[5],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Rock",9],col=2,xlim=range(d.music.sub[,9]),
     xlab=names(d.music.sub)[9],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Classical",5],col=2,
     xlim=range(d.music.sub[,5]),
     xlab=names(d.music.sub)[5],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Classical",9],col=2,
     xlim=range(d.music.sub[,9]),
     xlab=names(d.music.sub)[9],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Rock",6],col=2,xlim=range(d.music.sub[,6]),
     xlab=names(d.music.sub)[6],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Rock",10],col=2,xlim=range(d.music.sub[,10]),
     xlab=names(d.music.sub)[10],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Classical",6],col=2,
     xlim=range(d.music.sub[,6]),
     xlab=names(d.music.sub)[6],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Classical",10],col=2,
     xlim=range(d.music.sub[,10]),
     xlab=names(d.music.sub)[10],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Rock",11],col=2,xlim=range(d.music.sub[,11]),
     xlab=names(d.music.sub)[11],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Rock",12],col=2,xlim=range(d.music.sub[,12]),
     xlab=names(d.music.sub)[12],main="Rock")
hist(d.music.sub[d.music.sub[,2]=="Classical",11],col=2,
     xlim=range(d.music.sub[,11]),
     xlab=names(d.music.sub)[11],main="Classical")
hist(d.music.sub[d.music.sub[,2]=="Classical",12],col=2,
     xlim=range(d.music.sub[,12]),
     xlab=names(d.music.sub)[12],main="Classical")
pairs(d.music.sub[-c(54:57),c(3:6,11)],
pch=as.numeric(d.music.sub[-c(54:57),2]))

indx1<-c(sample(c(1:10),7),sample(27:36,7),sample(37:46,7))
indx2<-c(sample(c(11:20),7),sample(21:26,4),sample(47:54,6))
indx<-c(indx1,indx2)
sort(indx)
[26] 37 38 41 42 43 44 46 47 48 49 51 53 54
c(1:54)[-indx]
[1] 3 5 9 12 16 18 21 26 29 31 36 39 40 45 50 52

d.music.train<-d.music.sub[indx,]
d.music.test<-d.music.sub[-c(indx,55:57),]

#Trees
library(rpart)
music.rp<-rpart(d.music.train[,2]~.,data.frame(d.music.train[,c(3:6,11)]),
                       method="class",parms=list(split='information'))
music.rp

table(d.music.train[,2],
      predict(music.rp,data.frame(d.music.train[,c(3:6,11)]),type="class"))

table(d.music.test[,2],
predict(music.rp,data.frame(d.music.test[,c(3:6,11)]),type="class"))

par(mfrow=c(1,3),pty="m")
plot(music.rp)
text(music.rp)
par(pty="s")
plot(d.music.train[,5],d.music.train[,4],type="n",xlab="LMax",ylab="LAve",
     xlim=c(2900,32800),ylim=c(-98,217))
points(d.music.train[d.music.train[,2]="Rock",5],
       d.music.train[d.music.train[,2]="Rock",4],pch=3)
points(d.music.train[d.music.train[,2]="Classical",5],
       d.music.train[d.music.train[,2]="Classical",4],pch=1)
abline(v=27135.5)
lines(c(4000,27135.5),c(-5.676338,-5.676338))
title("Training data")

plot(d.music.test[,5],d.music.test[,4],type="n",xlab="LMax",ylab="LAve",
     xlim=c(2900,32800),ylim=c(-98,217))
points(d.music.test[d.music.test[,2]="Rock",5],
       d.music.test[d.music.test[,2]="Rock",4],pch=3)
points(d.music.test[d.music.test[,2]="Classical",5],
       d.music.test[d.music.test[,2]="Classical",4],pch=1)
abline(v=27135.5)
lines(c(4000,27135.5),c(-5.676338,-5.676338))
title("Test data")

# Random forests
library(randomForest)
music.rf2 <- randomForest(as.factor(d.music.train[,2]) ~ .,
data=data.frame(d.music.train[,c(3:6,11)]),
importance=TRUE,proximity=TRUE,mtry=3)
music.rf2
table(d.music.train[,2],predict(music.rf2, data=data.frame(d.music.train[,c(3:6,11)])))

table(d.music.test[,2],predict(music.rf2, newdata=data.frame(d.music.test[,c(3:6,11)]),type="class"))

music.rf2$importance[order(music.rf2$importance[,4],decreasing=T),4:5]

# LDA
library(MASS)
cls<-factor(d.music.train[,2],levels=c("Classical","Rock"))
music.lda<-lda(d.music.train[,c(3:6,11)],cls, prior=c(0.5,0.5))
table(cls, predict(music.lda,d.music.train[,c(3:6,11)],dimen=1)$class)
cls2<-factor(d.music.test[,2],levels=c("Classical","Rock"))
table(cls2, predict(music.lda,d.music.test[,c(3:6,11)],dimen=1)$class)

music.lda.xtr<-predict(music.lda,d.music.train[,c(3:6,11)],dimen=1)$x
music.lda.xts<-predict(music.lda,d.music.test[,c(3:6,11)],dimen=1)$x

par(mfrow=c(2,2))
hist(music.lda.xtr[d.music.train[,2]=="Classical"],breaks=seq(-6,6,by=0.5), col=2,xlim=c(-6,6),xlab="LDA Predicted",main="Classical Training")
abline(v=0)
hist(music.lda.xts[d.music.test[,2]=="Classical"],breaks=seq(-6,6,by=0.5), col=2,xlim=c(-6,6),xlab="LDA Predicted",main="Classical Test")
abline(v=0)
hist(music.lda.xtr[d.music.train[,2]=="Rock"],breaks=seq(-6,6,by=0.5), col=2,xlim=c(-6,6),xlab="LDA Predicted",main="Rock Training")
abline(v=0)
hist(music.lda.xts[d.music.test[,2]=="Rock"],breaks=seq(-6,6,by=0.5), col=2,xlim=c(-6,6),xlab="LDA Predicted",main="Rock Test")
abline(v=0)

for (i in c(3:6,11))
  cat(cor(d.music.train[,i],music.lda.xtr),"\n")

mn<-(music.lda$means[1,]+music.lda$means[2,])/2
sum(mn*music.lda$scaling)
prd<-as.matrix(d.music.train[,c(3:6,11)])%*%music.lda$scaling-15.9
prd[order(prd)]

# Predict Enya
predict(music.lda,d.music.sub[55:57,c(3:6,11)],dimen=1)$class
predict(music.lda,d.music.sub[55:57,c(3:6,11)],dimen=1)$x

# Predict new observations
d.music.new<-read.csv("music-new.csv")
d.music.new.vars<-cbind(d.music.new[,c(1:3,35)],
  apply(d.music.new[,19:33],1,median,na.rm=T))
dimnames(d.music.new.vars)[[2]][5]<-"LFreq"

predict(music.lda,d.music.new.vars,dimen=1)$class
predict(music.lda,d.music.new.vars,dimen=1)$x
x <- cbind(rep(NA, 5), rep(NA, 5), d.music.new.vars)
dimnames(x)[[2]][1] <- "Artist"
dimnames(x)[[2]][2] <- "Type"
d.music.plus <- rbind(d.music.sub[, c(1:6, 11)], x)

x <- as.numeric(d.music.plus[, 2])
x[is.na(x)] <- 4
pairs(d.music.plus[, 3:7], pch = x)

write.table(d.music.plus, "music-plusnew-sub.csv", append = T, quote = F,
            col.names = T, row.names = T, sep = ",")

Plots of the audio tracks here, but they are available on the course web site.