**Sample Corpus Text Analysis Report**

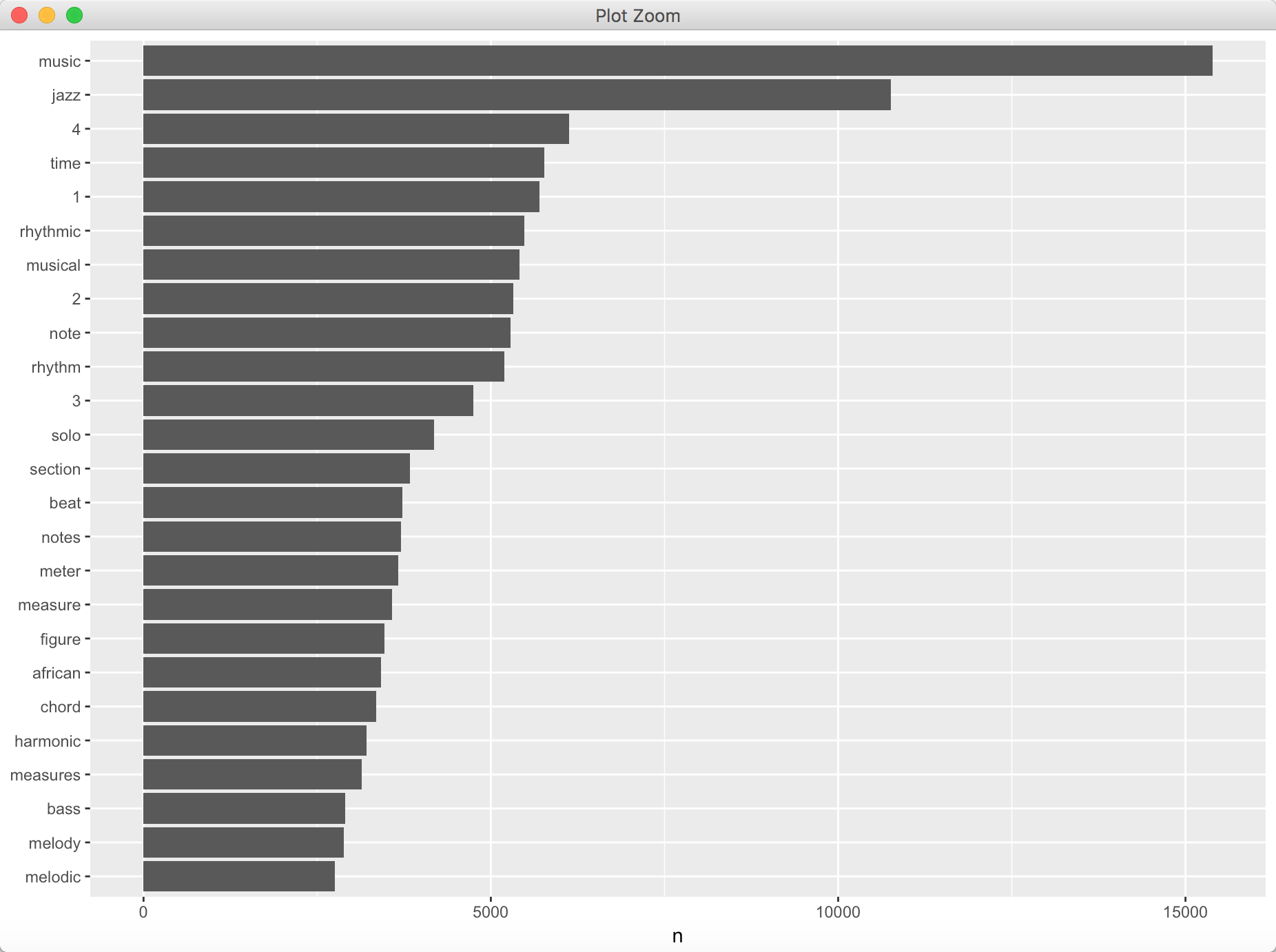
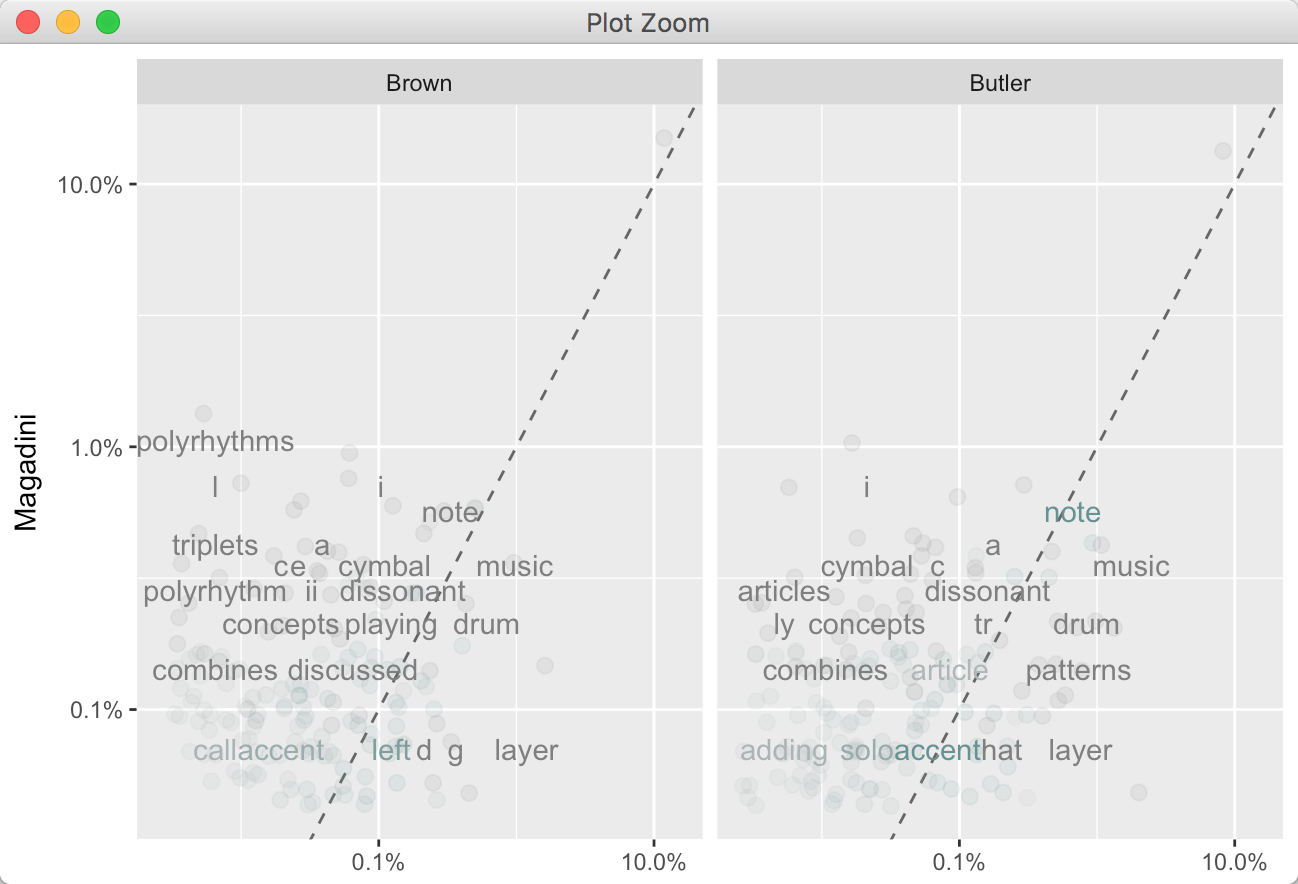
**Introduction**

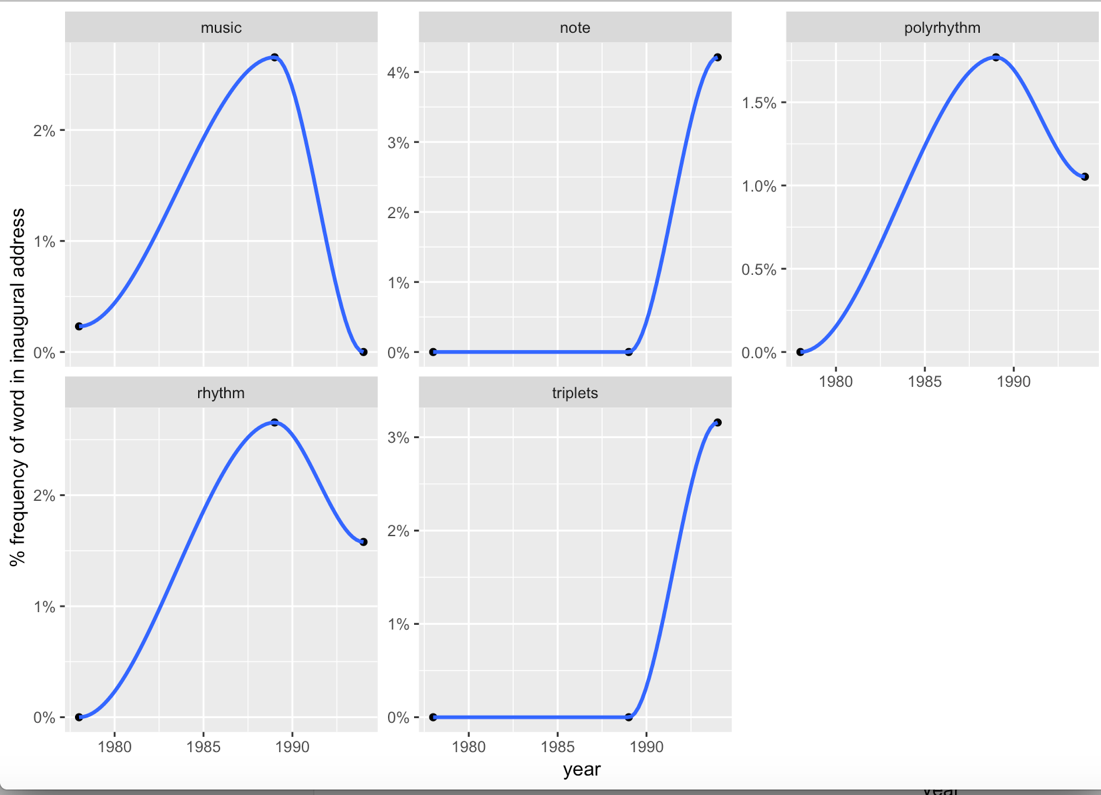
The goal of the current project is to explore the concept of rhythmic ambiguity in popular music (i.e., rock, metal, jazz, funk, rap, hip-hop and electronic dance music) through the lenses of audience-based comments. The project was conducted by text analysis in Sample Corpus using R programming language. Text analysis is the process of extracting useful and important information from text through computational approach, which can present new evidence to support a hypothesis. Appendix A collects many techniques of text analysis, including word frequency, topic modeling, sentimental analysis and so forth. All the analysis is done in R, a popular programming language, and easy to learn for literature researchers. Moreover, R is an open source language so that a variety of text mining packages are available in R. Therefore, R is a useful computational tool to explore literary question.

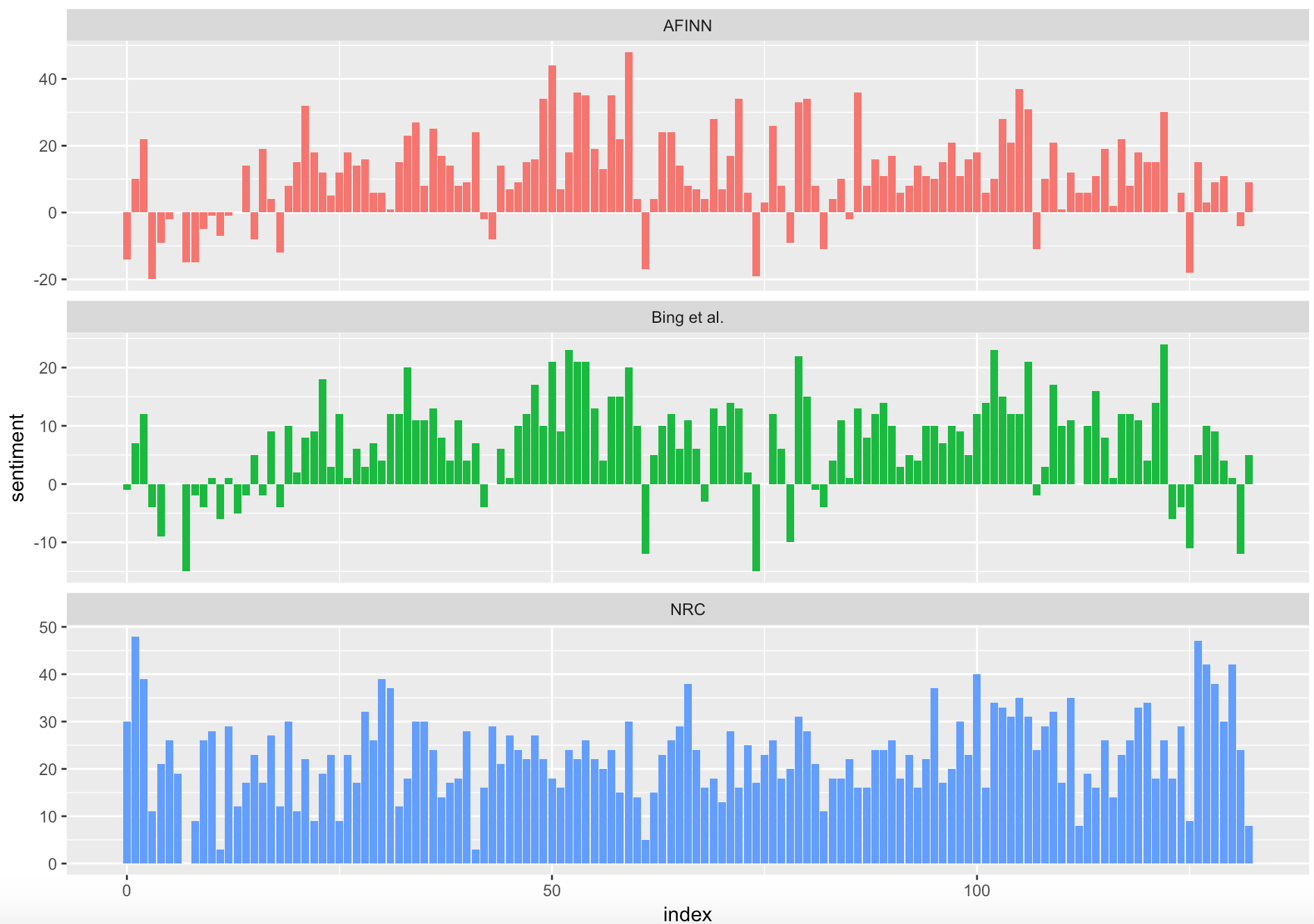
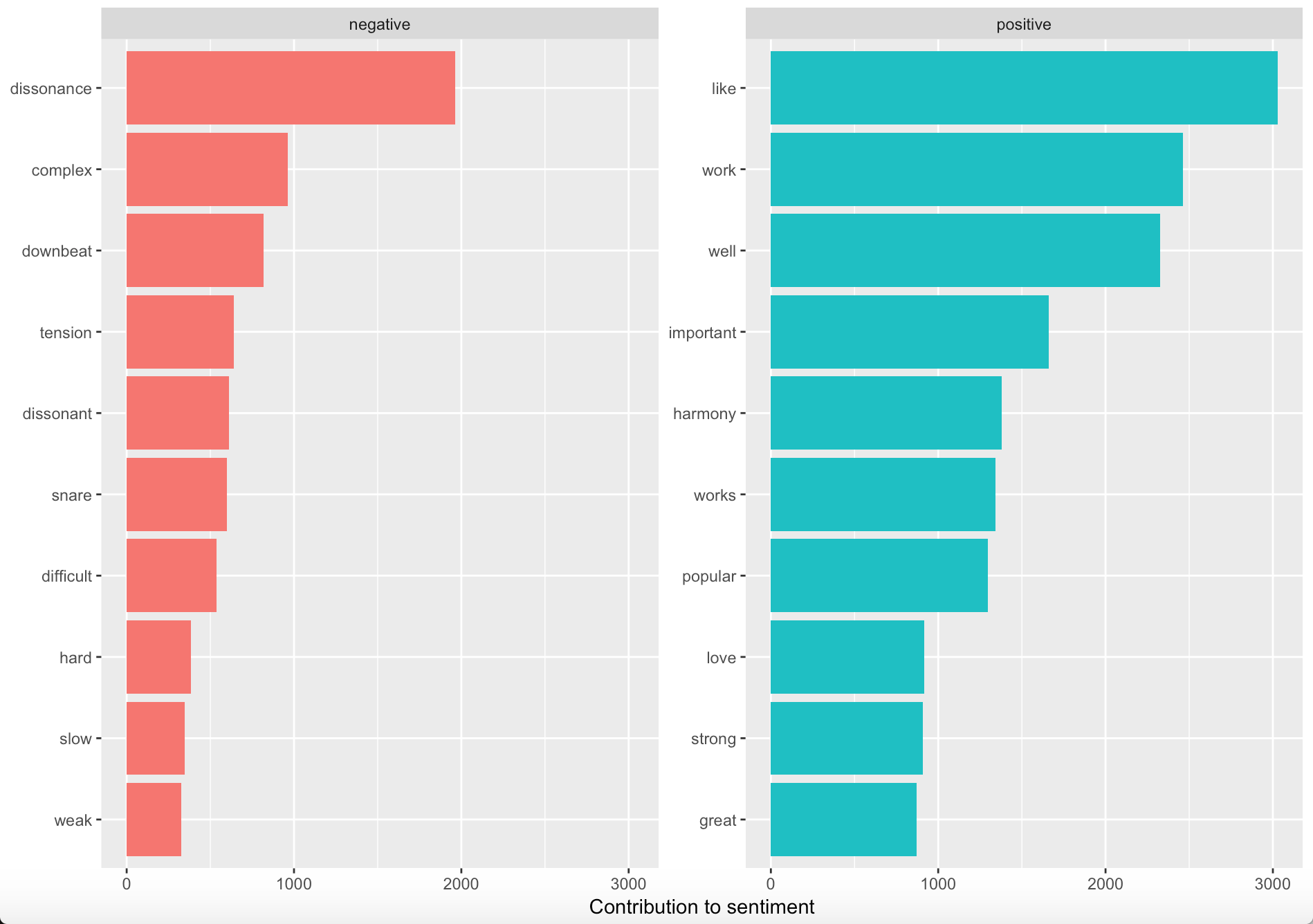
**Status Report**

1. R training: during summer, I completed reading “Text Analysis with R for Students of Literature (Jockers, 2014)”, “Text Mining with R (Silge and Robinson, 2018)” till chapter 8 and “Macroanalysis” till chapter 4.
   1. Exercises in “TAR” are done in “TextAnalysisWithR” folder: <https://app.box.com/folder/49733008484>
   2. I replicated tasks in two books and adapted code to analyze Sample Corpus
      1. Replicated code of “TAR” are in “TAR” folder: <https://app.box.com/folder/53735136968>
      2. Replicated code of “TMR” are in “TMR” folder: <https://app.box.com/folder/53725995888>
2. Updated Sample Corpus metadata in “Sample Corpus Metadata” workbook: <https://app.box.com/file/318963418350>
   1. Created “Short File ID” for each text in “Metadata (new)” worksheet.
   2. Updated new text name to the files in “Short\_ID\_Sample\_Corpus\_txt” folder: <https://app.box.com/folder/54618465334>
3. Database search results are in “Sample Corpus Metadata\_Database Search” spreadsheet: <https://app.box.com/file/319830734511>. The goal is to extract at least one song for each text. We still need to read song title from abstract in further steps.
4. Researched “Topic modeling” and “Unsupervised clustering” in “Topic modeling and unsupervised clustering.docx”: <https://app.box.com/file/326101928150>
5. Conducted exploratory analysis of “Sample Corpus”. Results are in “Exploratory\_Analysis” workbook: <https://app.box.com/file/326050329732>
   1. Generated hapax list for each text and “Sample Corpus”.
      1. Generated 165 hapax lists, one for each text in the “Sample Corpus”: “text-specific” worksheet. Code is in “hapax\_text\_specific”: <https://app.box.com/file/326136633064>.
      2. Generated a hapax for the entire corpus: “sample-corpus hapax” worksheet. Code is in “hapax\_corpus”: <https://app.box.com/file/326122046549>.
   2. Generated 25 most frequent words for each of 165 texts in the corpus: “26-frequent-words” worksheet. Code is in “frequent\_words”: <https://app.box.com/file/326131578810>.
   3. Generated a word count for each text of “Sample Corpus” in the “word-count” sheet. Code is in “word\_count\_each\_text”: <https://app.box.com/file/326145430851>.
   4. Performed unsupervised clustering on authorship by using high frequency words among the corpus. Results are in the “Unsupervised\_Clustering” folder: <https://app.box.com/folder/54618770182>. Code is in “unsupervised\_clustering”: <https://app.box.com/file/326120649411>.
   5. Conducted supervised classification for metadata of “Sample Corpus” such as gender and text’s style. A model is built based on existing data and category so that it can be used to classify further texts into gender or style. Code is in “supervised\_classification\_style” and “supervised\_classification\_gender”.
      1. A “Metadata (Modified)” worksheet is created to remove some items in “Gender”, “Author, Primary 1” or “Style, Primary” column for testing purposes. I used these removed thus unknown data as test data.
      2. Results are in the “gender-threshold-#words” worksheet” and “style-threshold-#words” worksheet. The spreadsheets contain the information of “Predication of Model Labels”, “Summary of Model Predication” and “Predication of Testdata”.
   6. Conducted topic modeling of “Sample Corpus”. Code is in “topic\_modeling\_xxx”.
      1. Uploaded a list of POS (part of speech such as verb, noun) in the file “POS tags.txt”: <https://app.box.com/file/326071274431>
      2. Generated 1, 25, 50, 100 untagged workc louds that each with 80 words by using Jocker’s stoplist under “Wordcloud\_Untagged” folder: <https://app.box.com/folder/50913483731>
      3. Generated 1, 25, 50, 100 tagged (adjectives, verbs and nouns) word clouds that each with 80 words by using Jocker’s stoplist under “Wordcloud\_Tagged” folder: <https://app.box.com/folder/54629756198>
         1. Name format for each word cloud: “topic#-mostfrequentword-second-third”
      4. Generated 165 tagged (nouns) topics information in the “165-tagged-topic info” sheet. Information is listed by column: Topic#; three most frequent words within each topic; three most representative texts in descending order.
   7. Conducted sentimental analysis on “Sample Corpus” by using existing sentimental database. Code is in “sentiment\_words”.
      1. 15 most common positive and negative words in randomly generated 11 groups of Sample Corpus by using “bing”dataset in the “15\_common\_pos\_neg\_words.csv” file: <https://app.box.com/file/326127695447>. Plots are in the “Common\_Sentimental\_Words” folder: <https://app.box.com/folder/54632057056>.
      2. Generated two preliminary lists of stopwords for musical research based on sentimental analysis. These two lists need to be gone through and updated. Files are in “Stoplist” folder. After that, a sentimental analysis can be performed by using our musical-research related stoplist.
   8. Explored Tsig pattern of “Sample Corpus” by using Jiayi’s pattern (Appendix E). The frequency and statistics are in the “Tsig.xlsx” spreadsheet: <https://app.box.com/file/326136532851>. Code is in “tsig\_frequency”: <https://app.box.com/file/326135524919>. Tsig pattern needs to be refined to restrict the numbers in a limited range.
      1. Generated Tsig with corresponding frequency in each text: “Text-specific Tsig”.
      2. Generated Tsig words with corresponding frequency across “Sample Corpus”: “Corpus Tsig”.
   9. Extracted proper nouns (Appendix D).
      1. Use Stanford Named Entity Recognizer (NER) to extract “people”, “location”, “organization” from one text to examine the validity of NER. The code is in “name\_entity\_recognition”: <https://app.box.com/folder/53573270000>. The results are named in “ner\_person/organization/location.csv”.
      2. Compare people’s name extracted from NER and capitalized words by finding “people’s item from NER” that partially match “capitalized words” in one text. The code is in “ner\_overlap”: <https://app.box.com/file/328299124101>. The result file is named “ner\_cap\_overlap.csv”: <https://app.box.com/file/328297476440>.
      3. Explore song title extraction by generating n-grams of capitalized words and searching keyword in context such as “album”. The code is in “song\_title\_extraction”: <https://app.box.com/folder/53573270000>. The exploration results are in the “Song\_Title\_Extraction” folder: <https://app.box.com/folder/54939915876>.
      4. Further work needs to be done to improve the method of song title extraction, for example, extract the content within double quotation in abstract from metadata (check Appendix D).
6. Conducted exploratory analysis on “Hesselink Corpus”. Results are in the “Hesselink\_Exploratory\_Analysis” folder: <https://app.box.com/folder/54940461172>.
   1. Experimental topic modeling for “Hesselink Corpus”. Code are named as “hesselink\_topic\_modeling\_untagged/adj/noun”.
      1. Generated Hesselink Corpus with each word with a tag in the “taggedHesselink” folder:
      2. Generated untagged word clouds in “Topic\_Modeling\_Untagged” folder: <https://app.box.com/folder/54941941240>.
      3. Generated tagged wordclouds (Nouns, Adjectiveves) in “Topic\_Modeling\_Untagged” folder: <https://app.box.com/folder/54941941240>.

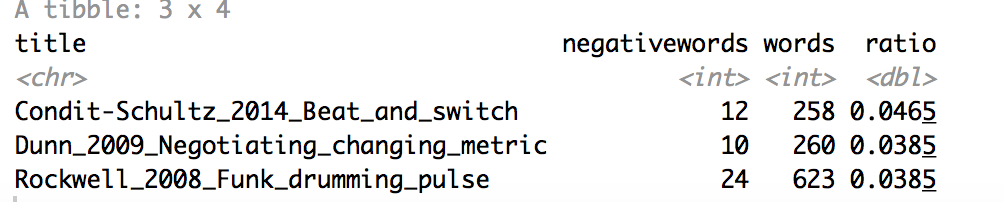
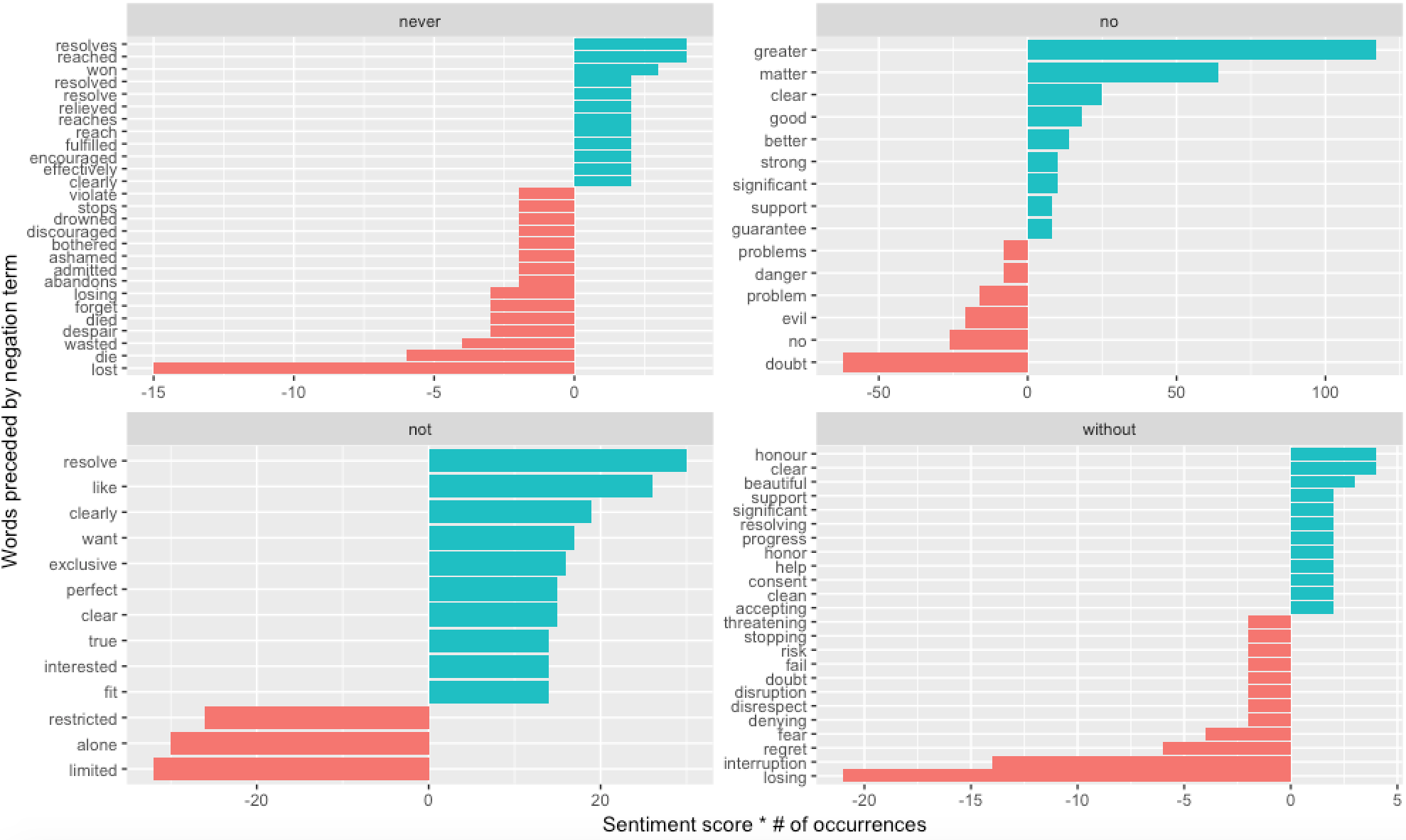
**Sample Corpus Findings Summary**

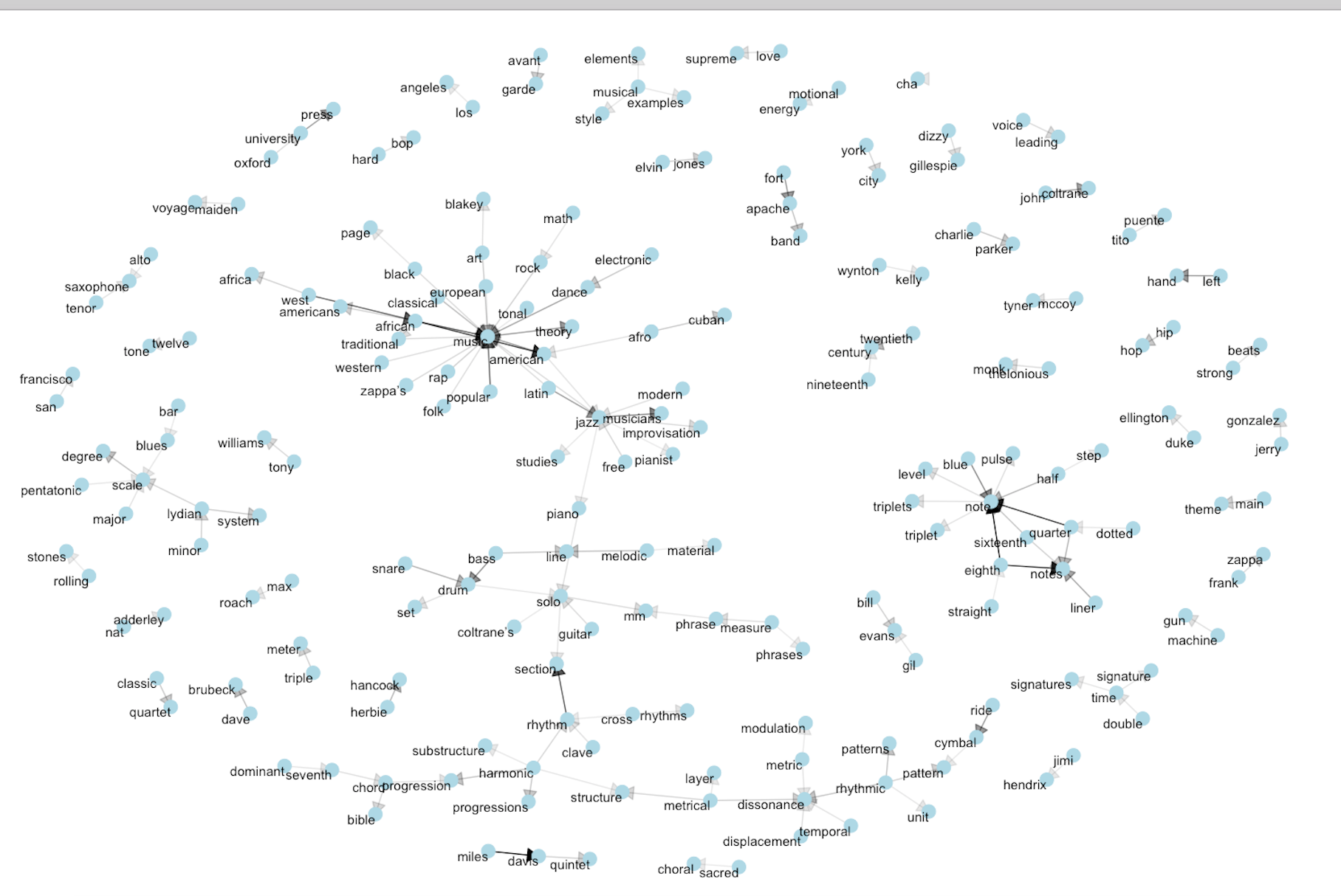
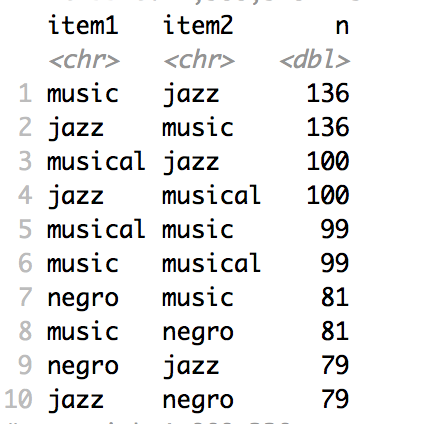
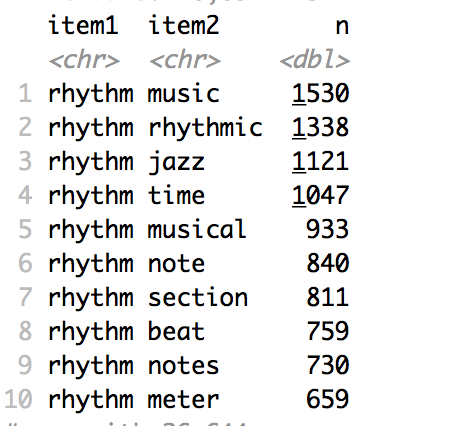
1. Frequency analysis:
   1. Identify the most common words across corpus (top 25 words). The file is “top\_25\_corpus\_words”: <https://app.box.com/file/326127695447>. 
   2. Compare word frequency across corpus vs. in-text word frequency. This allows us to compare strong deviations of word frequency within each text as compared to entire corpus. The correlation of word frequencies between the corpus and each text is in the “correlation\_of\_word\_freqs\_between\_corpus\_and\_each\_text”: <https://app.box.com/file/332845794416>. The mild to moderate correlations, which are all statistically significant (p-values < 0.0001), suggests that the relationship between the word frequencies is moderately similar across the corpus.
   3. Compare word frequency of different authors’ work: Brown, Butler and Magadini. The file is in “compare\_the\_word\_frequencies\_Maldini\_Brown\_Butler”: <https://app.box.com/file/332983422133>. 
      1. Words that are close to the line in these plots have similar frequencies in both sets of texts. For example, in both Magdini and Brown texts, words such as “note”, “music”, “dissonant” are fairly common and used with similar frequencies. Or in both Magadini and Butler texts, words such as “note”, “dissonant”, “article”, “accent” have similar frequencies.
      2. Words that are far from the line are words that are found more in one set of texts than another. For example, in the Magadini-Brown panel, words like “polyrythms”, “triplets” are found in Magadini texts but not much in the Brown texts, while words like “layer” are found in the Brown texts but not the Magadini texts. In comparing Magadini with Butler, Butler uses words like “layer”, “patterns” Magadini does not, while Magadini uses words like “cymbal” that Butler does not.
      3. Run correlation test between these set of word frequencies. See how correlated the word frequencies between different authors are. The correlatioin between Magadini and Brown is 0.9705953. The correlation between Magadini and Butler is 0.9584533. Therefore the word frequencies are more correlated between Magadini and Brown than Butler.
   4. Highest tf-idf terms in each text of “Sample Corpus” in the “highest\_tf-idf\_terms\_each\_text.pdf” file: <https://app.box.com/file/333010515520>.
   5. Highest tf-idf words in each style of “Sample Corpus”. Classify texts into groups based on their “Primary Style” from metadata. Counts, tf-idf are calculated based on style group, generate highest 15 tf-idf words for each style texts in “15\_highest\_tf-idf\_terms\_style.png”: <https://app.box.com/file/333012212145>. By looking at high tf-idf terms in each text, we can add less meaningful words into stop-words and re-generate the high tf-idf again.
   6. Highest tf-idf bigrams in each text of “Sample Corpus” is in the “high\_tf-idf\_bigrams\_each\_text.pdf” file: <https://app.box.com/file/306646374068>.
   7. Changes in word frequency over time within “Magadini”’s texts:

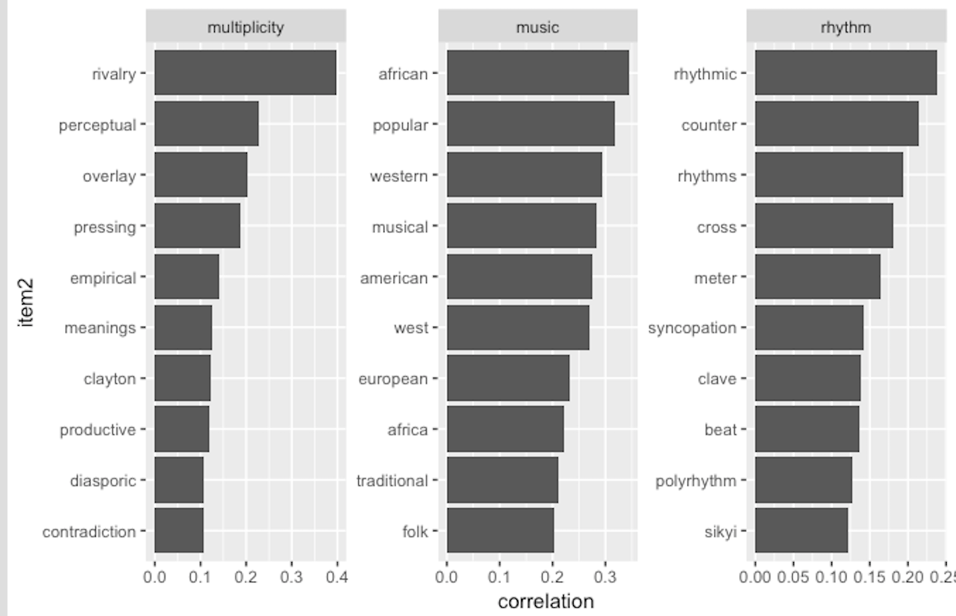


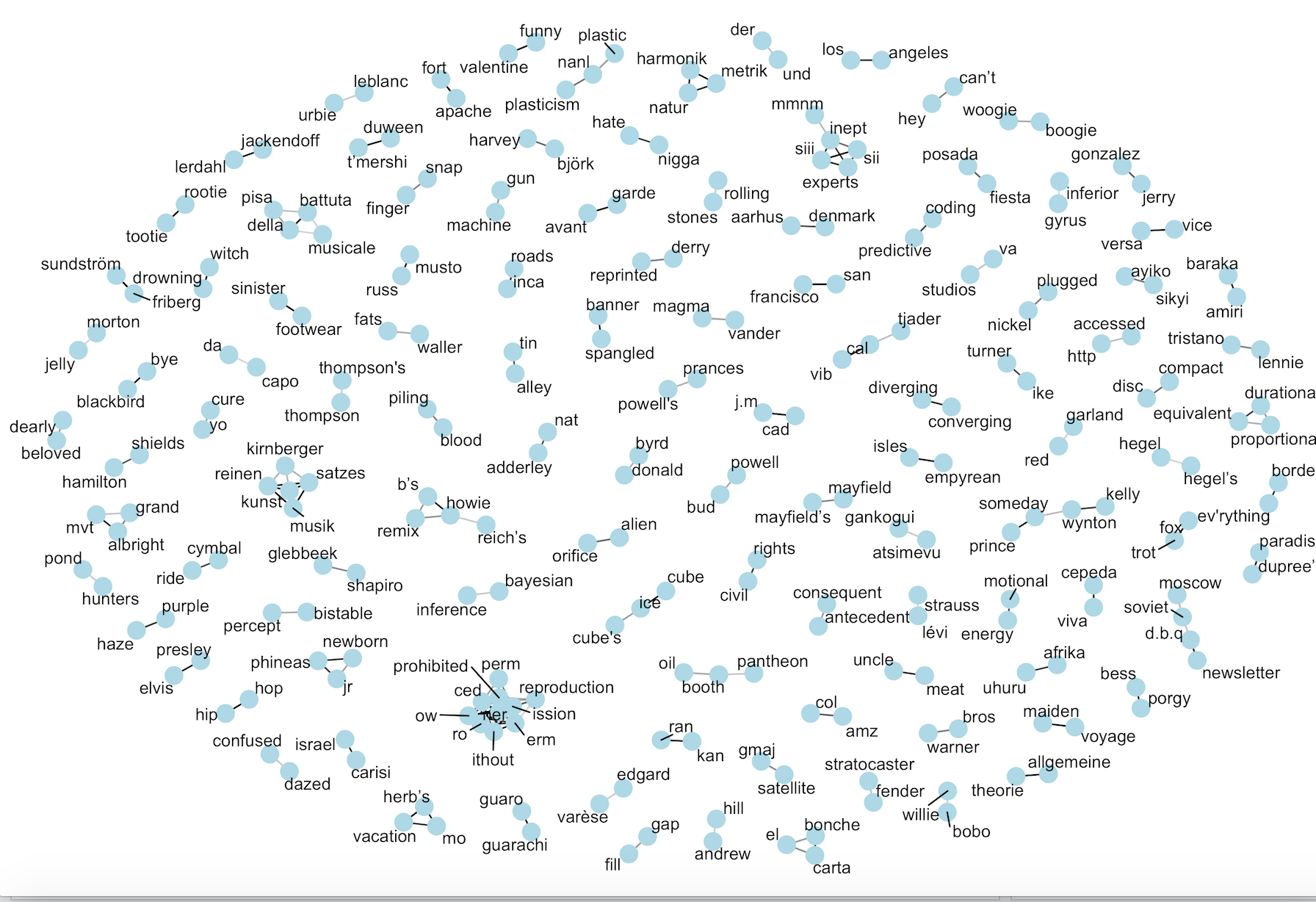
1. Sentimental analysis:
   1. Examine how sentiment changes through a text “Hester\_1997\_The\_melodic\_and” by using three different lexicons. There are 500 words per chunk because the text is really long. 
      1. The three different lexicons for calculating sentiment give results that are different in an absolute sense. AFINN and Bing et al. seems to have similar trends. The NRC lexicon gives the largest absolute values, with high positive values. The lexicon from Bing et al. has lower absolute values and seems to label larger blocks of contiguous positive or negative text. The NRC results are shifted higher relative to the other two, labeling the text more positively, but detects similar relative changes in the text.
   2. Find how much each word contribute to each sentiment in Sample Corpus, we use this to spot anomaly in the sentiment lexicon. This plot reveals the words that contribute to the positive and negative scores mostly in Sample Corpus. 
      1. We found there were many musical-related words that should be classified as neutral words but classified as positive or negative words by using provided lexicons.
      2. Solution: add inappropriately classified words into a custom stop-words list (Appendix C).
   3. Most common positive and negative words across corpus in a wordcloud: <https://app.box.com/file/332963759774>.

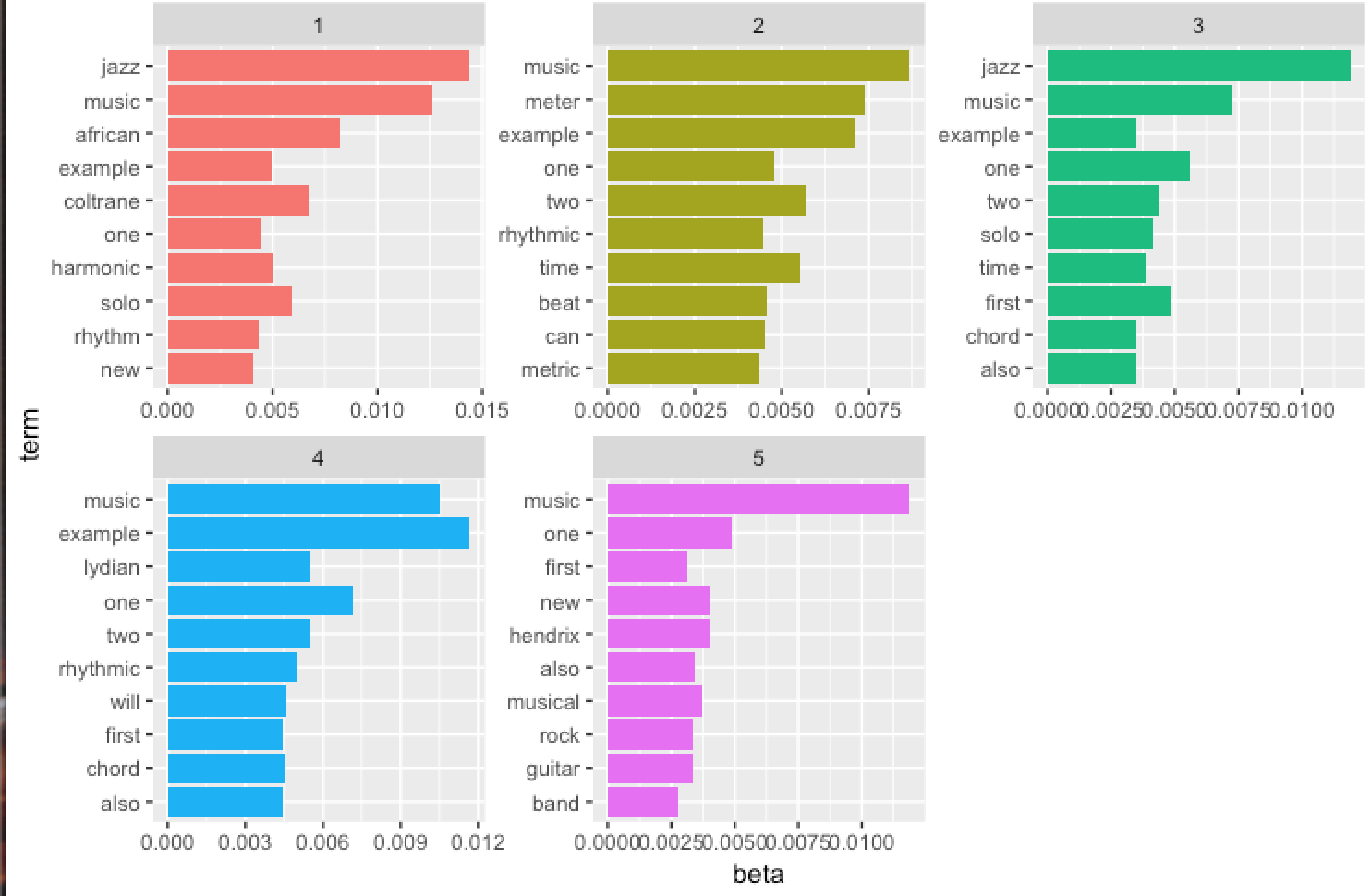
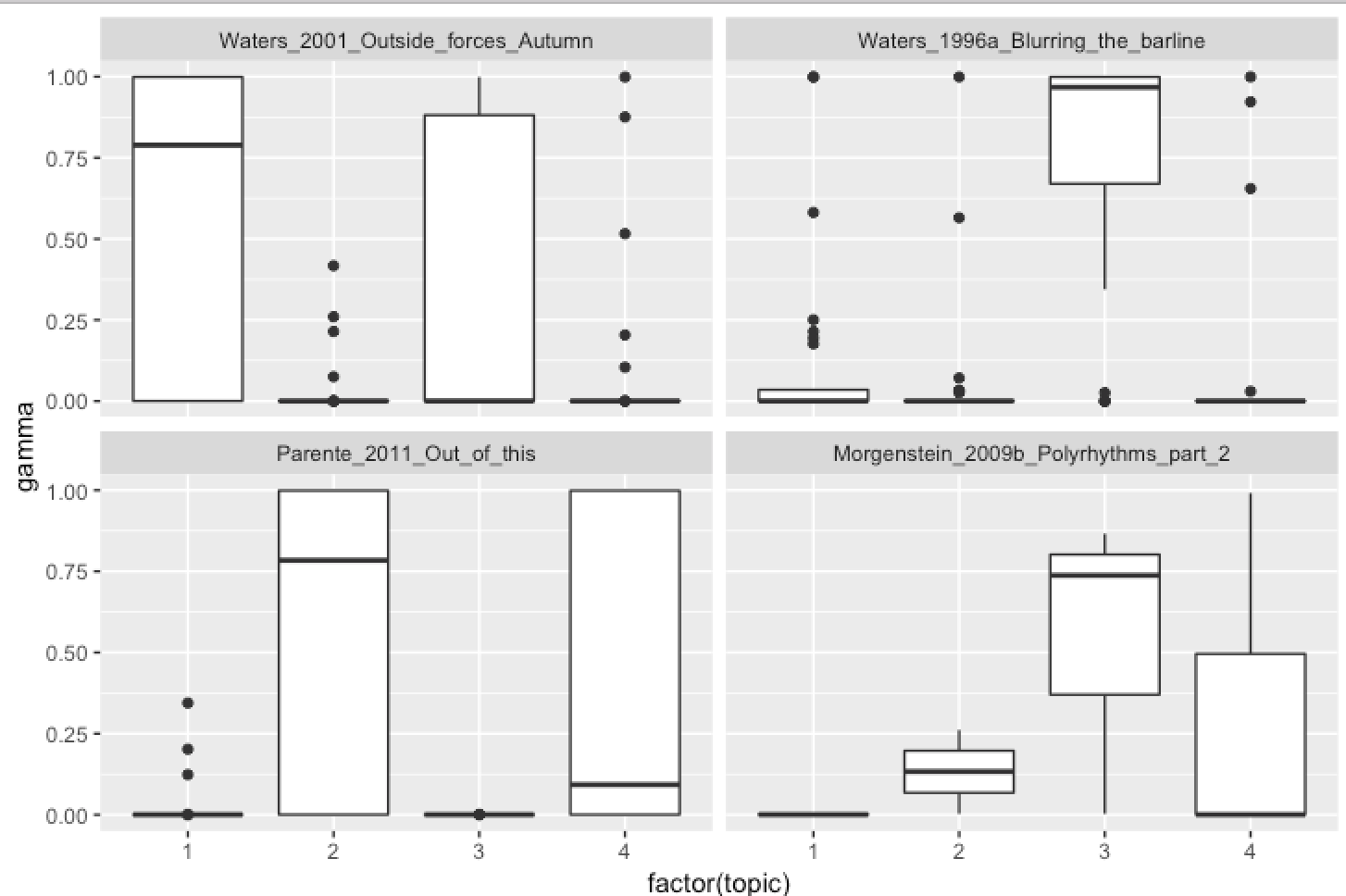
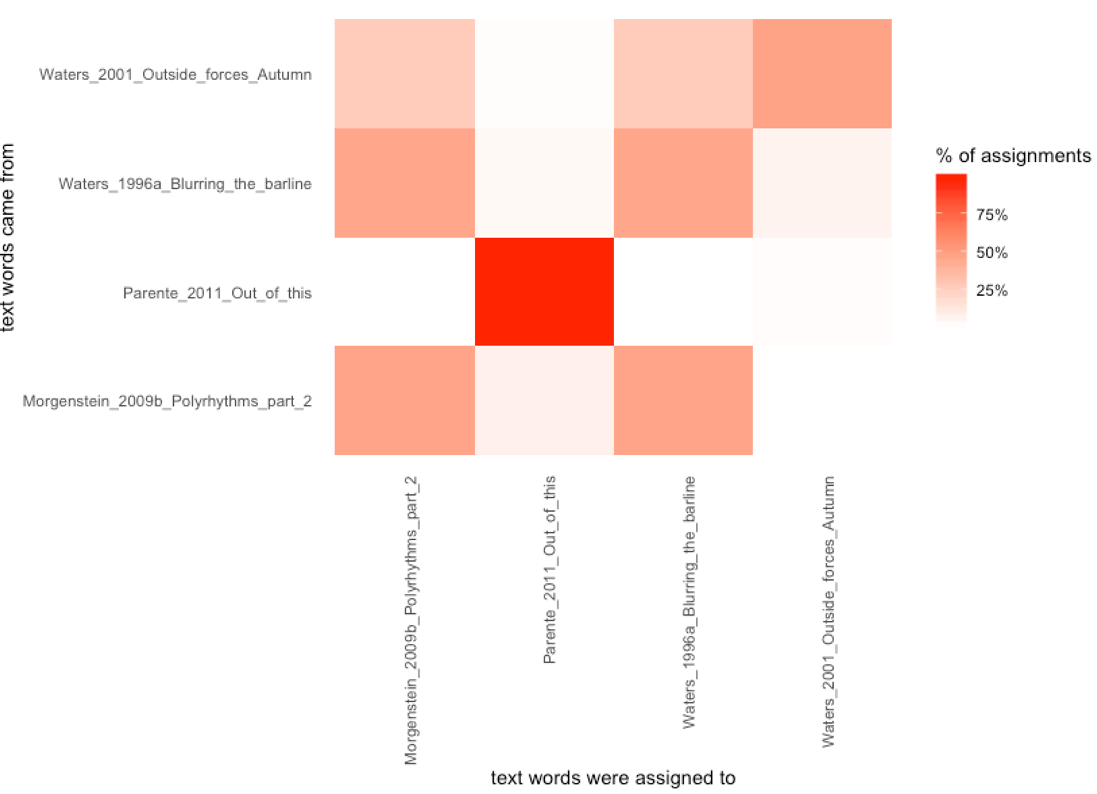
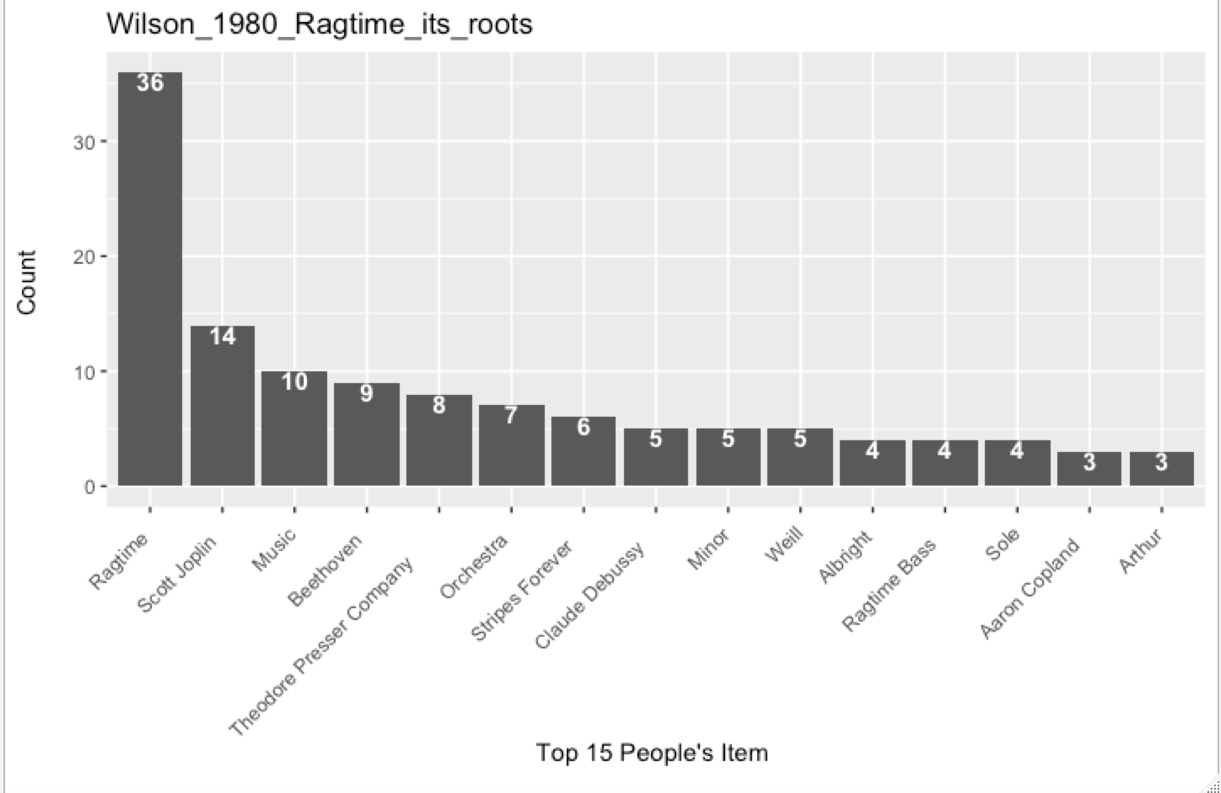


* 1. Find the most negative texts in the corpus. These are three texts with the saddest words across corpus, normalized for number of words in each text. 
  2. Explore sentimental related bigrams with negation. The 20 words preceded by ‘not’ that had the greatest contribution to sentiment scores in corpus, in either a positive or negative direction: <https://app.box.com/file/333021518807>. The most common positive or negative words to follow negations such as ‘never’, ‘no’, ‘not’, and ‘without’ in “common\_words\_follow\_negations.png”: <https://app.box.com/file/333013552061>. 

1. Word Relationship Analysis:
   1. Common bigrams in Corpus, showing those that occurred more than 100 times and where neither word was a stop-word and numbers in a network graph in the “common\_bigrams\_in\_corpus\_network\_graph.png”: <https://app.box.com/file/333017021359>. 
      1. Some details of the text structure can be visualized. For example, we see that “music”, “dissonance”, “note”, “solo” and “scale” form common centers of nodes. “Music” was followed by different genre or descriptive words such as “classical”, “rap”, “popular”, “folk”, “African”. We also see pairs or form individual names or proper nouns (“elvin jones”, “bill evans gil”, or “rolling stones”)
   2. Common pairs of words co-appearing within the same section (150 words/section) in “Sargeant\_1975\_Jazz\_hot\_and”: 
      1. The most common pair of words in a section is “music” and “jazz”. We can easily find the words that most often occur with “music”: jazz, musical, negro, type, American, hot, European, popular, characteristic, time.
   3. Find the words that most often occur with “rhythm” in corpus:
   4. Words were most correlated with “multiplicity”, “music” and “rhythm” in corpus: <https://app.box.com/file/333066642567>.



* 1. The network of correlated words greater than 0.7 coefficient that appearing within the same section in the “correlation\_network\_wordpairs\_corpus.png”: <https://app.box.com/file/333067822624>. 

1. To see where the certain texts rank in terms of average word use, I ranked their mean values in descending order. Large mean value means less lexical richness, and it means more repetition on the same word type. Small word value means high richness of vocabulary.
   1. Largest three mean values texts:
      1. Clement-2009-A\_study\_of\_the\_instrumental\_music\_of\_Frank\_Zappa: 29.726794
      2. Levy-2012-Harmonic\_and\_rhythmic\_interaction\_in\_the\_music\_of\_John\_Coltrane: 16.969869
      3. Wallmann-2010 The\_music\_of\_Herbie\_Hancock\_Composition\_and\_improvisation\_in\_the\_Blue\_Note\_years: 13.197486
   2. Smallest three mean values texts**:**
      1. Condit-SchultzArthur-2014-Beat\_and\_switch-Multi-stable\_rhythms\_metric\_ambiguity\_and\_rock\_&\_roll\_fake-outs: 1.632911
      2. Condit-Schultz-2013-A\_music\_theory\_of\_flow-The\_musicality\_of\_rap\_delivery:1.686747
      3. Peisner-2008-In\_the\_studio-Head\_injury\_inspires\_new\_melodic\_Mastodon\_album: 1.689655
2. Correlation between text length and the number of hapax is 0.5744496. The correlation is mild to modest. As text length gets longer, we can see an increase in the number of hapax.
3. Performed unsupervised clustering on authorship by using high frequency words among the corpus. It was using the high frequency words across the corpus. I did not remove stopwords. Three interesting thresholds: 0.075-170 words, 0.05-241 words, 0.025-570 words. I think 0.05 performs the best.
4. Conducted supervised classification for metadata. I Created a “Metadata (Modified)” worksheet: I removed some items in “Gender, Author, Primary 1” or “Style, Primary” column for testing purposes. I used these unknown data as test data. I did not remove stopwords by following the “TAR” book.
   1. Gender of Primary Author
      1. Class: male, female
      2. Results in sheets named “gender-threshold-#words”: Each sheet contains three tables
         1. **Prediction of Model Labels**: the detailed prediction generated from machine for each text in the model, the purpose is to test the accuracy of the model;
         2. **Summary of Model Predication**: The performance of our classification model on known data, it shows the predicted class versus actual class;
         3. **Prediction of Testdata**: Results of the data without gender labels in the "Metadata (modified)" by using trained model.
      3. When threshold set to 0.005, which reduces word set to the 18 most frequent words across corpus. The model is not really accurate. It can hardly classify female authors for known data. The prediction of testdata predicted two of three correctly (male authors), and one incorrectly (female author).
      4. When threshold set to 0.0001, which reduces word set to the 1244 most frequent words across corpus. The accuracy of model improves a lot. It can classify 65% female authors in training data. While the result of prediction is the same as threshold 0.005.
      5. When threshold set to 0.00005, which reduces word set to the 2252 most frequent words across corpus. The accuracy of model improves compared to 0.0001. It can classify 78% female authors in training data. But the prediction of testdata is the same as threshold 0.005.
   2. “Primary Style” of Music
      1. Class: Rap, Jazz, Rock, Metal, Electronic Dance Music, Electronica, Funk, Popular music, Hip-hop
      2. Results in sheets named “style-threshold-#words”: Each sheet contains three tables (Same as mentioned above).
      3. When threshold set to 0.005, which reduces word set to the 18 most frequent words across corpus. The model is not really accurate. It cannot classify Metal for training data. In the “Prediction of Testdata”, the model predicated that “Butler\_2006a\_Conceptualizing\_rhythm\_and” (its primary style is Electronic Dance Music) is most likely Electronic Dance Music, or Jazz, and it has small chance to be Rock. For a Jazz text - “Love\_2012\_An\_approach\_to”, the model predicted it to be Jazz. For a Rock text - “Temperley\_1999\_Syncopation\_in\_rock”, the model predicted it to be either Rock or Jazz.
      4. When threshold set to 0.0001, which reduces word set to the 1244 most frequent words across corpus. The accuracy of model improves a lot. It can mostly classify every style in training data. The prediction of testdata indicated that “Butler\_2006a\_Conceputalizing\_rhythm” is most likely Electronic Dance Music, and “Love\_2012\_An\_approach\_to” is Jazz, which are correct. But it indicated that Temperly\_1999\_Syncopation\_in\_rock is most likely Jazz, which is incorrect because this text is supposed to be Rock.
5. Topic Modeling
   1. Implement topic modeling in “Sample Corpus” by using LDA (TMR, Ch6). Set topic number to 5.
   2. Extract the per-topic-per-word probabilities. Visualize the most common 10 terms within each topic in “top\_10\_terms\_in\_5\_topics.png”: <https://app.box.com/file/333100090697>. 
   3. Application: Re-cluster disorganized sections from randomly picked four texts of corpus into four groups: Parente\_2011\_Out\_of\_this, Waters\_2001\_Outside\_forces\_Autumn, Morgenstein\_2009b\_Polyrhythms\_part\_2, Waters\_1996a\_Blurring\_the\_barline. First divide each text into multiple sections, each with 150 tokens. Treat each section as a document. Find document-word counts and convert word counts tidy form into dtm. Then create a four-topic model.
      1. Examine per-topic-per-word probabilities. Find top 10 terms within each topic: “recluster\_top\_10\_words\_in\_4\_topics.png” <https://app.box.com/file/333092386830>.
      2. Conduct per-document classification. The per-document-per-topic probabilities for each section within each text: “recluster\_prob\_each\_section\_within\_each\_text.png” <https://app.box.com/file/333100433579>. 
      3. Assign each word in each document to a topicConfusion matrix showing where LDA assigned the words from each text. Each row of this table represents the true text each word came from, and each column represents what text it as assigned to: <https://app.box.com/file/333126433120>.
         1. We notice that nearly all of words from “Parente\_2011\_Out\_of\_this” were correctly assigned, and most of “Waters\_2001\_Outside\_forces\_Autumn” were correctly assigned. While the other two texts had a fair number of misassigned words.
         2. Most common misclassified words: metric, bar, form, harmonic, meter (Some common musical technical terms)
6. Tsig Extraction (Appendix E).
7. Proper Noun Extraction
   1. Use Stanford Named Entity Recognizer to extract “people”, “location”, “organization”.
      1. Find “people’s item from NER” that partially match “capitalized words” in “Wilson\_1980\_Ragtime\_its\_roots”: <https://app.box.com/file/328297476440>.
      2. Generate plots of most 15 frequent items in “Wilson\_1980\_Ragtime\_its\_roots”: 
      3. Problem:
         1. NER is designed to recognize different kind of objects in English, we might not extract non-English names.
         2. NER will misclassified proper nouns as people’s name such as “Music”, “Ragtime”.
      4. Suggestion: train custom Stanford NER model to recognize non-English person names or other categories (Appendix D).

**Hesselink Findings Summary**

**References**

Silge, J., & Robinson, D., (2018). *Text Mining with R*. Retrieved from <https://www.tidytextmining.com/>

Jockers, M. L. (2014). *Text analysis with R for students of literature*. New York: Springer.

kwic. (n.d.). Retrieved from <https://www.rdocumentation.org/packages/quanteda/versions/1.3.4/topics/kwic>

Murphy, P., (2017, April 4). *Basic Text Mining in R*. Retrieved from <https://rstudio-pubs-static.s3.amazonaws.com/265713_cbef910aee7642dc8b62996e38d2825d.html>

Stanford Named Entity Recognizer (NER). (n.d.). Retrieved from <https://nlp.stanford.edu/software/CRF-NER.shtml>

Code Repository: In the “Code” folder - <https://app.box.com/folder/53573270000>.