Marynowicz Michael

Rapport : Information Visualisation

14 novembre 2022

VUE D'ENSEMBLE

L'objectif de ce projet est d'utiliser différentes techniques de visualisation afin de mettre en évidence certains pattern de la base de données WASABI. Dans ce projet, nous souhaitons mettre en avant les genres d'albums les plus produits dans chaque pays du monde. Les techniques de visualisation seront détaillées dans une future partie. Cependant, nous pouvons déjà dire que nous représenterons la répartition des genres par une carte du monde coloriée en fonction du genre le plus produit par pays. Nous pourrons aussi nous focaliser sur un pays afin d'obtenir la tendance des genres d'albums au cours des années.

En raison du grand nombre de données que possède la base de données WASABI, j'ai dû faire un choix pour déterminer les données que nous allons traiter. J'ai donc décidé de traiter les données des albums en fonction des genres. J'ai choisi de traiter les albums plutôt que les chansons car le fichier "songs_all_artists_3000.rds" qui contient l'intégralité des musiques , ne possède pas de colonne "genre". En d'autres termes, nous ne pouvons pas connaître le genre d'une musique précise. Cela explique mon choix de traiter les albums se trouvant dans le fichier "albums_all_artists_3000.rds".

Dans ce rapport, nous allons, tout d'abord, expliquer comment j'ai traité les données de la cette base de données pour y extraire les données nécessaires à notre représentation. Ensuite, nous expliquerons les techniques de visualisations. Enfin nous montrerons des scénarios mettant en scène l'utilisation de notre projet.

TRAITEMENT DES DONNÉES

Cette étape à sans doute été la plus longue et la plus fastidieuse du projet. Ceci est dû à plusieurs raisons. D'une part, il y a la découverte du langage. En effet, je n'avais jamais fait de R mais souhaitant apprendre j'ai fait le choix de ne pas changer. La découverte d'un nouveau langage est souvent longue car nous ne connaissons pas bien la documentation. Cela qui implique une longue phase de recherche. Heureusement des cours nous ont été dispensés ce qui a accéléré le processus d'apprentissage de ce langage.

Cette étape m’a permis d’extraire 2 fichiers :

1. locationInfo.csv ➜ Ce fichier contient le code des pays qui est représenté par la colonne “countryCode”, le nom des pays représenté par la colonne “countryName”, le genre de l’album qui a été produit dans ce pays représenté par la colonne “genre” et le nombre d’album qui a été produit dans le même pays et qui a le même genre représenté par la colonne “count”.
2. genreParAnnee.csv ➜ Ce fichier contient le code des pays qui est représenté par la colonne “country”. Le genre de l’album qui a été produit dans ce pays est représenté par la colonne “genre”. La colonne “count” représente la somme des albums qui ont la même date de publication, le même genre ainsi que le même pays.

Nous allons détailler la collecte des données de chaque data dans les parties ci-dessous :

Data “locationInfo”:

Lissage des données :

Pour commencer nous lisons le fichier “albums_all_artists_3000.rds”, grâce à la fonction readCSV. Nous stockons le résultat dans la data “albums”. De ce fait, la data “albums” contient toutes les informations possibles concernant les albums. Nous créons un nouvelle data nommée "locationInfo". Elle contiendra toutes les infos nécessaires afin de déterminer le nom des pays, les genres de chaque albums ainsi que le nombre d’albums du même genre dans un même pays. Cette data permettra de créer le fichier “locationInfo.csv”.

À sa création “locationInfo” contient la colonne “country” (contenant le code ISO2 des pays) et “genre” (contenant le genre de chaque albums). Ces colonnes sont issues de la data “albums”. Ceci nous permet de faire un premier tri concernant la data “albums”.

Désormais, nous avons tous les pays ainsi que les genres des albums produits. Cependant, en étudiant le résultat obtenu, beaucoup de lignes comportent des codes de pays ainsi que des genres égaux à NaN. De plus, certains genres musicaux sont les mêmes mais ils sont écrits différemment ex : “R&B” et “R&B” ou encore “Rock” et “rock”.

Pour pallier ce problème nous rentrons dans une première phase de lissage des données.

Premièrement, nous avons uniformisé les genres d'albums grâce à la fonction "filter" :

```
# fonction permettant d'unifier les genres
filterGenre <- function(x){
  x <- mutate(x,genre = str_replace(genre, ".*[rR]ock.*", "Rock")) %>%
    mutate(x,genre = str_replace(genre, ".*Rock.*", "Mika")) %>%
    mutate(x,genre = str_replace(genre, ".*[mM]etal.*", "Metal")) %>%
    mutate(x,genre = str_replace(genre, ".*[cC]ountry.*", "Country")) %>%
    mutate(x,genre = str_replace(genre, ".*Hip Hop.*", "Hip Hop")) %>%
    mutate(x,genre = str_replace(genre, ".*[pP]op.*", "Pop")) %>%
    mutate(x,genre = str_replace(genre, ".*Wave.*", "Wave")) %>%
    mutate(x,genre = str_replace(genre, ".*Electro.*", "Electro")) %>%
    mutate(x,genre = str_replace(genre, ".*[pP]unk.*", "Punk")) %>%
    mutate(x,genre = str_replace(genre, ".*[fF][oi]lk.*", "Folk")) %>%
    mutate(x,genre = str_replace(genre, ".*[jJ]azz.*", "Jazz")) %>%
    mutate(x,genre = str_replace(genre, ".*Visual Kei.*", "Visual Kei")) %>%
    mutate(x,genre = str_replace(genre, ".*Neue Deutsche.*", "Neue Deutsche")) %>%
    mutate(x,genre = str_replace(genre, ".*R&amp;B.*", "RandB"))

}
```

Cette fonction prend en paramètre une data, par exemple ici "locationInfo". La fonction va parcourir la colonne "genre" afin de remplacer l'expression contenue dans le deuxième paramètre de str_replace par le 3ème paramètre. Ceci permet d'unifier les noms des genres. Par exemple, avant d'exécuter cette fonction nous pouvons trouver un genre nommé "rock" et un autre nommé "Rock". Une fois la fonction exécutée, le genre sera unifié en "Rock".

Or certains noms contenaient des Hexa Caractère ex : "Children'''s Music".
Afin de traiter ces cas particuliers, j'ai dû parcourir la liste pour tous les trouver. Une fois identifiés j'ai pu une fois de plus unifier les genres de cette manière :

```
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Neue-Deutsche-H&#xE4;rte", "Neue Deutsche"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Neue Deutsche H&#xE4;rte", "Neue Deutsche"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, ".*Children'&amp;apos;'s Music.*", "Children Music"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Children&amp;apos;s Music", "Children Music"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, ".*Gothic Rock&#x200F;&#x200E;.*", "Gothic Rock"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, ".*Gothic Rock&#x200F;&#x200E;.*", "Gothic Rock"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Forr&#xF3;", "Forro"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Visual Kei&#x200E;", "Visual Kei"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Dark Wave&#x200F;&#x200E;", "Dark Wave"))
locationInfo <- mutate(locationInfo,genre = str_replace(genre, "Rock &amp;apos;N&amp;apos; Roll", "Rock And Roll"))
```

J’ai renommé la colonne “country” en “countryCode” car cela est plus compréhensible car cette colonne contient les codes des pays. De plus, nous voulons introduire le noms des pays donc afin d’être clair j’ai fait le choix de changer le nom.

Concernant la modification des éléments égaux à NaN ou NULL. Je procède de la sorte :

```
# suppression des lignes contenant des données nulles

locationInfo <- locationInfo %>% replace(.=="NULL", "Inconnu")
locationInfo <- locationInfo %>% replace(.=="", "Inconnu")
locationInfo$countryCode <- locationInfo$countryCode %>% replace_na('Inconnu')
```

Toutes les données de “locationInfo” ont été lissées. Nous pouvons donc commencer à incrémenter “locationInfo” afin de connaître le nom des pays.

Ajouts de données a la data “locationInfo”:

Tout d'abord, j’ai fait des recherches pour avoir une librairie qui convertit directement le code ISO2 en nom des pays. J’ai trouvé la library **countrycode** qui le fait très bien. Cependant, un certain nombre de pays n'étaient pas trouvés. J’ai donc chercher un fichier qui contient la totalité des codes contenus dans “locationInfo” afin de d’y associer le nom des pays. J’ai trouvé ce fichier sur ce lien, il contient l'intégralité des codes. La photo ci-dessous montre comment je traite ce fichier :

```
allLocation <- read.csv("~/Desktop/Visu/Project/Wasabi-dataset/country_lat_long.csv")
allLocation <- allLocation %>% rename(countryCode = ISO.3166.Country.Code,countryName = Country)
allLocation <- allLocation %>% mutate(countryCode = str_replace(countryCode, ".*PS.*", "XW"))

# merge de allLocation et de locationInfo pour avoir les données geographiques des pays
# les deux list ne contenaient pas les meme nombres de ligne donc cbind et left_join impossible
locationInfo <- merge(x = locationInfo, y = allLocation, by="countryCode", all.x = T)
```

Je stocke les données du fichier dans la data “allLocation”. J’ai renommé les colonnes d’une part pour améliorer la compréhension et d'autre part pour pouvoir faire le merge avec la data “allLocation” sur la colonne countryCode. Je peux désormais merge “locationInfo” avec “allLocation” afin d'associer le nom des pays à leurs codes.

Cette nouvelle colonne contient des noms de pays égaux à NaN car certains codes de pays sont égaux à Inconnu. De ce fait, de la même manière que nous avons procédé plus haut, nous allons remplacer les NaN par Inconnu.
Dû au merge, plusieurs colonnes se sont ajoutées. Je vais donc sélectionner seulement celles qui nous seront utiles, soit : countryCode, countryName, genre.

```
locationInfo$countryName <- locationInfo$countryName %>% replace_na('Inconnu')
locationInfo <- locationInfo %>% select(countryCode,countryName,genre)
```

De plus, nous voulons déterminer le genre majoritaire dans un pays. Cela implique que nous devons calculer la proportion de chaque genre dans chaque pays. C'est pour cela que nous faisons le group_by comme dans la photo ci-dessous. :

```
# on regroupe les lignes similaires et on crée une nouvelle colonne contenant la somme
locationInfo <- locationInfo %>% group_by(countryCode,countryName,genre) %>% summarise(count = n())
```

Cela nous permet de créer une nouvelle colonne nommée "count" contenant cette proportion de genre dans chaque pays.

Désormais nous avons toutes les données que nous souhaitons concernant cette première data "locationInfo". Nous écrivons donc un fichier csv les données de celle-ci..

```
write.csv(locationInfo,"/Users/maryno/Desktop/Visu/Project/Wasabi-dataset/locationInfo.csv", row.names = FALSE)
```

Data "genreParAnnee":

Nous commençons par créer une data "genreParAnnee". Ce sont les données de cette data qui seront stockées dans le fichier "genreParAnnee.csv". À sa création, je lui ai ajouté les colonnes "countryCode" et "genre" issues de la data "locationInfo". Ensuite, nous lui ajoutons grâce à "mutate" la colonne "publicationDate" provenant de la data "albums" qui contient toutes les infos des albums.

Concernant le lissage des données comme pour le traitement des valeurs null ou égales à NaN, je procède de la même manière que pour "locationInfo".

Enfin, je regroupe les lignes similaires et j'ajoute le résultat dans une nouvelle colonne nommée "count". Cette dernière me permettra de représenter la proportion des albums sortis dans un pays avec un genre et une date de publication similaires.

```
# on regroupe les albums qui ont le meme genre, qui sont sorti la même année et dans le meme pays
genreParAnnee <- genreParAnnee %>% group_by(country, genre, publicationDate) %>% summarise(count = n())
```

Désormais, la data "genreParAnnee" contient toutes les infos nécessaires à la réalisation de mon graphique. En effet, grâce aux données récoltées nous pouvons faire un graphique qui représente sur l'axe des x l'année de publication des albums et sur l'axe des y le nombre d'albums de ce genre dans le pays sélectionné.

Je peux donc écrire ces données dans un fichier csv nommé "genreParAnnee.csv" :

```
write.csv(genreParAnnee,"/Users/maryno/Desktop/Visu/Project/Wasabi-dataset/genreParAnnee.csv", row.names = FALSE)
```

TECHNIQUES DE VISUALISATIONS

Dans cette partie je vais tout d'abord vous parler du traitement en D3js des fichiers précédemment créés. Ensuite, je vous expliquerai les techniques de visualisations.

Traitement des fichiers csv :

Fichier genreParAnnee.csv :

Le traitement du fichier "genreParAnnee.csv" est fait dans la classe "DrawLineChart". L'objet "data" contient l'ensemble des données du fichier "genreParAnnee.csv".

L'objet "allGenre" est un set. Il a été créé afin d'avoir l'ensemble des genres sans répétition.

Lors de la création d'une instance de cette classe, nous créons 3 objets :

- "genreToPrint" ➔ Contient l'ensemble des genres qui sont affichés dans le diagramme
- "genreCanPrint" ➔ Contient l'ensemble des genres que l'utilisateur peut afficher grâce aux checkbox.
- "colors" ➔ Contient l'ensemble des couleurs des genres. En effet, à chaque genre contenu dans "genreCanPrint" est associé, au même index, une valeur dans "colors". C'est grâce à cela que nous pouvons colorier les points ainsi que les lignes du diagramme.

L'objet "minMax" permet de récupérer la plus ancienne "publicationDate" contenu dans "data" ainsi que la plus récente afin de déterminer le range de l'axe des x.

Enfin c'est grâce à la fonction "drawLineChart" que je dessine mon diagramme.

Fichier locationInfo.csv :

Le traitement du fichier "locationInfo.csv" est fait dans le fichier "map.js" L'objet "data" contient l'ensemble des données du fichier "locationInfo.csv".

Afin de tracer la carte je lis le fichier "world-countries-no-antartica.json" dont les données sont stockées dans l'objet "geojson". Le fichier et le code pour dessiner la carte ont été trouvés sur le site suivant : https://www.datavis.fr/d3js/map-improve

L'objet "colors" me permet de colorier de la même couleur les pays possédant le même genre d'albums majoritaires. Nous ajoutons une couleur dans l'objet "colors" à chaque fois que nous trouvons le genre majoritaire dans un pays. Les couleurs sont générées aléatoirement.

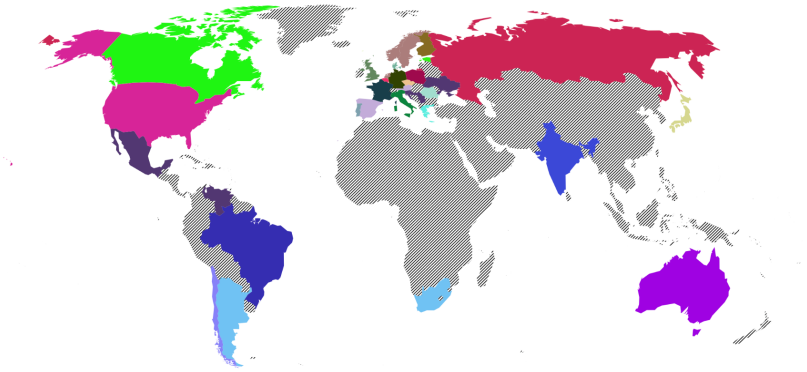
Les genres inconnus sont stockés dans l'objet "genreInconnu" afin de pouvoir les traiter séparément des autres genres. En effet, je vais traiter seulement les genres qui ne sont pas "Inconnu" afin de déterminer le genre majoritaire dans chaque pays. Ce traitement est fait dans la fonction "getAllMaxWithoutInconnu". De plus, dans cette fonction si je trouve un pays contenant un genre d'album "Inconnu" je l'ajoute dans "genreInconnu" . Ceci me permet d'afficher pour chaque pays sa proportion du genre d'albums majoritaire ainsi que le nombre d'albums de genre Inconnu.

L'objet "genreOfCountryInconnu" me permet d'afficher la liste des genres pour les pays ayant un countryCode égal à "Inconnu". Ces pays sont regroupés dans un cercle en haut à gauche de la carte.

Techniques de visualisations :

1. Overview :

   Dans la première partie la technique de visualisation utilisé est "Overview" en effet comme vous pouvez le voir ci dessous :



Nous pouvons voir que chaque pays est colorié dans une couleur. Cette couleur correspond au genre d'albums majoritaire dans ce pays. Les pays grisés et rayés sont les pays pour lesquels nous n'avons pas de données.

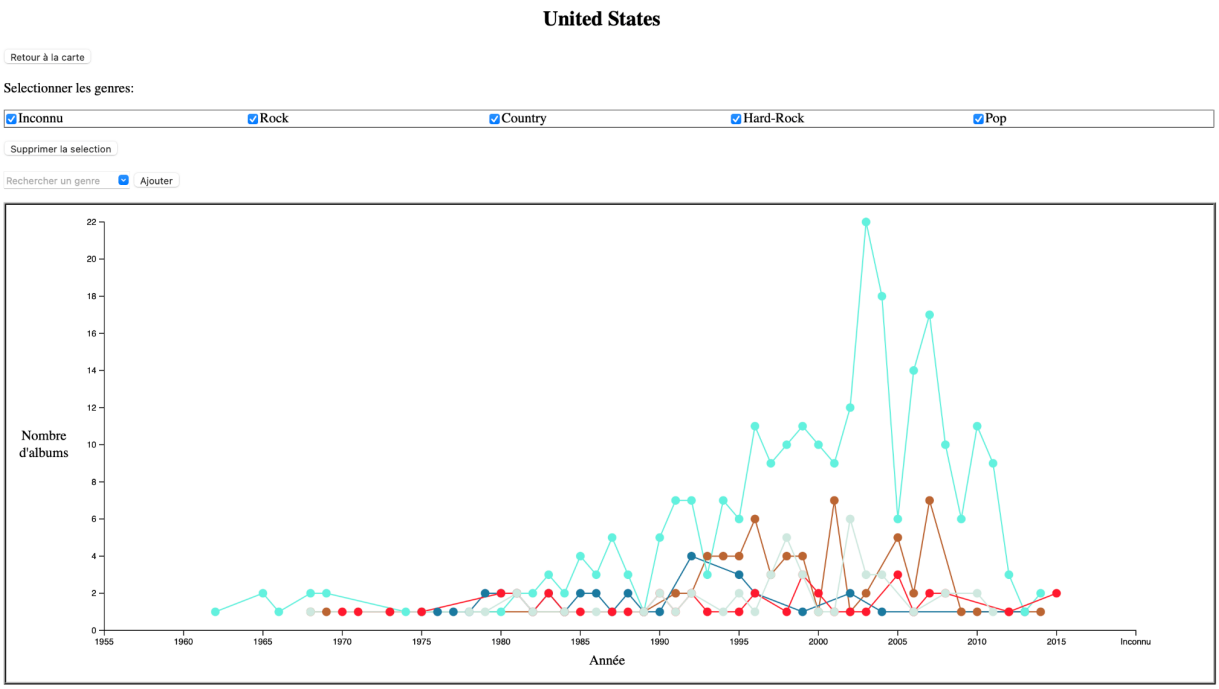
Cette technique de représentation nous permet de visualiser des pattern concernant les genres de musiques dans le monde. De cette manière, nous pouvons facilement avoir accès au genre majoritaire du pays que nous souhaitons. De plus, une fois ce genre identifié, nous pouvons repérer les pays colorés de la même couleur afin d'identifier les pays dont le genre majoritaire est identique.

De plus, cette visualisation de la carte est intuitive pour l'utilisateur, il sait facilement se repérer.

Lorsque vous voulez des détails sur les proportions des genres d'albums dans un pays, il vous suffit de cliquer sur le pays en question. Ceci vous donnera la deuxième technique de visualisation détaillée ci dessous :

2. Details and focus

   Dans cette seconde partie, nous nous focalisons sur les genres d'album dans un pays donné comme vous pouvez le voir dans l'exemple ci-dessous :



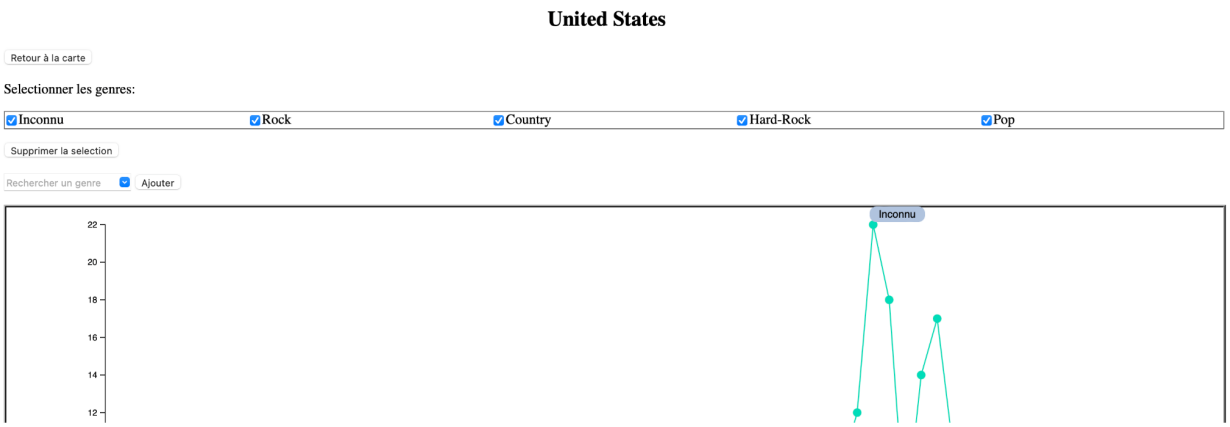
Nous représentons la variation des genres en fonction des années. Pour cela nous avons sur l'axe des x les années de publication des albums. Et sur l'axe des y le nombre d'albums de ce genre dans ce pays à une même date.

Initialement, il est affiché la tendance au cours du temps des 5 genres majoritaires dans le pays sélectionné. Grâce à cette représentation nous pouvons voir de manière plus précise si le genre majoritaire est l'est largement où s' il existe un genre proche de lui. Nous pouvons de même identifier les genres en pleine expansion.

Nous pouvons également filtrer les genres à afficher. Ceci est utile si par exemple nous voulons comparer 2 genres, le fait d'avoir seulement ces 2 genres affichés rend plus lisible la comparaison.

De même, nous pouvons ajouter de nouveaux genres à afficher. En effet, vous pouvez soit taper le genre que vous voulez ajouter dans l'input "Rechercher un genre" puis appuyer sur "Ajouter". Soit appuyer sur  À droite de l'input ce qui listera tous les genres possibles, vous pouvez ensuite le sélectionner puis cliquer sur "Ajouter" pour l'ajouter.

Afin de limiter la charge cognitive lors du survol de la souris sur un point du graphe, le genre de l'album représenté par le point est affiché comme vous pouvez le voir ci dessous :



SCÉNARIOS

Concernant l'usage de mon projet, deux scénarios principaux sont possibles :

1. Utilisation par un professionnel de musique :

    - Mon projet peut être utile pour un professionnel de musique pour plusieurs raisons. Par exemple, lors de la sortie d'un album, un professionnel peut consulter ma carte s' il souhaite étendre son public dans de nouveaux pays. En effet, il pourra voir si un pays possède comme genre majoritaire le même genre que celui de son album. Il pourra en conséquence décider de lancer une campagne de publicité dans ce pays.

    - De même, un professionnel de musique peut vouloir connaître la variation du genre de son prochain album dans le pays où il se trouve. Il pourra alors cliquer sur le pays afin d'afficher le diagramme des genres et ainsi connaître la variation du genre de son choix.

    - Enfin, un producteur peut consulter la carte quand il cherche à produire un nouveau talent. Il pourra comparer les tendances des albums dans le pays de son choix. Il pourra ainsi chercher à produire des artistes qui sont dans le genre le plus tendance.

2. Utilisation par un consommateur de musique :

    - En tant que consommateur de musique je peux être amené à regarder par curiosité la tendance de mon genre préféré dans le pays de mon choix.

    - De plus, je peux choisir le pays ou je me trouve actuellement afin de voir les tendances. Ce qui peut me faire découvrir un nouveau genre de musique.