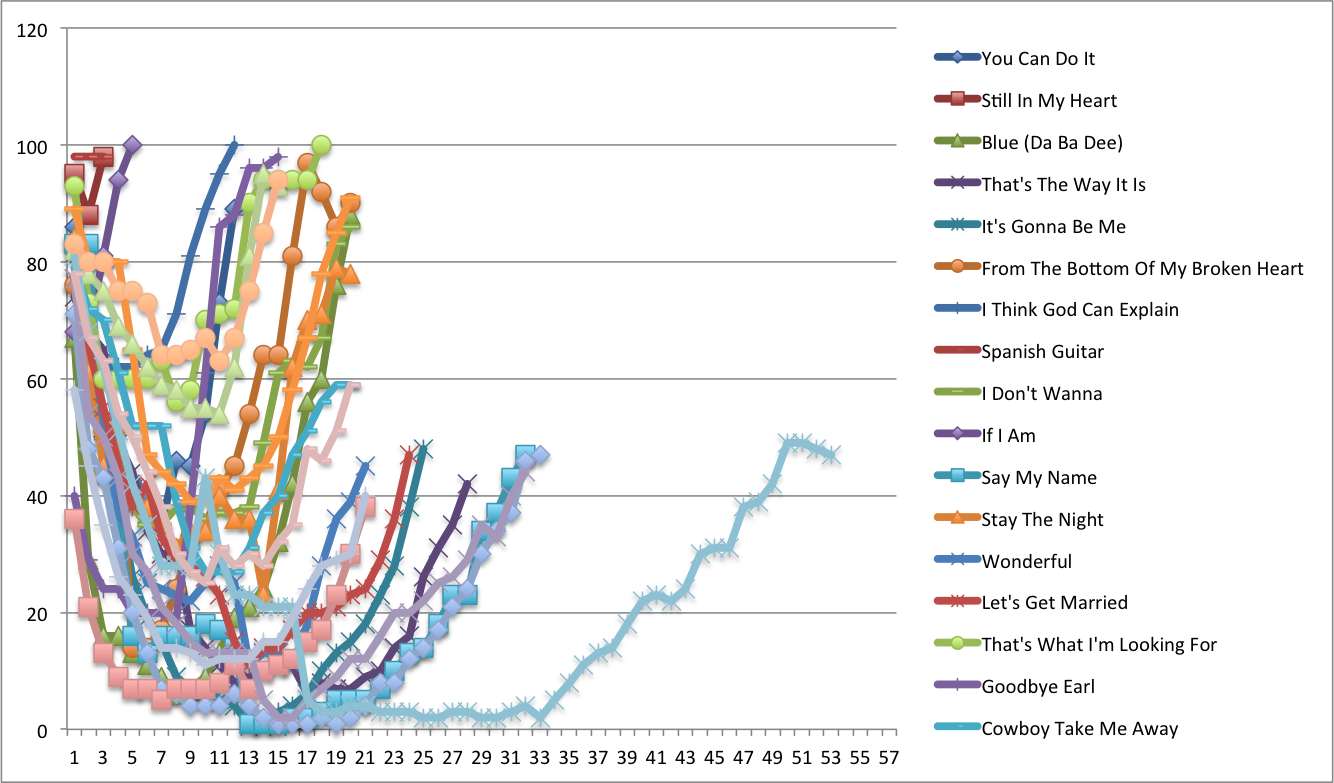
* The first step in analyzing the Billboard data was importing the raw CSV file into a Pandas dataframe so that it could be more easily manipulated. Once imported, I saw that the column names weren’t very pretty, so I renamed all of them by creating two new arrays filled with column names, one for the descriptive columns (Artist, Track Name, Track length, etc) and another array full of the numbered week columns which I created using a list comprehension as there were over 70 columns to be named.

Once the columns were cleaned up, I moved onto extracting the weekly ranking data to be cleaned as this would be at the heart of the analysis. Using the .iterrows() function with a for loop, I created a new dictionary using the track names as the keys, and using the weekly rankings as their values. I chose to use a dictionary because I’m simply more comfortable with them compared to a Pandas dataframe. Once in the dictionary, I was faced with the task of cleaning out all of the null values that were previously necessary to make the dataset compatible with a Pandas Dataframe (standardized row and column widths). I accomplished this with yet another loop, going through each of the dictionary entries using the .iteritems() function to go check each of the values for nulls using *if str(i) != 'nan'* and then appending the values in float format to a new “cleaned” version of the previous dictionary. I needed to use *str(i)* to be able to check all of the values which were previously all floats at the same time for nulls which are displayed as “NaN” and produce an error if used in any computational functions.

Although it was irrelevant to my method of data cleaning, the project requested us to melt the original pandas dataframe to eliminate all 76 week columns and stack them vertically, resulting in a single column containing all week rankings for every track. I did this using the pd.melt() function with ‘Week’ as my variable name, and ‘Ranking’ as my value name.

Now that I had clean data, I wanted to start my analysis. Aside from tracking weekly ranking, I wanted to track how quickly tracks were climbing or falling on the Billboard. I did this by calculating the weekly change in rank by subtracting each weekly ranking from the previous week’s ranking and then appending each of those calculated values to a new array which was then assigned as another value for each song in the dictionary. This allows me to call a single track in the dictionary and return the weekly rankings in one array and the change in weekly ranking directly below it in another array. This keeps everything iterable and easy to access through a simple dict.items() function. At this point I had all of the data I wanted to use, however this is where I ran into problems.

Because of my dictionary format, exporting to CSV format became a major roadblock. I couldn’t figure out how to export the whole dictionary at once with both the weekly rankings and the weekly change in ranking, so when I exported to CSV, I exported the dictionary in halves, one half containing the weekly rankings, and the other containing the weekly change. Once I got to this point I was able to correctly export to CSV, however I ran into another problem once in Tableau. Tableau didn’t understand the structuring of my columns for some reason and refused to see each tracks rankings as one cohesive set of values, meaning I couldn’t display the rankings in a continuous line plot. I was able to achieve this to a certain degree in Excel, however it started misbehaving when I would give it too many rows of data. Below is a sampling of the data visualized in Excel:

The most clearly visible trend is the rankings dropping off around week 20. This leads me to believe the typical lifecycle for modern music is around 5 months.