

Information Visualisation

Groupe BERNIGAUD-BOURDEAU-CORREIA-JEANNES :

Benjamin Bernigaud, Quentin Bourdeau

Ambre Correia, Theo Jeannes

Introduction

Pour ce projet de visualisation, nous allons nous concentrer sur le caractère explicite ou non des chansons et l'évolution de la représentation de ces chansons dans l'industrie musicale. Tout au long du projet, nous étudierons donc la place des chansons à paroles explicites dans le monde de la musique.

Dans un premier temps, nous présenterons l'évolution de la proportion de chansons explicites produites à travers les années et les genres. Pour cela, nous utiliserons un graphique spirale ce qui nous permettra de mettre en évidence une éventuelle tendance dans l'évolution du nombre de chansons avec des paroles explicites au fil du temps.

Nous chercherons également à identifier la distribution des chansons explicites à travers différentes catégories. Nous représenterons donc sur des ensembles parallèles les chansons regroupées par genre, par albums, par artistes, genre de l'artiste ou du groupe, ainsi que par caractère explicite. Nous utiliserons le nombre de chansons de chaque catégorie pour définir l'importance des catégories dans le diagramme.

Ensuite, nous afficherons les chansons dans une pyramide de population pour déterminer si le caractère explicite d'une chanson influence sa popularité à travers les années. Pour cela, nous représenterons l'année de publication en ordonnée et la popularité moyenne d'une chanson explicite ou non en abscisse. Un zoom sémantique permettra d'afficher la répartition des genres ou des artistes dans les années affichées. Cela nous permettra de comparer la popularité des chansons explicites avec les chansons tout publics, et de détecter un éventuel changement dans l'appréciation des chansons explicites à travers le temps.

Enfin, notre dernière représentation prendra la forme d'une carte à bulles, ce qui nous permettra d'identifier les pays produisant le plus de chansons explicites. Cela nous permettra également de constater d'éventuelles disparités entre les zones géographiques, vis-a-vis de leur proportion de chansons explicites produites.

Le code source du traitement de données et des visualisations se situe dans [l'annexe 1](#). Cette annexe inclut également un lien vers tous les fichiers .csv utilisés pendant le projet.

Type de données visualisées

Pour nos représentations, nous allons devoir créer deux structures à partir du jeu de données:

- Pour pouvoir faire une carte à bulles, nous allons avoir besoin d'associer des positions à chaque point de données. Cela nous amènera à créer un champ de données.
- Pour représenter le diagramme en spirale, la pyramide de population et les ensembles parallèles, il nous suffira d'utiliser les données sous forme de table.

Utilisateurs visés

Dans un premier temps, les labels, producteurs et artistes pourraient être intéressés par l'impact d'un contenu explicite sur sa popularité. Ils pourraient également chercher à savoir si une chanson explicite a autant de chances de devenir célèbre qu'une chanson sans paroles explicites.

Les législateurs et les autorités de régulation pourraient également être intéressés à comprendre dans quelle mesure les avertissements concernant les chansons explicites sont utiles. Ils pourraient également comparer l'évolution de la popularité des chansons explicites avec les réformes et lois sur la régulation de ces contenus pour évaluer l'efficacité de telles mesures.

Récupération des données depuis WASABI

La première étape de ce projet a consisté à récupérer les données. Les données fournies ne contenaient pas le champ *explicitLyrics* pour les chansons, ce qui est crucial à nos visualisations. Pour cette raison, nous avons dû récupérer les données nous-même. Nous avons donc récupéré les données à travers l'API fournie. Pour cela, nous avons écrit un script Python, *importDatas.py*, permettant de faire des requêtes pour récupérer toutes les données nécessaires en adaptant dynamiquement les paramètres fournis à la requête en fonction du résultat de la requête précédente.

La principale difficulté concernant cette récupération de données a été la limite de requête. L'API limitant à une trentaine de requête par seconde, et ne retournant les chansons que 200 par 200, nous avons implémenté dans un premier temps un délai entre chaque période de requête pour éviter les requêtes inutiles pendant le timeout.

Nous avons ensuite implémenté un système pour permettre de définir quelles parties du dataset télécharger, en définissant le numéro de la première et de la dernière chanson souhaitée, ainsi qu'un système de sauvegarde automatique, qui tire avantage des temps d'attente à cause du timeout pour enregistrer les données. Toutes ces mesures nous ont permis de répartir le téléchargement entre nos machines pour ensuite récupérer les données, en pré filtrant déjà les colonnes que nous savions inutiles à nos visualisations, comme les url par exemple.

Traitement des données

Pour commencer le traitement des données, nous avons étudié la base de données WASABI, et nous avons listé tous les champs dont nous aurions besoin pour nos représentations :

CHANSONS	ALBUMS	ARTISTES
<ul style="list-style-type: none"> • <i>releaseDate</i> • <i>title</i> • <i>publicationDate</i> • <i>genre</i> • <i>rank</i> • <i>explicitLyrics</i> 	<ul style="list-style-type: none"> • <i>X_id..oid</i> • <i>id_artist</i> • <i>country</i> • <i>language</i> • <i>deezerFans</i> 	<ul style="list-style-type: none"> • <i>X_id..oid</i> • <i>gender</i> (male/female/other/""') • <i>type</i> • <i>members</i> • <i>name</i> • <i>location.country</i>

Nous avons ensuite effectué tous les traitements nécessaires pour rendre les données exploitables en R. Nous nous sommes d'abord rendus compte que la base de données contenait énormément de champs vides, et de nombreuses incohérences au niveau du format des données. Notre première action a été de filtrer les fichiers CSV, pour ne garder que les colonnes citées ci-dessus, afin d'éviter de conserver des informations inutiles qui alourdissent les traitements.

Nous avons commencé par supprimer les rares chansons sans titre, en considérant qu'il s'agissait d'erreurs de saisie. Ensuite, nous avons ajouté un champ avec une année associée à chaque chanson en fonction des dates de publications et de sortie de chaque chanson.

Pour obtenir une année exploitable, nous avons commencé par standardiser le format des dates dans les colonnes *releaseDate* et *publicationDate*. Nous avons supprimés les caractères ", [,], et ' qui n'étaient pas nécessaires puis nous avons passés toutes les dates au format ISO 8601 (YYYY-mm-DD) lorsque cela était possible, et en attribuant NA, qui correspond à la valeur nulle en R, aux chansons

dont l'année n'était pas définie. Une fois cela fait, nous avons extrait uniquement l'année des dates formatées, puisque le reste des dates ne nous intéressent pas.

Lors de l'extraction, nous nous sommes rendues compte que certaines dates avaient été mal saisies, car les années de publications étaient supérieures à 2023, comme 2048 ou 2100. Nous avons donc corrigé ces dates à l'aide d'une table de correspondance entre les dates mal saisies et les dates réelles, que nous avons construit en vérifiant les dates de sorties des chansons. Cela nous a permis de corriger 2100, qui correspondait à 2010, ou 2048 qui correspondait à 2018. Nous avons ensuite rempli la nouvelle colonne *year*, en prenant la valeur dans *publicationDate* si elle était définie, ou dans *releaseDate* à défaut. Toutes les valeurs non définies ont simplement été remplies avec *NA*, pour faciliter les traitements futurs. Nous avons ensuite fini cette partie du traitement en supprimant ces deux colonnes qui n'étaient plus nécessaires.

Nous avons ensuite dû trouver un moyen de rendre utilisable le genre d'une chanson. En effet, c'est une information cruciale que nous utilisons dans 3 de nos visualisations. Ces genres étaient stockés sous la forme d'une liste, ce qui les rendait impossibles à utiliser en l'état.

Pour les rendre utilisables, la première chose a été de fusionner les listes de tous les genres inscrits dans le jeu de données. Pour cela, nous avons retiré tous les caractères parasites, puisque certaines listes ont été écrites avec des guillemets autour des mots, que certaines listes comportent des crochets en début et en fin de liste, alors que d'autres listes ne comportent pas ces éléments. Une fois les listes standardisées, nous les avons fusionnées, et nous avons affiché le nombre d'occurrences de chaque genre dans un histogramme.

Le but de l'histogramme, qui est situé dans le dossier *visualisation/public/distrib* était de visualiser les genres les plus représentés, ce qui permet de constater que la distribution des genres était très inégale. Nous avons choisi de conserver les 20 genres les plus fréquents sur 656, car ces 20 genres représentent plus de 75% des genres présents en fréquence d'apparition. Nous avons ensuite regroupé les différentes variations de genres, en créant 20 listes correspondant aux genres majoritaires. Par exemple, les genres 'Soft Rock', 'Hard Rock' et 'Folk Rock' ont été placés dans la liste 'Rock', tandis que les genres 'Jazz standard', 'Jazz Fusion' et 'Smooth Jazz' ont été placés dans la liste 'Jazz'. Tous les genres qui ne pouvaient pas être regroupés ont été placés dans une liste 'Other', qui permet d'identifier que la chanson a un genre renseigné qui est minoritaire comme 'Garage' ou 'Adult Contemporary Music'.

Une fois cette liste de genre établie, nous avons pu parcourir le genre pour lui assigner le genre majoritaire qui correspond au genre de sa liste. Si une chanson appartient à plusieurs genres, son genre correspondra au genre le plus représenté dans le jeu de données. Pour les chansons qui n'ont pas de genre définies, nous avons créé la catégorie 'Undefined'. Ce traitement nous a permis d'obtenir un genre par chanson, et de réduire significativement les genres, de 656 à 22, ce qui rend les visualisations basées sur les genres possibles, et permet d'éviter les genres dupliqués ou les variantes trop peu significatives.

Parallel set : Ambre Correia

Traitement des données

Tout d'abord, le fichier *parallel_set_preprocess.R* s'occupe du pré-traitement des données. Dans un premier temps, les colonnes utiles ont été extraites des fichiers *songs_cleaned_date_genre* (*explicitLyrics*, *artist_name*, *id_album*, *genre*), de *artists.csv* (*name*, *type*, *gender*), et *albums.csv* (*X_id...oid*, *explicitLyrics*).

Pour plus de clarté, *X_id...oid* a été renommé en *id_album*, et *explicitLyrics* en *explicitAlbum*. De même, le nom des artistes a été renommé en *artist_name* pour effectuer la jointure plus facilement.

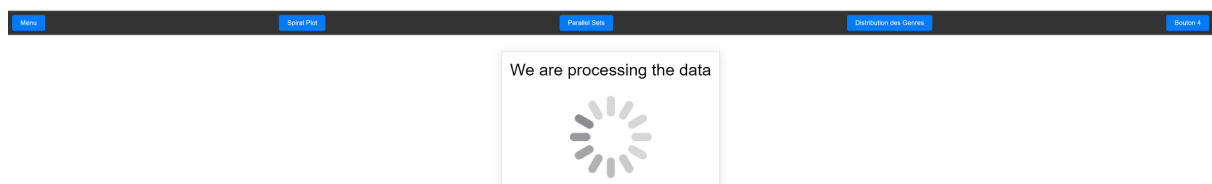
Concernant la jointure, elle a été effectuée entre les musiques et les artistes sur *artist_name*, et sur *id_album* pour les albums. Les colonnes servant pour la jointure ont ensuite été supprimées, car elles n'étaient plus utiles.

Ensuite, le traitement principal consiste à supprimer les lignes vides, ou celles qui avaient des valeurs manquantes dans *type*, *gender* et *explicitAlbum*, car les données seraient inutilisables. Après cette première étape, les valeurs manquantes ont été complétées selon le processus suivant :

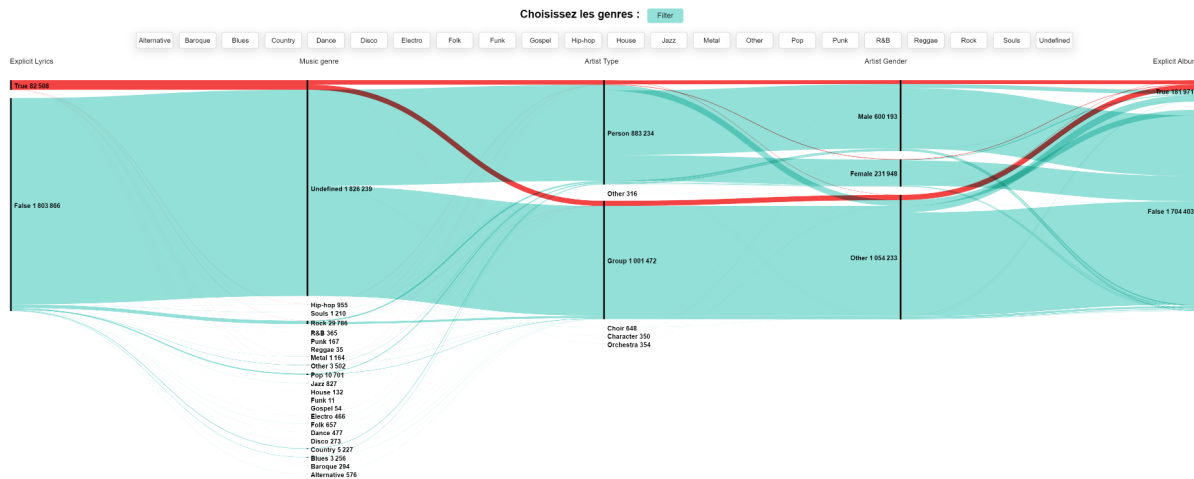
- Si le *gender* est à *male* ou *female*, alors le *type* est complété par *person*, sinon par *group*
- Si l'album contient une musique explicite, alors il est également explicite, sinon il est à *false*
- *Gender* est complété par *other*

Visualisation

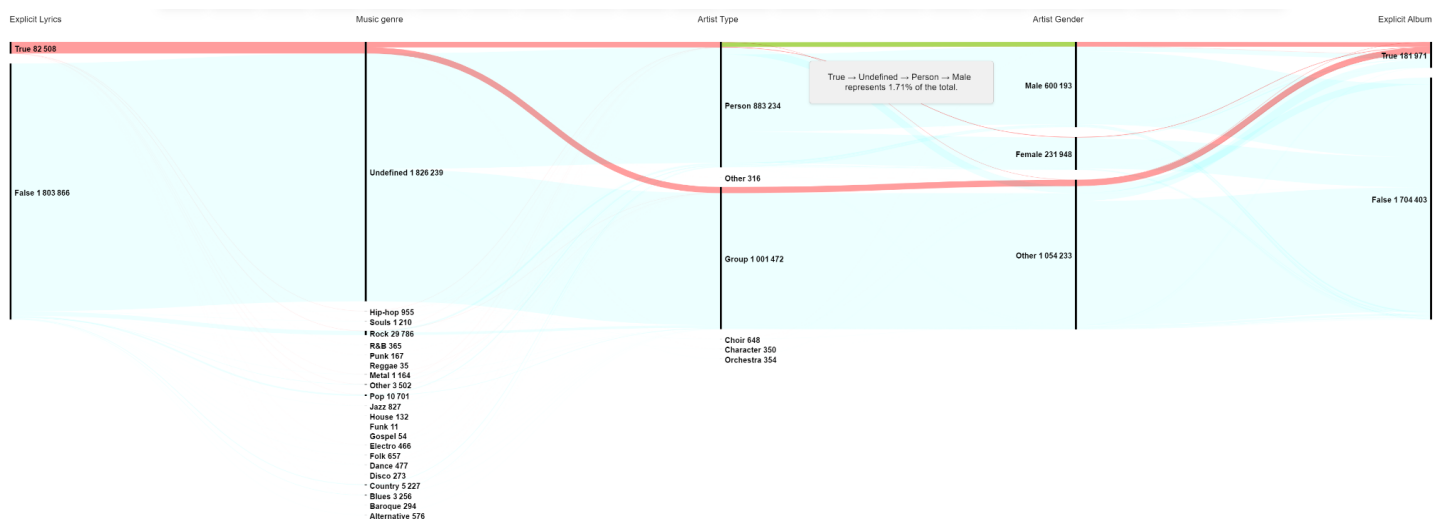
Quand nous arrivons sur la page du parallel sets, un message nous informe que les données sont en train d'être traitées, comme montré dans l'image ci-dessous. En effet, les fichiers CSV font près de 2 millions de lignes et ont besoin d'un petit délai pour être traités.



Pour les deux niveaux de visualisation, nous avons tout d'abord le graphe, qui est affiché avec la couleur rouge pour les *explicitLyrics*, et bleu pour le reste. Les chiffres associés à chaque section du graphe sont également affichés de base pour plus de clarté.



Lorsque l'on passe sur le graphe, un tooltip s'affiche en hover et il contient le chemin exact ainsi que le nombre d'occurrence. Donc, par exemple, *male* affiche 600 193 nombre d'occurrence, mais le tooltip du path *True -> Undefined -> Person -> Male* va afficher 32 295, ce qui est une information différente. Enfin, lorsque l'on clique sur une section du graphe, cette section est mise en valeur et une popup s'affiche, donnant le chemin exact des informations et le pourcentage que cette section représente.



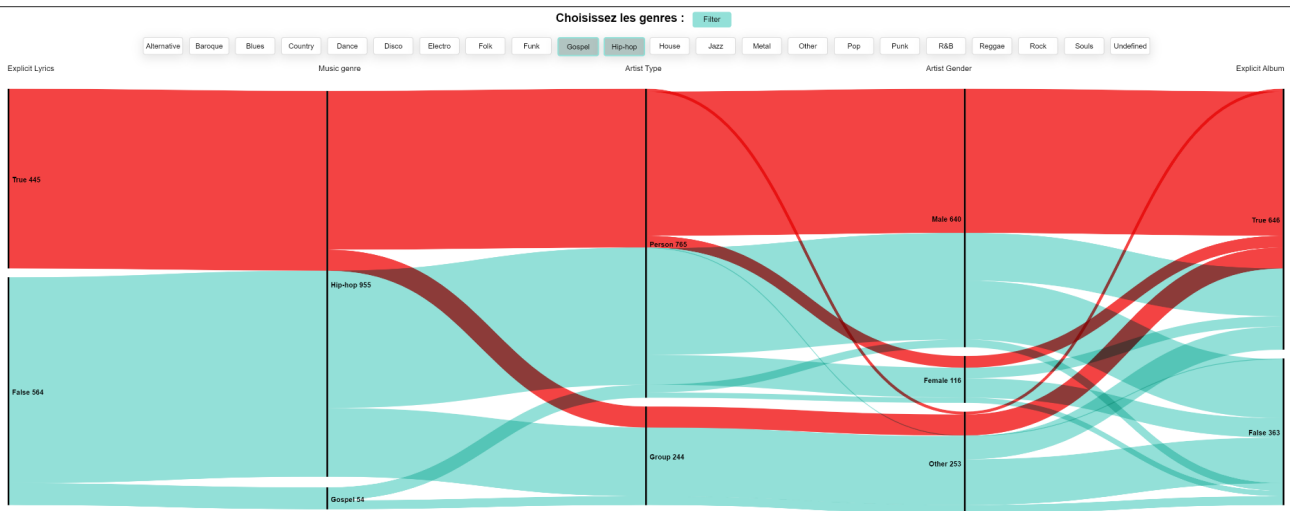
La popup qui s'affiche peut être déplacée à l'aide de la souris pour ne pas gêner la visualisation. Pour la faire disparaître, il suffit de re-cliquer sur la section de graphe sélectionnée pour revenir à l'état normal.

Pour la tâche interactive, nous retrouvons un filtre, en haut de la page, dans lequel il est possible de sélectionner le ou les genres que l'on veut filtrer. Pour cela, il suffit de les choisir puis d'appuyer sur *filtrer*. A nouveau, un message de chargement des données sera affiché à l'utilisateur.

Choisissez les genres : Filter

Alternative Baroque Blues Country Dance Disco Electro Folk Funk Gospel Hip-hop House Jazz Metal Other Pop Punk R&B Reggae Rock Souls Undefined

On peut retrouver ci-dessous un exemple du graphe une fois qu'un filtre est appliqué.



Spiral Plot : Théo Jeannes

Ma représentation consiste à afficher le nombre de chansons par genre à travers les années, en différenciant le caractère explicite de ces chansons, afin de mettre en évidence d'éventuelles tendances dans l'évolution des chansons explicites à travers les genres et les années.

Traitements des données

Toutes les fonctions ayant servi à préparer les données du spiral plot, ou graphique en spirale, sont dans le fichier *spiral_plot_preprocess.R*. A partir des données préparées en amont, je n'ai eu besoin que des données des chansons.

La première étape a été de filtrer les chansons pour ne garder que les années, les genres et le caractère explicite ou non d'une chanson. J'ai ensuite enlevé toutes les lignes sans année, puisque ma visualisation ne permet pas de représenter ces lignes. J'ai également limité le jeu de données aux années entre 1962 et 2017, les années en dehors de cette période ne comportant que moins d'une dizaine de chansons pour certaines, et pas du tout pour le reste des années.

J'ai fini par associer à chaque ligne une valeur correspondant à son nombre d'occurrences dans mon jeu de données, pour pouvoir mesurer le nombre de chansons par année par genre et par caractère explicite ou non. Les données finales ont donc quatre colonnes, l'année *year*, le genre musical *genre*, le caractère explicite *explicitLyrics* et le nombre de chansons dans cette catégorie, *value*. Ces données sont contenues dans le fichier *spiral_plot_count.csv*.

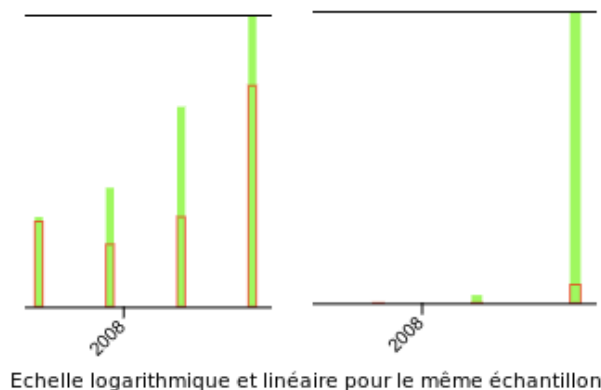
Visualisation

Pour ma visualisation, je suis parti du travail d'Arpit Narechania ([Annexe 2](#)), qui implémente un spiral plot. La première chose a été de remplacer les données factices par mes propres données.

Pour faire apparaître les données sur l'axe, j'ai dû adapter les échelles. La position de chaque barre sur la spirale était définie uniquement en fonction de la date. Avec mes données, cela aurait amené tous les genres à se superposer plutôt qu'à être placé côte à côte. Pour éviter cela, une map associant le genre à un nombre entre 0 et 1 est calculée à chaque affichage des données. La position correspond maintenant à l'année sommée avec la valeur du genre dans la map. Cette technique m'a permis de positionner tous les genres d'une même année côte à côte.

Pour identifier plus facilement que des genres font partie d'une même année, les genres sont regroupés par groupe en fonction de leur année, et chaque groupe se voit attribuer une couleur. J'ai choisi seulement 6 couleurs, pour ne pas surcharger visuellement le graphique. La palette de couleurs a été générée avec I Want Hue(Annexe 3), ce qui permet d'obtenir des couleurs les plus différentes possibles, afin de réduire les chances que l'utilisateur puisse confondre deux couleurs.

J'ai ensuite fait le choix d'afficher le nombre de chansons, soit la hauteur de chaque barre sur une échelle logarithmique. Sans celle-ci, avec une échelle linéaire, le genre 'Undefined' qui représente la vaste majorité de chaque année aurait été si



grand que les autres barres n'auraient été affichées que sur cinq ou dix pour cent de la hauteur. L'inconvénient de cette échelle est sa difficulté de compréhension, les comparaisons étant bien plus faciles sur une échelle linéaire. Cependant au vu de la place limitée sur l'écran, une échelle logarithmique permet d'afficher des différences importantes facilement, ce qui est le but recherché puisque nous voulons identifier des tendances.

Pour identifier les chansons à caractères explicite, j'ai choisi de les encadrer en rouge, couleur qui contraste avec toutes les couleurs présentes dans la palette, et de diminuer légèrement l'opacité. Cela permet de distinguer facilement si un genre sur une année comporte plus de chansons explicites que de chansons implicites.

J'ai par la suite modifié le tooltip déjà existant, pour faire correspondre les labels avec nos données, et pour ajouter des informations sur la proportion. En effet, puisque la comparaison avec une échelle



First Year 1962 Last Year 2017

- | | |
|-------------|-----------|
| Alternative | Baroque |
| Blues | Country |
| Dance | Disco |
| Electro | Folk |
| Funk | Gospel |
| Hip-hop | House |
| Jazz | Metal |
| Other | Pop |
| Punk | R&B |
| Reggae | Rock |
| Souls | Undefined |

logarithmique n'est pas intuitive, j'ai ajouté une tooltip, affichée lorsque que l'utilisateur survole une barre, qui contient à la fois le nombre de chansons de la catégorie et le nombre de chansons pour ce genre de cette année, mais également un pourcentage représentant cette valeur, pour faciliter les comparaisons entre les années.

J'ai ensuite ajouté plusieurs filtres différents, qui permettent de restreindre les données affichées pour faciliter l'exploration, ou pour n'afficher que les

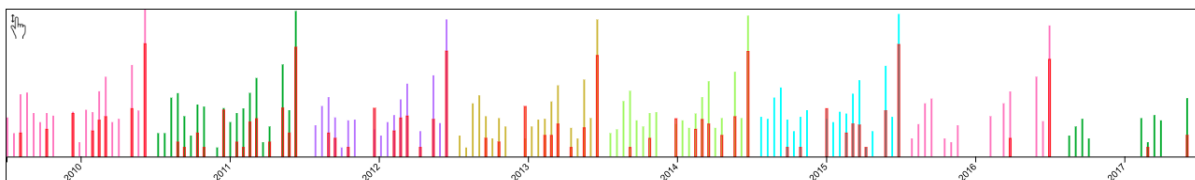
☒ Show explicit songs ☒ Show implicit songs

Filter

données souhaitées si l'utilisateur sait déjà ce qu'il cherche et souhaite simplement localiser l'information. Le premier filtre est un double slider qui permet de définir la période à afficher sur le graphique. J'ai ajouté deux champs pour saisir la date manuellement si l'utilisateur trouve cette saisie plus confortable. Il y a ensuite une liste des différents genres, pour permettre à l'utilisateur de choisir les genres affichés, en bleu des genres qui ne doivent pas l'être, en blanc. Pour finir, deux cases à cocher permettent de filtrer les chansons par caractère explicite ou non, si l'utilisateur ne souhaite comparer que le nombre de chansons explicites en fonction des années par exemple. Le bouton *Filter*, tout en bas du panneau avec les filtres, permet de valider les changements, pour afficher les données filtrées.

L'étape suivante a été d'adapter dynamiquement la taille à la fois du graphique, et la largeur de chaque barre. Pour adapter la taille du graphique et du panneau de filtres, j'ai simplement défini des proportions en CSS, pour rendre l'affichage responsive. La largeur de chaque barre est quant à elle calculée dynamiquement en fonction du nombre d'informations à afficher, afin de s'assurer que deux barres ne rentrent pas en collision, ce qui créerait une occlusion non désirée.

La dernière chose à faire a été d'ajouter un deuxième niveau de visualisation, pour pouvoir voir les proportions, et la répartition entre explicite et implicite correctement, sans perdre l'information globale. Pour cela, j'ai introduit un histogramme sous le spiral plot, pour permettre une visualisation globale, et de zoomer sur une partie spécifique du spiral plot en même temps.



Cet histogramme reprend les mêmes règles d'affichage que le spiral plot, et permet de zoomer ou de dézoomer à l'aide de la molette, comme indiqué par l'icône en haut à gauche du graphique, pour afficher plus ou moins d'informations. Lorsque l'utilisateur clique sur une barre, que ce soit dans le spiral plot ou dans l'histogramme, les données affichées dans l'histogramme se centre instantanément sur la barre cible, ce qui permet à l'utilisateur de visualiser les proportions du genre et de la période choisie, et son voisinage proche, tout en gardant une vue globale dans le spiral plot, ce qui permet de remettre les détails dans le contexte.

J'ai du pour finir adapter les labels, pour placer l'année au milieu de son genre, et modifier la position de la tooltip par rapport au curseur, afin de la placer au-dessus du curseur, pour ne pas cacher la visualisation.

Modifications par rapport à ce qui est prévu

L'objectif de la visualisation était de représenter les genres à travers les années, en différenciant les chansons explicites ou non. Cela n'a pas changé, la représentation finale permet bien d'afficher ces informations.

Les tâches supportés devaient être d'identifier une tendance temporelle dans le développement des chansons explicites, pouvoir faire un zoom sur une période temporelle, filtrer différents genre de musique, pouvoir afficher plus d'informations au survol, et être capable d'utiliser une vue en détails par rapport à la vue globale. Toutes ces tâches sont supportées par ma visualisation.

Ma visualisation permet d'explorer les données, de localiser une catégorie précise en filtrant les genres et la période comme anticipé, mais elle permet également de catégoriser et de distinguer les chansons explicites des chansons implicites, grâce aux checkboxes.

Bubble Map : Quentin BOURDEAU

Cette représentation permet de voir le nombre de chansons qui a été créée en fonction de leur localisation géographique. La taille des cercles correspond au volume de données correspondantes et la couleur à la quantité relative de chansons explicites ou non ce qui permet de se rendre rapidement compte de la répartition du caractère explicite au travers du globe.

Traitements des données

Plusieurs fichiers sont utilisés pour créer mes données car il est plus simple de passer par des fichiers intermédiaires pour obtenir mes données finales.

Il a tout d'abord fallu éliminer toutes les lignes qui ne m'intéressaient pas :

- Les chansons n'ayant pas de pays ou un pays non défini par les normes conventionnelles internationales
- les chansons n'ayant pas leur caractère explicite défini (ou attend True ou False au final)
- les chansons n'ayant pas d'années de sortie

Ensuite j'ai combiné mes données à un autre CSV trouvé sur le lien suivant :

https://developers.google.com/public-data/docs/canonical/countries_csv?hl=en

Cela m'a permis d'avoir le nom complet des pays correspondant, ce qui est utile à l'utilisateur final, mais aussi d'avoir les coordonnées des pays pour savoir où afficher les données correspondantes.

J'ai aussi associé à chaque chanson sa décennie de sortie, pour pouvoir grouper plus facilement par la suite.

Pour la dernière étape il m'a fallu compter le nombre de chansons partageant le même pays (et donc les mêmes coordonnées) et les regrouper en une seule ligne comportant les nouvelles informations suivantes :

- Le nombre total de chanson
- Le nombre de chanson avec un caractère explicite
- Le nombre de chanson sans caractère explicite
- Le ratio de chanson explicite / non explicite

Pour la gestion des décennies, cela pourrait être géré de manière dynamique en JS mais le traitement prend beaucoup de temps. J'ai donc préféré créer des presets contenant les données des différentes époques disponibles.

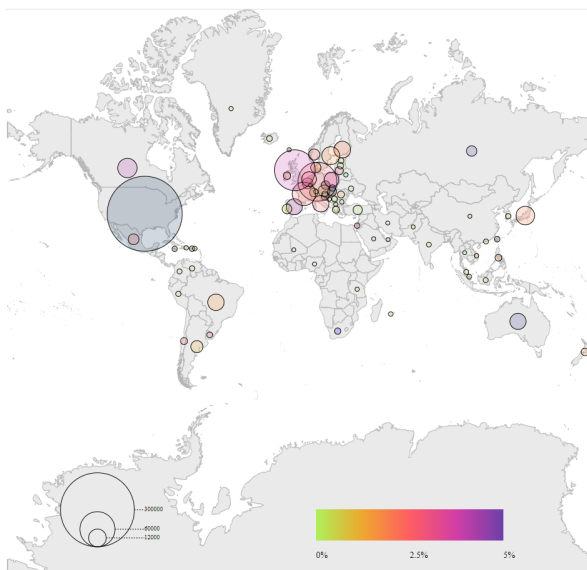
Visualisation

Pour la visualisation je suis parti de plusieurs tutoriels disponibles notamment un des plus complets : https://d3-graph-gallery.com/graph/bubblemap_template.html

La plupart des visualisations de bubble map sont adaptées à d3 v4 ou v6. Je me suis basé sur les tutoriels de v4 en traduisant pour la v5 utilisée dans ce projet.

Ma visualisation se sépare en 3 parties :

- La map



- Les filtres

☐ Remove US:

Decade: All

- Les informations sélectionnées

Country Name: Australia

Volume of songs: 9469

Volume of explicit songs: 527

Volume of non-explicit songs: 8942

% of explicit songs: 5.57%

Country Name: United States

Volume of songs: 309020

Volume of explicit songs: 19925

Volume of non-explicit songs: 289095

% of explicit songs: 6.45%

La map contient évidemment les bulles d'informations déjà évoquées, en plus d'une légende qui dépend des filtres utilisées (la taille des bulles s'adaptent aux données).

Pour les filtres il y en a 2 :

Tout d'abord le "remove US", en effet puisque toute la représentation s'adapte à la taille du cercle le plus gros et que les US ont sensiblement plus de données que les autres, l'avoir peut parfois rendre la visualisation moins intéressante.

Enlever ce point permet de rééquilibrer la taille des plus petits pays, notamment ceux d'Afrique, Amérique du sud ou encore d'Australie

Ensuite le filtre de décennie, qui peut trier parmi les décennies intéressantes, ou toutes les chansons avant les années 90.

La 3ème partie est la sélection d'informations : elle est formée de deux cadres comprenant les informations que l'on souhaite voir (volume de données explicite ou non, pays, ratio de données explicites).

Le premier cadre est mis à jour lorsque l'on passe sur un cercle avec la souris tandis que le deuxième est mis à jour lorsque l'on clique sur une bulle.

Cela permet de facilement visualiser une comparaison de deux pays : on clique d'abord sur les US pour garder les informations dans le deuxième cadre, puis on passe la souris sur l'Australie. Cela affichera alors les deux informations côte à côte, permettant de comparer efficacement. La map permet donc à la fois d'avoir une vue globale grâce à la taille des cercles et leur couleurs, et à la fois une vue plus locale en comparant directement 2 lieux.

Population Pyramide : Benjamin Bernigaud

Ma représentation consiste à afficher le nombre de chansons à travers les années, en différenciant le caractère explicite de ces chansons, afin de mettre en évidence d'éventuelles tendances dans l'évolution des chansons explicites à travers les genres et les années. Il est également possible de regarder les genres, les artistes et les pays les plus prolifiques au long des années.

Traitements des données

En premier, le script CountByExpAndYear permet de trier les données. celui-ci accomplit deux choses : en premier lieu, il permet d'avoir le nombre de chansons explicite par an, trier par explicite ou non. Il permet aussi de trier les musiques par an et par genre. ici, on ne garde les champ ExplicitLyrics, year et Count

Dans un deuxième temps, ce script effectue la même action, mais en plus va trier les données par leur genre.

Ensuite, on utilise le script classeurGenreSur10Ans, qui permet de classer les genres les plus représentés sur 10 ans. Celui-ci va créer un CSV avec la période, l'explicité, le genre et le nombre de musique.

Après cela, on va utiliser les scripts group_by_country et bestAuthorByDecade avec les albums en données d'entrées.

Ceux-ci servent respectivement à associer à chaque période, pour les musique explicites ou non, quels sont les groupes les plus populaires ainsi que les pays cumulant le plus de vue sur leur musique. Cette comparaison est faite en regardant le nombre de fans deezer. on garde ensuite les noms des pays, l'explicité et la période dans le premier et les noms des groupes, l'explicité et la période dans le second.

Visualisation

La visualisation se fait en deux temps principaux : tout d'abord, le modèle de base est une pyramide de population. Les dates les plus anciennes sont en bas, et on utilise une échelle logarithmique afin de s'assurer de la visibilité des données les plus faibles en valeur. On affiche des blocs



proportionnels aux valeurs de la catégorie. Ici, la pyramide est divisée entre explicite et non explicite, avec des périodes de 10 ans.

Lorsqu'on parcourt la pyramide, les différentes cases s'allument, afin de confirmer à l'utilisateur qu'il peut interagir avec cette case, et l'assurer qu'il est bien sur la case où se trouve sa souris. L'utilisateur peut ensuite cliquer sur cette case, afin d'avoir des informations additionnelles.



On a une bulle d'informations qui s'affiche, afin d'apporter des précisions. Tout d'abord, la valeur de la case. Bien que celle-ci soit visible dans le graphique, il est difficile de voir la valeur exacte, puisque les valeurs sont élevées. Ensuite, on peut voir s'afficher les groupes les plus populaires dans ce type (explicite ou non) et cette période, ce qui permet de voir quels sont les groupes ayant marqué

l'époque. De même, on a accès au pays et genre les plus populaire, ce qui permet notamment de voir quelles sont les tendances de l'époque, et éventuellement les comparer à celle des autres cases, afin de voir l'évolution de celle-ci.

Modifications par rapport à ce qui est prévu

Un certain nombre de modifications ont été effectuées par rapport à l'objectif d'origine. Celles-ci sont toutes liées aux quantités d'informations affichées. En effet, il était prévu d'afficher les 5 pays les plus populaire ainsi que les 5 groupe les plus populaire, néanmoins seulement 3 de chaque sont affiché dans le rendu final, en effet ce choix a été fait pour des raisons de clarté, le but de ce graphique étant notamment de garder en visibilité pour tout publique, avoir plus de données que nécessaire réduisait la clarté sans forcément délivrer des informations réellement utiles.

Annexes

Annexe 1:

- <https://github.com/JeannesTheo/infovis> : Github avec le code source
- <https://drive.google.com/drive/folders/1KMQ-7-d675h42WQPgVOmDHN4jC1Lqb27?usp=sharing> : Intégralité des fichiers CSV utilisés durant le projet.

Annexe 2:

- <https://gist.github.com/arpitnarechania/027e163073864ef2ac4ceb5c2c0bf616>

Spiral Plot créé par Arpit Narechania, avec un tooltip et des données factices.

Annexe 3:

- <https://medialab.github.io/iwanthue/>

Site permettant de choisir une palette de couleur optimales, ou chaque couleur est le plus éloigné possibles des autres couleurs