With this project my aim is to improve my data analysis and data science skills and to create a public repository that I can use to showcase my capabilities.

For this first project I’ve chosen a topic that I’m not familiar with and that I hope it will give me the opportunity to try different data analysis tools that I haven’t been able to try on my date to date job yet. As a little bit of background, I am a manufacturing engineer with extensive experience in the aerospace sector, specifically in manufacturing.

Coming back to the project, I will split this week into different sections.

**DATA COLLECTION**

I have segregated the data collection part into a data\_creation module for tidiness purposes.

I check whether the data has already been collected (which would be stored in parquet file on the current working directory) and if not I collect the data from the following webpage using pandas:

<http://www.umdmusic.com/default.asp?Lang=English&Chart=D&ChDay=1&ChMonth=2&ChYear=1999&ChBand=&ChSong>=

I decided to collect data using parquet files from 1970s to 2020s to be able to see if there is any change in the behaviour of the data across time. I chose to scrape the data 4 times for every month in that year range.

The reason on one hand is to create enough granularity of the data, since the charts are on a weekly basis, I need to collect the data weekly. I also want to create a big enough dataset to be able to apply descriptive analysis and machine learning methods.

**DATA PREPARATION**

My initial objective is to find out the characteristics that differentiate the top songs (the ones that reach the number 1 position in the charts) from the rest of the songs.

I try to answer the following 3 questions:

* How long do n1 songs last in charts vs the rest of the songs?
* How many positions do n1 songs climb vs rest of the songs?

Check that bottom position is before n1 song reaching n1

* How many positions do n1 songs climb from starting position to n1 position?
* What the starting positions of n1 songs are in charts vs the rest of the songs?

To be able to compare the n1 songs population with the rest of the population I differentiate the Artist and Title of each song that has ever reached a number one position, this is, where the column “this week position” has ever been 1.

I work using a summary table to which I can pin the attributes of the different songs. Because what I want to do is analyse the n1 songs vs the rest of the songs, the summary table includes all songs and to this I will be adding columns.

I create the attributes (that don’t exist yet) and are required to answer the questions stated above:

* The maximum date in the charts
* The bottom position of each of the songs in the charts
* The top position of each song in charts

Once the attributes are calculated, I clean the data through the following 2 steps:

1. I select the songs where the OverallPeakPos (which came with the original data) and the calculated peak position match. I have realized that if these 2 attributes don’t match, it is because the peak position of that song took place outside of the arbitrary time range (1970 to 2020) that was selected, and therefore the calculated attributes will not be a true reflection of the song’s real attributes.
2. I select the songs where the Entry in Charts or LastDate in charts reflect that the song’s run in the charts didn´t start before or didn´t potentially finish after the time analysis range.
3. I select the songs where the original length of days in charts is at least equal or bigger than the calculated time in charts. This is because if a song has had multiple re-entries across the years in the charts, the calculated time in charts can through silly numbers, where a song originally entered the charts in 1973 and the maximum date of the last time it entered the charts is 2016.

I realized that there is a mismatch between the Overall Total Weeks (the time that the song has been in charts according to the original data) and the calculated time in charts (by subtracting the last parsing date of that song with the entry date of the song).

For example, Mariah Carey’s song “All I want for Christmas is you” entered the charts in 2018-12-01 and exited the charts in 2019-01-01, spanning a total of 31 days. However, the original column specifies a total of 19 weeks in charts, which I think includes for future appearances in following Christmas periods and would not be suitable for the current analysis.

**DATA ANALYSIS**

For the data analysis I generate a combined chart of Probability Density Function and Cumulative Density Functions for each of the attributes that I wanted to analyse.

To create the graphs I used a function to first create the data for plotting the graphs, and a summary table to append under the graphs with further data. The next is an example of the graphs generated:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

I used these graphs to draw conclusions over the following questions:

* How long do n1 songs last in charts vs the rest of the songs?
* How many positions do n1 songs climb vs rest of the songs?
* What the starting positions of n1 songs are in charts vs the rest of the songs?

**Length of stay in charts**

When analysing the data for each of the decades, both the mean and the max length increases the closer you get to today’s date and that the max length of also increase.

Where in the 1970 to 1980 period the graph shows an average of 55.92 and very few songs going past 200 days in the charts (4 times the mean), in the 2010 to 2019 a number of songs are seen to have an average around 400.

I think that not only the average stay of the song in the charts has increased, but this increase has also in part been driven because of extreme cases where the songs stay for a significant amount of time when compared to the mean.

Gráfico, Gráfico de líneas

Descripción generada automáticamente



For the specific case of 2010 to 2019, it can clearly be seen that the shape of the distribution between the n1 songs population and the rest of the songs is different. The rest of the population ranges from 0 all the way to 585 days with a mean of 80 days, whereas the n1 songs population start at 100 days and has a max of 460 days. This is, a n1 song will as a minimum stay in the charts for 100 days.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Climb of songs in charts**

In the case of positions climbed, there also is a difference in distributions among the different decades, albeit much less discernible. There is a trend where the max in the 1970s is 123 and the max in the 2010s is 99. This could potentially point to a higher starting point of the n1 songs in the charts.

When comparing the distributions between the n1 population and the rest in this case as well we can see a remarkable difference and in this case as well there is a cut up point of 99 positions, this means that a song will not achieve n1 unless it starts in a position 100 or higher.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Further analysis

Analyse if there is a statistical difference between decades for:

* Length of stay of songs in n1 position
* Positions climbed from start to top
* Positions climbed from bottom to top

Is there any useful conclusion?

* A song will not reach n1 if it doesn´t start in the charts at position X
* A song that is located beyond position X will not become n1
* A song that has reached n1 will stay in the charts for at least X time

What are other attributes that can determine if a song will be a n1?

* Words in lyrics
* Music genre
* Artists
* Topics of music

Get the lyrics for the songs and the music genre and find the most repeated words, the most common music genres, artists or topics.

Try machine learning to do a sentiment analysis over the data.