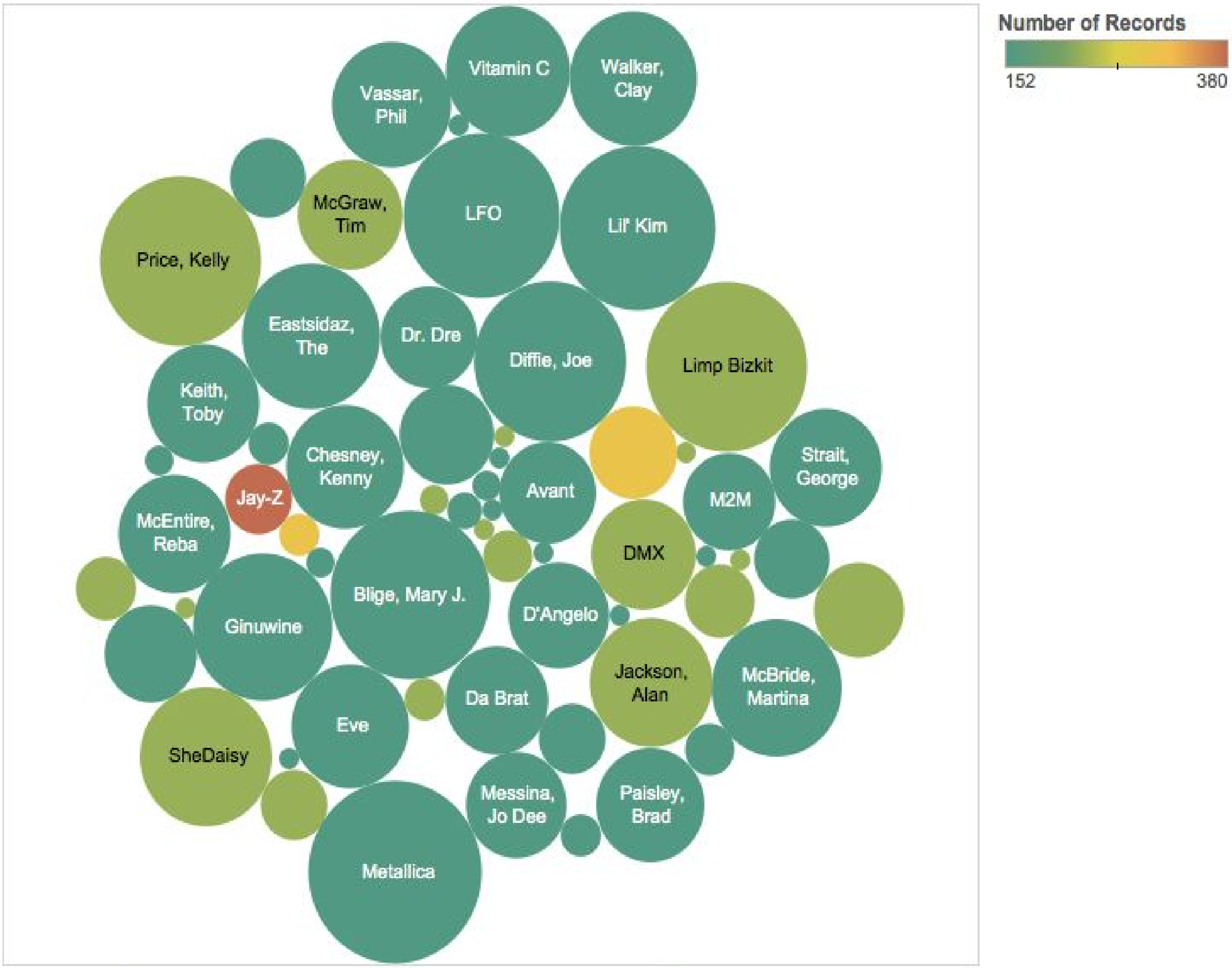
Data Science Immersive

Project 2: Data analysis and cleaning



Artist. Color shows sum of Number of Records. Size shows minimum of Rank. The marks are labeled by Artist. The data is filtered on sum of Rank, which keeps non-Null values only. The view is filtered on sum of Number of Records and minimum of Rank. The sum of Number of Records filter ranges from 150 to 380. The minimum of Rank filter ranges from 1.00 to 75.20.

Introduction

This week we explored the Billboard Music Charts top 100 list in 2000. The raw data was stored in a csv file and consisted of 316 artists that made the Billboard Music Charts at some point during a 76-week period. From the original assignment, we were asked to clean the data before hand and then come up with a problem statement. Because the changes made to this assignment occurred on Friday after I've done the preliminary data cleaning and analysis, I can't simply ignore what I've learned and unsee the data. The remainder of this paper will explain the process I've implemented to present the data in an understandable format.

Problem Statement

After being asked to explore the data, I noticed several patterns with the data itself. First, most artists who make it onto the Billboard Music Chart for 2000 didn't necessarily stay in the top 100 list for a long time. From just eyeballing the data, most appear to drop off between weeks 20 and 30. However, because the values that follow are all "NaN", we do not truly know whether the artists dropped from the charts or whether data was omitted or failed to be collected. Either way, we do not have sufficient data to make much of an argument here.

The second trend I noticed is that some artists are much more prolific at releasing tracks than others. However, more track release does not necessarily mean these tracks top the charts or stay on the chart for a long time. Furthermore, because most of the weekly ranking data were unknown values, it appears that the artist or track simply disappear after reaching a certain ranking and are never to be seen again, which is highly unusual if the data was being collected adequately.

The last trend that I observed was the genre and how many artist of each genre made it to the Billboard Music Charts. My alternative hypothesis for the dataset is that songs released in the rock genre are more likely to make it onto the Billboard Top 100 and also more likely to be in the top positions on the chart. The null hypothesis, then, is that no one genre dominates the Billboard Music Charts. Despite being able to derive a hypothesis to test against the dataset, the large quantity of unknown values does compromise any conclusion drawn from the analysis of the data.

Exploratory analysis of the data

Data exploration is the first step to data analysis and usually involves summarizing the main characteristics of a dataset, according to TechTarget's Search Business Analytics. I began with the raw csv file, which was then read into Pandas, one of Python's modules. The raw data was converted into a dataframe. From there, the dataframe's head (which contains the first 5 rows of data of the dataframe), tail (which contains the last 5 rows of data), and summary were extracted and explored.

jupyter project2_script Last Checkpoint: Last Thursday at 11:28 PM (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 2

```
import pandas as pd
import numpy as np
billboard = pd.read_csv('https://raw.githubusercontent.com/ga-students/DSI-DC-1/master/week-02/project-02/assets/billboard.csv?tok
pd.set_option('display.width', 1000) # sets so that all columns have appropriate width
pd.set_option('display.max_columns', 30) # sets so that all columns are displayed
pd.set_option('display.max_rows', 50) #sets so that all rows are displayed
billboard.shape #assesses the total size of the dataset
billboard #prints out original dataset
```

Out[2]:

	year	artist.inverted	track	time	genre	date.entered	date.peaked	x1st.week	x2nd.week	x3rd.week	x4th.week	x5th.week	x6th.wee
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	78	63.0	49.0	33.0	23.0	15.0
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	15	8.0	6.0	5.0	2.0	3.0
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	71	48.0	43.0	31.0	20.0	13.0
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	41	23.0	18.0	14.0	2.0	1.0
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	57	47.0	45.0	29.0	23.0	18.0
5	2000	Janet	Doesn't Really	4:17	Rock	2000-06-17	2000-08-26	59	52.0	43.0	30.0	29.0	22.0

jupyter project2_script Last Checkpoint: Last Thursday at 11:28 PM (autosaved)

File Edit View Insert Cell Kernel Help | Python 2

In [3]:

```
bill_head = billboard.head()
pd.set_option('display.max_columns', 30)
bill_head
```

Out[3]:

	year	artist.inverted	track	time	genre	date.entered	date.peaked	x1st.week	x2nd.week	x3rd.week	x4th.week	x5th.week	x6th.week
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	78	63.0	49.0	33.0	23.0	15.0
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	15	8.0	6.0	5.0	2.0	3.0
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	71	48.0	43.0	31.0	20.0	13.0
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	41	23.0	18.0	14.0	2.0	1.0
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	57	47.0	45.0	29.0	23.0	18.0

5 rows × 83 columns

In [4]:

```
bill_tail = billboard.tail()
bill_tail
```

jupyter project2_script Last Checkpoint: Last Thursday at 11:28 PM (autosaved)

File Edit View Insert Cell Kernel Help | Python 2

In [5]:
```
bill_describe = billboard.describe()
bill_describe
```

Out[5]:

	year	x1st.week	x2nd.week	x3rd.week	x4th.week	x5th.week	x6th.week	x7th.week	x8th.week	x9th.week	x10th.week	x11th.we
count	317.0	317.000000	312.000000	307.000000	300.000000	292.000000	280.000000	269.000000	260.000000	253.000000	244.000000	236.0000
mean	2000.0	79.958991	71.173077	65.045603	59.763333	56.339041	52.360714	49.219331	47.119231	46.343874	45.786885	45.4745
std	0.0	14.686865	18.200443	20.752302	22.324619	23.780022	24.473273	25.654279	26.370782	27.136419	28.152357	29.0605
min	2000.0	15.000000	8.000000	6.000000	5.000000	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000
25%	2000.0	74.000000	63.000000	53.000000	44.750000	38.750000	33.750000	30.000000	27.000000	26.000000	24.750000	22.0000
50%	2000.0	81.000000	73.000000	66.000000	61.000000	57.000000	51.500000	47.000000	45.500000	42.000000	40.000000	42.5000
75%	2000.0	91.000000	84.000000	79.000000	76.000000	73.250000	72.250000	67.000000	67.000000	67.000000	69.000000	69.2500
max	2000.0	100.000000	100.000000	100.000000	100.000000	100.000000	99.000000	100.000000	99.000000	100.000000	100.000000	100.000

8 rows × 77 columns

In [6]:
```
pd.set_option('display.max_rows', 30)
billboard.iloc[:, [2, 1, 3, 5, 6, 7, 82]]
```

Out[6]:

	track	artist.inverted	time	date.entered	date.peaked	x1st.week	x76th.week
0	Independent Women Part I	Destiny's Child	3:38	2000-09-23	2000-11-18	78	NaN
1	Maria, Maria	Santana	4:18	2000-02-12	2000-04-08	15	NaN

After the initial phase of exploring the dataset, I performed a variety of functions, some of which included listing the columns out and finding the datatype, renamed column headings that were poorly or improperly named, and used the melt function to pivot the weekly ranking arrangement from wide to long and added an average ranking column.

jupyter project2_script Last Checkpoint: Last Thursday at 11:28 PM (autosaved)

File Edit View Insert Cell Kernel Help | Python 2

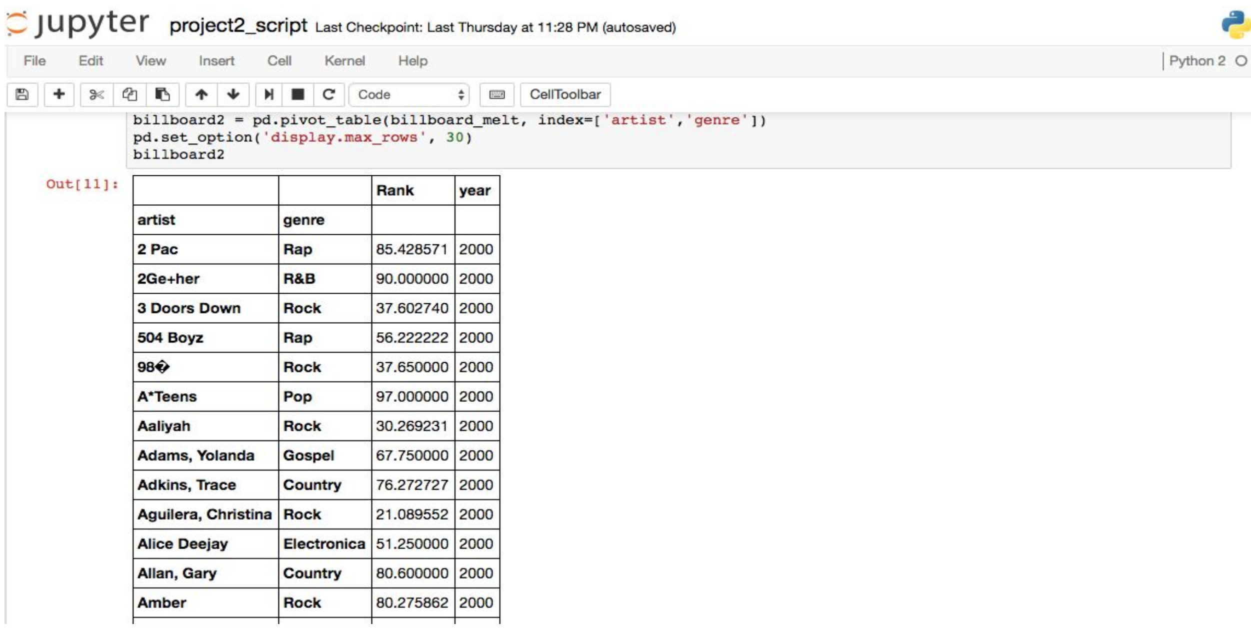
In [10]:
```
# Using Pandas' built in melt function, pivot the weekly ranking data to be long rather than wide. As a result,
# you will have removed the 72 'week' columns and replace it with two: Week and Ranking. There will now be multiple
# entries for each song, one for each week on the Billboard rankings.
billboard_melt = pd.melt(billboard_copy, id_vars=['year','artist','track','track.length','genre','date.entered','date.peak
pd.set_option('display.max_rows', 30)
billboard_melt
```

Out[10]:

	year	artist	track	track.length	genre	date.entered	date.peaked	Week	Rank
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	x1st.week	78.0
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	x1st.week	15.0
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	x1st.week	71.0
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	x1st.week	41.0
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	x1st.week	57.0
5	2000	Janet	Doesn't Really Matter	4:17	Rock	2000-06-17	2000-08-26	x1st.week	59.0
6	2000	Destiny's Child	Say My Name	4:31	Rock	1999-12-25	2000-03-18	x1st.week	83.0
7	2000	Iglesias, Enrique	Be With You	3:36	Latin	2000-04-01	2000-06-24	x1st.week	63.0
8	2000	Sisqo	Incomplete	3:52	Rock	2000-06-24	2000-08-12	x1st.week	77.0
9	2000	Lonestar	Amazed	4:25	Country	1999-06-05	2000-03-04	x1st.week	81.0
10	2000	N'Sync	It's Gonna Be Me	3:10	Rock	2000-05-06	2000-07-29	x1st.week	82.0
11	2000	Aguilera, Christina	What A Girl Wants	3:18	Rock	1999-11-27	2000-01-15	x1st.week	71.0

The reasons behind all the data exploration and rearrangement is so that I can get a clearer picture of the dataset as a whole. Column names were changed because they were unclear, datatypes for some columns had to be changed because they weren't integers of floating numbers which would make data visualization difficult, the multiple columns of weekly ranking had to be sliced or melted because the original display lacked clarity and organization, etc. Ideally, the more refined one can make the charts, the cleaner and more organized the data becomes for statistical analysis.

Being able to create pivot tables of the given dataset is another great way to organize the data in a much more readable format. As seen below, after melting the weekly ranking columns into one, I used the pivot table function in panda to create a chart that is much more readable and understandable, indexed by artist and genre. The columns relevant to testing my hypothesis remain, mainly that of average ranking of the artists and genre.




```
billboard2 = pd.pivot_table(billboard_melt, index=['artist','genre'])
pd.set_option('display.max_rows', 30)
billboard2
```

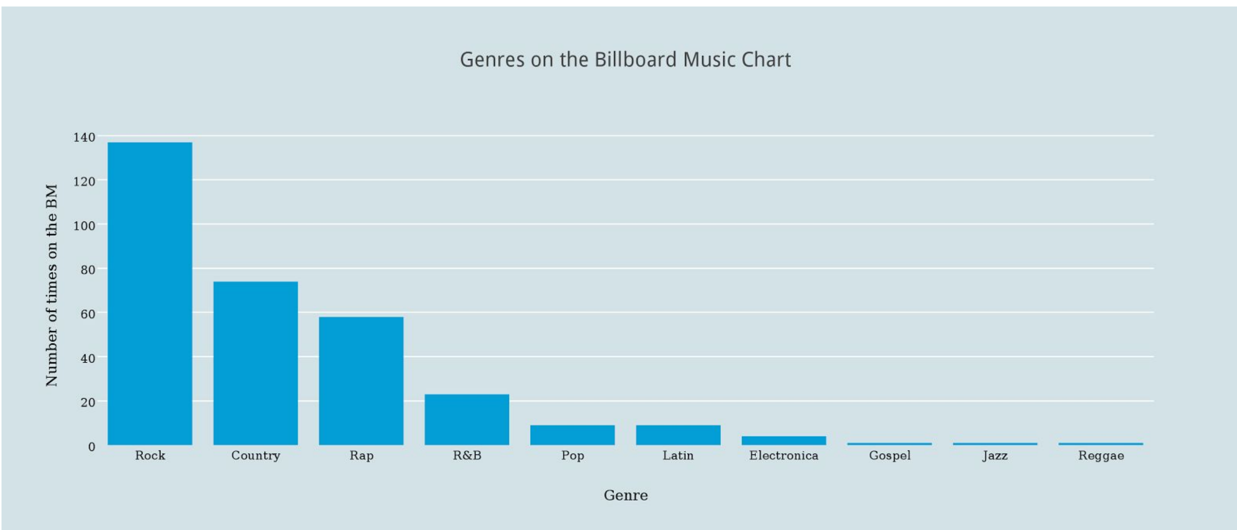
		Rank	year
artist	genre		
2 Pac	Rap	85.428571	2000
2Ge+her	R&B	90.000000	2000
3 Doors Down	Rock	37.602740	2000
504 Boyz	Rap	56.222222	2000
98�	Rock	37.650000	2000
A*Teens	Pop	97.000000	2000
Aaliyah	Rock	30.269231	2000
Adams, Yolanda	Gospel	67.750000	2000
Adkins, Trace	Country	76.272727	2000
Aguilera, Christina	Rock	21.089552	2000
Alice Deejay	Electronica	51.250000	2000
Allan, Gary	Country	80.600000	2000
Amber	Rock	80.275862	2000

A major problem with this dataset is the exhaustive amount of NaN (not a number) values found on multiple columns and rows. Because these values are unknown or missing, they compromise the ranking mean.

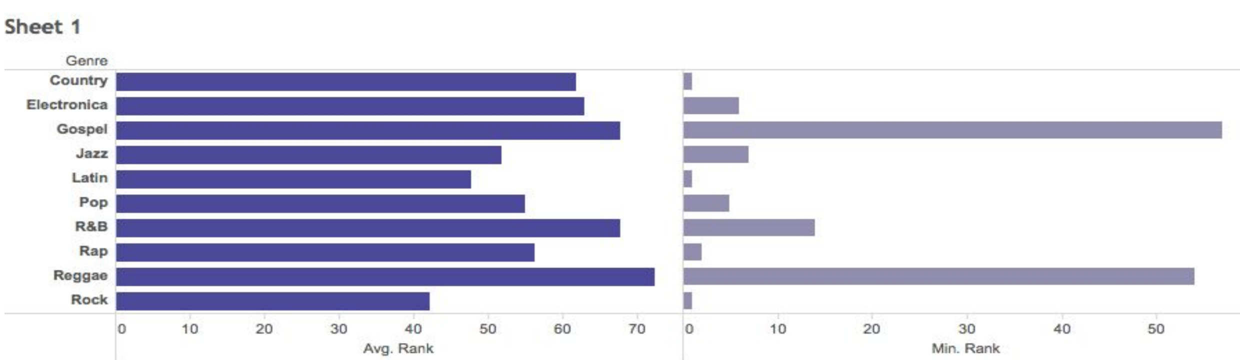
Data Visualization

After combing through the dataset, we should ideally find a way to present the information in a meaningful, succinct and appealing way. This is the process of data visualization. There are various libraries in Python that can perform data visualization such as Matplotlib, Pyplot, Plotly, and Seaborn. There are proprietary tools outside of Python that can do the same, one of which is Tableau. The advantages to using libraries in Python is that they're open-source, and readily accessible. Tableau is a proprietary software that costs $1000+ per person. Tableau does not require extensive knowledge of programming to use, but does require training to use its full suite of functions. Tableau also uses a graphical user interface, which may seem easier to use. However, once a user knows how to utilize the various functions of Python and can program well, the toolkits that come with the Python libraries are just as extensive as Tableau.

Below are some of the major graphs and charts created to visualize the data collected. From Plotly, a very simple bar chart is graphed showing how each genre performed on the Billboard Music Chart. Songs categorized as “Rock” appear on the Billboard Music Chart more often than other genres. However, this is not a telling conclusion since we know the dataset had missing values, which shouldn’t be tampered with by adding values that aren’t necessarily true nor recorded in place of the “NaN” values.



Using Tableau, another comparison was made between the genre of music against both the average ranking and the minimum ranking.



Average of Rank and minimum of Rank for each Genre.

Because this is the Billboard Music Charts, having a lower number on the ranking scale actually means better performance since it means that particular song made it to the number 1 or 2 spots on the list out of 100. Based on the comparisons above songs from the Rock genre had the lowest average ranking as well as being one of 3 genres that made the chart at number 1.

Conclusion

The write-up above demonstrates how important the process of data cleaning and exploration is to the analysis and presentation of the data. Without methods to filter the incoherent and messy raw data, a data scientist would never be able to make sense of the data and present it to stakeholders. Without clear presentation material to tell a story, the data collected is essentially useless--they exist but aren't given meaning, and data is capable of giving a lot of meaning whatever subject matter we are studying.

At the same time, if there is insufficient data such as missing or unknown values as in the case of the dataset given for this project, faulty methodology in the collection of data during the experimental design phase of a study, or if improper programming techniques were utilized to clean the data, the analysis and any conclusions drawn from that analysis would be faulty and not reproducible. Each step of the analysis process is important to maintaining the integrity of the presentation of such data.