Phase 2 - INFO 2950 Final Project

Luke Ellis

Introduction

Music has long been a passion of mine, so my mind instantly focused on the analysis of music when the open-ended nature of this project was presented. Well I certainly do not regret choosing this topic, I do realize that it is much more difficult than initially expected. Like most art forms, Music is an extremely subjective field, and the more quantifiable parts of music present many computational challenges to obtain. For example, to create a large dataset of songs with their tempos (beats per minute) is a difficult task requiring a powerful computer and a lot of time. Or if one wished to classify the genre of 200,000 songs, they would run into issues of classification before they even start thinking of how to handle such a task computationally. Many songs do not simply and obviously fall into one category of music. Different listeners across different regions classify the same piece of music as different styles or genres.

So for this project, I decided to not lose focus trying to solve a controversial aspect of music before even starting my analyses. Instead, I decided to use the Billboard Hot 100 Chart, with data over the past 60 or so years, as the centerpiece of my investigations. Since the Billboard Hot 100 represents a significant part of the culture of the United States, I figured I would attempt to cross reference this data with other major datasets, such as those relating to the economy or politics. And I'm sure between now and the final deadline of the project, I will think of a few more datasets with which I'd like to cross reference. My goal in the next couple weeks is to figure out how to merge the lyrics of the songs to the dataset, as this will open new doors for analysis.

Research Questions

- What trends does the Billboard Hot 100 contain within itself?
- Does the volatility of the Billboard Hot 100 correlate inversely with the volatility of the stock market?
- How do the lyrics/content of songs change during election years?
- Are songs about romance more likely to land a spot on the Billboard Hot 100?
- What are the most common lyrics in songs on the Hot 100?

Data Appendix

Here I'll explain the current state of my data. I have plans to merge more data in the future.

In [3]:
```
import pandas as pd
import numpy as np
import datetime
```

```
from datetime import date
import matplotlib.pyplot as plt
```

In [4]:

```
charts = pd.read_csv("Hot Stuff.csv")
vix = pd.read_csv("VIX.csv")
```

- Hot Stuff.csv is a file of the Billboard Hot 100's entries from August 2, 1958 to May 29, 2021
- VIX.csv is a table of daily data for the VIX Volatility Index from January 3, 1990 to October 19, 2021

In [5]:

```
charts.WeekID = pd.to_datetime(charts.WeekID, errors='ignore', infer_datetime_fo
vix.Date = pd.to_datetime(vix.Date, errors='ignore', infer_datetime_format=True)
```

After importing the CSVs to DataFrames, I converted each tables "date" columns to the Python datetime library.

In [6]:

```
start = datetime.datetime(1990, 1, 9) ## Earliest VIX data
end = datetime.datetime(2021, 5, 30) ## Latest Billboard Hot 100 data

shortChart = charts.loc[charts.WeekID > start]
shortChart.sort_values(by='WeekID')
```

Out[6]:

	url	WeekID	Week Position	Song	Performer	SongID
288514	http://www.billboard.com/charts/hot-100/1990-0...	1990-01-13	95	Cover Girl	New Kids On The Block	Cover GirlNew Kids On The Block
90453	http://www.billboard.com/charts/hot-100/1990-0...	1990-01-13	15	Two To Make It Right	Seduction	Two To Make It RightSeduction
318630	http://www.billboard.com/charts/hot-100/1990-0...	1990-01-13	81	Get On Your Feet	Gloria Estefan	Get On Your FeetGloria Estefan
318227	http://www.billboard.com/charts/hot-100/1990-0...	1990-01-13	64	Leave A Light On	Belinda Carlisle	Leave A Light OnBelinda Carlisle
158287	http://www.billboard.com/charts/hot-100/1990-0...	1990-01-13	43	Nothin' To Hide	Poco	Nothin' To HidePoco
...	...	...	...	...	...	...
19008	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	67	His & Hers	Internet Money, Don Toliver & Lil Uzi Vert Fea...	His & HersInternet Money, Don Toliver & Lil Uz...
204390	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	54	Your Power	Billie Eilish	Your PowerBillie Eilish

	url	WeekID	Week Position	Song	Performer	SongID
1793	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	58	Build A Bitch	Bella Poarch	Build A BitchBella Poarch
224183	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	74	Hold On	Justin Bieber	Hold OnJustin Bieber
74511	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	63	Ski	Young Thug & Gunna	SkiYoung Thug & Gunna

163795 rows × 10 columns

I then shortened the original charts DataFrame (which had over 300,000 entries) to just cover the time period that the VIX and Hot 100 datasets overlapped.

In [9]:

```
## Unique weeks on the Hot 100 in our time range
chartWeeks = sorted(shortChart.WeekID.unique())

chartV = []
for week in chartWeeks:
    songsThatWeek = shortChart.loc[shortChart.WeekID == week]
    chartV.append(songsThatWeek['Weeks on Chart'].sum())

print("First 5 weeks, Chart Volatility Score: ", chartV[:5])
```

```
First 5 weeks, Chart Volatility Score:  [1030, 1018, 1012, 999, 996]
```

Above I created the ***Chart Volatility Score*** for the Billboard Hot 100 data. In the for loop, I found the 100 songs on the chart for each week and then took the sum of their "Weeks on Chart" data. The higher the score, the less volatile the chart was that week. When a new song enters the chart, it has a "Weeks on Chart" value of 1. So the weeks with lower chart volatility scores have more songs that are newer to the Hot 100.

In [11]:

```
fiveDayAvg = [0, 0, 0, 0]
for i in range(4, vix.shape[0]):
    fiveDayTotal = vix.at[i, 'Close'] + vix.at[i-1, 'Close'] + vix.at[i-2, 'Clos
    fiveDayAvg.append(fiveDayTotal / 5)

vix['FiveDayAvg'] = fiveDayAvg
vix.head(10)
```

Out[11]:

	Date	Open	High	Low	Close	Adj Close	Volume	FiveDayAvg
0	1990-01-03	18.190001	18.190001	18.190001	18.190001	18.190001	0.0	0.000
1	1990-01-04	19.219999	19.219999	19.219999	19.219999	19.219999	0.0	0.000
2	1990-01-05	20.110001	20.110001	20.110001	20.110001	20.110001	0.0	0.000

	Date	Open	High	Low	Close	Adj Close	Volume	FiveDayAvg
3	1990-01-08	20.260000	20.260000	20.260000	20.260000	20.260000	0.0	0.000
4	1990-01-09	22.200001	22.200001	22.200001	22.200001	22.200001	0.0	19.996
5	1990-01-10	22.440001	22.440001	22.440001	22.440001	22.440001	0.0	20.846
6	1990-01-11	20.049999	20.049999	20.049999	20.049999	20.049999	0.0	21.012
7	1990-01-12	24.639999	24.639999	24.639999	24.639999	24.639999	0.0	21.918
8	1990-01-15	26.340000	26.340000	26.340000	26.340000	26.340000	0.0	23.134
9	1990-01-16	24.180000	24.180000	24.180000	24.180000	24.180000	0.0	23.530

To prepare the VIX data, I added a Five Day Average column that took the average of the closing prices of previous five open market days. This would hopefully smooth the trends in the market and help to alleviate outliers. It would also make the merging of the data a little cleaner and more accurate since the Billboard Hot 100 is also measured over a week.

In [12]:

```
volatileTable = pd.DataFrame(columns=['Date', 'ChartVScore'])
volatileTable.Date = chartWeeks
volatileTable.ChartVScore = chartV
volatileTable.head()
```

Out[12]:

	Date	ChartVScore
0	1990-01-13	1030
1	1990-01-20	1018
2	1990-01-27	1012
3	1990-02-03	999
4	1990-02-10	996

In [13]:

```
volatileTable = pd.merge_asof(volatileTable, vix, on="Date")
volatileTable.head()
```

Out[13]:

	Date	ChartVScore	Open	High	Low	Close	Adj Close	Volume	FiveD
0	1990-01-13	1030	24.639999	24.639999	24.639999	24.639999	24.639999	0.0	21.9
1	1990-01-20	1018	22.500000	22.500000	22.500000	22.500000	22.500000	0.0	24.3
2	1990-01-27	1012	26.280001	26.280001	26.280001	26.280001	26.280001	0.0	25.7
3	1990-02-03	999	24.320000	24.320000	24.320000	24.320000	24.320000	0.0	25.6

	Date	ChartVScore	Open	High	Low	Close	Adj Close	Volume	FiveD
4	1990-02-10	996	23.690001	23.690001	23.690001	23.690001	23.690001	0.0	24.1

I then made a new table to handle the volatility measures. Since the Hot 100 is usually posted on days the stock market is closed (i.e. no VIX score for that day), I merged the nearest Five Day Average to the chart posting day. So the Date column here has the date of the Billboard's chart posting, but the Five Day Average is probably from the Friday before the date. Now this dataframe is ready for some testing.

In [16]:
```
peakCharts = charts
peakCharts.sort_values('Peak Position', ascending=True).sort_values('Weeks on Ch
```

Out[16]:

	url	WeekID	Week Position	Song	Performer	
302681	http://www.billboard.com/charts/hot-100/2014-0...	2014-05-10	49	Radioactive	Imagine Dragons	Radioacti
302673	http://www.billboard.com/charts/hot-100/2014-0...	2014-03-22	45	Sail	AWOLNATION	SailAW(
302665	https://www.billboard.com/charts/hot-100/2021-...	2021-05-29	23	Blinding Lights	The Weeknd	Blinding
278572	http://www.billboard.com/charts/hot-100/2009-1...	2009-10-10	48	I'm Yours	Jason Mraz	I'm Y
278565	http://www.billboard.com/charts/hot-100/1998-1...	1998-10-10	45	How Do I Live	LeAnn Rimes	LiveLe
...	...	...	...	...	...	
69378	https://www.billboard.com/charts/hot-100/2019-...	2019-11-30	66	The Take	Tory Lanez Featuring Chris Brown	Th Lanez Cl
67753	http://www.billboard.com/charts/hot-100/2011-0...	2011-03-26	66	The Race	Wiz Khalifa	Th
69995	https://www.billboard.com/charts/hot-100/2020-...	2020-06-13	66	TKN	ROSALIA & Travis Scott	TKNF Tı
70452	http://www.billboard.com/charts/hot-100/2011-0...	2011-07-02	66	Today Is Your Day	Shania Twain	Toc DaySh
87239	http://www.billboard.com/charts/hot-100/2015-1...	2015-11-28	87	Traveller	Chris Stapleton	Tra

29389 rows × 10 columns

Finally, I made this ***Peak Charts*** reduction from the full charts data. This DataFrame contains the most useful row for each song as it grabs the row with the highest "Weeks on Chart" value. So it is the last week the song was on the chart, so it has each song's true "peak" position and true "Weeks on Chart" value. When it comes to analysis involving the date, this DataFrame isn't the best as it does not account for split weeks on the chart (i.e. a song leaving the Hot 100 for 2 weeks then coming back). It is also under 10% the size of the original DataFrame.

Data Description

Composition

The data is made up of two main CSV files and reorganized into several DataFrames.

Motivation

Funding

In [ ]:

```

```

Collection Process

All data used so far has not been collected from indivduals. They are all based on calculations with a variety of quantitative inputs. The Billboard Hot 100 is determined by an equation involving music sales, views, listens, and much more. The VIX is calculated automatically taking into consideration prices across markets and historical data. No data has been collected manually to my knowledge.

Preprocessing and Cleaning

The details of preprocessing and cleaning have been thoroughly examined in the "Data Appendix" section. No existing data has been altered, but certain records have been removed. When looking at both the VIX data and Hot 100 data, the Hot 100 data had to be reduced in order for the dates of the two datasets to overlap fully. The first week of overlapping VIX and Hot 100 data was also dropped since the five day average metric could not be calculated accurately.

Privacy Statment

No personal data has been collected from an individual at any point in this process. Human contributions to this data have only been data enterers working for either Yahoo Finance or the Billboard. No data is at risk of being stolen as all of it is already in the public domain.

Link to Source Data

Both the VIX and Hot 100 CSV files can be found and downloaded here:
https://drive.google.com/drive/folders/1a3jU_kq8fJVpKr-ltrCXVu6l2q8LdTuP?usp=sharing

Data Limitations

Exploratory Data Analysis

In [14]:

```
charts.describe()
```

Out[14]:

	Week Position	Instance	Previous Week Position	Peak Position	Weeks on Chart

	Week Position	Instance	Previous Week Position	Peak Position	Weeks on Chart
count	327895.000000	327895.000000	295941.000000	327895.000000	327895.000000
mean	50.499309	1.072538	47.604066	41.358307	9.153793
std	28.865707	0.334188	28.056915	29.542497	7.590281
min	1.000000	1.000000	1.000000	1.000000	1.000000
25%	25.500000	1.000000	23.000000	14.000000	4.000000
50%	50.000000	1.000000	47.000000	39.000000	7.000000
75%	75.000000	1.000000	72.000000	66.000000	13.000000
max	100.000000	10.000000	100.000000	100.000000	87.000000

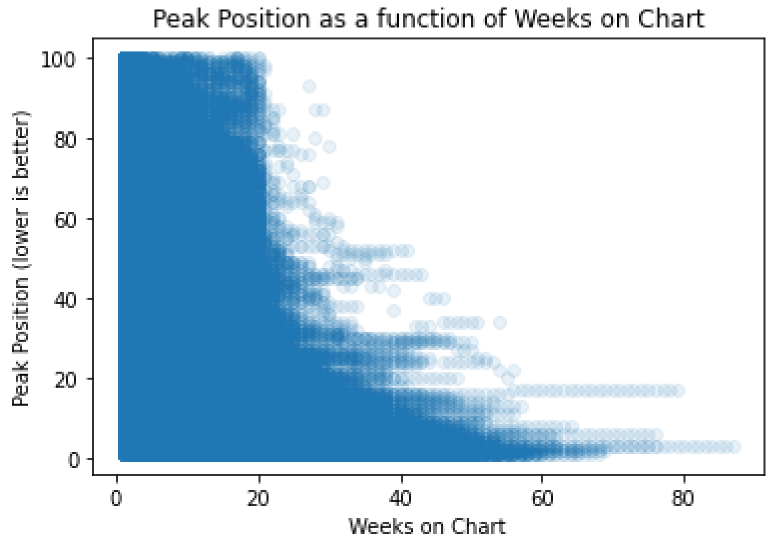
In [15]:
```
vix.describe()
```

Out[15]:

	Open	High	Low	Close	Adj Close	Volume	FiveDayAvg
count	8011.000000	8011.000000	8011.000000	8011.000000	8011.000000	8011.0	8009.000000
mean	19.557959	20.346117	18.820846	19.481100	19.481100	0.0	19.472629
std	8.105947	8.562156	7.594262	8.032406	8.032406	0.0	7.929560
min	9.010000	9.310000	8.560000	9.140000	9.140000	0.0	0.000000
25%	13.780000	14.395000	13.260000	13.750000	13.750000	0.0	13.758000
50%	17.620001	18.209999	16.969999	17.559999	17.559999	0.0	17.576000
75%	22.959999	23.780000	22.190001	22.840000	22.840000	0.0	22.756000
max	82.690002	89.529999	72.760002	82.690002	82.690002	0.0	74.618001

I first ran describe just to make sure things were working and in order. The values in the Hot 100 DataFrame are especially telling that the data is correct. The maximum Week position is 100 (as it always should be), the minimum position is 1, and the mean is about 50.5, halfway between 1 and 100. The other numbers all make sense too. The VIX data is a little less telling, but the minimums and maximums make sense.

In [17]:
```
plt.scatter(peakCharts['Weeks on Chart'], peakCharts['Peak Position'], alpha=0.1
plt.xlabel("Weeks on Chart")
plt.ylabel("Peak Position (lower is better)")
plt.title("Peak Position as a function of Weeks on Chart")
plt.show()
```



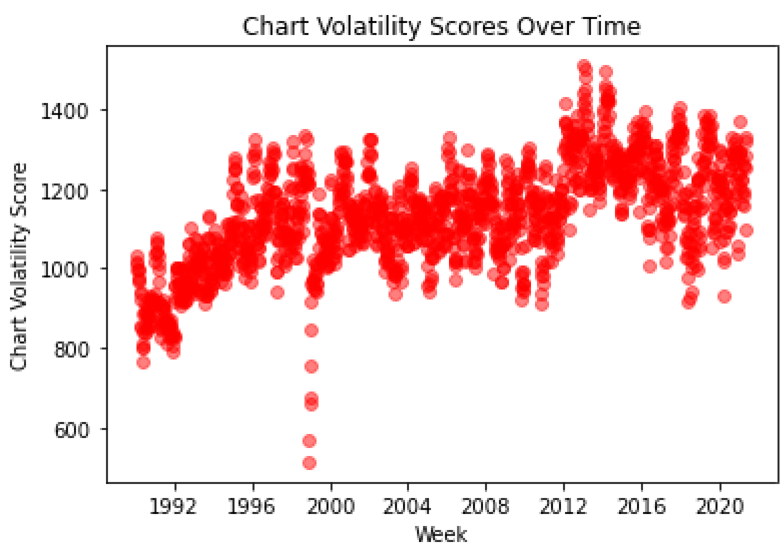
This first graph shows Peak Position (as explained in the Peak Charts DataFrame) as a function of how many weeks a song has spent on the Hot 100. As is visible, it appears songs that spend many weeks on the chart often have a higher peak. It's interesting how this scatter plot almost ended up looking like a histogram, but I guess that is what happens when you have 30,000 data points. There appears to be some sort of correlation here, so I will work on making that clearer and mathematically concrete.

In [21]:

```
plt.scatter(volatileTable.Date, volatileTable.ChartVScore, alpha=0.5, c="red")
plt.xlabel("Week")
plt.ylabel("Chart Volatility Score")
plt.title("Chart Volatility Scores Over Time")
```

Out[21]:

```
Text(0.5, 1.0, 'Chart Volatility Scores Over Time')
```

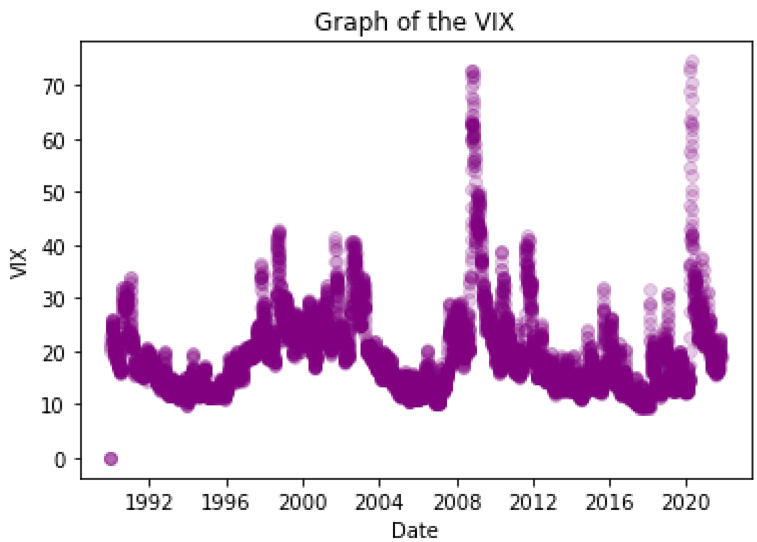


This graph shows the **Chart Volatility Scores** over time. I will probably also produce one for the whole dataset. I find it interesting how the volatility score fluctuates pretty rapidly. This will require further investigation to see if perhaps there are consistent fluctuations during a calendar or fiscal year.

In [24]:

```
plt.scatter(vix.Date, vix.FiveDayAvg, alpha=0.2, c="purple")
```

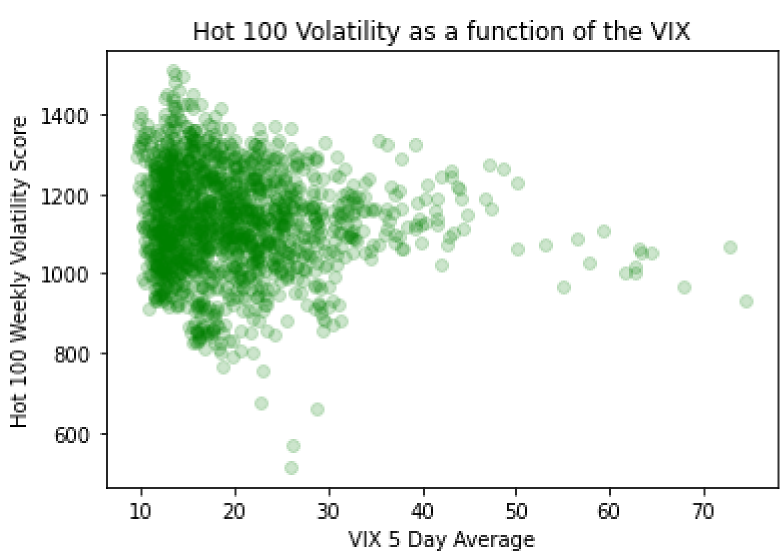
```
plt.xlabel("Date")
plt.ylabel("VIX")
plt.title("Graph of the VIX")
plt.show()
```



This graph simply shows the VIX index overtime. The VIX stands for Volatility Index, and it measures general market volatility in the U.S. Stock Exchange. The higher the index, the more volatile the markets are, and vice versa. As you can see, there is a spike in the VIX around the 2008 Recession and around the start of the COVID-19 Pandemic, both of which were turbulent times for the U.S. economy.

In [25]:

```
plt.scatter(volatileTable.FiveDayAvg, volatileTable.ChartVScore, alpha=0.2, c="g
plt.ylabel("Hot 100 Weekly Volatility Score")
plt.xlabel("VIX 5 Day Average")
plt.title("Hot 100 Volatility as a function of the VIX")
plt.show()
```



This graph is where I've compared the VIX and my own Chart Volatility Score. As you can see, it is mostly a cluster of points with a tail trailing off to the right. The cluster is very heavy though, so I'm not sure if the points going off to the right will have much influence on a fit.

I was expecting to see an inverse correlation between the VIX and Volatility Score. My informal hypothesis was that when the economy is less stable, people listen to music they are more comfortable with (i.e. high VIX means high Volatility Score). If parts of their lives are unsteady or unsure, they will take comfort in the areas they can control. I'm not going to count this thinking out just yet as I would like to play with the scaling and scoring a bit more.

Questions for Reviewers

- Am I doing too much? This is only a portion of the final report I plan to submit, but I'm wondering if including even more datasets might cause the project to lose focus?
- Would adding lyrics to my Hot 100 chart data be too cumbersome?

In [ ]:

```
```