Data Visualization

The Evolution of Hip Hop in The World

Aglind Reka
Hed Derbel
Youssef Bouchida
Paul Peyssard

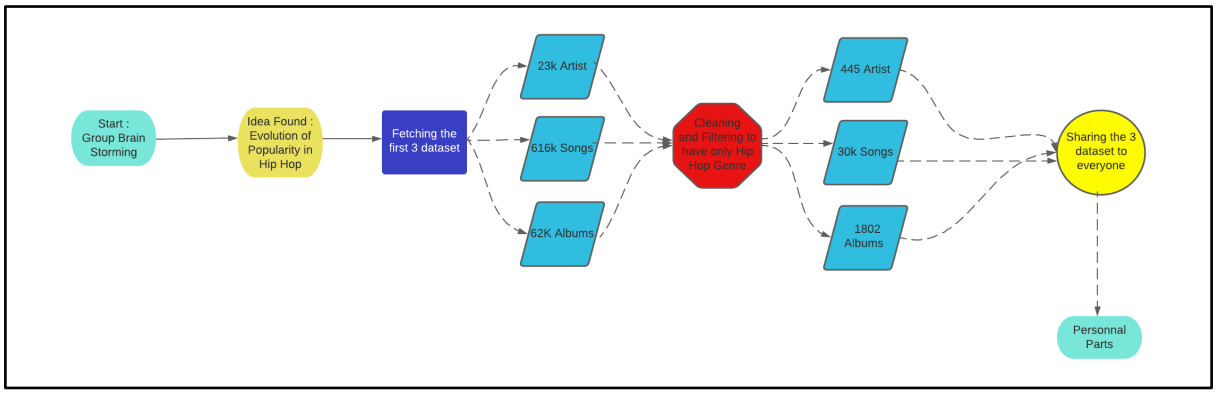
December 12, 2022

MSc 1 Data Science & Artificial Intelligence
Marco Winckler
Aline Menin

Summary

Introduction

This document presents our project for the class Data Visualization where we have to visualize the WASABI Dataset, a database gathering many information about deezer's artist, songs and albums.
As we had a lot of information, we wanted to concentrate on something narrower. We decided to work on the evolution of the Hip Hop genre. The evolutions in a large field, each of us has its own definitions of evolution that he will explain in his personal part.
In this report, you will see how we managed to fetch, select and clean the WASABI dataset in order to visualize it. It was not an easy task as a big part of the dataset was unknown data.
You can see below a workflow that represents our group work for this project.



You can find all the code for this project here.

I Processing the data

In order to fetch the data, we changed the offset to 0 and the number of loops to 100 to be able to fetch more data.

```
offset = 0 # where to start searching for data in the dataset ; here start the search from the 72000th record

get_url <- "wasabi.i3s.unice.fr/api/v1/artist_all/" # this url gets data for all artists in the wasabi API
# see https://wasabi.i3s.unice.fr/apidoc/ to know what pattern to use to recover the data you want

albums_df <- tibble()
songs_df <- tibble()
artist_df <- tibble()

for (i in 1:100) {
  print(paste(i, " requesting data..."))
```

After this step, we had around 23 000 artists, 616 000 sounds and 62 000 albums in our dataset.
The next step was to select only the columns we were interested in.

```
artists=select(artist_df,"id","name","locationInfo","genres","labels","type","lifeSpan.ended","lifeSpan.beg
albums=select(albums_df,"_id","name","genre","id_artist","title","publicationDate","deezerFans","country","
songs=select(songs_df,"id", "position","lyrics","id_album","isClassic","title","length","explicitLyrics","r
```

According to our theme and the visualization that each of us wanted, who chose to keep only theses columns :

- For the artists : id, name, locationInfo, genres, labels, type, lifeSpan.ended, lifeSpan.begin, lifeSpan.end, gender deezerFans, id_artist_discogs, dpg_genre

- For the albums : _id, name, genre, id_artist, title, publicationDate, deezerFans, country, dateRelease, language

- For the songs : position, lyrics, id_album, isClassic, title, length, explicitLyrics, rank, publicationDate, language, language_detect, repeatedness, genre, recordLabel, recorded, runtime, award, subject

As we wanted only the Hip Hop songs, albums and artists, the next step was to get rid of every other genre in the three dataset. For this, analyzed the unique value of each dataset in their genre category and select only the genre we were interested in :

- For the albums :

```
unique(album$genre)
HipHopAlbums=c("Hip Hop","R&amp;B","Southern Hip Hop","East Coast Hip Hop","West Coast Hip Hop","Nerdcore Hip
Hop","Rap Rock","French Pop","Rap Metal","Underground Hip Hop","Gangsta Rap","Experimental Hip Hop","French
Hip Hop","Australian Hip Hop","Hardcore Hip Hop","French Pop&#x200E;","Political Hip Hop")
albumRap=albums[albums$genre %in% HipHopAlbums,]
```

It was much more complicated for the songs and artists because there were a lot more genres so we used the function grepl to see which genre were containing keywords we could use (Rap, RAP, rnb…)

- For the Artists :

```
HipHopArtists=c()
for (i in 1:length(unique(artists$genre))){
  if (grepl('Hip Hop',unique(artists$genre)[i])
      || grepl('rap',unique(artists$genre)[i] )
      ||grepl('RAP',unique(artists$genre)[i] )
      ||grepl('Rap',unique(artists$genre)[i] )
      ||grepl('hip hop',unique(artists$genre)[i] )
      ||grepl('Hip-Hop',unique(artists$genre)[i] )
      ||grepl('R&B',unique(artists$genre)[i] )
      ){
    HipHopArtists=append(HipHopArtists,unique(artists$genre)[i])
  }
}
ArtistRap=artists[artists$genre %in% HipHopArtists,]
```

After this part, we had our new database with the columns we were interested in and just from the Hip hop scene. Then we saved it in another csv file in order to share it and start our personnel parts.

As we could not match some artists ID with some of our songs or albums, some of us fetched more data just in order to get the name of the song's artist or album's artist, but we will see it in the next part.

III Personal parts

a) Paul Peyssard : The evolution of popularity in Hip Hop

1) Processing of data

For my part, I chose to visualize the evolution of popularity in Hip Hop music in the WASABI dataset. The popularity represents different things, I chose especially the popularity in terms of rank for the songs and Deezer fans for albums and artists.
First, as we fetched and processed the data on my computer, I had to erase a lot of the memory with *rm(list=ls())* as my project was becoming too heavy with many datasets. After this, I imported again only the datasets I needed so I could start to work on my personnel part.
In order to have datasets with known rows, I put every empty string, na value on NULL and non valid values as “NULL” so I could have it all together.
After that, i got rid of some columns i did not want in my personal part :

```
song=select(song,-position,-lyrics,-explicitLyrics)
song=select(song,-isClassic,-repeatedness,-subject)
artist=select(artist,-id_artist_discogs,-dbp_genre)
```

My goal is to work on the popularity of songs, artists and albums, so a lot of my work will be on the rank for the songs and the deezerFans for the albums and artists. In order to have a better vision of the dataset, I then ordered theses 3 dataset by these columns :

```
song=song %>% arrange(rank)
album=album %>% arrange(desc(as.numeric(deezerFans)))
artist=artist %>% arrange(desc(as.numeric(deezerFans)))
```

To have even a cleaner view, I put these same columns as the first columns of the datasets.

```
artist <- artist %>%
  dplyr::select("deezerFans", everything())
album <- album %>%
  dplyr::select("deezerFans", everything())
song <- song %>%
  dplyr::select("rank", everything())
```

In the song dataset, there were some songs that had multiple occurrences with the same id rank etc.. So i erased the duplicates with :

```
song <- song[!duplicated(song$rank), ]
```

After this step, the cleaning of my data was more or less done, but now I want to add some columns in order to be able to display better visualizations. In the song and album dataset for exemple, I wanted to have a column “Artist Name” and “Album Name” in order to associate artists to their song. For that, I used the song, albums and artist IDs.

```
song_albumID=c(song$id_album)
albumName=c()
for (i in 1:length(song_albumID)){
  if(song_albumID[i] %in% album$X_id){
    albumName=append(albumName,album$title[album$X_id==song_albumID[i]])
    idArtist=album$id_artist[album$X_id==song_albumID[i]]

  }else{
    albumName=append(albumName,"unknown")
  }
}
song=song%>%mutate(song,Album_Name=albumName)
```

Here, I created a vector with all the IDs of albums in the song dataset in order to compare it to the album dataset and get their titles. After that, I created a new column “Album_Name” with *mutate()*. If album id did not match anything in the album dataset, their name would be set to “unknown”.

As we dont have the artist id in the song dataset, it was a little bit more difficult to get the artist name, I had to go from the song to the album and then to the artist with more or less the same process.

I also did the same thing to have the artist name in the album dataset :

```
id_artist3=c(albumtest$id_artist)
artistName3=c()
for (i in 1:length(id_artist3)){
  if(id_artist3[i] %in% moreArtist$id){
    artistName3=append(artistName3,moreArtist$name[moreArtist$id==id_artist3[i]])
  }else{
    artistName3=append(artistName3,"unknown")
  }
}

album=album%>%mutate(album,Artist_name=artistName3)
```

The problem was that a lot of artists and albums had unknown names now, as we did not fetch enough data to proceed to the comparaision and a lot of top artist or top songs were left without too much information, it was a problem for me as I wanted to see the popularity and needed the top of the Hip Hop scene. The solution I found was to fetch all the WASABIi dataset in order to match the three datasets IDs. It took me around 10 hours overnight to fetch the whole WASABI dataset. After fetching, I did the same process as shown before and I managed to retrieve all the artist and album names I wanted.

I need to have a column with only the year of release of the songs or albums, so I decided to create a new column, year_publication.

```
years=c()

for (i in 1:nrow(song)){
  years=append(years,substr(song$publicationDate[i],1,4)
)
}
song=song%>%mutate(year_publication=years)
```

Also, as I wanted to compare hip hop songs only between them, I had to create another rank columns corresponding to the rank inside Hip Hop genre : rankHere

```
song=song%>%mutate(rankHere=c(1:601))
```

It was very easy as my songs were ordered in rank, I just had to increase the number by one in each song (601 being the number of rows).

2) Shiny UI

For the Shiny UI, I used a fluidPage type of page as I wanted to have a single shiny App for my whole project. In this page, I put all the variables I wanted to use in order to filter my plots according to the user's will. I used many filters including :

-One select input to change the plot

-One slider input to change the years in plot 1 and 2

-One Radio button to change the color in plot 1 and 2

-Two Radio button to change the gender in plot 3 and 4 (different because options were not the same)

In order to show, hide, enable and disabled the filter according to the plot, I used the library shinyjs with the function shinyjs::enable and others.

In my main panel, I put different output according to my needs :

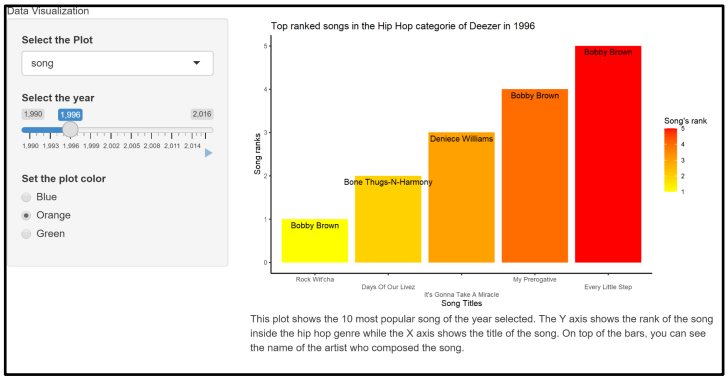
-One plotOutput for the plot 1,2 and 4
-One plotlyOutput for the interactive plotly number 3
-One tableOutput to display an interactive table with the plot 4 brusher
-One textOutput to display the descriptions of the plots

3) Shiny Server and Visualization

I then put all my output on the server. In order to be able to change the plot for the plotOutPut, I did a condition observer that changed the plot displayed according to the select input seen previously. The same process was done to show or hide the non needed filters and to change the description.

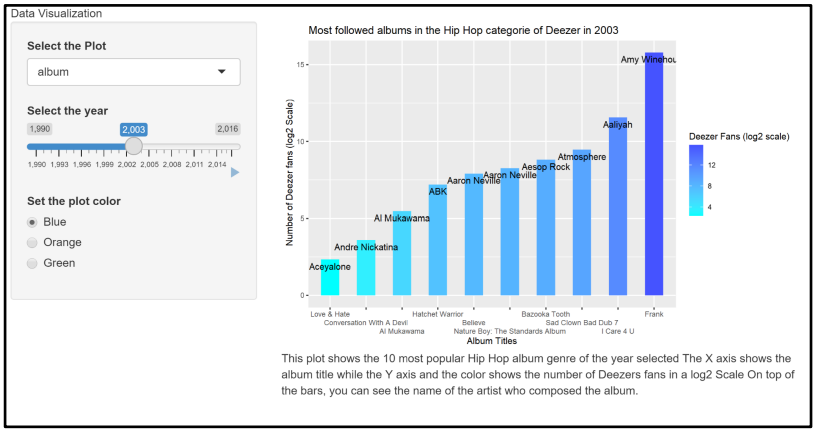
1st plot : Evolution of song's rank over the year :

The first bar plot represents the evolution of the 10 best songs over the year in terms of Deezer Fans. Here, I created a new variable "rankHere" to have only the top in the Hip Hop category. When you change the year's slider, the year of the plot changes. The longer the bar and the more intense the color, the better the sound is ranked. On the Y axis you can find the rank from 1 to 10, on the X axis you can find the name of the song and on top of the bar you can find the name of the artist.



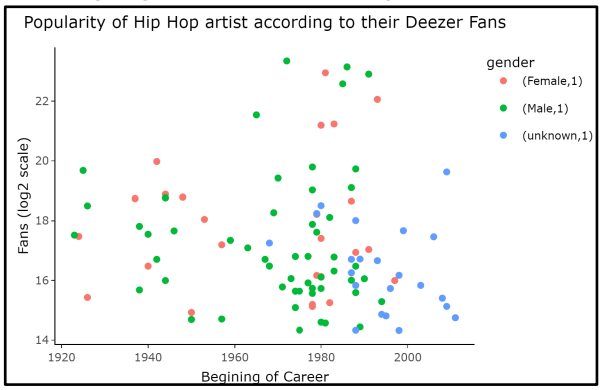
2nd plot : Evolution of album's Deezer fan's over the year :

The second bar plot represents the evolution of the 10 best albums over the year in terms of Deezer fans.
Using the same process as for the 1st plot, I chose to plot the number of Deezer Fans by album in a log2 Scale. On this vertical barchart, you can see on the X axis the album title, on top of the bars you can see the artist name and on the Y axis is the number of Deezer Fans. You can also choose the year and the color you want to have. The code is more or less the same than the previous one but simpler.



3rd plot : Interactive scatter plot of artist's fans according to the beginning of their career :

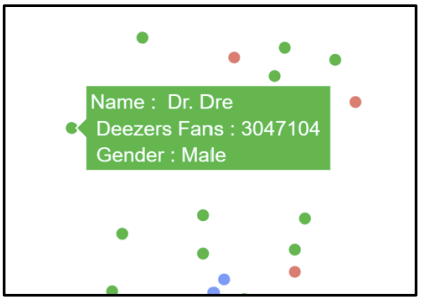
For this plot, I used ggplotly in order to have more information with hover and tooltips.
In order to be able to compare the popularity of artists and their time, I chose here to do a scatter plot with, in the X axis, the year they started their career and in the Y axis, the number of Deezer Fans (log2 scale). The color represents the gender of the artist, however, you can choose to display only one single gender or multiple gender.



As I want the most popular artist, I chose to take the 25 most popular artist of this new dataset and make a scatter plot :

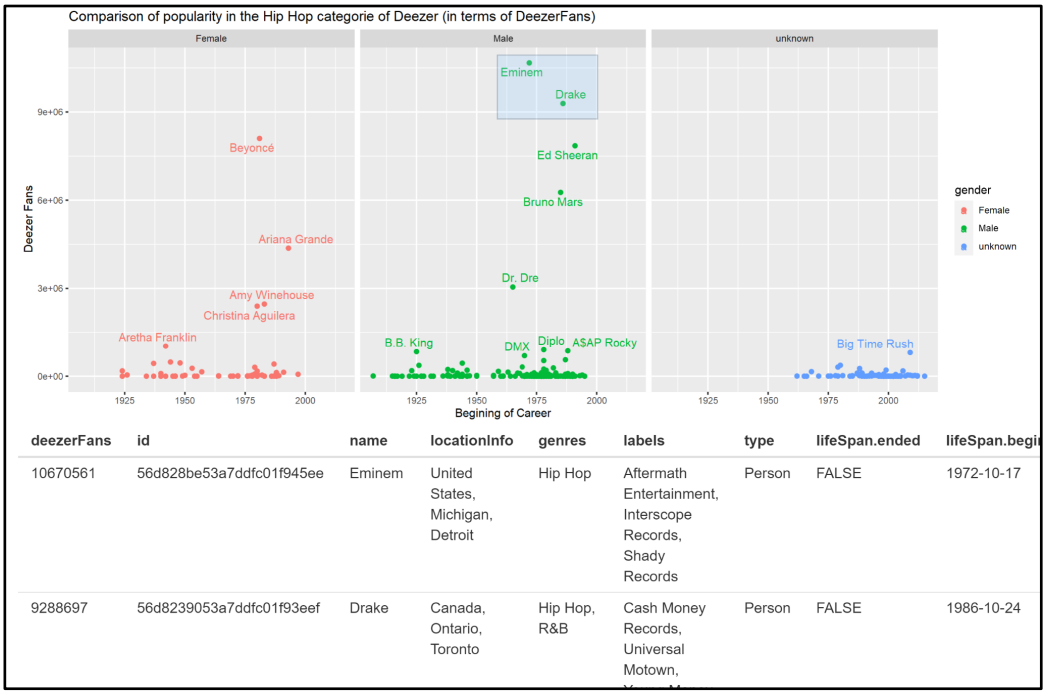
```
head(dfbySex,25)%>%ggplot(aes(y=log2(as.numeric(deezerFans)),x=substr(lifeSpan.begin,1,4),color=gender))+
  geom_point()+
  labs(title="Artist with the most deezer Fans",x="Begining of Career",y="Fans (log2 scale)")+
  scale_fill_manual(values = c("#f9ebaf", "#1f2c33","#1f2c33"))+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_text_repel(aes(label = name))
```

The ggplotly tooltips shows more informations about points of the plot :



4th plot : Interactive Scatter plot : Comparison of popularity between gender :

This plot is more or less the same type of data but with a different representation. Here, I wanted to show the comparison of popularity between genders with more artists than in the previous plot but with the same axis. Also, to have an interactive aspect, I added a table on the bottom of the plot. When you select data inside the plot, it changes the table in order to show the artist's information.



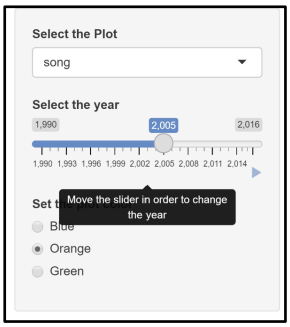
deezerFans	id	name	locationInfo	genres	labels	type	lifeSpan.ended	lifeSpan.begi
10670561	56d828be53a7ddfc01f945ee	Eminem	United States, Michigan, Detroit	Hip Hop	Aftermath Entertainment, Interscope Records, Shady Records	Person	FALSE	1972-10-17
9288697	56d8239053a7ddfc01f93eef	Drake	Canada, Ontario, Toronto	Hip Hop, R&B	Cash Money Records, Universal Motown,	Person	FALSE	1986-10-24

The code to display this plot is the same as the previous but with some the addition of facet:

```
artistKnown2%>%ggplot(aes(x=as.integer(substr(lifeSpan.begin,1,4)),y=as.integer(deezerFans),color=gender))+
  geom_point()+
  facet_wrap(facets=~ gender)+
  labs(title="Comparaision of popularity by gender",x="Begining of Career",y="Deezer Fans")+
  geom_text_repel(aes(label = name))
```

Additional Tooltip :

I also put a ToolTip in order to describe the sliderInput :



b) Youssef Bouchida :

i) Hip-Hop Artists With Most DeezerFans Per Country

1) Data Cleaning

For this visualization I needed 3 main fields which are the following :

- Artist's Name
- Artist's DeezerFans
- Artist's Country

Since we retrieved our personal data filtered to only the Hip-Hop genre, it was easier to go through the cleaning phase and perform the requested visualization. First, I started by loading the dataset on RStudio:

```r
artists <- read.csv("/Users/youssefbouchida/Downloads/M1 DSAI/DATA VIZ/DataBaseHipHopOnly/Artist_HipHop_Only.csv")
head(artists)
```

Description: df [6 × 13]

	id <chr>	name <chr>	locationInfo <chr>
1	56d7e9246b60c09814f93e65	A Broken Silence	Australia, New South Wales, Sydney
2	56d7e9446b60c09814f93ebf	A Gun Called Tension	United States, Washington
3	56d7e9cc6b60c09814f93f74	A$AP Rocky	United States, New York, New York City
4	56d7e9d76b60c09814f93f8e	A.B.N.	United States, Texas, Houston
5	56d7e9e36b60c09814f93faa	A.L.T.	United States, California, El Monte
6	56d7e9ea6b60c09814f93fbb	A04	Finland, Uusimaa, Porvoo

Removing all the unnecessary columns:

```r
#artists <- artists[, -1]
artists <- artists[, -c(4:9)]
head(artists)
```

Retrieving only the Country name from the "locationInfo" field and excluding the NA and NULL values for the DeezerFans column, since in my case I want to represent the top 5 artists per country:

```r
artists$locationInfo <- sapply(strsplit(artists$locationInfo, ","), "[",1 )
artists <- artists[!artists$deezerFans== "NULL", ]
artists <- artists %>% drop_na()
head(artists,100)
nrow(artists)
```

Now the most important part is grouping all the cleaned artists dataset by country and rank in order to get one single artist per country which is in this case the most followed artists.

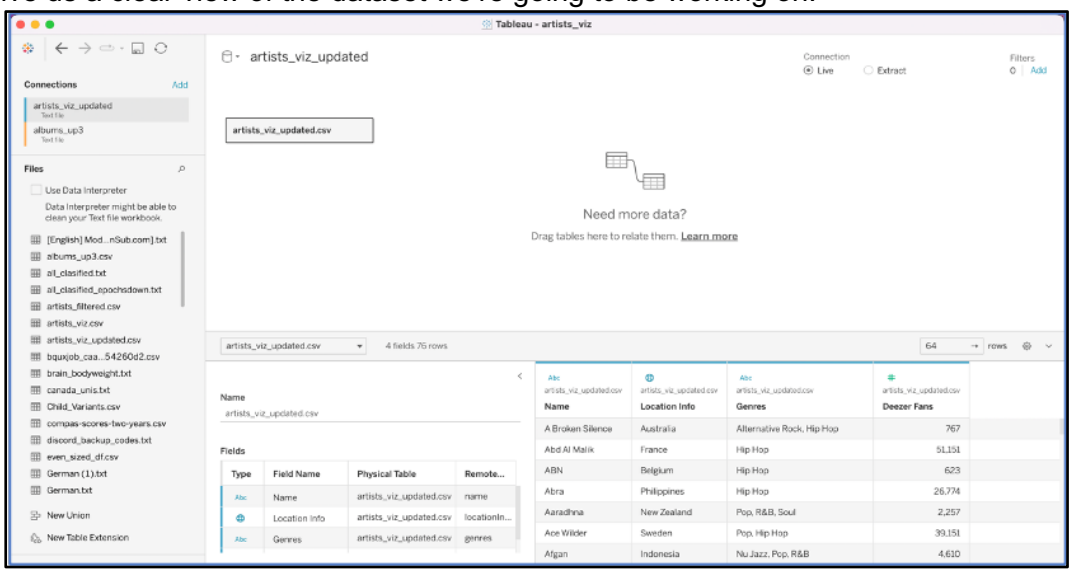
```r
library(dplyr)
artists3 <- artists %>% group_by(locationInfo) %>% top_n(1, deezerFans)
#artists %>% filter(locationInfo == "United States") %>% arrange(deezerFans)
```

Finally exporting the CSV file in order to start working on the visualization on Tableau.

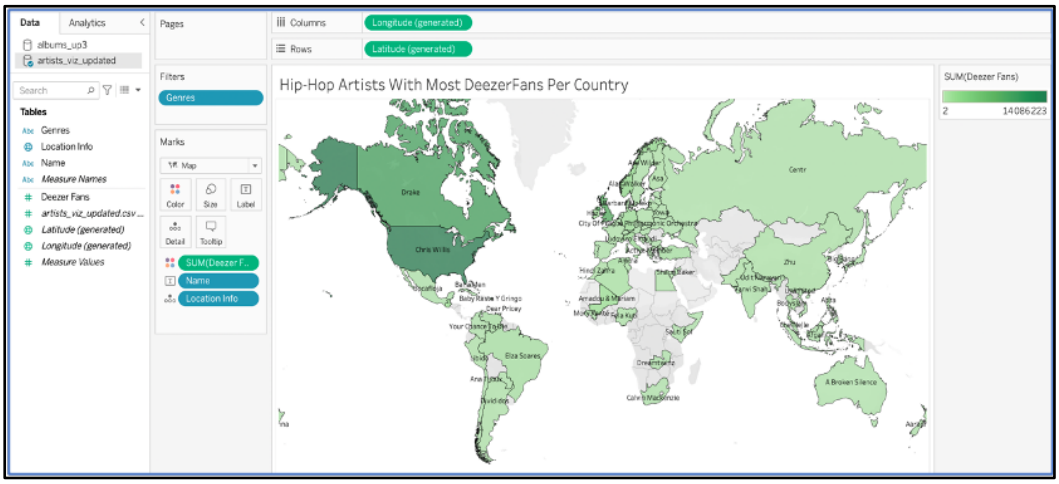
```r
write.csv(artists2,"./Downloads/artists_viz_up.csv", row.names = FALSE)
```

2) Data Visualization on Tableau:

First of all started by loading the dataset on Tableau, which thanks to its easy and remarkable UI give us a clear view of the dataset we're going to be working on:



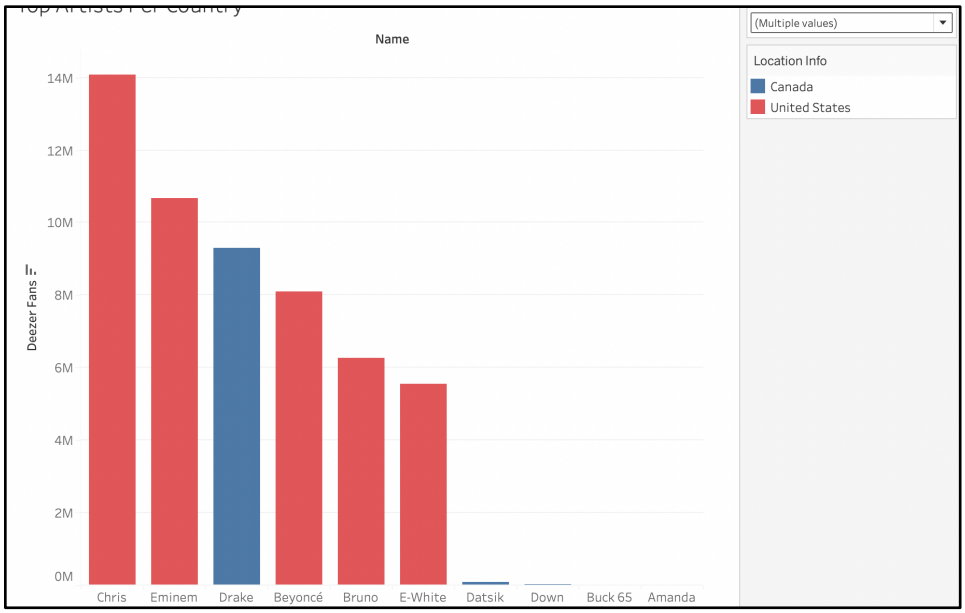
Following that, I created a new sheet where I specified that the "LocationInfo" represents Countries in order to get the possibility from tableau to make a map visualization (as you can see in the picture below). In order to achieve the following visualization, I chose the DeezerFans field as a color variable so that the user can easily distinguish which is the artist with most DeezerFans and where is he/she located. In order to make the visualization easier to understand, I added the artist name on top of his country.



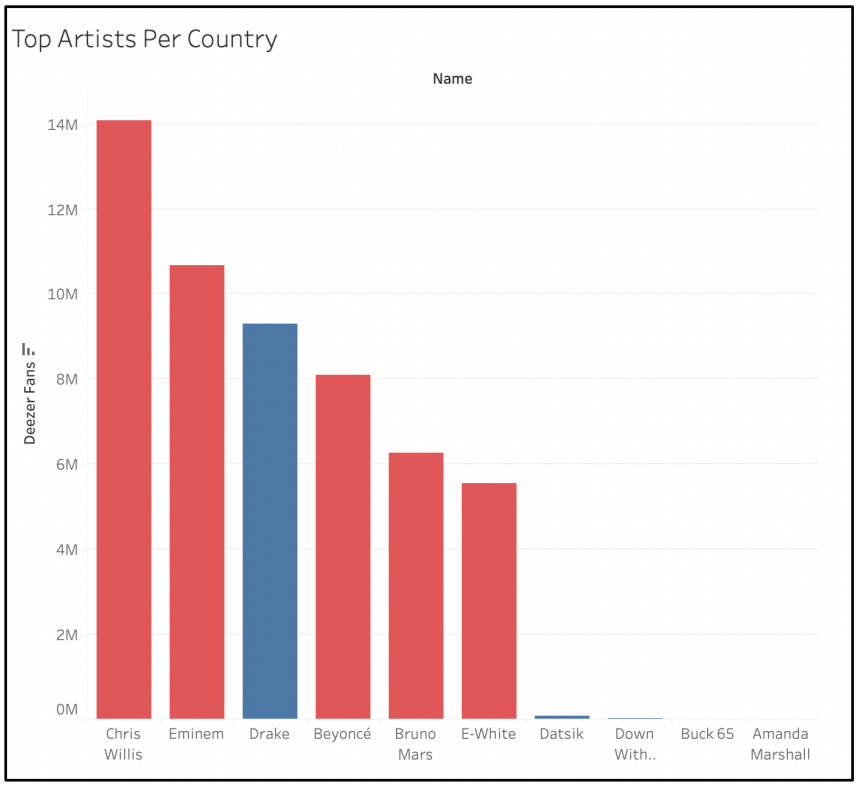
Furthermore, the user can have simple access to the artist's country, name and the number of his DeezerFans simply by hovering over any of the wanted countries thanks to the tooltip available on Tableau.

Making the visualization interactive was the whole point of this project, so in order to give the user a more interactive experience with the first visualization (the map), I thought about adding an action in order that

when any area of the map is clicked it'll take the user directly to another sheet where he'll be able to choose one or multiple countries from a dropdown list and get a detailed view of the top artists within the chosen country:



And in case the user chooses more than one country, he'll have the ability to visualize the difference between the top artists from each of the selected countries, with the ability to hover over any of the artists bar to get a full detail about their country,name and number of DeezerFans :

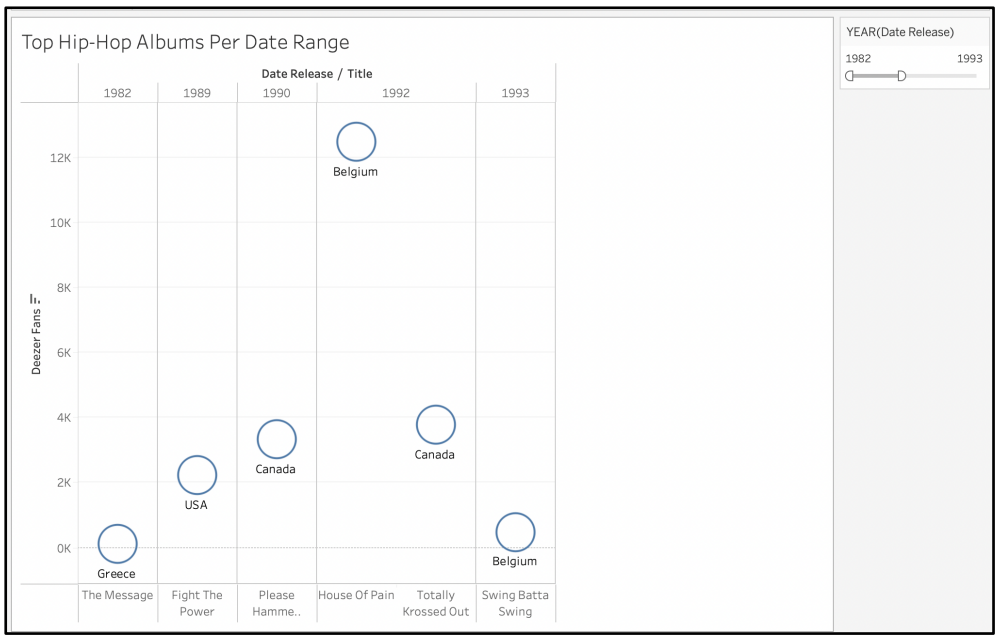


ii) Top Hip-Hop Albums Per Date Range

For this visualization I needed 3 main fields which are the following :

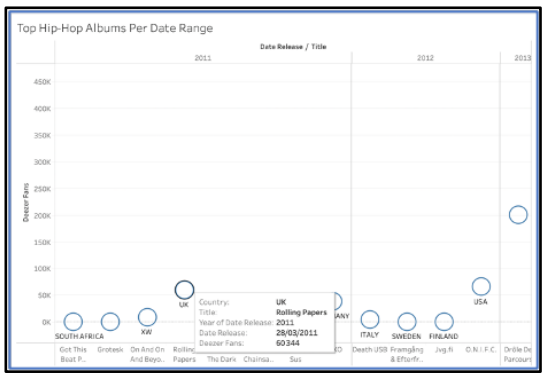
- Albums's Date of Release
- Albums's DeezerFans
- Albums's Name
- Albums's Country

First, I started by loading the cleaned data for the albums (which was the same exact procedure as for the artists) on Tableau, then proceeded to create a new sheet on Tableau, with a different style of visualization this time in order to give a user an interactive interface to be able to visualize the top albums in the Hip-Hop genre in a specific time range that he chooses using the Date range slider on the top right of the visualization:



I provided the user with a simple interactive data range slider in which he'll easily be able to choose a range of dates and the top albums per the selected date range will show up.

The visualization gives a clear view to the user of the name of the albums on the x-axis and the number of DeezerFans for that album on the y-axis while also displaying the original country of the album. And in addition to that, by sampling hover the mouse over the desired circle, the user can get a clear understanding of the name, DeezerFans and country of the chosen album.



c) Aglind Reka : The popularity of Hip Hop in years

1) Processing of data

After we downloaded and processed the data like is explained in the first part it was the part of each of us. At the beginning we had difficulty with the WASABI dataset because we had too much missing data, but we managed to deal with that.
My part is done in R shiny and I choose to show/visualize the popularity of HIP HOP in years. For processing the data first I needed the albums with name, genre, publicationDate, deezerFans, country so I can show in the range of years and countries, information of the names, genres, publicationDate, deezerFans and country of every album.

First I loaded the data, only albums with HipHop genre.

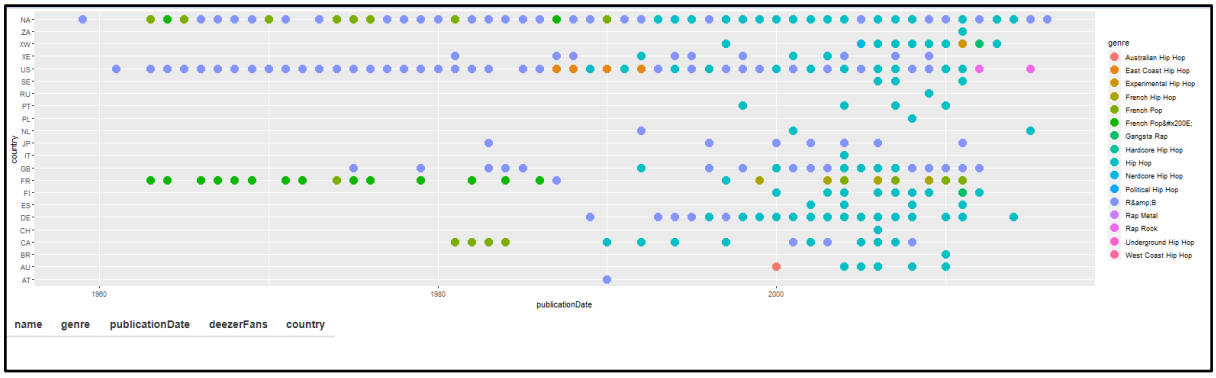
```
Album_HipHop <- read.csv("Album_HipHop_Only.csv")
Album_hh <- Album_HipHop %>% select(name, genre, publicationDate, deezerFans, country)
Album_hh[Album_hh == 'NULL'] <- NA
Album_hh[Album_hh == ''] <- NA
Album_hh <- filter(Album_hh, !is.na(publicationDate))
Album_hh$publicationDate <- strtoi(Album_hh$publicationDate)
Album_hh$deezerFans <- strtoi(Album_hh$deezerFans)
```

Second I chose only the columns I was interested in, name, genre, publicationDate, deezerFans and Country. The null values and empty I changed with not Available Values.
I cleaned only the publicationDate, so I took only the rows with the date.
I converted the publication date and deezerFans in integer for facilitating my work with the visualization after because I had problems in visualization if I didn't change the two columns in integer.
For the visualization part I choose two different visualizations so to be more interactive as possible.

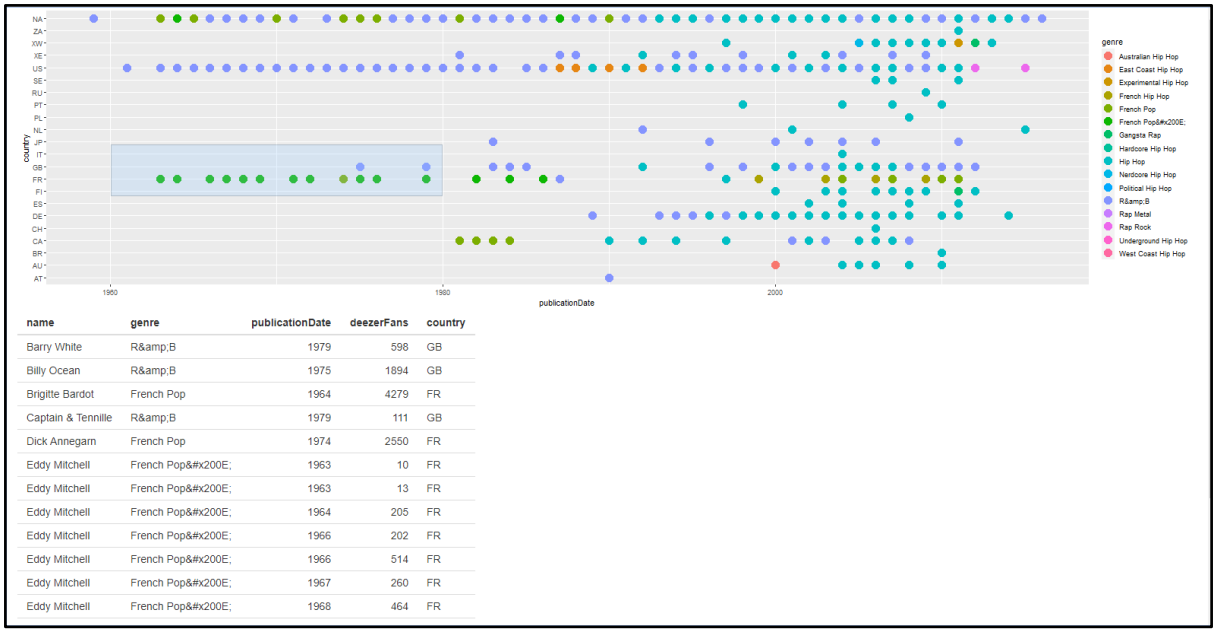
2) Data Visualization on R shiny:

For the visualization part I use three different libraries:
library(dplyr)
library(ggplot2)
library(shiny)

First Visualization:
My idea is to show the popularity of Hip Hop in a range of years.

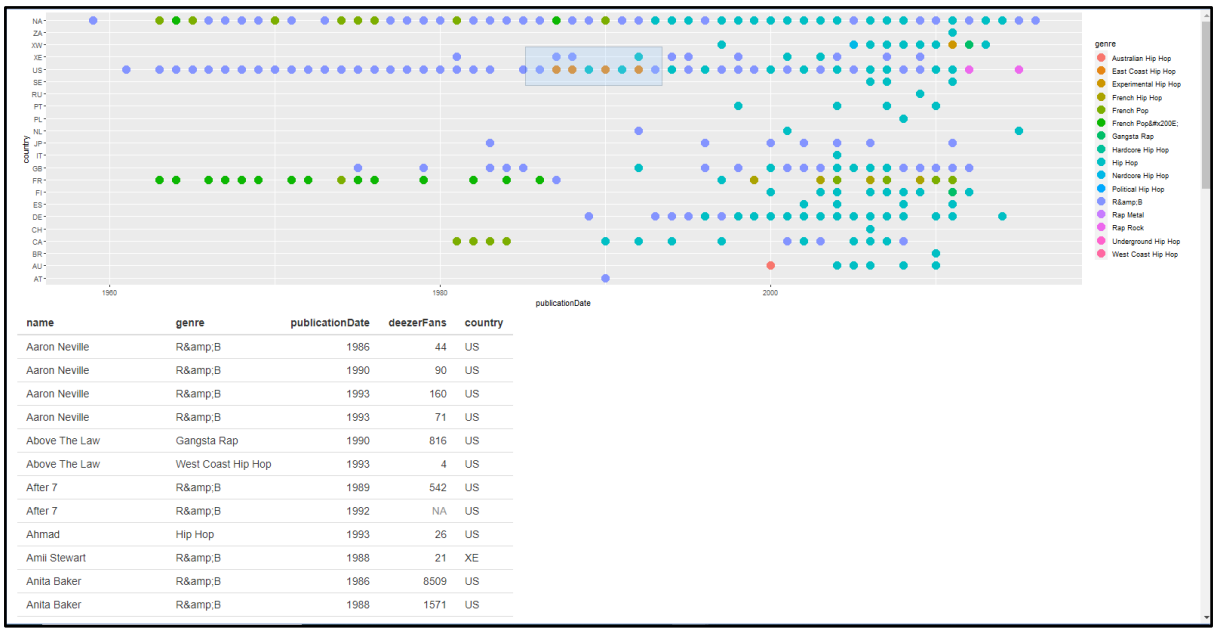


In the Xasis I've the publication date of the album, in the Yasis I've the country, and the points are colored by genre.
I've a representation of albums starting from year 1959 until 2016.



name	genre	publicationDate	deezerFans	country
Barry White	R&B	1979	598	GB
Billy Ocean	R&B	1975	1894	GB
Brigitte Bardot	French Pop	1964	4279	FR
Captain & Tennille	R&B	1979	111	GB
Dick Annegarn	French Pop	1974	2550	FR
Eddy Mitchell	French Pop‎	1963	10	FR
Eddy Mitchell	French Pop‎	1963	13	FR
Eddy Mitchell	French Pop‎	1964	205	FR
Eddy Mitchell	French Pop‎	1966	202	FR
Eddy Mitchell	French Pop‎	1966	514	FR
Eddy Mitchell	French Pop‎	1967	260	FR
Eddy Mitchell	French Pop‎	1968	464	FR

In the example above I've selected starting from year 1960 to 1980 and only the countries Italy, France, Gb, Finland, Jepan.
In the example above we can see all information of this selection, in the first column I've the names, in the second column I've the genre,third column publication date,fourth column fans and in the last column country.

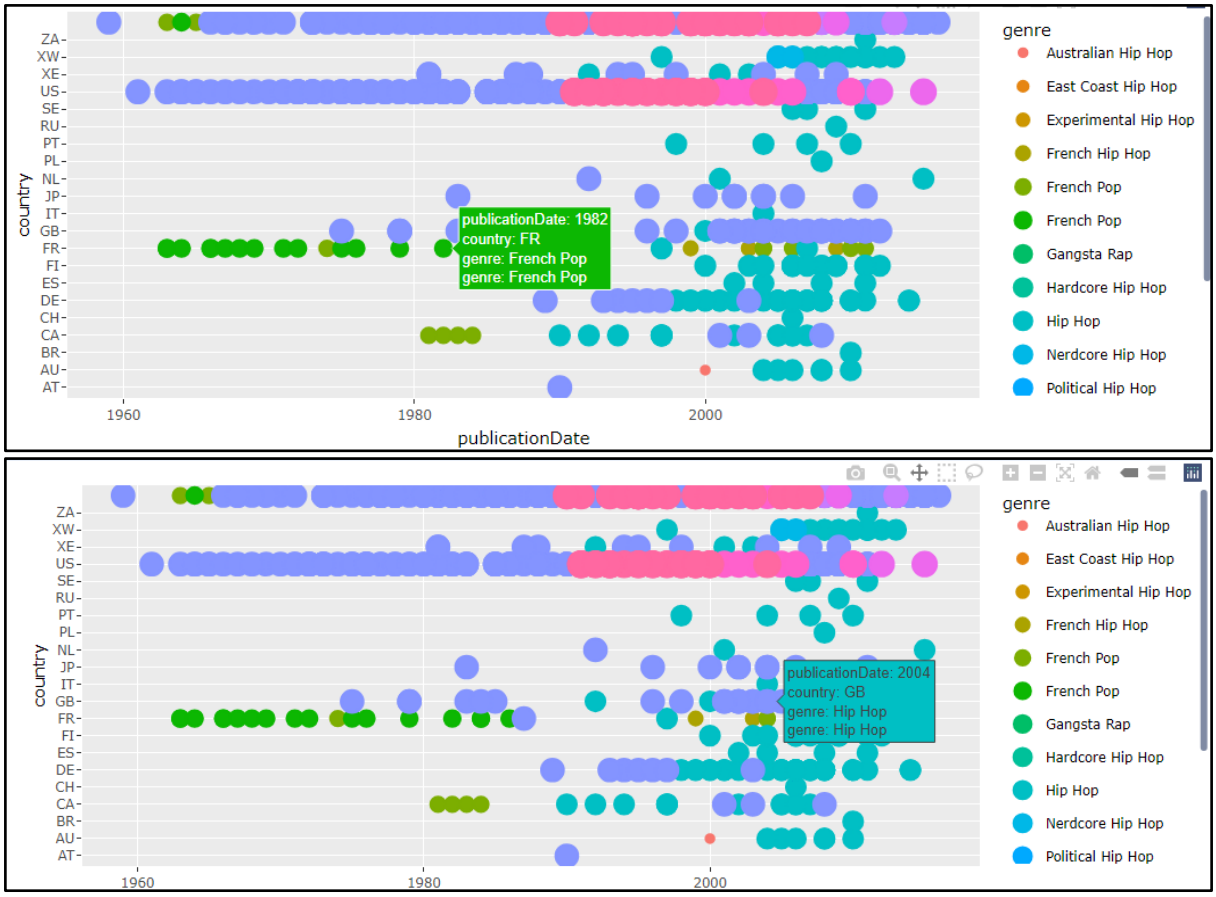


name	genre	publicationDate	deezerFans	country
Aaron Neville	R&B	1986	44	US
Aaron Neville	R&B	1990	90	US
Aaron Neville	R&B	1993	160	US
Aaron Neville	R&B	1993	71	US
Above The Law	Gangsta Rap	1990	816	US
Above The Law	West Coast Hip Hop	1993	4	US
After 7	R&B	1989	542	US
After 7	R&B	1992	NA	US
Ahmad	Hip Hop	1993	26	US
Amil Stewart	R&B	1988	21	XE
Anita Baker	R&B	1986	8509	US
Anita Baker	R&B	1988	1571	US

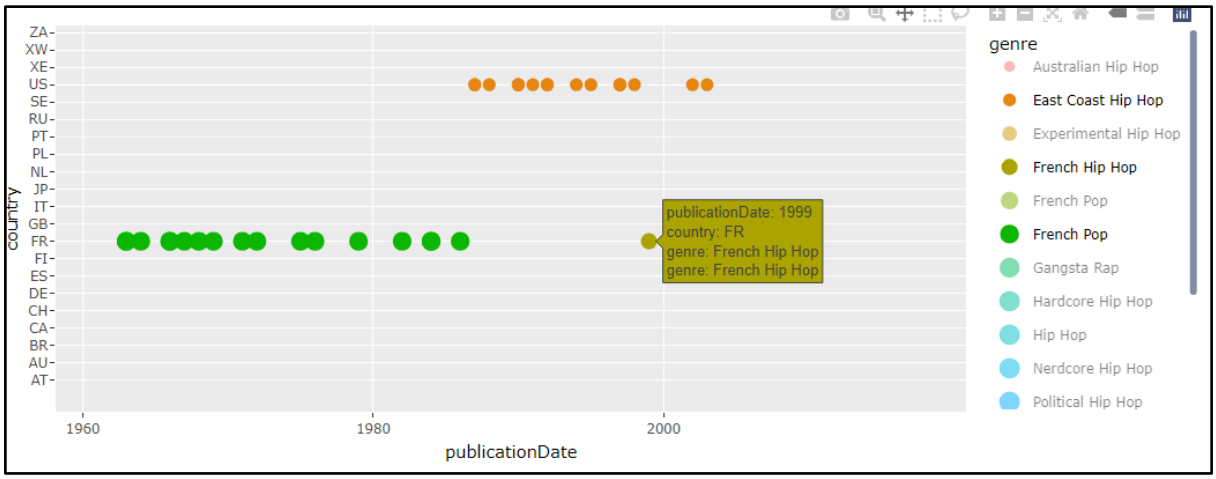
This is the second example where I take a different range of years and countries, and like I see it shows me different data.

Second Visualization:

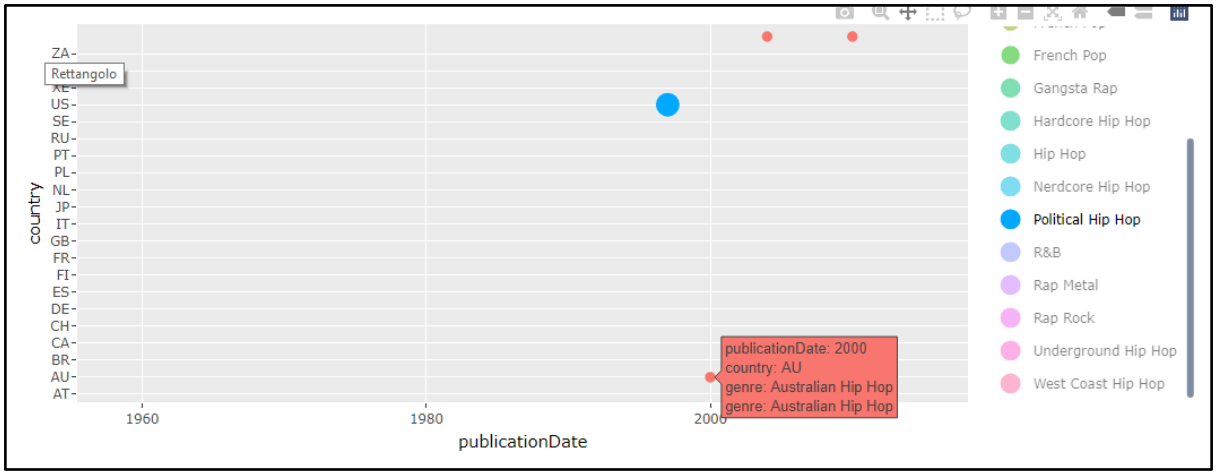
The second visualization is different, in the Xasis and Yasis I've publication date and country, and the colors are represented by the genre.



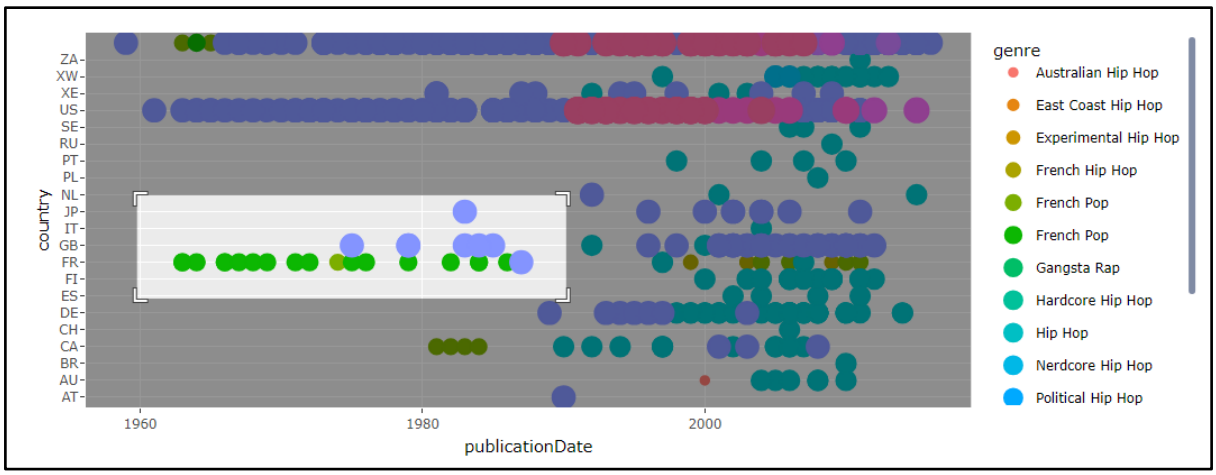
In the two examples above it's shown how I move the mouse in the dots and it shows me the information of every dot.

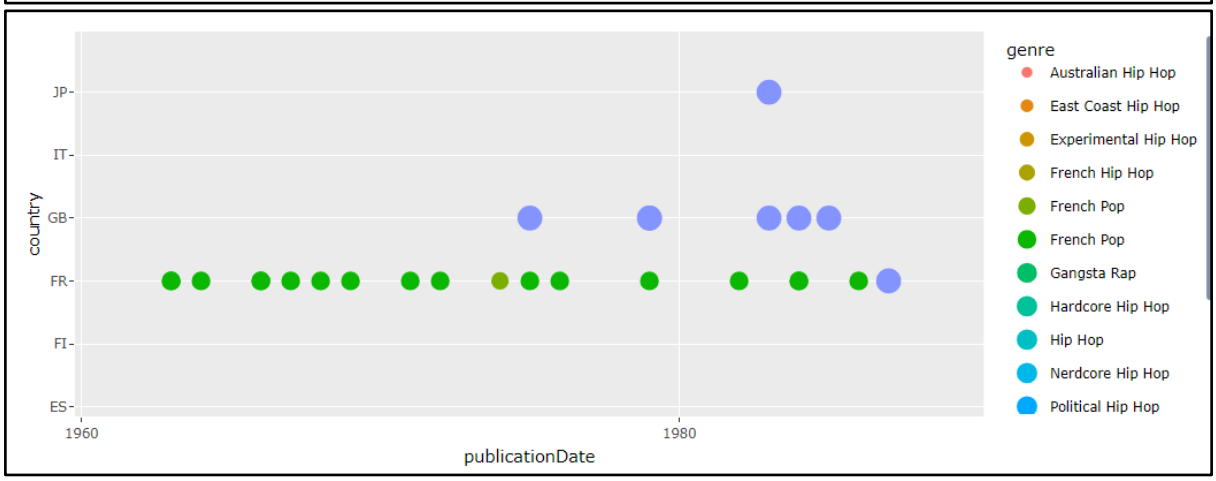


In this example I've chosen only the genres East Coast Hip Hop, French Hip Hop, French Pop, and it shows only the date related with these three genres.



In these examples I show only a range of dots, for every genre.





d) Hedi Derbel :

1) Data processing

After collecting the data to work on our own, some data processing had to be done. At first I wanted to visualize the evolution of Hip Hop popularity by the number of streaming of each album per year. Since the Wasabi Dataset had no such data on it, I chose to visualize the evolution of Hip-Hop through the years via the number of albums released.

I focused on the last 30 years of the data collection in the wasabi dataset. Since 2016 had very little data (I suppose they stopped collecting data), I selected the interval 1985-2015.

From the Albums file, I filtered my data to select realiables features and filter the time period and created a new file.

```
albumsProject=read.csv("albumsProjectModified.csv")
albumsProject= albumsProject %>%
  select(name,genre,title,publicationDate,
         deezerFans,country,dateRelease,language) %>% filter(publicationDate>1984 & publicationDate< 2016 )

write.csv(albumsProject,"albumsProjectModified.csv",row.names =F )
```

After that, since a big part of the dataset had “” or NULL values on the “genre” feature, I had to unify them all under the value “unknown”

```
dim1=dim(albumsProject)

for (i in 1:dim1[1]){
  if (albumsProject$genre[i]=="") {
    albumsProject$genre[i]="unknow"
  }

}
```

2) Album released by genres from 1985 to 2015:

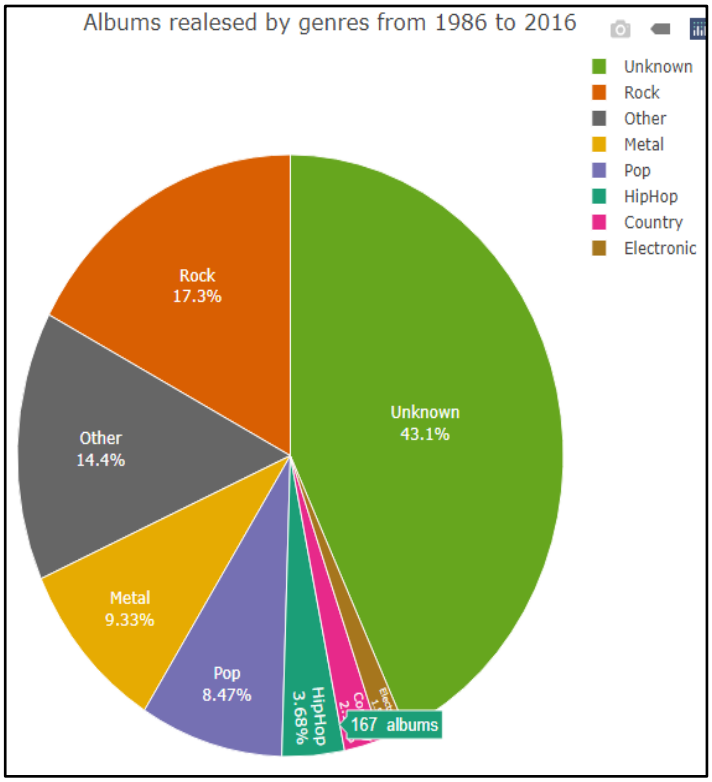
To realize the pie chart, I had to go through all genres in the “genre” column and regroup them all under their main genre. To do that, I created a dataframe for each of the 7 main genres and regrouped the others in “Others'” as they didn’t have a main genre.

```
HipHop=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Hop|Hip|Rap|R&B|Rap",Var1))
HipHop  %>%
  add_row(Var1="HipHop",Freq=sum(HipHop$Freq))

Rock=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Rock",Var1))
Rock  %>%
  add_row(Var1="Rock",Freq=sum(Rock$Freq))

Pop=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Pop|pop",Var1))
Pop  %>%
  add_row(Var1="Pop",Freq=sum(Pop$Freq))
```

```
Country=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Country",Var1))
Country %>%
  add_row(Var1="Country",Freq=sum(Country$Freq))


Unknown=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("unknow",Var1))
Unknown  %>%
  add_row(Var1="Unknown",Freq=sum(Unknown$Freq))


Metal=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Metal",Var1))
Metal  %>%
  add_row(Var1="Metal",Freq=sum(Metal$Freq))
```

```
Electronic=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(grepl("Electronic",Var1))
Electronic %>%
  add_row(Var1="Electronic",Freq=sum(Electronic$Freq))


Other=as.data.frame(table(albumsProject$genre)) %>%
  arrange(Freq) %>%
  filter(!grepl("Electronic|Metal|unknow|Pop|pop|Hop|Hip|Rap|R&B|Rap|Country|Rock",Var1))
Other  %>%
  add_row(Var1="Other",Freq=sum(Other$Freq))



Sums=c(sum(HipHop$Freq),sum(Rock$Freq),sum(Pop$Freq),
       sum(Country$Freq),sum(Unknown$Freq),sum(Metal$Freq),
       sum(Electronic$Freq),sum(Other$Freq))

Idx=c("HipHop","Rock","Pop","Country","Unknown","Metal","Electronic","Other")


#pie chart Albums realesed by genres from 1986 to 2016

Top8Genre=data.frame(Idx,Sums)
```

After that, in the Top Genre DataFrame, I collected the number of albums released in each genre in the wasabi dataset. Then came the pie chart with the percentage of Hip Hop among other genres.

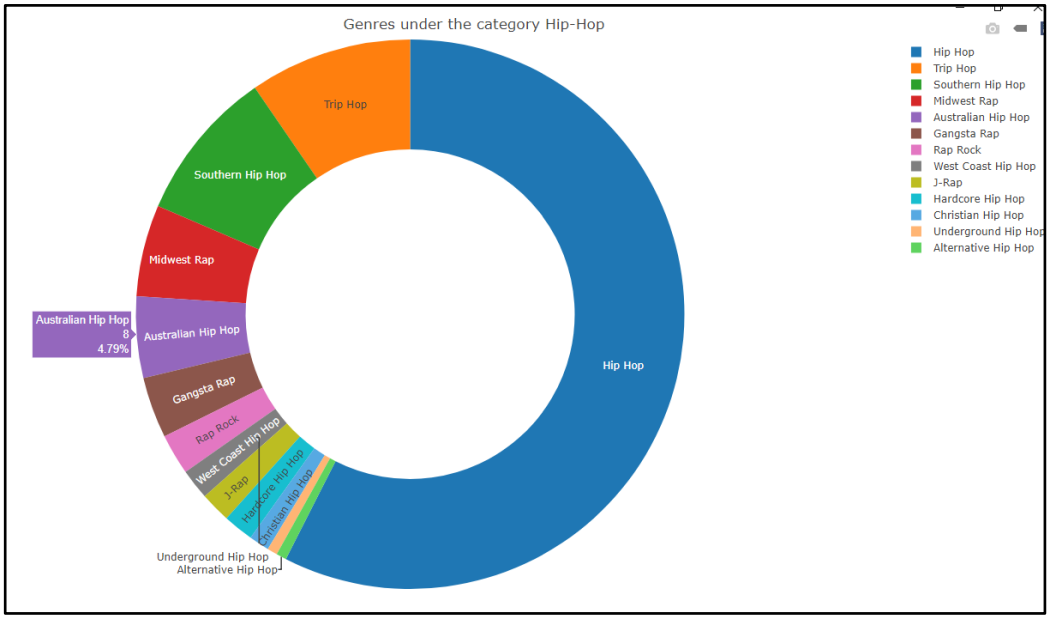


With the library plotly and RColorBrewer, I plotted a pie chart where you can visualize the popularity of Hip Hop among the other genres. We remark that in that period, the Hip-Hop only presented 3.68% of total albums released with 167 albums.

Even if we remove the “unknown” from the pie, Hip-Hop still represents only 3.68% of total albums.

3) Genres under the category Hip-Hop

To understand more what are the different genres under the Hip-Hop category, we illustrated through a donut chart the repartition of the genres in Hip-Hop and their popularity depending on their number of released albums.

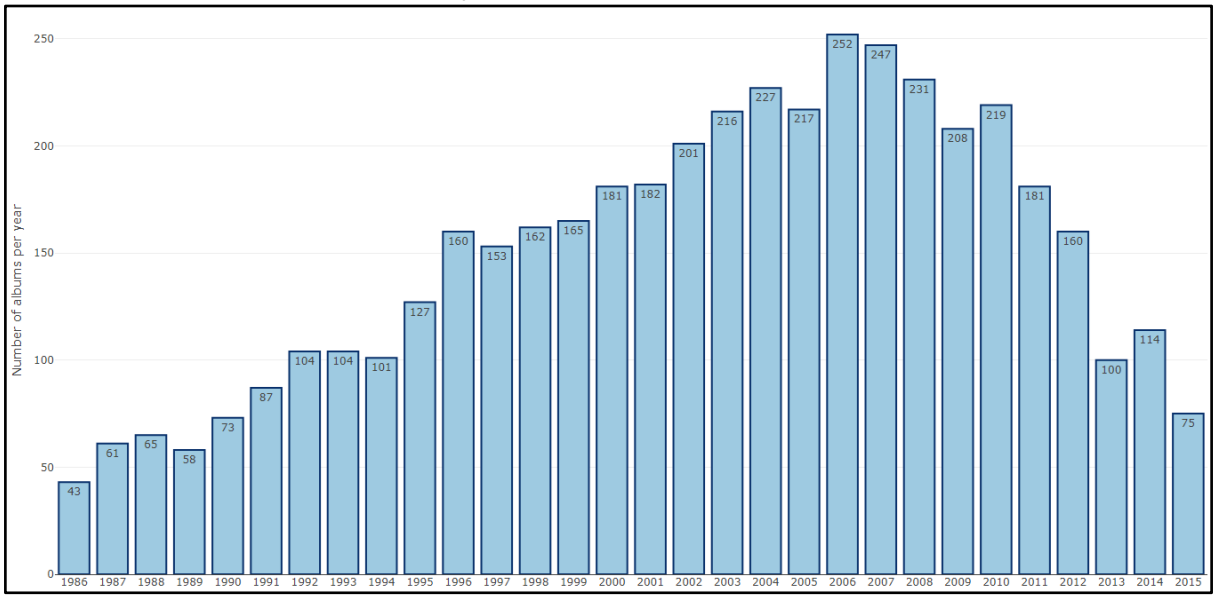


4)Released Hip-Hop albums in the last 30 years

To vizualise the progress through the 1985-2015 period, I collected through the publication date in the albumsProject dataset the publication Date of each album. After that, I created a dataframe with each year and the corresponding number of albums released that year

```
#publication date
publicationDate=table(albumsProject$publicationDate)
publicationDate=as.data.frame(publicationDate)   #year of publication Date of albums


#year of publication Date Graph
x <- publicationDate$Var1
y <- publicationDate$Freq
data <- data.frame(x, y)

fig1 <- plot_ly(data, x = ~x, y = ~y, type = 'bar',
                text = y,textposition = 'auto',
                marker = list(color = 'rgb(158,202,225)',
                              line = list(color = 'rgb(8,48,107)', width = 2)))
fig1 <- fig1 %>% layout(title = "Released Hip Hop albums in the last 30 years",
                        yaxis = list(title = "Number of albums per year"),
                        xaxis = list(title = "Year of release"))
fig1
```

This was the result of the plot. On the x axis, the year of release of album and on the y-axis the number albums released displayed also on the barchart



Conclusion

The project entitled The evolution of Hip Hop in the world was envisioned with the goal of providing the user with an interactive interface to visualize the data retrieved from the WASABI dataset and understand it better.

This project allowed us to gain practical knowledge on several topics such as fetching our data, cleaning the data, brainstorming within our group, working with Shiny on R for the first time and making interactive visualizations using Tableau.

In addition, the project allowed us to understand the development phases of a project development life cycle, as well as the methodology needed to overcome the various problems encountered.

This project gave us great satisfaction in having designed different visualizations that can be implemented in any type of user who's interested in this subject, through simple modifications.

It was a great challenge to learn and create different visualizations using a new technology.