Group 8
Saurabh Donthin
Jamie Wheeler

Computer Blue
An Analysis of Prince's 80's Music

1. Problem description

Our project investigated the music artist, Rogers Prince Nelson, also known as Prince. Only albums that Prince produced during the eighties were considered. The reason for doing so was that the eighties was when most of Prince's iconic albums were released. In other words, there was a higher probability of understanding which song would be a hit and which song would not be a hit, simply because of the number of songs that were a hit during the 1980s. The purpose was to understand if there were any defining patterns and critical features in his songs that could be used to determine if the song would be a hit or not and if there is a relation between the sentiment of his songs and the song being a hit or not. Lastly, all other endogenous metrics were investigated and modeled to see if there was any predictive power from them for the if a song was a hit and to what extent.

The ability to classify songs can promote subscriber retention or value to users for streaming and music recommendation systems. The ability to identify factors of a hit is valuable for artist and record label song promotion and identifying potentially new and successful artists. Further, the ability to quantify things such as sentiment and album structure may offer new art or cultural study of an artist, song, album, and/or genre. Typically, these studies have been done almost solely by opinion, case study, and relative comparison without much quantification.

2. Data Description

There was no dataset to begin with, and had to gather all data from primary or solid secondary sources. What follows is a description of the different variables considered, the method that was used to extract the data, and the data cleaning methods that were used to create the dataset.

Variable Name: Song name	**Source:** Wikipedia
Variable Description: Name of the song	
Extraction Method: Each album was searched on google the Wikipedia entry for the album located. Next, the song name, the position on the album, and the total song length were extracted using import.io*.	
Cleaning Operations Performed: The Wikipedia entries had a lot of special characters that were also imported and had to be removed using Microsoft Excel's find and replace function.	

Variable Name: Album Name	**Source:** Wikipedia
Variable Description: The name of the album in which each song is present	
Extraction Method: Each album was searched on google the Wikipedia entry for the album located. Next, the song name, the position on the album, and the total song length were extracted using import.io*.	
Cleaning Operations Performed: The Wikipedia entries had a lot of special characters that were also imported and had to be removed using Microsoft Excel's find and replace function.	

Variable Name: Position on Album	**Source:** Wikipedia
Variable Description: The position of the song in the Album	
Extraction Method: Each album was searched on google the Wikipedia entry for the album located. Next, the song name, the position on the album, and the total song length were extracted using import.io*.	
Cleaning Operations Performed: The Wikipedia entries had a lot of special characters that were also imported and had to be removed using Microsoft Excel's find and replace function.	

Variable Name: In Hot 100 Billboard	**Source:** http://www.billboard.com/artist/351039/prince/chart
Variable Description: If the song was in the Hot 100 = yes, if not = no	
Extraction Method: The language that was used to extract these values was Python. The packages that were used were urllib2 and lxml. The name of the song and the position were extracted.	
Cleaning Operations Performed: There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information needed.	

Variable Name: Position on top 100 billboard	**Source:** http://www.billboard.com/artist/351039/prince/chart
Variable Description: The highest possible position in the hot 100 billboard list. If not, it was given a value of 0.	
Extraction Method: The songs were extracted from http://www.billboard.com/artist/351039/prince/chart . The language that was used to extract these values was Python. The packages that were used were urllib2 and lxml. The name of the song and the position were extracted.	
Cleaning Operations Performed: There was no cleaning required as Urllib2 and lxml can be used to extract the exact pieces of information needed.	

Variable Name: Year	**Source:** Wikipedia
Variable Description: The year in which the song was released.	
Extraction Method: The year in which the album was extracted manually. There was no need to automate the process as the number of albums was limited.	
Cleaning Operations Performed: No cleaning operations were necessary.	

Variable Name: Song length in seconds	**Source:** Wikipedia
Variable Description: The song length in seconds	
Extraction Method: The song length was extracted from Wikipedia. After extraction, the total time was converted from Min:Sec format to seconds by performing a basic conversion in Microsoft Excel.	
Cleaning Operations Performed: No cleaning operations were necessary.	

Variable Name: BPM Value	**Source:** https://www.cs.ubc.ca/~davet/music/track/PRINCE___PRN/, https://jog.fm/workout-songs/by/prince?order=desc&sort=bpm,
Variable Description: The BPM (Beats per minute) value of the song.	
Extraction Method: The BPM values were extracted using multiple methods, because all of them were not available at one place. Some were extracted using import.io, while others were extracted manually because each song's BPM value had to be searched for.	
Cleaning Operations Performed: For the values extracted using import.io, manual cleaning was performed in Microsoft Excel to remove special characters.	

Variable Name: Key	**Source:** https://www.cs.ubc.ca/~davet/music/track/PRINCE___PRN/, http://prince.org/msg/7/124627, https://beatportcharts.com/track/1701012/
Variable Description: The key of the song.	
Extraction Method: The key of each song was not available at a single location, so a combination of import.io and manual extraction was used.	
Cleaning Operations Performed: The values obtained through import.io were manually cleaned in Microsoft Excel for removing special characters.	

Variable Name: Lyrics	**Source:** www.princelyrics.co.uk
Variable Description: The lyrics for each song.	
Extraction Method: The lyrics were extracted using import.io* from princelyrics.co.uk. The song lyrics were not in a format which allowed them to be processed.	
Cleaning Operations Performed: Stopwords like "I", "me" etc were removed. Special characters like ".",",","!" were also removed using basic find and replace operations in excel.	

Variable Name: Word Count	**Source:** www.princelyrics.co.uk
Variable Description: The word count for each song.	
Extraction Method: The word count was extracted from the lyrics. First, the excel transpose function was used to put all of the song lines into a single cell. Then the formula =IF(LEN(TRIM(A2))=0,0,LEN(TRIM(A2))-LEN(SUBSTITUTE(A2," ",""))+1) was used to get the word count.	
Cleaning Operations Performed: Stop words and special characters had been removed.	

Variable Name: Sentiment	**Source:** www.princelyrics.co.uk
Variable Description: The overall sentiment of the song. Process explained in in 3. Sentiment Analysis	
Extraction Method: Explained below in 3. Sentiment Analysis.	
Cleaning Operations Performed: Explained in 3. Sentiment Analysis	

Variable Name: Category	**Source:** www.princelyrics.co.uk
Variable Description: Based on the overall sentiment value, the song is classified into one of four categories	
Extraction Method: The basis for the category value was the sentiment score that was obtained, which means that the data extraction had already been done.	
Cleaning Operations Performed: No cleaning operations were necessary.	

*Note: Import.io is a standalone desktop application through which it is possible to extract data from websites without any code necessary. It is also possible to save the functionality as an API so the same operations can be performed on multiple web pages by teaching the application to extract data by highlighting examples by which the software understands and generalizes from these examples to extract data from the web pages.

3. Sentiment Analysis

Lyrics were saved for each song in a separate csv file after each song was cleaned by removing all the stop words and special characters. Next, the file was loaded into the workspace in R. A corpus of positive and negative keywords was loaded into the database and compared the lyrics against the corpus. For every positive word that the lyrics had, score was increased by +1. The same was done with the negative words corpus by giving a score of -1. The total score was calculated by summing the positive and negative score then a net sentiment score was calculated by dividing the sentiment score with the total number of words meaning that sentiment can only be between -100 and +100.

The sentiment rating method here works very well within a single artist's song library as the assumption that their writing sentiment for any given song is relative to their writing sentiment for all their other songs. The sentiment rating method may not work for comparing an artist to a different artist without finding some method to normalize their lyrics.

4. Data Modeling

The data was modeled using a series of methods including logistic regression, liner regression, classification, and decision tree. For logistic regression, it was attempted to identify if it was a in the hot 100 or not. Linear regressions were performed for other album position, chart position, BPM, song length, and word count.

Every combination of dependent variable was attempted and very little that was statistically significant was found and even when it was statistically significant, it was of minor impact. The R code is provided in the appendix. The methods were also attempted with Taylor Swift songs from a couple of albums and more was found that was statistically significant, so it is likely that the models are fine. It is simply that there are too many exogenous variables and potentially randomness to make it feasible to model these factors for Prince's song writing.

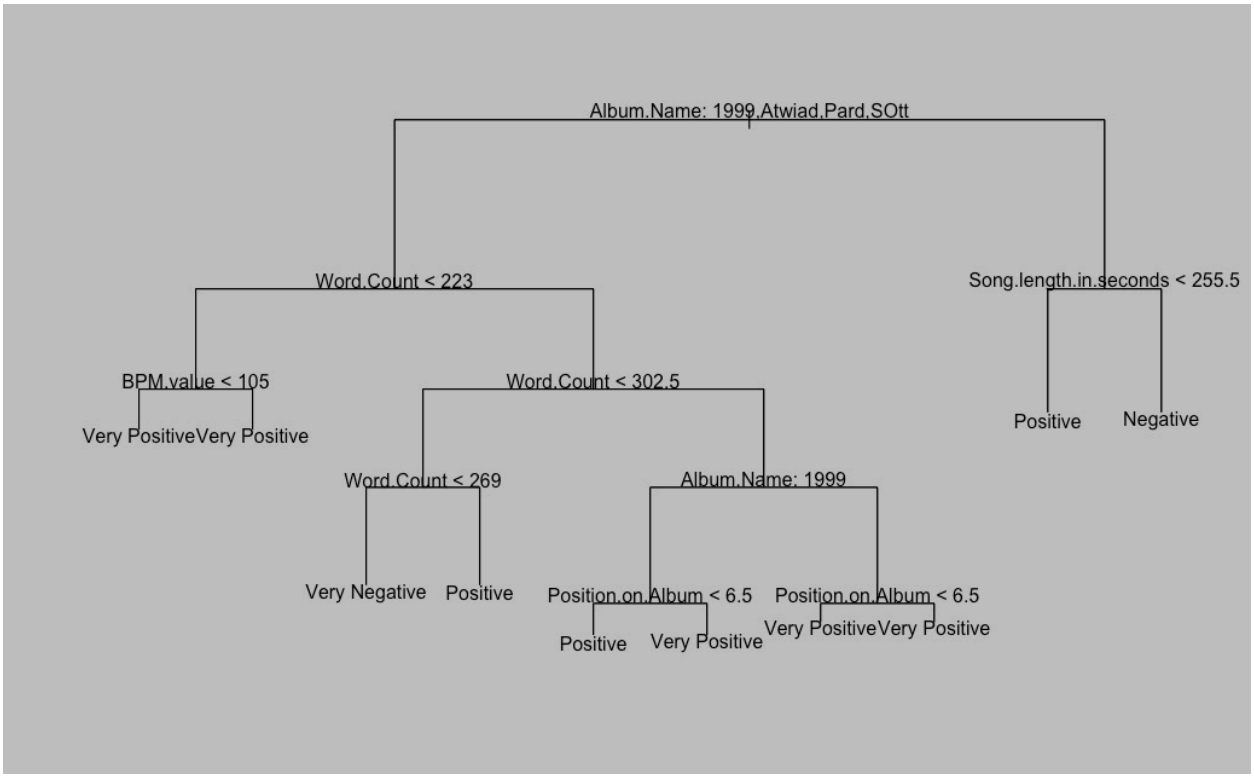
A classification model was also created with the dependent variable of In.100.Hot.Billboard. The dataset was separated into a training and validation set at 60% and 40% respectively. The results were:

	Observation		**Measures**
Prediction	**0**	**1**	Accuracy: 48.15%
0	1116	187	Sensitivity: 18.75%
1	11	20	Overall Error: 59.26%

The mediocre accuracy, low sensitivity, and high error rate indicate the classification model is not very good at predicting if a song would be a hit.

The decision tree gave us the best model of deciding if his songs would be a hit. Early models crashed R. It is believed the crash is due to his song names are almost as unique as the key the song is in making too many viable branches. Keys were removed and the buckets created

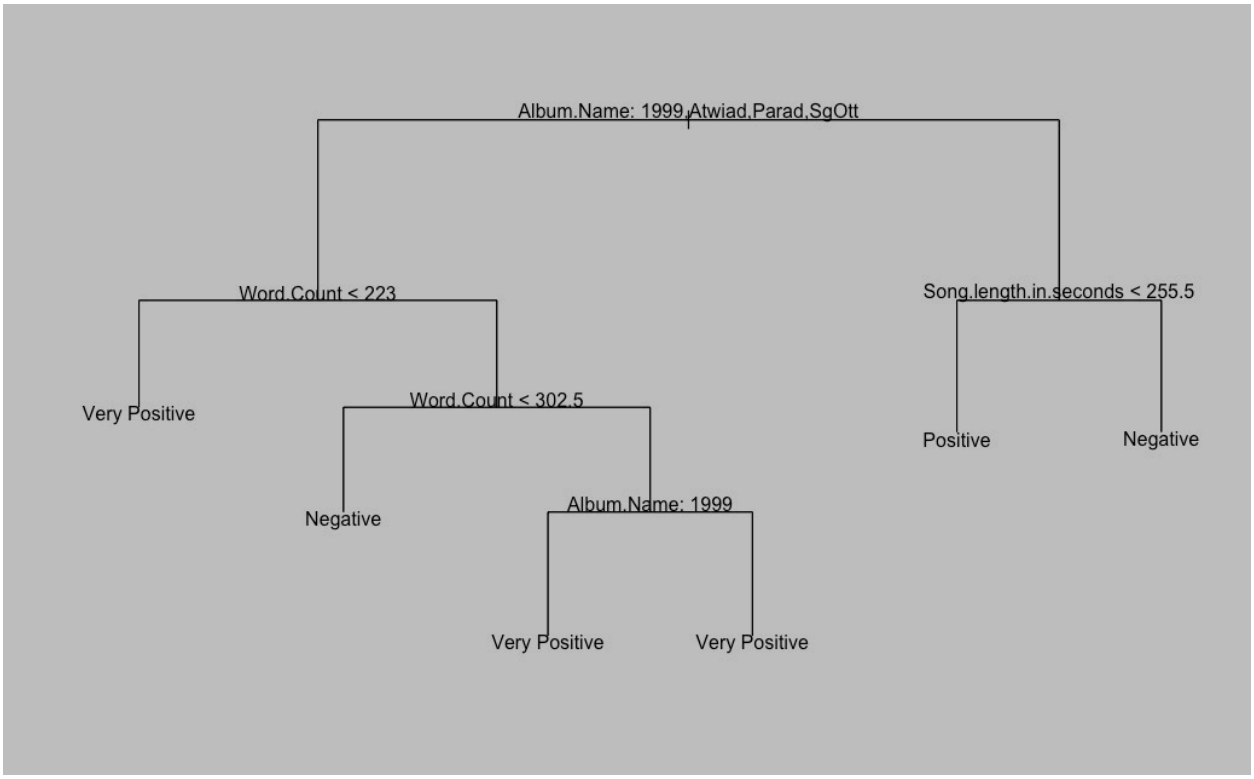
which classified the sentiment as very negative (-100 to -51), negative (-50 to -1), neutral (0), positive (1 to 50), or very positive (51-100). The decision tree gave very clear results:



Unpruned Decision Tree

Decision trees appear to be good at identifying if a song was a hit how one could identify it, unfortunately, given the dependence upon album, it is not a very good predictive tool. It is reasonable that album would be the strongest predictor of past data since there are albums with clearly higher sentiment than others.

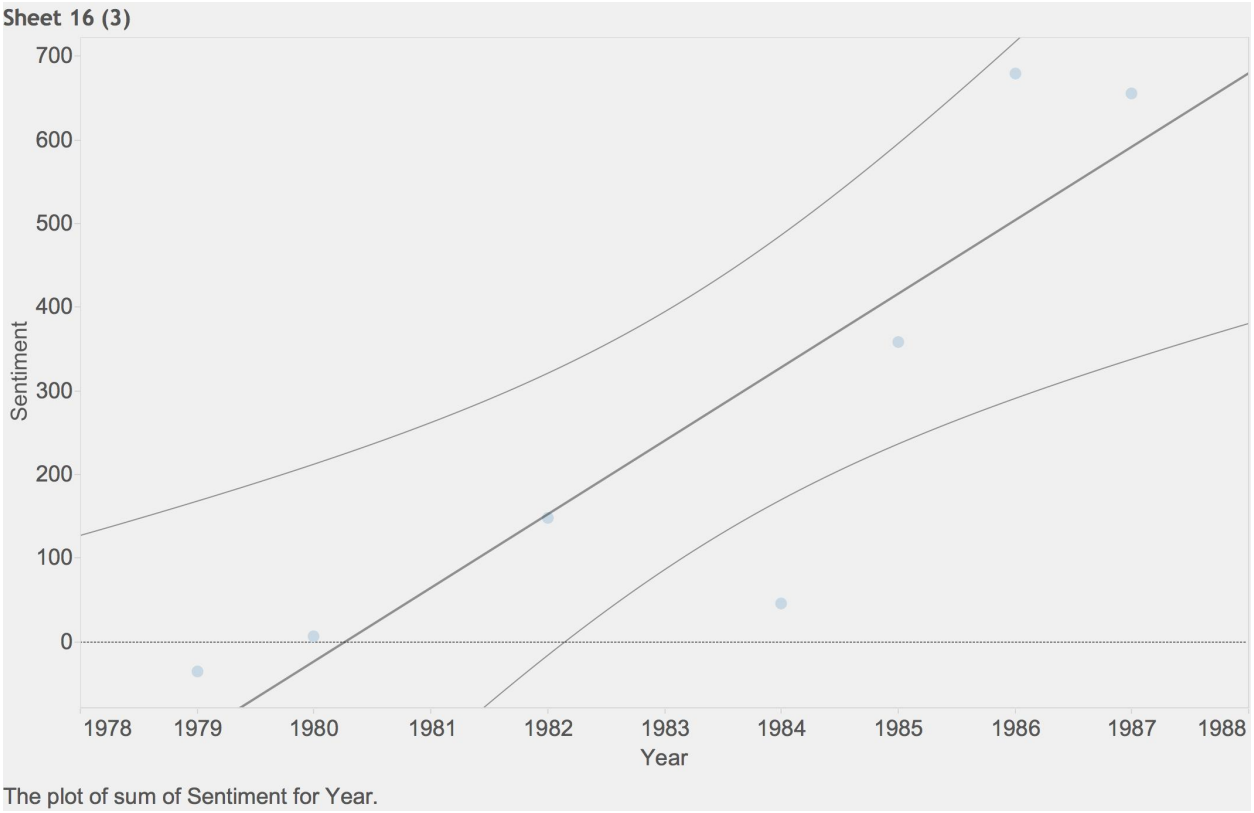
The pruned tree informs us of something interesting as well.



Pruned Decision Tree

The most interesting part of the pruned decision tree is that albums other than *1999, Around the World in a Day*, and *Sign O the Times* can be categorized almost strictly by song length in seconds indicating they may not have been as diverse. Also, it appears that there was a preference for shorter songs to be positive. It should be noted that the node "Album Name: 1999" has two branches that both terminate for very positive sentiment and is in fact correct as all other options would have been diverted by other branches.

Finally, one relationship was identified within the data set. As year increased, Prince's writing became more positively sentimental. With a p-value of 0.009, well less than the 0.05 significance level, and an R-squared of 77.5, the trend is both statistically significant and a good fit. The next page shows the sentiment, year, and line of best fit.

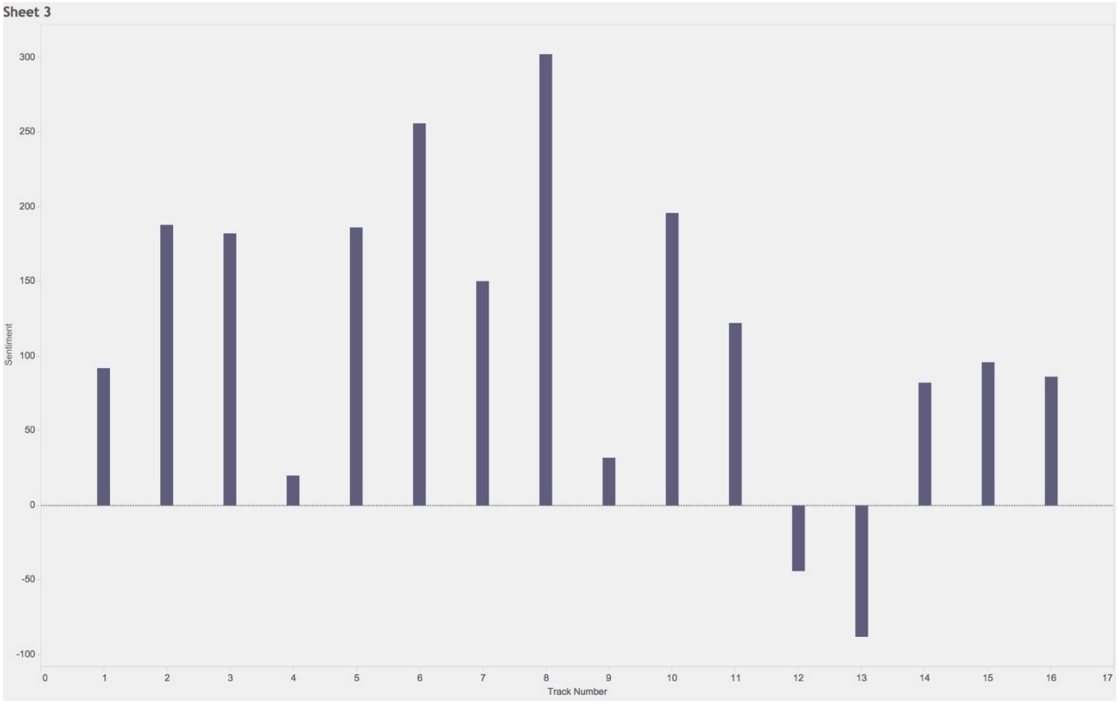


Sentiment by Album Year

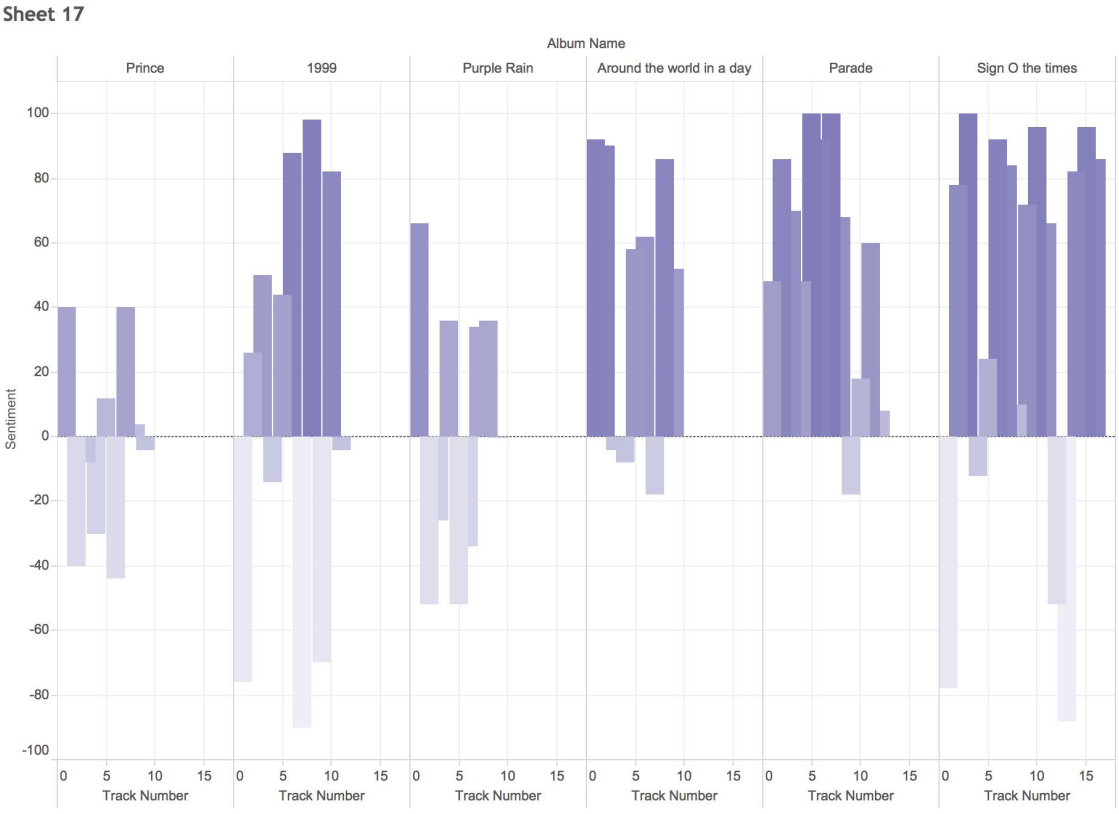
5. Visualization and Analysis

When the sentiment analysis was performed, visualizations were applied to analyze potential patterns related to the sentiment. There appeared to be a general trend toward increasing sentiment that dropped off after track 8. However, the higher track numbers are a bit misleading because there are only three albums have more than nine tracks and only one has more than twelve tracks.

When sentiment rating is separated by album, there is a slight trend toward increasing sentiment as the album continues with his lower sentiment songs clearly being more skewed toward the beginning of the album. Additionally, a pattern of overall album sentiment appearing with there being two eras can be observed; an early neutral sentiment era and a later positive sentiment era.

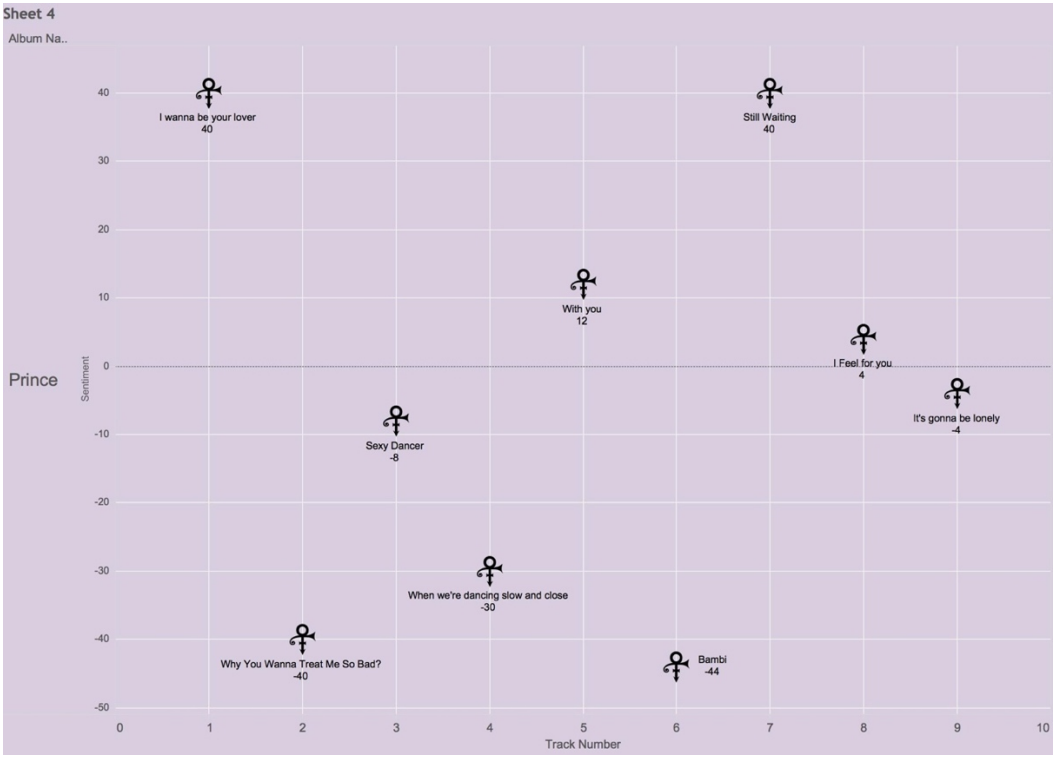


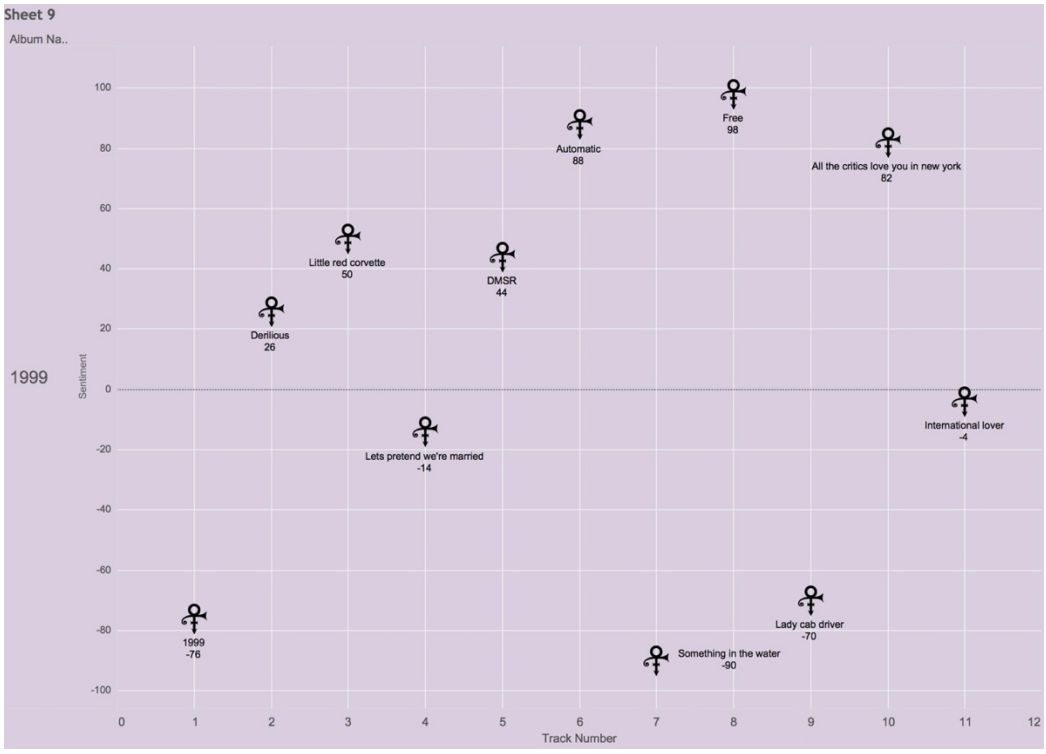
Song Sentiment by Track



Song Sentiment by Album and Track

A weakness of the above graphs is the inability to easily identify neutral sentiment. The following visualizations were done to identify any neutral sentiment songs on albums.





From the above three visualizations that there are in fact some songs that are nearly neutral sentiment can be observed. The one that stands out the most, though, is "Purple Rain". "Purple Rain" has a sentiment rating of 0. The result was unexpected due to the results of the sentiment analysis. While most sentiments were rounded to the nearest integer, they were at least four significant figures. "Purple Rain" is not rounded. It is actual zero. A further analysis of "Purple Rain" was done to explore the curiosity.

I never meant to cause you any **sorrow**
I never meant to cause you any **pain**
I only wanted to one time to see you **laughing**
I only wanted to see you
Laughing in the purple rain

[Chorus]

I never wanted to be your weekend **lover**
I only wanted to be some kind of **friend**
Baby, I could never **steal** you from another
It's such a **shame** our friendship had to end

Positive Sentiment Word

Negative Sentiment Word

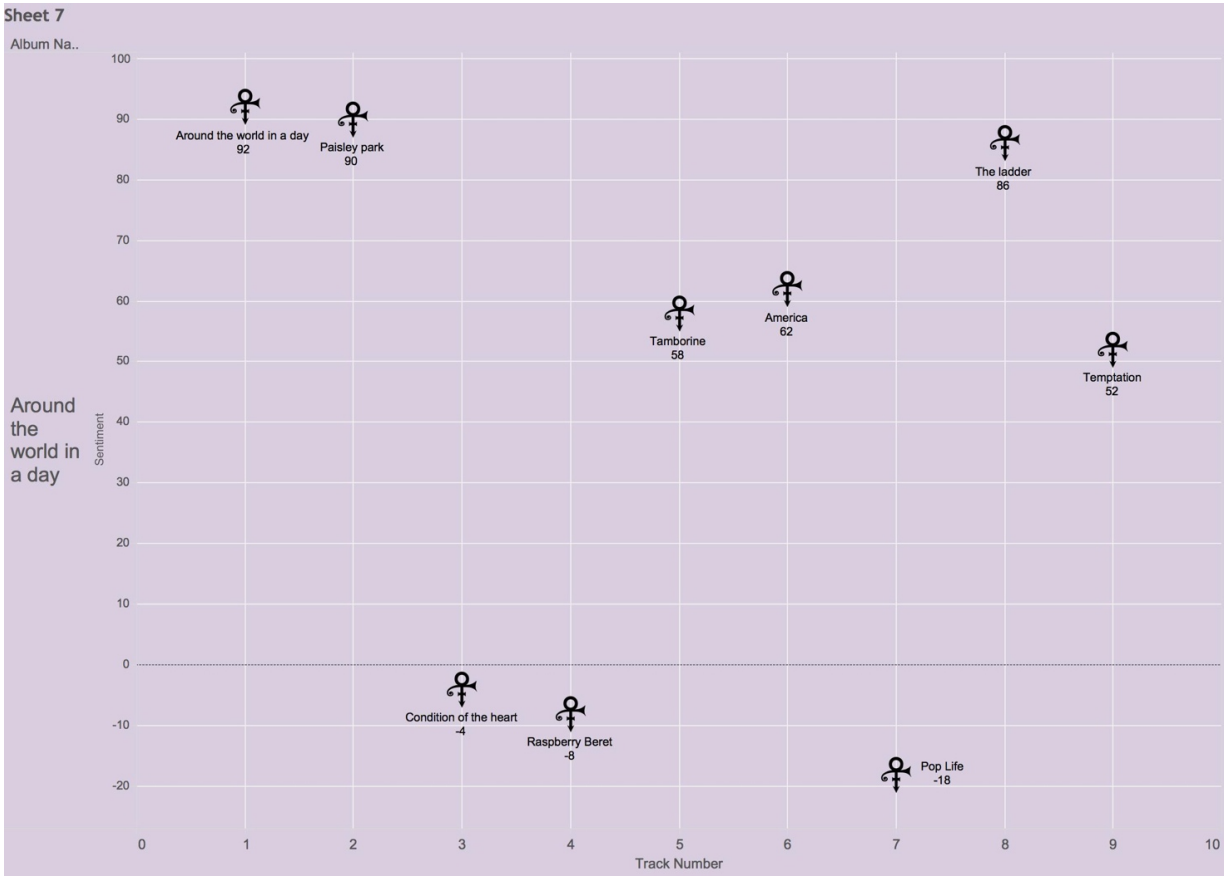
-sorrow + -pain + laughing + Laughing + lover + friend + -steal + -shame =

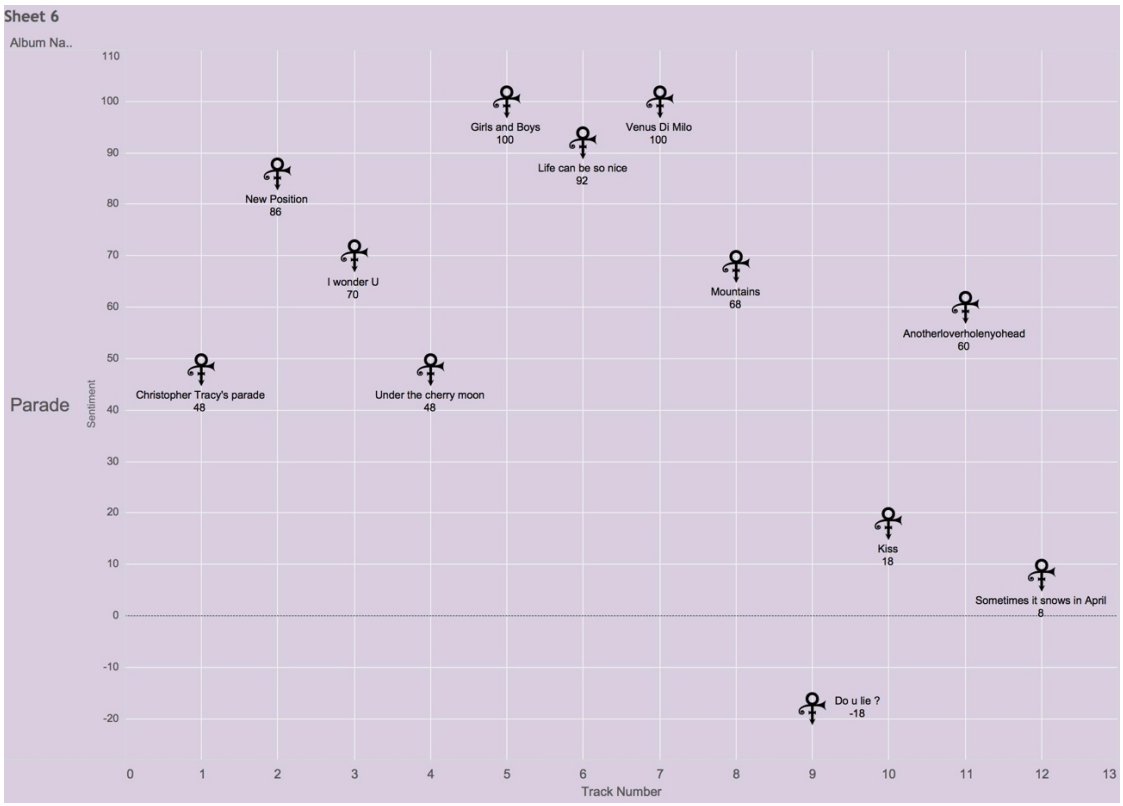
-1-1+1+1+1+1-1-1 = 0

Not only is “Purple Rain” sentimentally neutral, it is sentimentally symmetrical. It is unlikely that perfect symmetry was accidental. It may be that Prince structured intentionally. In fact, it seems likely that some aspects were intentionally structured.

The finding that “Purple Rain” is sentimentally neutral is significant because it appears intentional and it is a completely unique finding.

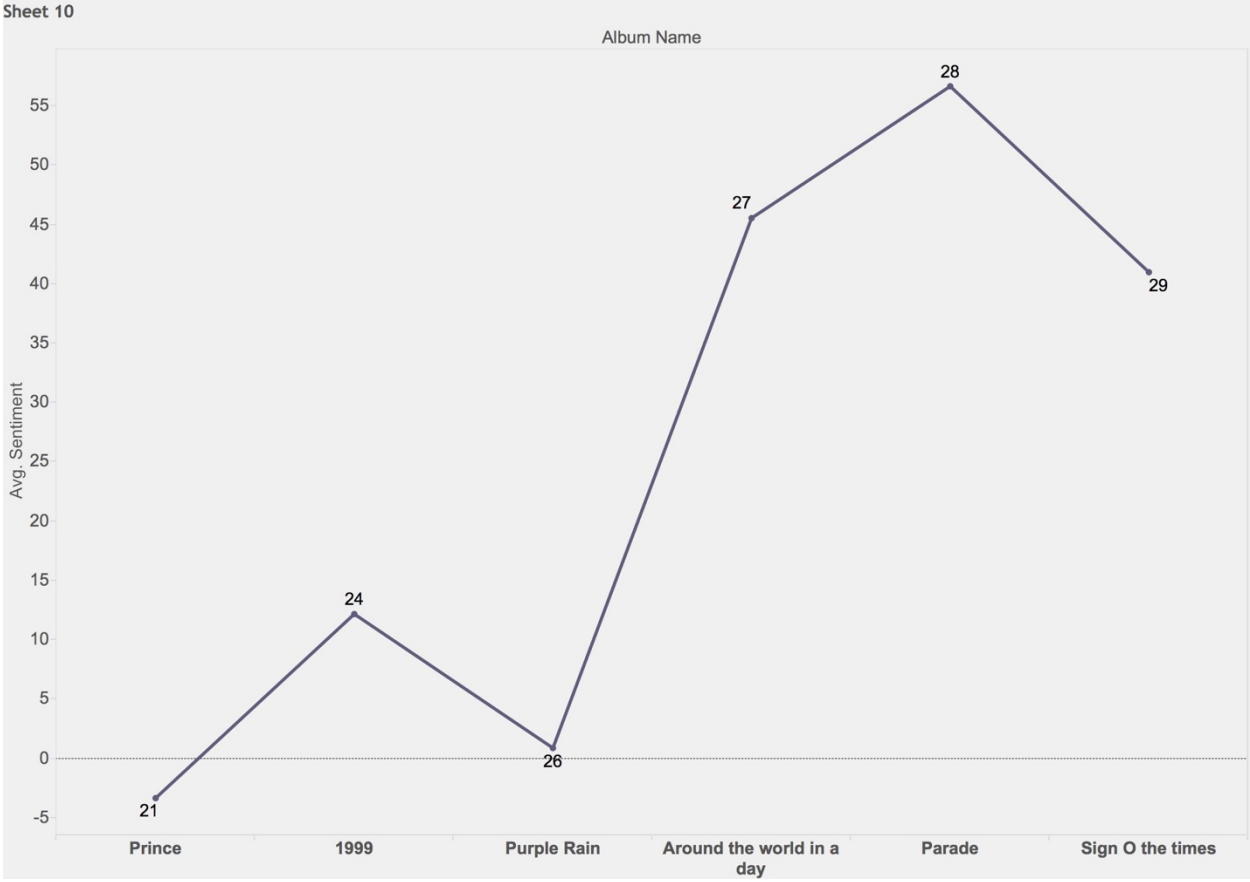
It can be seen in albums after *Purple Rain* that overall sentiment was overall much more positive.




Sheet 6
Album Na..
Parade
Sentiment
Track Number
Girls and Boys
100
Venus Di Milo
100
Life can be so nice
92
New Position
86
I wonder U
70
Mountains
68
Anotherloverholenyohead
60
Christopher Tracy's parade
48
Under the cherry moon
48
Kiss
18
Sometimes it snows in April
8
Do u lie ?
-18

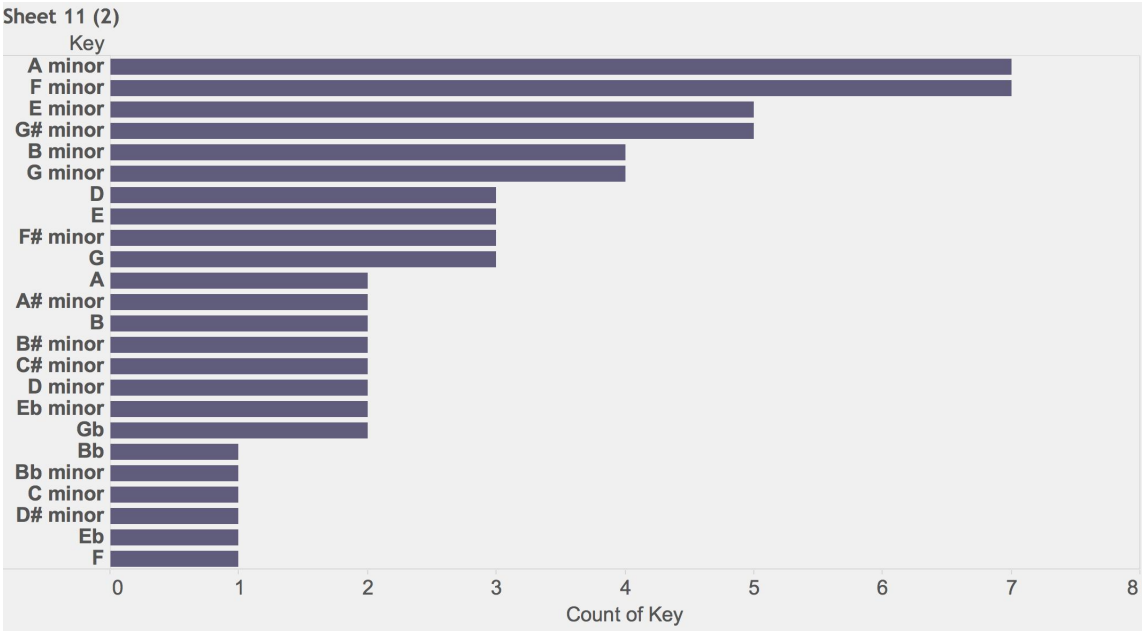

Sheet 5
Album Na..
Sign O the times
Sentiment
Track Number
Housequake
100
Starfish and coffee
92
Its gonna be a beautiful night
96
Play in the sunshine
78
Slow Love
84
The cross
82
Adore
86
Forever In my life
72
It
24
Hot thing
10
The ballad of dorothy parker
-12
Strange relationships
-52
Sign O the times
-78
I could never take the place of your man
-88

The change in overall sentiment from neutral for the first three albums to more positive is best shown in the following visualization:



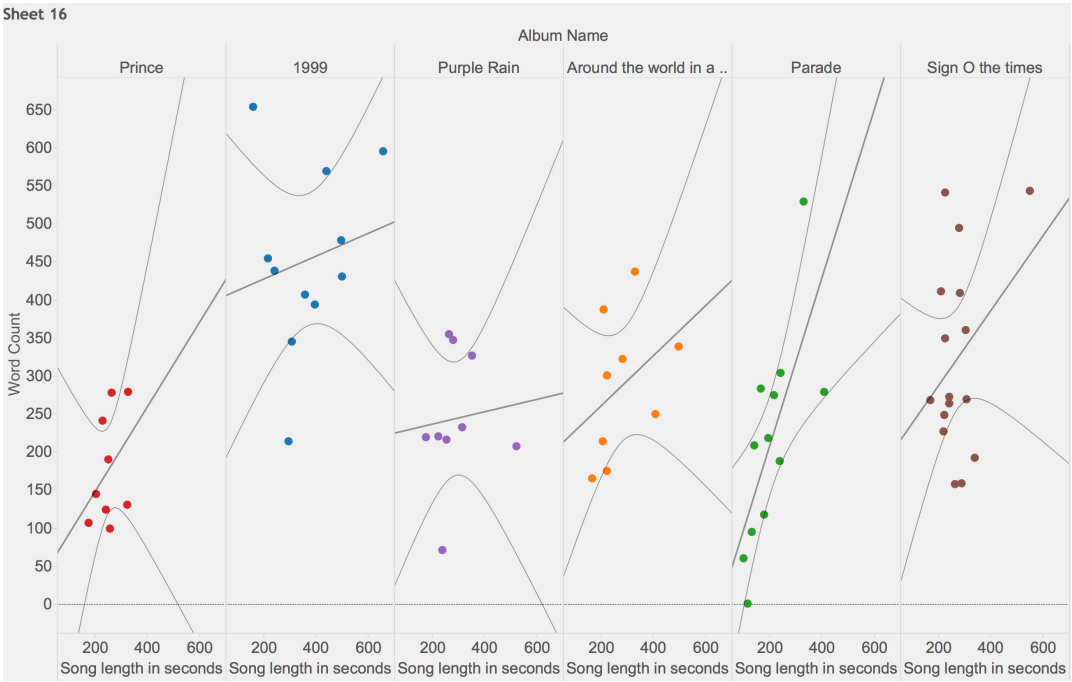
In the above visualization., the two distinct eras in the 80s are far more apparent.

Additional relationships were explored including musical ones. While statistically strong correlations could not be found for Prince's preference for song keys for sentiment or placement on album, what is clear is that Prince did have preferences for certain keys as shown in the visualization shown on the next page:



Prince seemed to have a strong preference for songs in the minor key with 51 of 69 songs in the minor key, his six (6) most preferred key of 26 keys used being minor keys, and 15 of 24 keys used being minor.

Another relationship that was identified through visualization was the unexpected relationship between word count and length of songs as shown in the following visualization:



6. Findings and Conclusion

It appears that to it is difficult to create a predictive model to determine if any of Prince's songs were going to be a hit, at least with the dataset created. However, there were still interesting patterns to be found. Finding that Prince wrote more positively as he got older is very interesting as it may be possible to identify eras in Prince's writing. From era finding and associations, additional models may be made or previous models amended to find more trends. Sentiment eras can be used to investigate what may have been happening in Prince's life at the time to determine some correlation between types of events and the impact upon his song writing.

What is most clear is that given the various nature of Prince's writing, it becomes very difficult to quantify his song writing in such a manner as to predict how the listening audience will receive the song. Prince was chosen because of his great variety and how prolific he was. What was found was a great variety of unique song writing but what else was found is that his song writing may have reflected his state of mind at the time, and this may be measureable and quantifiable.

We are left with interesting questions about the state of mind of Prince as he wrote and a new and unique dataset with which to continue to explore as well as new techniques to explore these relationships with.

The most interesting finding is the neutral and symmetrical sentiment found in "Purple Rain". The finding that it is not only as close to mathematically neutral as a song can be quantified to be, but it is also perfectly symmetrical is not interesting, but moreover, it is a completely unique finding and possibly important finding.

References

Wikipedia. "Prince Albums Discography." Wikipedia. Wikimedia Foundation, n.d. Web. 13 Dec. 2016.

"Prince." Prince - Chart History | Billboard. Billboard Inc., 2016. Web. 13 Dec. 2016.

Tompkins, Dave. "Prince :: Purple Rain [PRINCE___PRN-09]." Https://www.cs.ubc.ca/~davet/music/. University of British Columbia, 2016. Web. 13 Dec. 2016.

Jog.fm. "Running Songs by Prince by BPM." Jog.fm. Jog.fm, 2016. Web. 13 Dec. 2016.

Prince.org. "Songs in the Key of Prince?" Prince.org. Prince.org, 12 Mar. 2004. Web. 13 Dec. 2016.

Beatport Charts. "Prince - Strange Relationship (LP Version)." Prince - Strange Relationship (LP Version). Beatport Charts, 2016. Web. 13 Dec. 2016.

"Prince Lyrics @ Www.princelyrics.co.uk." Prince Lyrics. Princelyrics.co.uk, 2016. Web. 13 Dec. 2016.

```
install.packages("ISLR")
library("ISLR")
install.packages("tree")
library("tree")
install.packages("SDMTools")
library("SDMTools")

df <- read.csv("~/Google Drive/GMU/Current Classes/STAT
515/Final Project/Data Set/prince_csv.csv")
install.packages("plyr")
library("plyr")
install.packages("stringr")
library("stringr")

#load song
Dataset1 = read.csv("~/Google Drive/GMU/Current Classes/STAT
515/Final Project/Data Set/1-1.csv",header = FALSE,sep = ",")
#View(Dataset)

Dataset=c(Dataset1)
View(Dataset)

pos.words <- scan("~/Google Drive/GMU/Current Classes/STAT
515/Final Project/positive-words.txt", what='character',
comment.char=';')
neg.words <- scan("~/Google Drive/GMU/Current Classes/STAT
515/Final Project/negative-words.txt", what='character',
comment.char=';')
pos.matches <- pmatch(Dataset, pos.words)
```

```
neg.matches <- match(Dataset, neg.words)
View(pos.matches)
pos.matches1 <- !is.na(pos.matches)
neg.matches1 <- !is.na(neg.matches)
View(pos.matches1)
score <- sum(pos.matches1) - sum(neg.matches1)
score

# predicting sentiment
df.lm = df[-c(1)]
#View(df.lm)
model <- lm(Sentiment ~.,data = df.lm)
summary(model)

# predicting billboard position
df.billboard = df[-c(1,4)]
model <- lm(Position ~.,data = df.billboard)
summary(model)

# predicting if it is a hit or not
df.logistic = df[-c(1,5)]
df.log = glm(In.Hot.100.Billboard~., data=df.logistic,
family=binomial)
summary(df.log)
coef(df.log)
summary(df.log)$coef

# predicting position on album

model <- lm(Position.on.Album ~.,data = df.lm)
summary(model)
```

```
# predicting song length

model <- lm(Song.length.in.seconds ~.,data = df.lm)
summary(model)

# predicting word count

model <- lm(Word.Count ~.,data = df.lm)
summary(model)

# predicting BPM
model <- lm(BPM.value ~.,data = df.lm)
summary(model)

# categorization
df.cat=df[-c(1,2,3,5,9,11)]
row<-nrow(df.cat)
set.seed(12345)
trainindex <- sample(row, 0.6*row, replace=FALSE)
training <- df.cat[trainindex,]
show(training)
validation <- df.cat[-trainindex,]
mylogit<-glm(In.Hot.100.Billboard ~ .,data=training,
family=binomial)
summary(mylogit)
coef(mylogit)
exp(coef(mylogit))
step(mylogit)
valC<-predict(mylogit,validation,type="response")
valC[1:5]
```

```
matrixC =
confusion.matrix(validation$In.Hot.100.Billboard,valC,threshold=
0.5)
matrixC

##### decision trees #####
df.tree=df[-c(1,5,9,10)]
#View(df.tree)
set.seed(12345)
tree.df=tree(Category~.,df.tree)
par(bg = "Grey")

plot(tree.df)
text(tree.df,pretty=4)
summary(tree.df)

#cv.df=cv.tree(tree.df)
#plot(cv.df$size,cv.df$dev,type='b')
prune.df=prune.tree(tree.df,best=5)
plot(prune.df)
text(prune.df,pretty=5)
```