MA678: Midterm Project Final Report

Yudi (Kenny) Mao

Problem Statement and Background

Music industry has been transforming from traditional record distribution to modern digital distribution in recent decades. As digital records make success, people are much easier to get, listen and comment on different types, genres of music. Meanwhile, music publishers like Apple, Spotify and Pandora are able to recommend music to their subscribers, based on their appetite and habit, as well as music trend. Thus, how to measure the popularity of different music effectively and correctly remains a big problem to music producers and publishers.

The Million Song Dataset (MSD), which this analysis project based on, was created under a grant from the National Science Foundation, project IIS-0713334. It is a free-available collection of audio features and metadata from a million contemporary popular music tracks, provided by The Echo Nest. Besides the entire dataset, Labrosa also provides a subset of 10,000 songs (1%, 1.8GB), which this project is based on.

I analyze the Million Song Dataset Subset to find the correspondence between music hotness and various features, including artist hotness, popularity, song duration, etc. Linear model and multilevel linear models are used to fit this analysis, with model check procedure. From the models, I expected to see some basic relationship between various features and to figure out whether the relationship meets people's common sense.

Data Collection and Wrangling

The Million Song Dataset Subset (10,000 files) is downloaded in HDF5 (.h5) files. HDF5 is a unique technology suite that makes possible the management of extremely large and complex data collections. Each file represents on song track with all related information, and the raw data is 2.76GB.

Labrosa page provides wrappers with various languages, but except R. In this project, I use R package “rhdf5” to read song characteristics into R. A 3.9MB CSV file with 10,000 observations and 53 variables is created for further wrangling, which is also on my GitHub page.

Although the CSV file is comparatively small, I still need to select and filter the data for modeling stage. The wrangling procedure contains:

1) Drop variable columns which only have NA and zeros;
2) Drop location variables since they have too many NA;
3) Drop most nametags, Id numbers, only leave three for tracking;

4) Filter rows with NA ;
5) Drop rows where zeros do not make sense: *tempo, time_signature*; Keep variable *year* although it has many zeros (transform later);

After these procedures, I get a dataset of 5637 observations and 15 variables (12 descriptive variables for analysis and 3 id variables for tracking). The Labrosa page provides an example track information, so it is able to find the legend of some of these variables. And the rest 5 are easy to guess. They are:

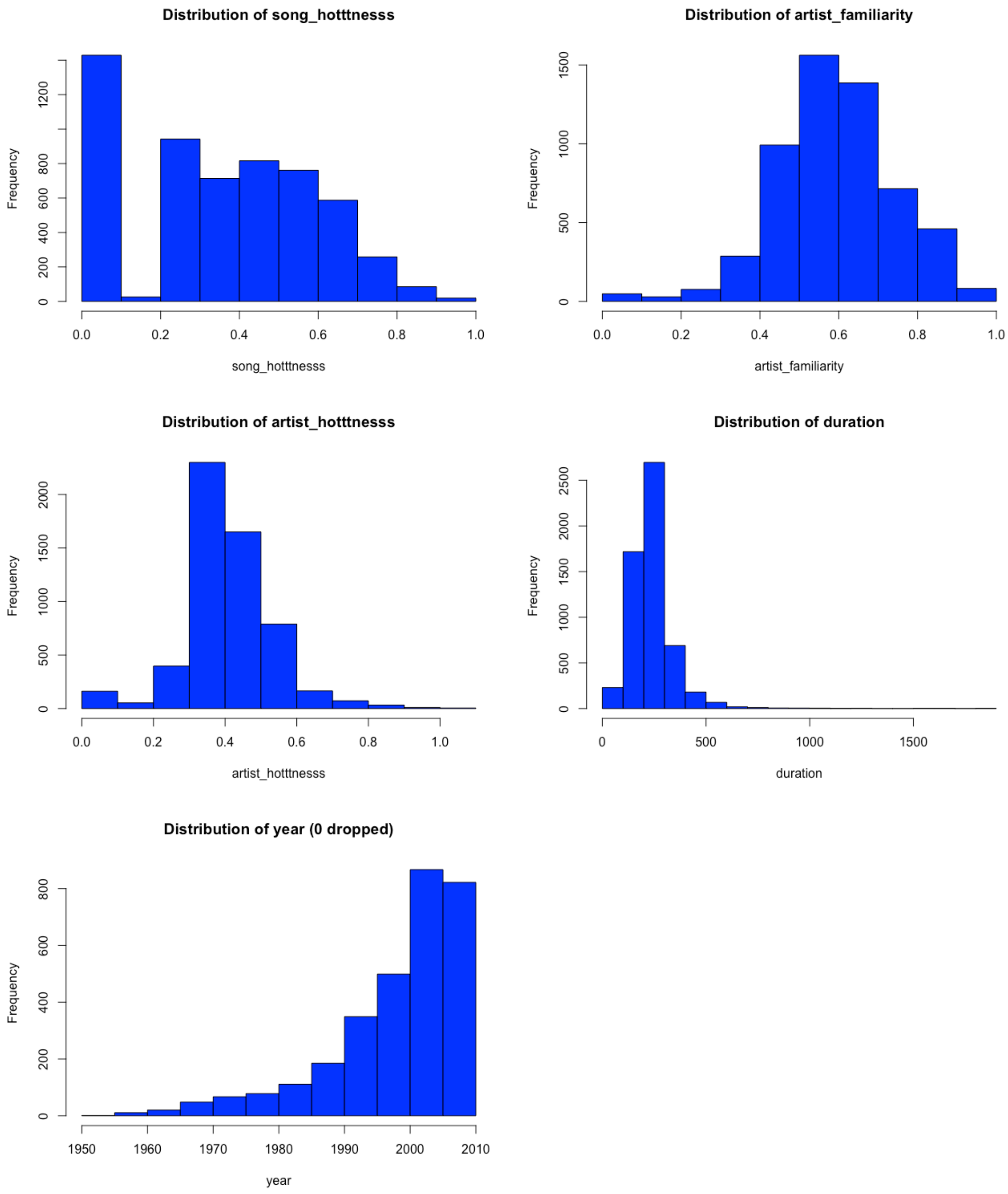
1) *Duration:* duration of the track in seconds
2) *End_of_fade_in:* time of the end of the fade in, at the beginning of the song, according to The Echo Nest
3) *Key:* estimation of the key the song is in by The Echo Nest
4) *Key_confidence:* confidence of the key estimation
5) *Loudness:* general loudness of the track
6) *Mode:* estimation of the mode the song is in by The Echo Nest
7) *Mode_confidence:* confidence of the model estimation
8) *Start_of_fade_out:* start time of the fade out, in seconds, at the end of the song, according to The Echo Nest
9) *Tempo:* tempo in BPM according to The Echo Nest
10) *Time_signature:* time signature of the song according to The Echo Nest, the i.e. usual number of beats per bar
11) *Time_signature_confidence:* confidence of the time signature estimation
12) *Track_id:* The Echo Nest ID of this particular track on which the analysis was done
13) *Artist_familiarity*
14) *Artist_hotttnesss*
15) *Artist_id*
16) *Song_hotttnesss*
17) *Song_id*
18) *Year:* year when this song was released, according to musicbrainz.org

Basic statistics and EDA

Before fitting models for the cleaned data, I want to look at the basic statistics of these variables and perform some EDA to get a better sense of the data.

First of all, I confirm no more NA would show up in any observation or variable. Then I check how many zeros appear in each column, to make sure they make sense. After that, I try to get a sense of the data through plotting histograms and ggplot2 point graphs.

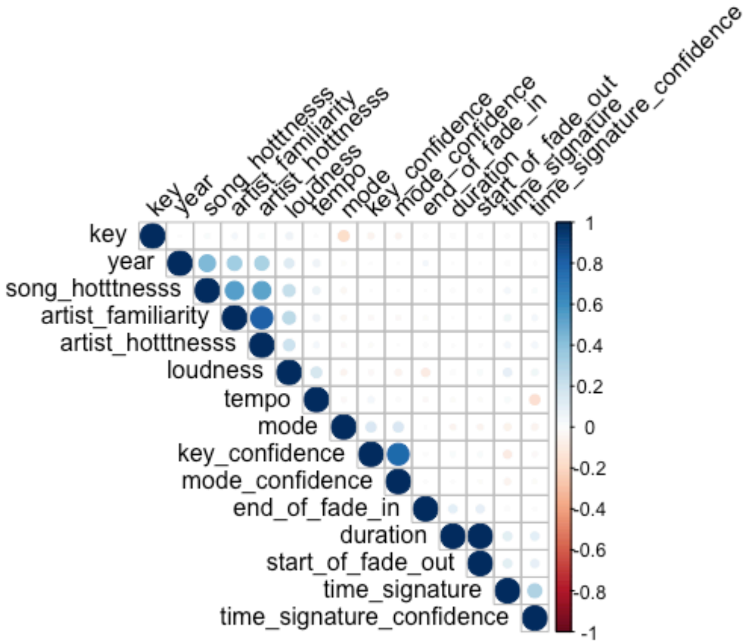
Zeros in column *year* significantly affect the plot, so I dropped zeros first. It would be transformed when modeling. And many zeros are observed in *song_hotttnesss*, which is the outcome.



p1-5. histograms of the distribution of variables

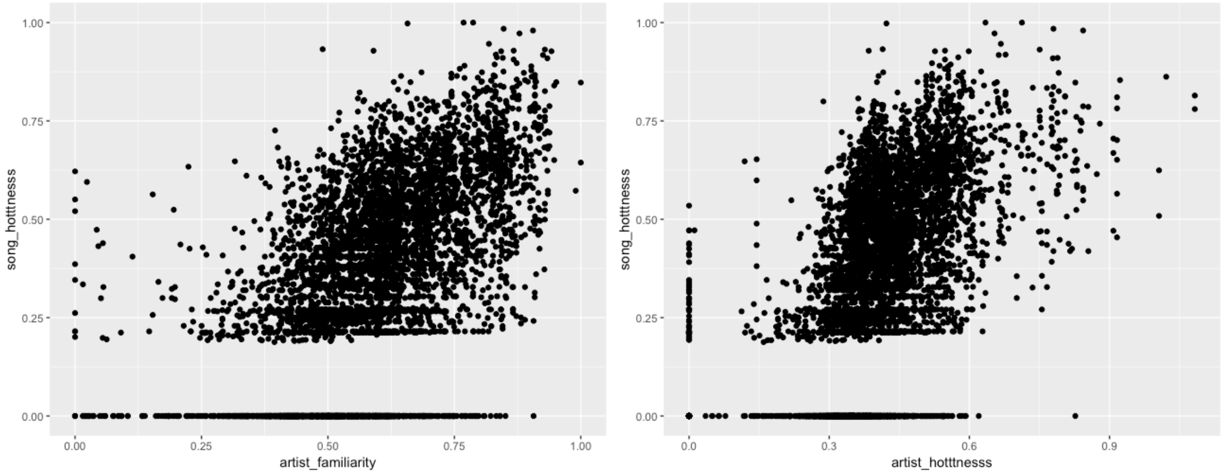
Meanwhile, I visualize the correlation among variables using "corrplot" package because:

1) It basically shows the relationship between predictors and outcome (*song_hotttnesss*);
2) It shows the correlation to be contained in models.



p6. Correlation plot

We have a large correlation between these pairs of variables: *song_hotttnesss* and *artist_familiarity, artist_hotttnesss, year*; *artist_familiarity* and *artist_hotttnesss; key_confidence* and *mode_confidence*. Thus, I also plot point graphs between *song_hotttnesss* and *artist_familiarity, artist_hotttnesss.* Due to the graphs, a basic increasing trend could be observed in both relationships, and zeros in *song_hotttness* seem weird. They are far from other values. In this project, I decided to drop these zeros either.



P7-8 point plots between song hotness and artist variables

Modeling and Model Check

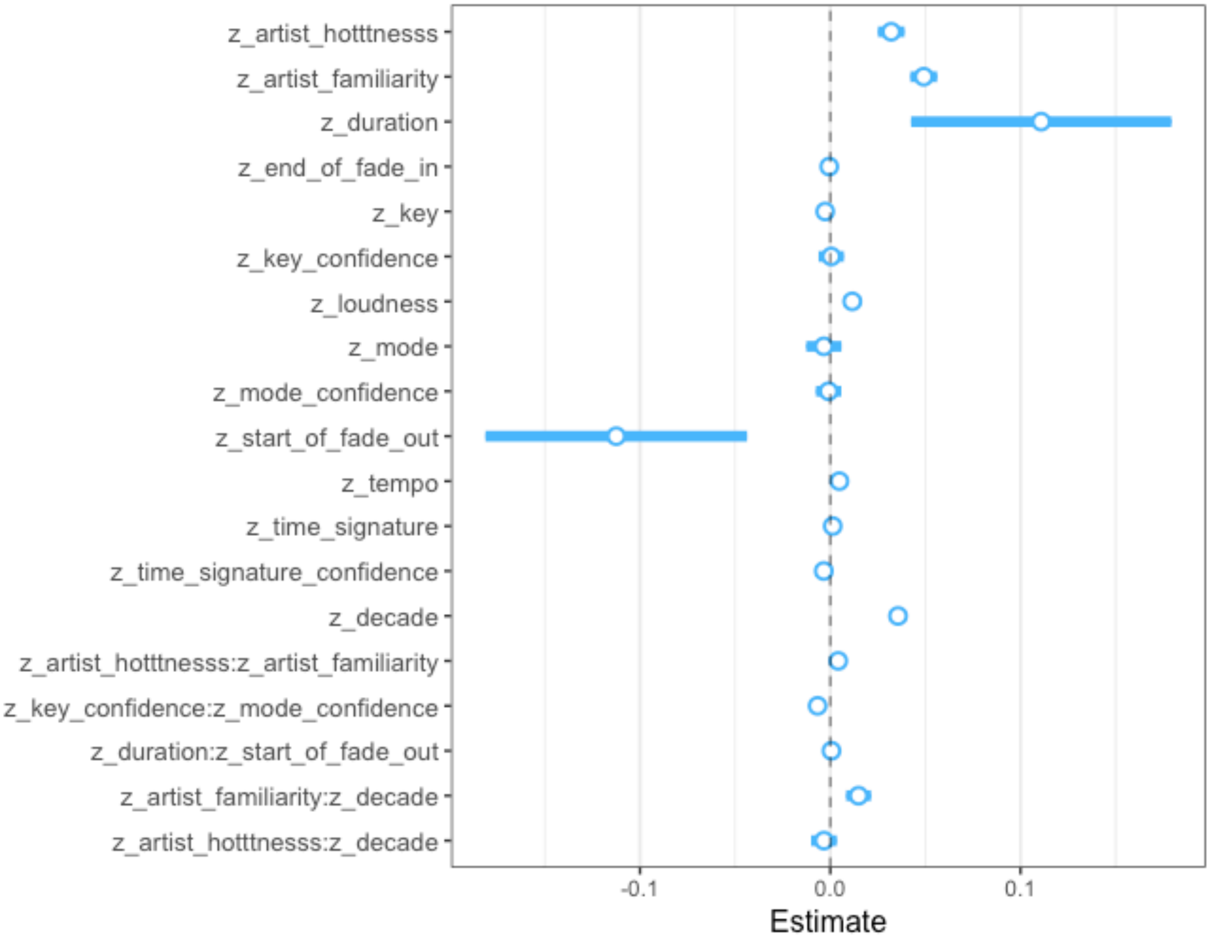
Due to the result of EDA above, I decide to do more wrangling and transformation on variables:

1) Drop rows with zeros in *song_hotttness*;
2) Regard zeros in *year* to be 1950 (the smallest value in the data);
3) Transform *year* to *decade* for easier modeling and interpretation;
4) Transform variables to their z scores.

Linear model and multilevel linear models are applied. I pick predictors that have a significant effect on the outcome to change the model fit. Visualization is also very important to help interpreting models.

Linear model

Significant predictors are: *artist_hotttnesss, artist_familiarity, duration, loudness, start_of_fade_out, tempo, decade.* (Some are not easy to be observed in the graph).



P9 linear regression model

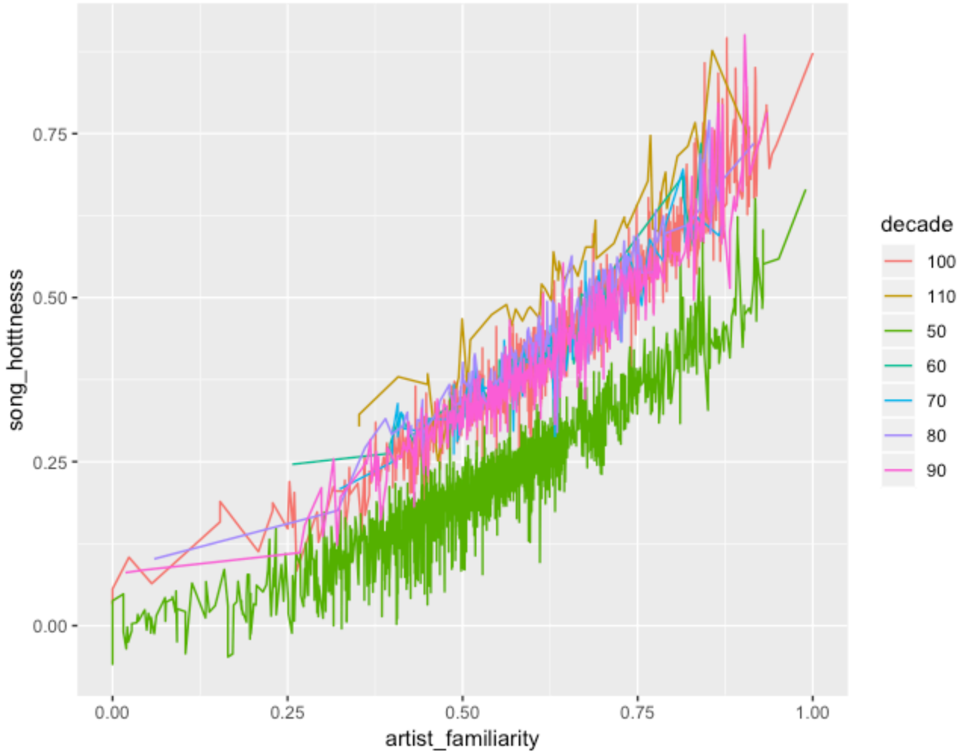
However, F-statistics of the raw linear model is **111.1**, while that of the model only contains significant variables is **210.4**. That indicates that the model fit of the raw model is better than the latter one (though both of them are bad).

Since z-scores (standardized variables) of each variable is used in the model, we can directly compare coefficients. According to **P9** above, we can group predictors into three levels:

1) Most effect: *duration, start of fade out* (negative effect)
2) Significant effect: *artist hotness, artist familiarity, decade*
3) Others: *loudness,* etc.

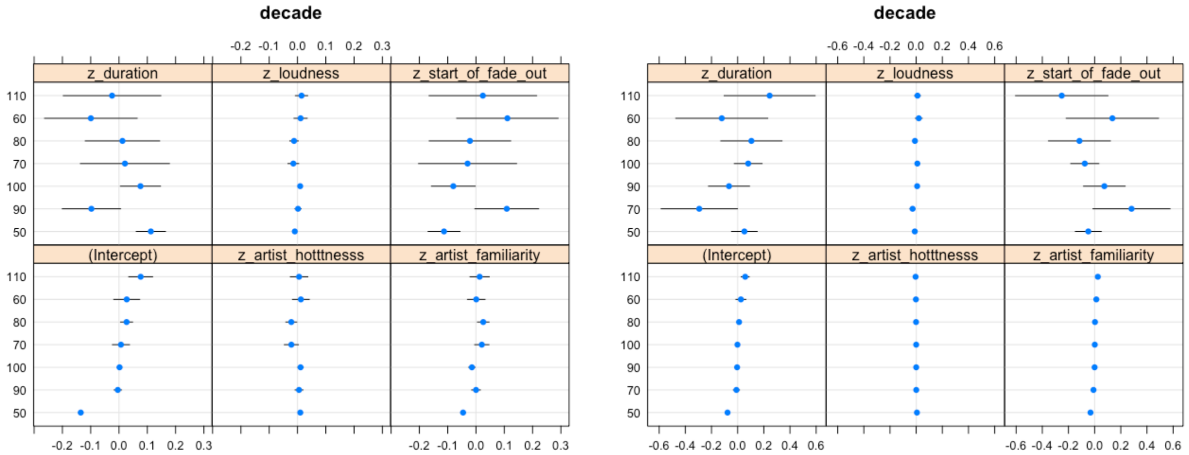
Multilevel linear model

From the above analysis, I find several key predictors. Among these predictors, *decade* is not continuous but grouped. Songs in the same decade might be mutually dependent. So, we can treat *decade* as a random effect. Here I use an EDA plot to help visualize the relationship between song hotness and artist familiarity as an example.



P10 EDA grouped by decade

From the mixed model with varying intercept, I find 5 important predictors: *artist_hotttnesss, artist_familiarity, duration, loudness, start_of_fade_out.* Since calculation for multilevel linear models in R is comparatively bigger than linear models, I use these 5 predictors above for model with varying intercept and slope, rather than using all predictors.



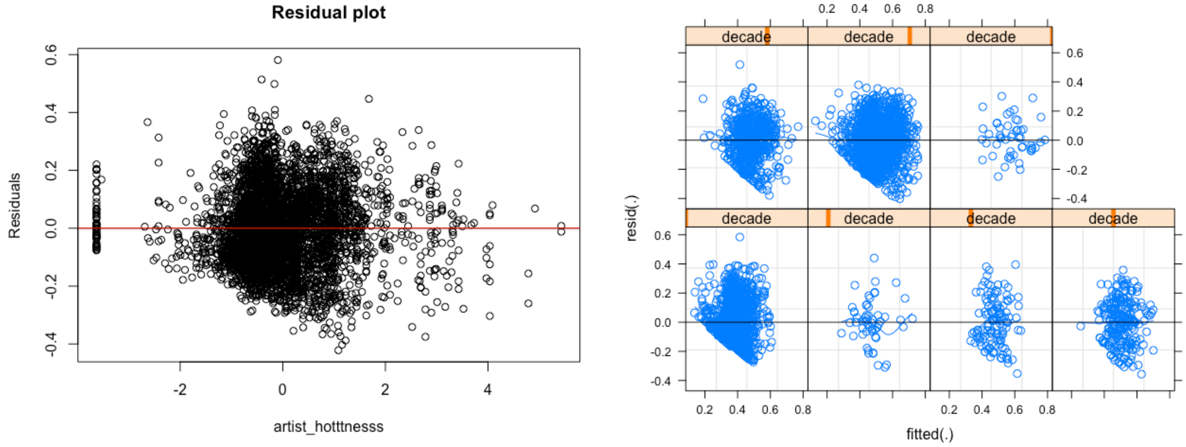
P11-12 random effect

P11 and P12 are plots for the random effect. The difference is P11 is based on data that zeros in song hotness are not dropped, while P12 is based on data that zeros have been dropped.

We can see in P11, random effect in 50 (which means songs in the 1950s) is not significant. That is understandable because zeros in raw year data are transformed to 1950, and in decade 50. They are not naturally grouped, which makes the random effect in 50 not significant. However, in P12 random effect in the 1950s looks better. That indicates data with zeros in *song_hotttnesss* overlaps with data which has zeros in *year*.

We can also see a similar result from the linear model that *duration* and *start_of_fade_out* are two most important predictors.

To check the residual plot of both two models, we have P13-14. Still, large variance appears among different groups of decades. It is not able to say either linear regression model or multilevel linear model fits the data well.



P13-14 residual plot

Conclusion and Discussion

In this project, I use Million Song Dataset Subset to discover the relationship between song hotness and various features. Firstly, I import the HDF5 data into R, clean the data for further analysis. Secondly, I check basic statistics and perform EDA to get a sense of the data. Then I apply linear models and multilevel linear models with different predictors/Random effects. Finally, I try to check the model fit and interpret models to draw the conclusion.

We can draw several conclusions from the Million Song Dataset:

1) Song hotness is most related to song duration and the start time of fade out. **Longer songs have more hotness, and songs fade out later have more hotness.** Actually, these two features are also related, that is, longer songs tend to fade out later.
2) Song hotness is also related to hotness and familiarity of its artist. **More famous artists create songs with more hotness.** And they are all related to years. **Recent songs, as well as artists, tend to gain more attention.**
3) Compared to the large data, especially if we want to apply models on the whole MSD (1 million songs), both linear regression model and multilevel linear model are limited. They do not fit very well, though it is able to interpret overall trends and relationships of data.

The Million Song Dataset is really an interesting dataset. It contains more than these 20 features used in this project. For example, it also has characteristic features, which we could do text analysis. It has the weight of different genres of each song. Thus, this project can be extended to another one with larger data (whole dataset rather than the subset), more features (genres, song names) and more complex models (machine learning models).

Acknowledgments

The Million Song Dataset was created under a grant from the National Science Foundation, project IIS-0713334. The original data was contributed by The Echo Nest, as part of an NSF-sponsored GOALI collaboration. Subsequent donations from SecondHandSongs.com, musiXmatch.com, and last.fm, as well as further donations from The Echo Nest, are gratefully acknowledged.

https://labrosa.ee.columbia.edu/millionsong/

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere.
The Million Song Dataset. In Proceedings of the 12th International Society
for Music Information Retrieval Conference (ISMIR 2011), 2011.