All Pop Music Sounds the Same These Days: A Statistical Investigation Into Chord Progressions In Popular Music

Kelsey Evans

5/7/21

Contents

Supplementary info:

- hook data cleaning.R is the web scrape for the chordData csv
- WEBPAGES is a folder of .html files used for this webscrape
- ALL DATA CLEANING HERE.R is all the data cleaning to create total.csv
- hook.csv is the chord progression data
- billboardHot100_1999-2019.csv is the first set of Billboard data
- Hot Stuff.csv is the second set of Billboard data
- artistDf.csv is the artist democraphic data
- total.csv is a merge of all of the above .csv files
- Final project.Rmd contains all code
- Final project.pdf does not contain code

Introduction

I used to play classical piano, attend competitions, and practice daily after school. These days, I only play pop music for my friends. I pull up the chords, get the rhythm down, and sing over top of the simple structure I've created. I found that many songs use the same chords, so after learning one set, I'm able to play five or ten different songs. For instance, "I'm Yours" by Jason Mraz, "Umbrella" by Rihanna, and "Hey Soul Sister" by Train can all be played using the chords of C, G, A minor, and F.

For those unfamiliar with music, "chords" are the bones or structure behind a song, which the melody is added on top of. Chords can be recorded in two different ways - using letters or using numbers. The above example using letters means that the song has four chords, made up of three notes each (a chord cannot be "consonant"/"sound good" with more than three distinct notes, similar to how more than three points do not always create a plane. Not all songs follow this convention, because sometimes dissonance can add to a song musically, but most of but most of the time it holds).

The more common way to record chords uses numbers instead. The chords C, G, Am, and F become I, V, vi, IV. This is to take into account "transposition." In other words, if one played the song higher or lower, the sound, or letter notes, would be different, but the underlying structure would be the same. I could sing "I'm Yours" in the key of C, using C, G, Am, and F, or in the key of, say, G, using G, D, Em, and C. Both of these sound the same, so they are both notated I, V, vi, IV.

A final note is that the lowercase refers to a minor, or "sad sounding" chord, while the uppercase refers to a major, or more "happy sounding" chord. There are other types of chords other than repeats of four numbers, but it isn't necessary to understand the exact meaning of each one, just that each set of numbers refers to the different ways a song can be structured.

I chose to investigate how these different chord structures might relate to a song's popularity. For instance, the chord I listed above, the I, V, vi, IV, is colloquially known to be the most catchy progression in all of pop music. Is this true? Do songs using this progression do better on the music charts? Do they stay there longer? What about when you take into account the demographics of the artist, the year the song was released, or the listed genre of the song? All of these are exploratory questions that I wondered before digging into the data.

The Data

I used four different sources for data, which I will cite at the end of the document. The first is a data set that I created using quite a bit of html web scraping, from a music theory website called "hooktheory." This website contains crowd-sourced chord progression data on about 2,000 songs. Here is a sample of that data -

Chord data:

	Title	Artist	ChordProg
491	Live While We're Young	One Direction	vi.V.IV.V
368	Sweet Donuts	Perfume	I.V.vi.IV
1484	Somebody To Love	Queen	IV.I6.V
1608	Read My Mind	The Killers	IV.V.V6-vi.vi

The next two sets of data are Billboard 100 data sets. The first is from 1999- 2020, but includes information on the genre of the songs. The second is from 1958-present, but does not include genre labels. Again, a sample from each data set -

1999-2019 data:

	Artists	Name	Weekly.rank	Peak.position	Weeks.on.chart	Week	Genre
5152	Jake Owen	I Was Jack	75	75	5	2018-07-07	Country
79819	P!nk	Family Portrait	80	20	18	2003-03-11	R&;B,Pop
85676	Fabolous	Young'n	81	81	3	2001-12-11	Rap
7798	Zayn	Dusk Till Dawn	85	44	14	2017-12-30	Pop

1958-2020 data:

	WeekID	Week.Position	Performer	Previous.Week.Position	Peak.Position	Weeks.on.Chart
308175	12/26/1981	5	Rod Stewart	5	5	11
54621	8/31/2013	35	Sage The Gemini Featuring IamSu!	32	32	4
36044	5/12/2001	84	Tim Rushlow	67	59	9
277801	2/20/1982	15	Sheena Easton	17	15	13

The final data set is some collected information about artist demographics. That set looks like this -

Artist data:

	Artist	Followers	YearFirstAlbum	Gender	Group.Solo
67	Bing Crosby	250331	1932	M	Solo
57	Bobby Helms	10470	1957	M	Solo
934	Jon B	590156	1995	M	Solo
559	Celine Dion	2903347	1987	F	Solo

There were a lot of problems when it came to merging these data sets. Many songs were covered by other artists, so didn't intially make it into the merge. Out of the 2000 songs from the chord data, I had to fix about 200 of them by hand. I did a left merge between the chord data and the billboard data. To do this, I had to hand-code a few things: If a song was in the chord data but not the billboard charts, I marked it as having been on the charts for zero weeks, and I marked its peak and weekly positions on the chart as 101 for a placeholder (the lowest place on the chart is 100). This way, I didn't have to lose any of the chord data. I also did a left merge between the billboard data and the artist demographic data/genre data, which didn't exist for all of the 1958- present billboard songs. This left the merged data with NAs in spots, but keeping as much data as I could. The merged data set has the following categories:

```
##  [1] "Artist"                "Title"
##  [3] "ChordProg"             "WeekID"
##  [5] "Week.Position"         "Instance"
##  [7] "Previous.Week.Position" "Peak.Position"
##  [9] "Weeks.on.Chart"        "Genre"
## [11] "Followers"             "NumAlbums"
## [13] "Gender"                "Group.Solo"
```

These are as follows:

- "Artist" is the singer or band
- "Title" is the name of the song
- "ChordProg" is the progression from the chord data
- "WeekID" is the week that the song was on the chart (songs will appear in the data set more than once if they were on the chart for multiple weeks)
- "Week.Position" is the position on the chart that week

- "Instance" is the number of distinct times the song has been on the charts
- "Previous.Week.Position" is where the song was on the charts the previous week
- "Peak.Position" is the highest the song has ever been on the charts
- "Weeks.on.Chart" is how many weeks, at the point of the WeekID, a song has been on the charts
- "Genre" is a list of genres the song could fall under
- "Followers" is the number of Spotify followers the artist has
- "NumAlbums" is the number of albums the artist has
- "Gender" is the gender of the artist
- "Group.Solo" is whether the artist is a solo artist or a group/band

Visualizations/Transformations

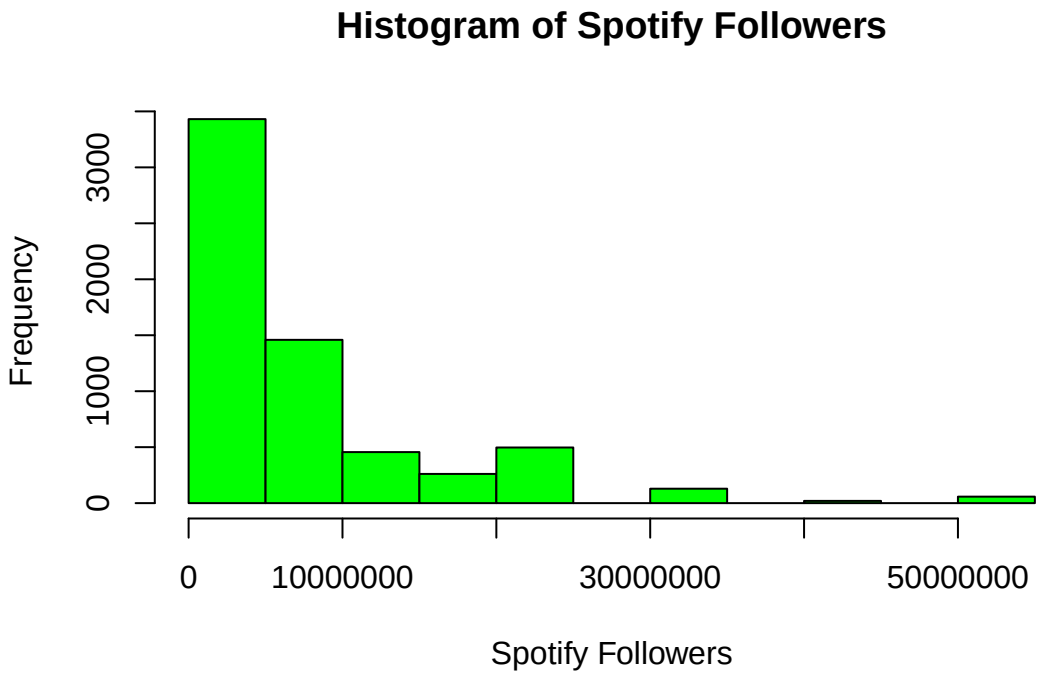
I will lay out what some of the variables look like. There are 21 types of progressions, with the combinations of I, V, vi, and VI making up most of them, albeit in different orders:

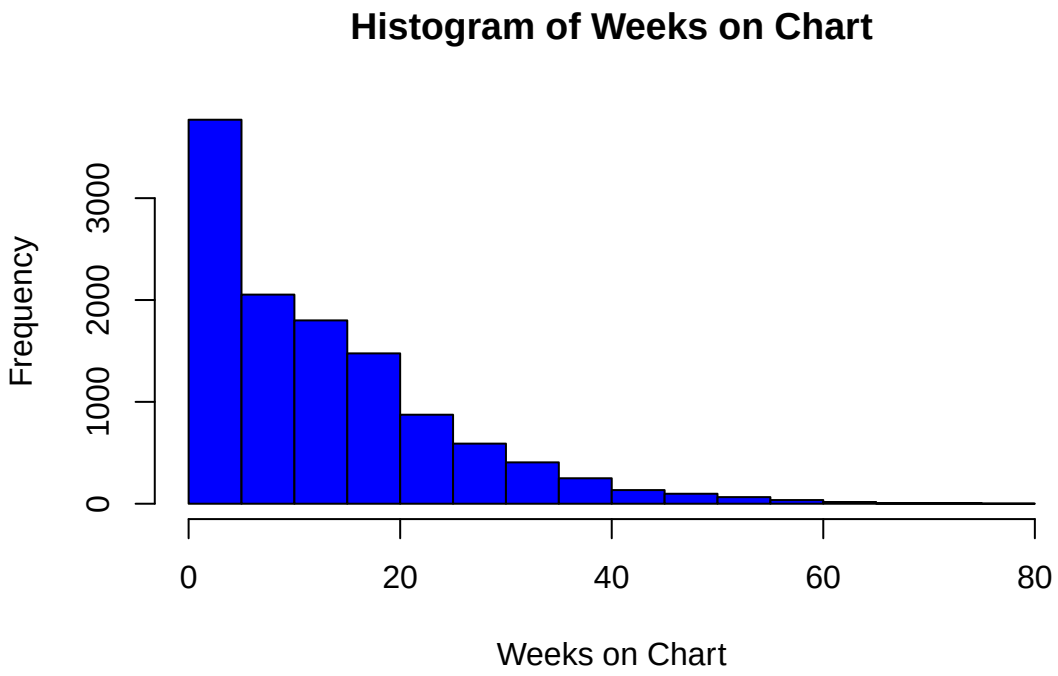
Var1	Freq
I.bVI.V	119
I.bVII.IV.I	298
I.ii7.I6.IV	85
I.iii64.vi.IV	214
I.IV.vi.V	875
I.V-vi.vi	611
I.V.IV.bVII.I	474
I.V.IV.V	521
I.V.vi.iii.IV	108
I.V.vi.IV	3239
I.V6.vi.V	1038
I.V7-IV.IV	233
I.vi.IV.V	849
IV.I6.ii	286
IV.I6.V	200
IV.ii.I64.V.I	635
IV.iv.I	106
IV.V.V6-vi.vi	78
V.vii-vi.vi	192
vi.V.IV.V	1286
vi.vi42.IV	133

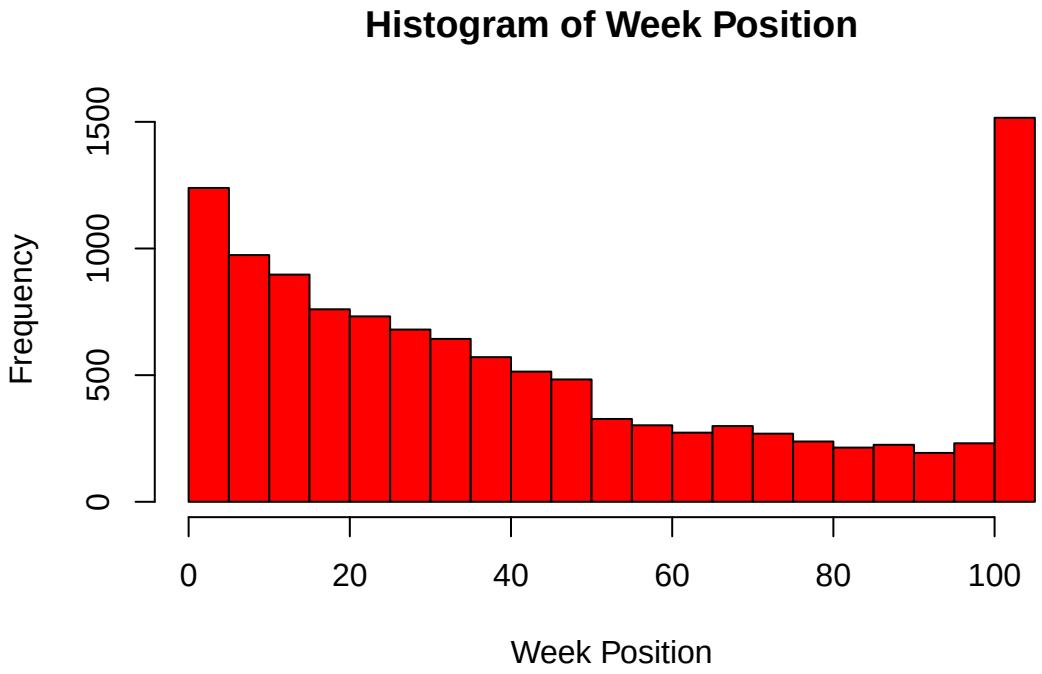
The gender of the artists skews slightly towards more male artists:

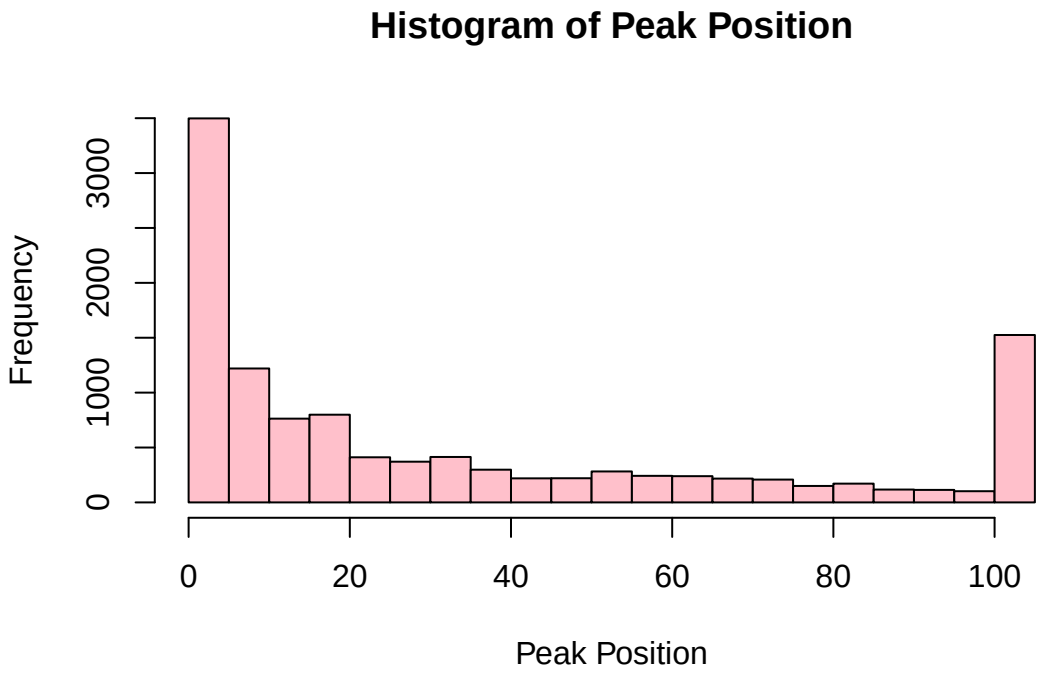
Var1	Freq
	423
F	2532
M	3356

All of the numeric variables skew extremely right:

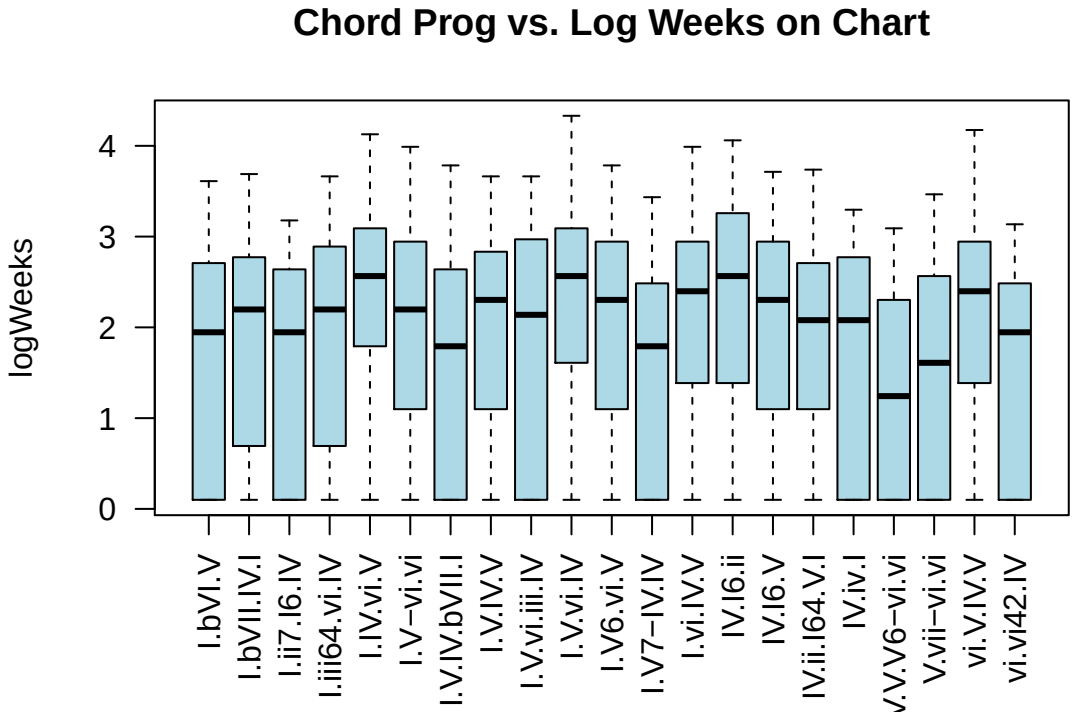


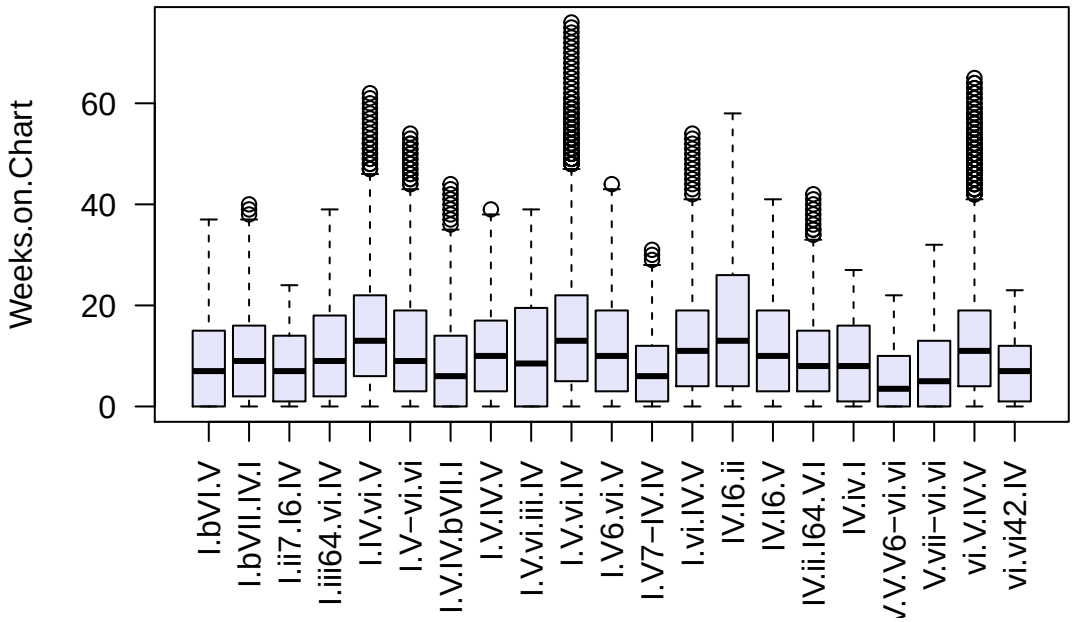






Because of this, I chose to also create a logged version of each numeric variable. This helps not only the right skew situation, but also puts the numeric variables on a comparable scale for analysis. (I changed -Inf to zero for zero weeks on the chart, and 0 to 0.1 for one week on the chart, to create a difference between 0 and 1 week). and Here is an example of some boxplots looking at weeks on chart grouped by the chord progression, in the logged and not-logged versions:



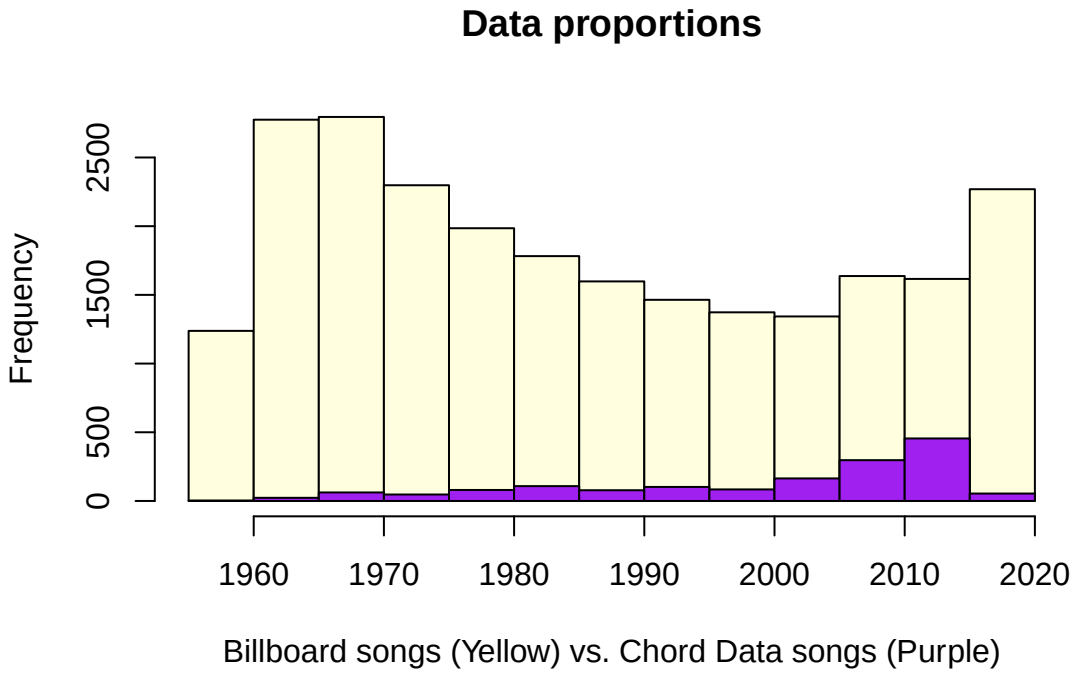


Here is a correlation matrix of the logged numeric variables:

	logWeeks	logWeekPosition	logPeakPosition	logFollowers
logWeeks	1.0000000	-0.2921688	-0.6249975	0.0539344
logWeekPosition	-0.2921688	1.0000000	0.6940584	-0.1640038
logPeakPosition	-0.6249975	0.6940584	1.0000000	-0.2450761
logFollowers	0.0539344	-0.1640038	-0.2450761	1.0000000

As expected, "Weeks on chart" is negatively correlated with position on the chart and positively correlated with "Followers", which is also negatively correlated with the position variables.

The final thing to note about the data is that there is a slight concern that the data for which there is chord information is not distributed the same as the billboard data. Because the data is crowdsourced on the internet, it skews towards a younger audience, specifically, from a glance, probably millennial or older Gen-Z individuals (the chord data peaks in the early 2010's).



The shape of the billboard data (yellow) with fewer individual songs around the 90s/00s/10s could mean that songs are staying on the charts for longer during that period of time. (Ordinarily, the "unit" in the billboard data is a song*year, but in this histogram I am looking only at the data distribution of each unique song, not each unique observation.)

Results

Genres

To begin constructing some models, I first wanted to make sure I had as much predicting power as possible. One thing I did was split up the "Genre" column, which was a separate comma-separated vector for each row, into a matrix of TRUE/FALSE for each genre. I then restricted to only genres which were tagged on more than ten songs. This way, the model isn't overfit because it takes in some genres that were only tagged on a song or two. After working on genres, the columns included in the data are as follows:

```
##  [1] "Artist"                "Title"
##  [3] "ChordProg"             "WeekID"
##  [5] "Week.Position"         "Instance"
##  [7] "Previous.Week.Position" "Peak.Position"
##  [9] "Weeks.on.Chart"        "Genre"
## [11] "Followers"             "NumAlbums"
## [13] "Gender"                "Group.Solo"
## [15] "logWeeks"              "logWeekPosition"
## [17] "logPeakPosition"       "logFollowers"
## [19] "Pop-Punk"              "Soul Pop"
## [21] "Indie Pop"             "Synth-Pop"
```

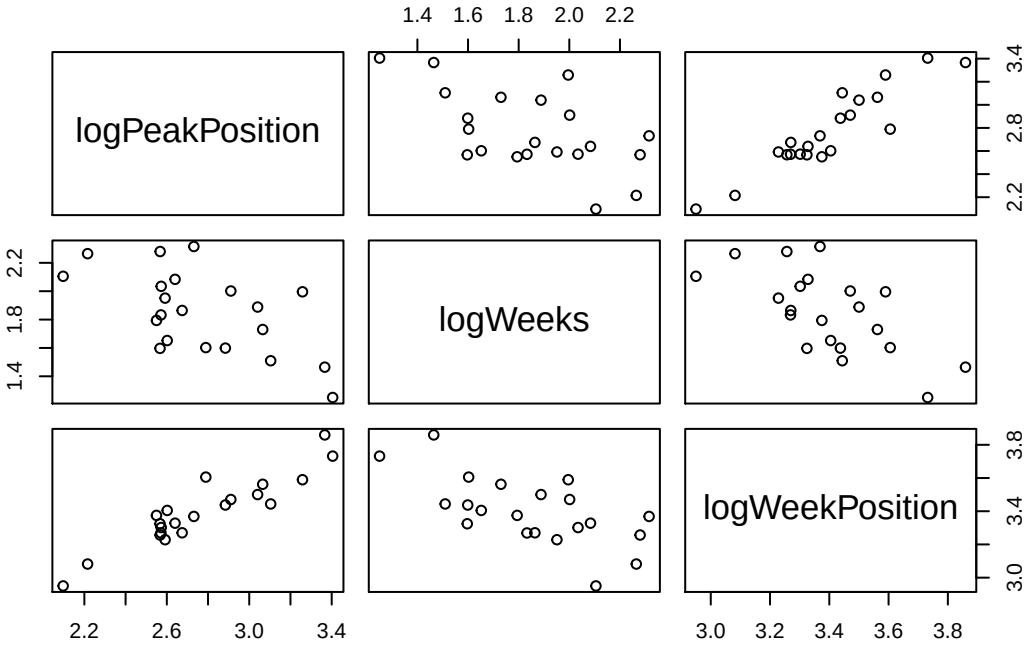
```
## [23] "Alternative"            "Indie Rock"
## [25] "Memes"                  "Piano"
## [27] "Soul"                   "Soundtrack"
## [29] "Dance-Pop"              "Electronic"
## [31] "Electro-Pop"            "UK"
## [33] "Country"                "Ballad"
## [35] "Adult Contemporary"     "R&B"
## [37] "Rap"                    "Alternative Rock"
## [39] "Adult Alternative"      "Pop-Rock"
## [41] "Rock"                   "Pop"
```

Cluster Analysis

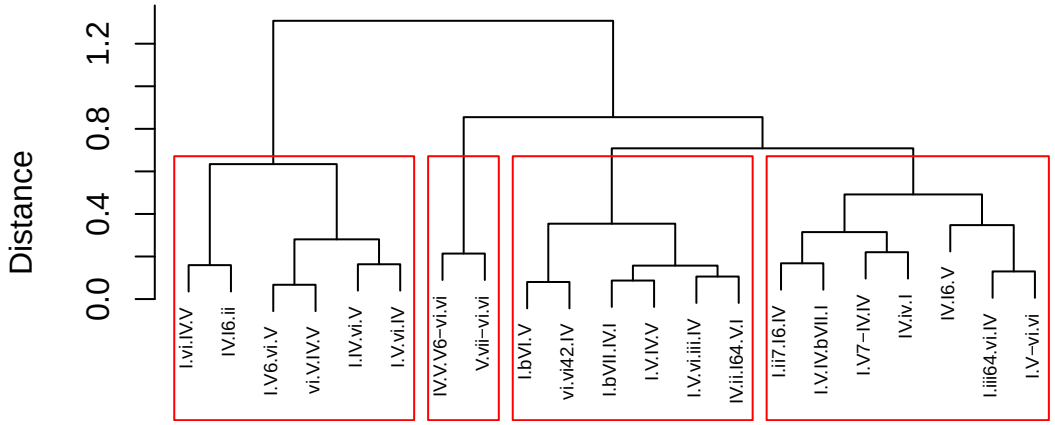
I was curious about clustering the data both on a chord level and a song level. In terms of on the chord level, I wanted to see, for instance, if songs with the same chords, but simply in another order, seemed to be grouped together. In terms of on the song level, I wanted to see if the number of groupings seemed to line up in any way with the number of chord progressions. I tried many different methods of clustering. I chose to reproduce as an example here the method using maximum difference and complete clustering, but the different methods all give fairly similar answers. The numeric variables I used were the logs of Week Position, Peak Position, and Weeks on chart.

First, I clustered the data on a chord level. The pairs plot does not show obvious clusters. The dendrogram also does not show extremely obvious clusters, but when arbitrarily picking, for instance, four clusters, chords that are the same but in another order do seem to be grouped together, which is very interesting indeed! This is true for various methods of clustering that I tried. In the left cluster, we can see four different permutations of the I.V.vi.IV progression. These show up in the DA plot as well.

This isn't necessarily extremely useful for other analyses, but it is very neat to see that in these visualizations, the iterations of the chord progression mentioned in the beginning seem to stick together.

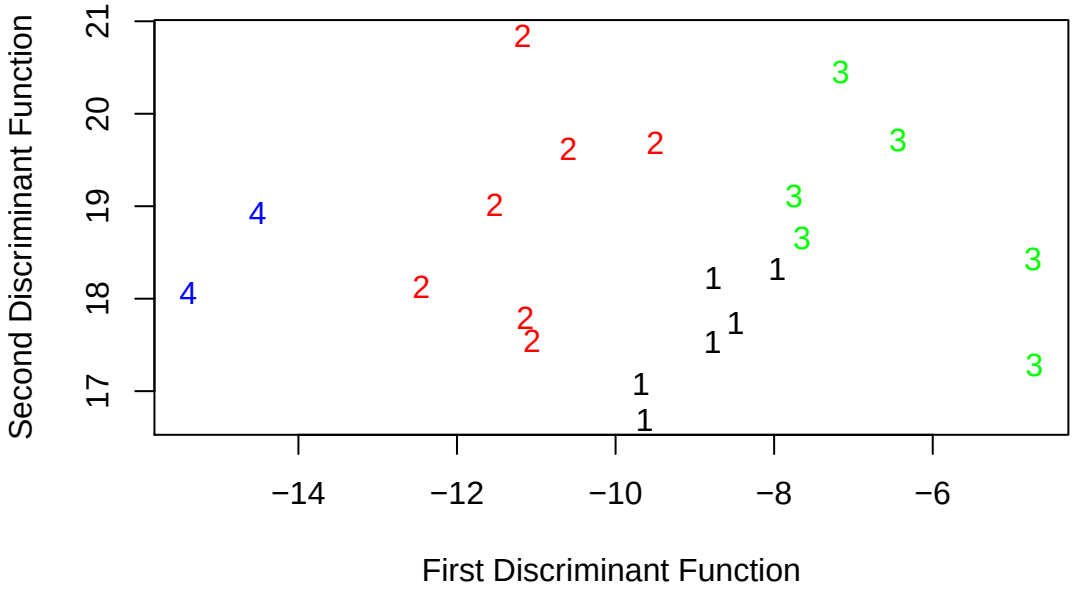


Clustering of ChordProgs, Maximum & Complete, 4 clusters

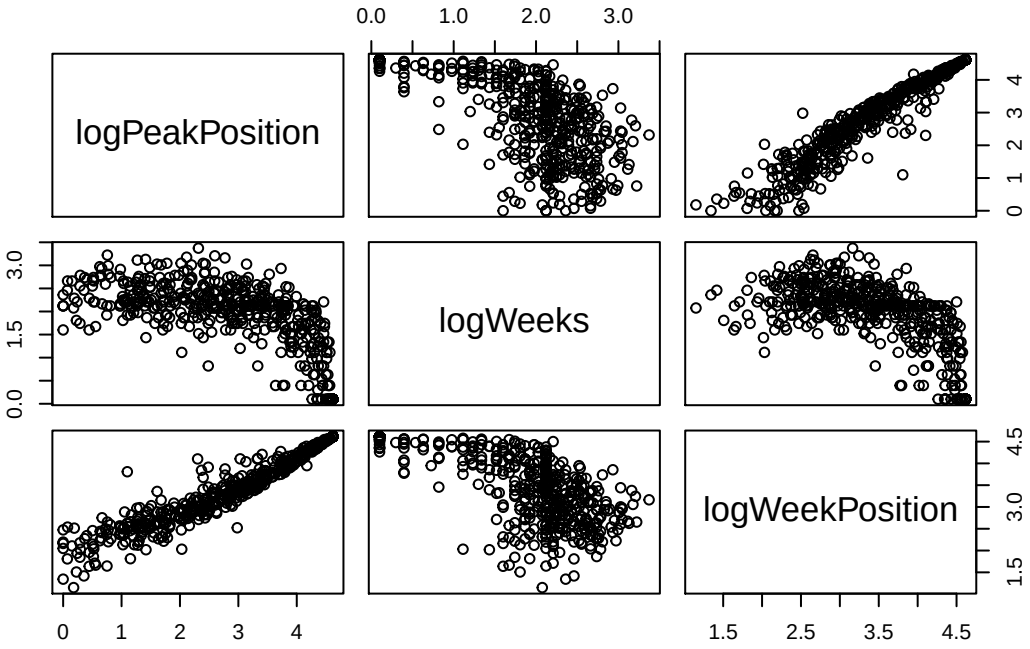


Clusters
hclust (*, "complete")

4 Cluster Solution in DA Space



Next, I clustered the data on the song level.



Here, there also do not seem to be obvious clusters at all - rather, it just looks like a fairly smooth correlation graph. It is possible that the songs cluster by chord and that cannot be seen with this visualization. In the linear model section, we will be able to see whether or not the songs using various chords are statistically significantly different from one another.

Linear Model

I created a linear model using the following variables:

The reason that the genres are each their own column while the chord progression is a single factor is because there can only be one chord progression per song, but there can be up to 15 genres, based on the Billboard tagging system.

Here are the results of the model predicting Weeks on Chart using all the other variables. The R^2 is 54%. I am able to achieve a much higher R^2 using all the genres as predictors, but it is artificially high because some genres are only attached to a song or two. In this model as well as rearranging the model to a contrast sum, I was pleased to see that songs using several of the chords that fall into the I.V.vi.IV category (in that order and others) perform statistically significantly better than the average song. The confidence interval for the coefficient on I.V.vi.IV is (0.099, 0.289). This is of course the logged version, so the actual number of weeks on the chart would be between one and two weeks more than the average song.

Variables	Estimate	Std_Error	t_value	probability	significant
(Intercept)	2.768	0.240	11.556	0.000	***
ChordProgI.bVII.IV.I	0.083	0.112	0.742	0.458	
ChordProgI.ii7.I6.IV	0.123	0.137	0.896	0.370	
ChordProgI.iii64.vi.IV	0.297	0.111	2.687	0.007	**
ChordProgI.IV.vi.V	0.295	0.099	2.984	0.003	**
ChordProgI.V-vi.vi	0.066	0.103	0.635	0.525	
ChordProgI.V.IV.bVII.I	0.010	0.113	0.087	0.931	
ChordProgI.V.IV.V	0.114	0.104	1.091	0.275	
ChordProgI.V.vi.iii.IV	0.018	0.126	0.147	0.883	
ChordProgI.V.vi.IV	0.194	0.095	2.044	0.041	*
ChordProgI.V6.vi.V	0.001	0.099	0.005	0.996	
ChordProgI.V7-IV.IV	0.070	0.126	0.558	0.577	
ChordProgI.vi.IV.V	-0.192	0.099	-1.946	0.052	
ChordProgIV.I6.ii	0.074	0.110	0.677	0.498	
ChordProgIV.I6.V	0.288	0.108	2.681	0.007	**
ChordProgIV.ii.I64.V.I	-0.069	0.103	-0.673	0.501	
ChordProgIV.iv.I	0.453	0.129	3.516	0.000	***
ChordProgIV.V.V6-vi.vi	0.379	0.182	2.084	0.037	*
ChordProgV.vii-vi.vi	0.060	0.137	0.439	0.661	
ChordProgvi.V.IV.V	0.138	0.098	1.404	0.160	
ChordProgvi.vi42.IV	-0.160	0.132	-1.209	0.227	
WeekID	0.000	0.000	6.399	0.000	***
Instance	0.274	0.015	18.507	0.000	***
NumAlbums	0.004	0.001	3.934	0.000	***
GenderF	-0.200	0.048	-4.145	0.000	***
GenderM	-0.026	0.042	-0.630	0.529	
Group.SoloGroup	0.399	0.188	2.122	0.034	*
Group.SoloSolo	0.229	0.190	1.203	0.229	
logWeekPosition	0.295	0.010	28.654	0.000	***
logPeakPosition	-0.607	0.008	-73.345	0.000	***
logFollowers	-0.073	0.007	-9.986	0.000	***
‘Pop-Punk‘TRUE	-0.203	0.058	-3.474	0.001	***
‘Soul Pop‘TRUE	-0.361	0.072	-5.024	0.000	***
‘Indie Pop‘TRUE	-0.023	0.074	-0.315	0.753	
‘Synth-Pop‘TRUE	0.038	0.065	0.587	0.557	
AlternativeTRUE	-0.157	0.074	-2.120	0.034	*
‘Indie Rock‘TRUE	0.408	0.091	4.476	0.000	***
MemesTRUE	0.011	0.045	0.251	0.802	
PianoTRUE	0.143	0.051	2.832	0.005	**
SoulTRUE	0.153	0.052	2.916	0.004	**
SoundtrackTRUE	0.047	0.043	1.095	0.273	
‘Dance-Pop‘TRUE	-0.111	0.041	-2.692	0.007	**
ElectronicTRUE	-0.135	0.053	-2.562	0.010	*
‘Electro-Pop‘TRUE	-0.176	0.052	-3.396	0.001	***
UKTRUE	-0.075	0.045	-1.663	0.096	
CountryTRUE	0.469	0.033	14.262	0.000	***
BalladTRUE	-0.324	0.042	-7.732	0.000	***
‘Adult Contemporary‘TRUE	-0.007	0.044	-0.150	0.881	
‘R&B‘TRUE	0.021	0.038	0.555	0.579	
RapTRUE	-0.148	0.036	-4.091	0.000	***
‘Alternative Rock‘TRUE	0.021	0.047	0.453	0.651	
‘Adult Alternative‘TRUE	0.040	0.043	0.937	0.349	
‘Pop-Rock‘TRUE	0.060	0.029	2.090	0.037	*
RockTRUE	0.220	0.027	8.221	0.000	***
PopTRUE	0.249	0.027	9.120	0.000	***

R^2:

```
## [1] 0.5451694
```

In a second model, I did the same thing, but with Peak Position as the response variable instead:

Variables	Estimate	Std_Error	t_value	probability	significant
(Intercept)	2.625	0.272	9.653	0.000	***
ChordProgI.bVII.IV.I	0.091	0.127	0.718	0.473	
ChordProgI.ii7.I6.IV	0.386	0.155	2.480	0.013	*
ChordProgI.iii64.vi.IV	0.494	0.125	3.946	0.000	***
ChordProgI.IV.vi.V	0.511	0.112	4.563	0.000	***
ChordProgI.V-vi.vi	0.211	0.117	1.800	0.072	.
ChordProgI.V.IV.bVII.I	0.027	0.128	0.207	0.836	
ChordProgI.V.IV.V	0.359	0.118	3.038	0.002	**
ChordProgI.V.vi.iii.IV	-0.062	0.142	-0.434	0.664	
ChordProgI.V.vi.IV	0.350	0.107	3.255	0.001	**
ChordProgI.V6.vi.V	0.056	0.112	0.502	0.616	
ChordProgI.V7-IV.IV	0.331	0.142	2.322	0.020	*
ChordProgI.vi.IV.V	-0.131	0.112	-1.175	0.240	
ChordProgIV.I6.ii	-0.028	0.124	-0.228	0.820	
ChordProgIV.I6.V	0.540	0.122	4.441	0.000	***
ChordProgIV.ii.I64.V.I	0.053	0.117	0.453	0.650	
ChordProgIV.iv.I	0.705	0.146	4.839	0.000	***
ChordProgIV.V.V6-vi.vi	0.714	0.205	3.473	0.001	***
ChordProgV.vii-vi.vi	0.368	0.154	2.383	0.017	*
ChordProgvi.V.IV.V	0.290	0.111	2.612	0.009	**
ChordProgvi.vi42.IV	0.082	0.150	0.550	0.582	
WeekID	0.000	0.000	5.220	0.000	***
Instance	0.278	0.017	16.535	0.000	***
NumAlbums	0.007	0.001	5.910	0.000	***
GenderF	-0.104	0.055	-1.910	0.056	.
GenderM	0.090	0.047	1.895	0.058	.
Group.SoloGroup	0.414	0.213	1.947	0.052	.
Group.SoloSolo	0.070	0.215	0.326	0.745	
logWeeks	-0.777	0.011	-73.345	0.000	***
logWeekPosition	0.636	0.009	68.328	0.000	***
logFollowers	-0.116	0.008	-14.062	0.000	***
‘Pop-Punk‘TRUE	-0.301	0.066	-4.557	0.000	***
‘Soul Pop‘TRUE	-0.468	0.081	-5.759	0.000	***
‘Indie Pop‘TRUE	-0.106	0.083	-1.274	0.203	
‘Synth-Pop‘TRUE	0.222	0.073	3.037	0.002	**
AlternativeTRUE	-0.140	0.084	-1.673	0.094	.
‘Indie Rock‘TRUE	0.696	0.103	6.763	0.000	***
MemesTRUE	-0.066	0.051	-1.308	0.191	
PianoTRUE	0.199	0.057	3.492	0.000	***
SoulTRUE	0.135	0.059	2.283	0.022	*
SoundtrackTRUE	0.034	0.048	0.704	0.482	
‘Dance-Pop‘TRUE	-0.112	0.047	-2.395	0.017	*
ElectronicTRUE	-0.073	0.060	-1.224	0.221	
‘Electro-Pop‘TRUE	-0.364	0.059	-6.200	0.000	***
UKTRUE	-0.072	0.051	-1.408	0.159	
CountryTRUE	0.552	0.037	14.878	0.000	***
BalladTRUE	-0.402	0.047	-8.491	0.000	***
‘Adult Contemporary‘TRUE	-0.147	0.049	-2.980	0.003	**
‘R&B‘TRUE	0.112	0.042	2.632	0.009	**
RapTRUE	-0.246	0.041	-6.027	0.000	***
‘Alternative Rock‘TRUE	-0.025	0.053	-0.464	0.642	
‘Adult Alternative‘TRUE	-0.029	0.048	-0.601	0.548	
‘Pop-Rock‘TRUE	0.082	0.032	2.527	0.012	*
RockTRUE	0.267	0.030	8.833	0.000	***
PopTRUE	0.214	0.031	6.904	0.000	***

R^2:

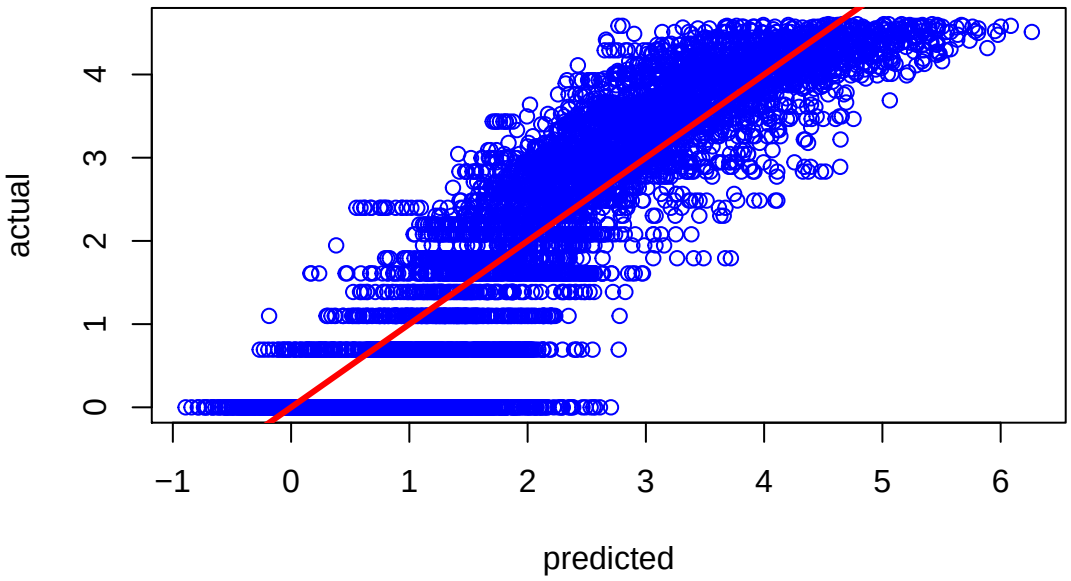
```
## [1] 0.7576472
```

Adjusted R^2:

```
## [1] 0.755474
```

Using this response instead, a higher R^2 of 76% is achieved. Interestingly, here it doesn't seem like the chords that I have suggested are uniformly doing better (in this case would mean negative coefficients). I wonder if all the songs tagged as being in "101st" place on the chart could be affecting this in any way.

Here is a chart of the predicted vs. actual values using this model:



Positioning of Non-Billboard Songs

I wondered about was, regarding the position on the chart, whether the maximum position matters. I had tagged all NON-billboard songs as having a position of 101 on the chart, but this would mathematically imply something like, that they "almost" made it on the chart, which is almost certainly not true for all of the songs. I decided to re-code them as 200, to see what this would do to the model and correlations. I re-code to 200, then took the log again. Here are the correlations:

```
##  [1] "Artist"                "Title"
##  [3] "ChordProg"             "WeekID"
##  [5] "Week.Position"         "Instance"
##  [7] "Previous.Week.Position" "Peak.Position"
##  [9] "Weeks.on.Chart"        "Genre"
## [11] "Followers"             "NumAlbums"
## [13] "Gender"                "Group.Solo"
## [15] "logWeeks"              "logWeekPosition"
## [17] "logPeakPosition"       "logFollowers"
## [19] "Pop-Punk"              "Soul Pop"
## [21] "Indie Pop"             "Synth-Pop"
## [23] "Alternative"           "Indie Rock"
## [25] "Memes"                 "Piano"
## [27] "Soul"                  "Soundtrack"
## [29] "Dance-Pop"             "Electronic"
## [31] "Electro-Pop"           "UK"
## [33] "Country"               "Ballad"
```

```
## [35] "Adult Contemporary"     "R&B"
## [37] "Rap"                    "Alternative Rock"
## [39] "Adult Alternative"      "Pop-Rock"
## [41] "Rock"                   "Pop"
## [43] "Peak.Position2"         "Week.Position2"
## [45] "logWeekPosition2"       "logPeakPosition2"
```

	logWeeks	logWeekPosition2	logPeakPosition2	logFollowers
logWeeks	1.0000000	-0.3301873	-0.6443814	0.0539344
logWeekPosition2	-0.3301873	1.0000000	0.7120253	-0.1628723
logPeakPosition2	-0.6443814	0.7120253	1.0000000	-0.2413795
logFollowers	0.0539344	-0.1628723	-0.2413795	1.0000000

	logWeeks	logWeekPosition	logPeakPosition	logFollowers
logWeeks	1.0000000	-0.2921688	-0.6249975	0.0539344
logWeekPosition	-0.2921688	1.0000000	0.6940584	-0.1640038
logPeakPosition	-0.6249975	0.6940584	1.0000000	-0.2450761
logFollowers	0.0539344	-0.1640038	-0.2450761	1.0000000

The correlations remain pretty much exactly the same. I ran the model predicting peak position again, and the model also changes essentially not at all. The R^2 remains the same:

```
## [1] 0.7576472
```

One more model I wanted to try was actually without the chord data - to see if it indeed added any predicting power to the model. I used the peak position variable with the original tagging as 101 for non-Billboard songs:

R^2:

```
## [1] 0.7431645
```

Adjusted R^2:

```
## [1] 0.7417192
```

The predicting power only decreased from 76 to 74 percent, so the chord progressions are not doing the heavy lifting, but they are contributing. The adjusted R^2 is also smaller, which is good because it means adding the chords is not overfitting the model.

Bootstrap on Gender?

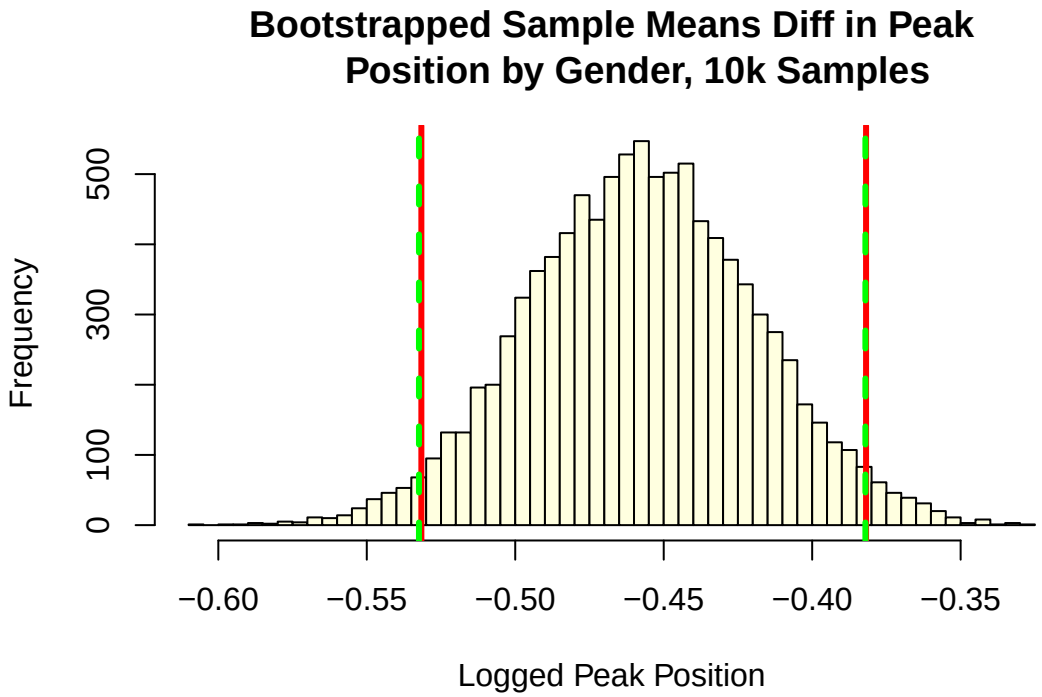
The final thing that piqued my interest was the gender variable. In the first model, where weeks on chart is the response variable, it seems that both male and female have negative coefficients. This, I would imagine, means that songs that were tagged with an artist gender at all (and not all of them are), do worse than songs that are not. I'm not quite sure of an easy way to see in what other ways these songs differ, but it is interesting. In the second model, where the response variable is the peak position, it seems that the female factor does better (negative coefficient). Neither are statistically significant at the 0.05 level, but almost, so I decided to do a bootstrap test of just those songs tagged with a gender, to see if I could glean anything about the difference.

This is a simple t-test for the difference in means for the logged peak chart position based on gender. The confidence interval does not contain zero:

```
## [1] "F" "M"
```

```
## [1] -0.5324271 -0.3820717
## attr(,"conf.level")
## [1] 0.95
```

Similarly, running a bootstrap test on the same data:



This gives almost the exact same confidence interval, and again does not contain zero. This would imply that songs by women actually make it higher on the charts. (negative difference), contrary to what I would think based on the regression output. This could mean that songs by women are actually making it higher on the charts, but it could also just be that the songs that are tagged with gender in the first place are biased in some way that is difficult to see. When doing a t-test for the number of weeks on the chart, we actually seem to get the same result - that songs by women are staying on the charts longer.

```
## [1] "F" "M"
```

```
## [1] 0.09720624 0.19967011
## attr(,"conf.level")
## [1] 0.95
```

Taken at face value, this would mean that songs by women do better, but once again I am more suspicious of the gender data, because Billboard only provides this data for some of the artists.

Conclusion

This was a very interesting and entertaining project. I really enjoyed getting to learn more about music data as I constructed these models. Overall, it seems that the chord data adds at least a little bit of predicting power to the data, which is exciting, and that the chord progression I mentioned, the I.V.vi.IV, does better than other chords by some metrics, such as the number of weeks on the chart. Chords of that form, but a different order, also seem to cluster together using various methods of cluster analysis. I would love to run all of this again with chord progressions and artist demographics available for all the Billboard songs, because I think that would give a much clearer picture. Overall, I was able to get about 55% predicting power for a model predicting weeks on chart, and 76% predicting power for a model predicting peak position.

Data Credit

- One set of Billboard data and the artist data was compiled by Daniel DeFoe and Charlie Liu on Kaggle: https://www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019
- The other set of Billboard data was compiled by Sean Miller on Data World: https://data.world/kcmillersean/billboard-hot-100-1958-2017
- The hooktheory data set was crowd-sourced on the internet. The website was put up by Dave Carlton, who gives permission for others to use it, and compiled into a dataset by me.