Homework 1, I546/N546: Symbolic Music Informatics

For each problem submit the R code you created with any necessary written annotations, as well as relevant plots. These problems use the midi data available on the class website
http://www.music.informatics.indiana.edu/courses/I546

1. Produce a pitch class histogram of the pitches contained in

   (a) The Joplin Rag, ragime/silvswan.dat, and

   (b) the Haydn piano sonata movement classical/hy-ps-07.dat

   Using just your histogram determine the keys of these pieces.

2. Examine the files Classical/ty_mai.dat and Classical/ty_juli.dat on the class web site. Perform a rhythmic analysis of these piece *without listening to the midi files* to determine the rhytmic structure. Submit a plot of onset times modulo the longest meaningful unit as well as your factoring of the rhythmic structure as done in class. What is the time signature?

3. The file "dorabella.ost" on the class web page has two columns giving the onset times of notes in

   (a) musical units (measure number + measure position)

   (b) time units with 8000/256 (about 31) time units per second.

   Create a plot of this data with musical time on the x axis and real time on the y axis. What is the average tempo for the first 4 measures? Explain any sudden tempo changes in this plot in musical terms. It may help to know that the piece is in 6/8 time.

4. Create a piano-roll representation of a section of a piece of music using one of the ".dat" files on the class web page. The R command

   ```
   > segments(x1,y1,x2,y2)
   ```

   adds a line segment to the existing plot connecting the point (x1,y1) to (x2,y2). Use this command to create "voices" in your plot by connecting notes that are likely to belong to the same "voice" or musical part. Generally speaking, notes are likely to belong to the same voice if their onset times and pitches are near one another, though, of course the reality is more subtle than this. For each note, i.e. (time, midi) pair (t_i, m_i), find the "best successor candidate" (t_j, m_j) by choosing the note that minimizes $a|t_i - t_j| + b|m_i - m_j|$ where a and b are constants of your choosing. Of course, you should only consider notes j having $t_j > t_i$. If the best candidate's score is less than some threshold T, also of your choosing, connect the notes by a line segment. Choose the constants so that the image connects notes that lie in the same voice as best as possible, subjectively speaking.

   While there are many ways to do things in R, you may wish to use the commands `min` or `which.min`. `min` returns the minimum element of a vector, while `which.min` returns the *index* of the element. So, for instance, `min(c(3.3,7.7,1.1,2.2)` gives 1.1, while `which.min(c(3.3,7.7,1.1,2.2)` gives 3 (since the 3rd element is the smallest).