

SAE 2.04

Exploitation d'une base de données

Projet **OCTAVE**

Contexte :

Nous allons vous présenter une **Analyse du Marché pour une Plateforme de Streaming Musical**.

Octave est notre plateforme de streaming musical. (*fictive*)

Nous souhaitons obtenir des financements afin de créer une application Musical à l'image de ce que propose nos principaux concurrents **DEEZER** et **SPOTIFY**.

Nous allons donc vous proposer une **étude de marché**. Issu de la création d'une base de données composées de 3 **DataSet**.

- [Global Music Streaming Trends & Listener Insights](#)
- [Music Sales by Format and Year](#)
- [Global Internet users](#)

Présentation des DataSet :

[Global Music Streaming Trends & Listener Insights.](#)

Le jeu de données est composé de plusieurs variables qui couvrent différents aspects des habitudes d'écoute des utilisateurs de musique en streaming. Parmi les principales informations, on trouve des données concernant :

- Les **utilisateurs** : leurs profils, démographies (âge, sexe, etc.) et leurs activités d'écoute.
- Les **chansons** : les titres de musique écoutés, les artistes, les albums, et les genres musicaux associés.
- Les **statistiques d'écoute** : le nombre de streams, la durée d'écoute, les tendances par genre musical, les classements des chansons les plus populaires, etc.
- Les **plateformes de streaming** : information sur la plateforme (par exemple, Spotify, Apple Music, etc.), les appareils utilisés pour l'écoute (smartphones, ordinateurs, etc.).

Types de données présents dans ce jeu de données :

- **Données numériques** : Ces données comprennent des valeurs numériques telles que le nombre de streams, la durée d'écoute, ou la position d'un titre dans les classements.
 - **Données catégorielles** : Par exemple, le genre musical (pop, rock, hip-hop), les plateformes de streaming, les pays ou régions d'écoute, ou encore les artistes et albums.
 - **Données temporelles** : Des informations sur les dates et heures d'écoute, ce qui permet de suivre les tendances temporelles des habitudes d'écoute (par exemple, la popularité d'une chanson à un moment donné).
 - **Données textuelles** : Le nom des artistes, des chansons et des albums, ainsi que des informations liées aux commentaires ou aux métadonnées associées.
-

Music Sales by Format and Year.

Le jeu de données **Music Sales by Format and Year** contient des informations détaillées sur les ventes de musique à travers différents formats et sur plusieurs années. Il permet d'analyser l'évolution des ventes de musique en fonction des différents moyens de distribution (CD, vinyle, téléchargement numérique, streaming, etc.) et des tendances de consommation sur plusieurs années.

Le jeu de données contient des informations sur :

- **Les ventes de musique** : Le nombre d'exemplaires vendus, ainsi que les revenus générés par ces ventes pour chaque format de distribution (CD, vinyle, téléchargement numérique, streaming).
- **Les formats de vente** : Les données sont classées par format de distribution, permettant d'analyser la popularité de chaque format au fil du temps.
- **Les années de vente** : Chaque entrée est associée à une année particulière, ce qui permet d'observer l'évolution des ventes et des revenus sur plusieurs années.
- **Les genres musicaux** : Les ventes sont également ventilées par genre musical, fournissant un aperçu des préférences musicales des consommateurs en fonction des formats de vente.
- **Les artistes et albums** : Le jeu de données inclut des informations sur les artistes et albums, permettant d'identifier les plus populaires et leur impact sur les ventes.

Types de données présents dans ce jeu de données :

- **Données numériques** :
 - Nombre d'exemplaires vendus pour chaque format et année.
 - Revenus générés par chaque format de vente.
- **Données catégorielles** :
 - Formats de vente (CD, vinyle, téléchargement numérique, streaming).
 - Genres musicaux (pop, rock, hip-hop, etc.).
 - Artistes et albums associés à chaque vente.
- **Données temporelles** :
 - Années de vente, permettant d'analyser les tendances et l'évolution des ventes de musique au fil des années.
- **Données textuelles** :

- Les noms des artistes, des albums et des genres musicaux, utilisés pour l'analyse qualitative des tendances dans l'industrie musicale.
-

Global Internet Users.

Le jeu de données **Global Internet Users** contient des informations détaillées sur l'utilisation d'Internet dans le monde entier. Il permet d'analyser la croissance et les tendances de l'internet en fonction de différents critères, tels que les régions, les pays, les années, et d'autres facteurs démographiques.

Le jeu de données contient principalement des informations sur :

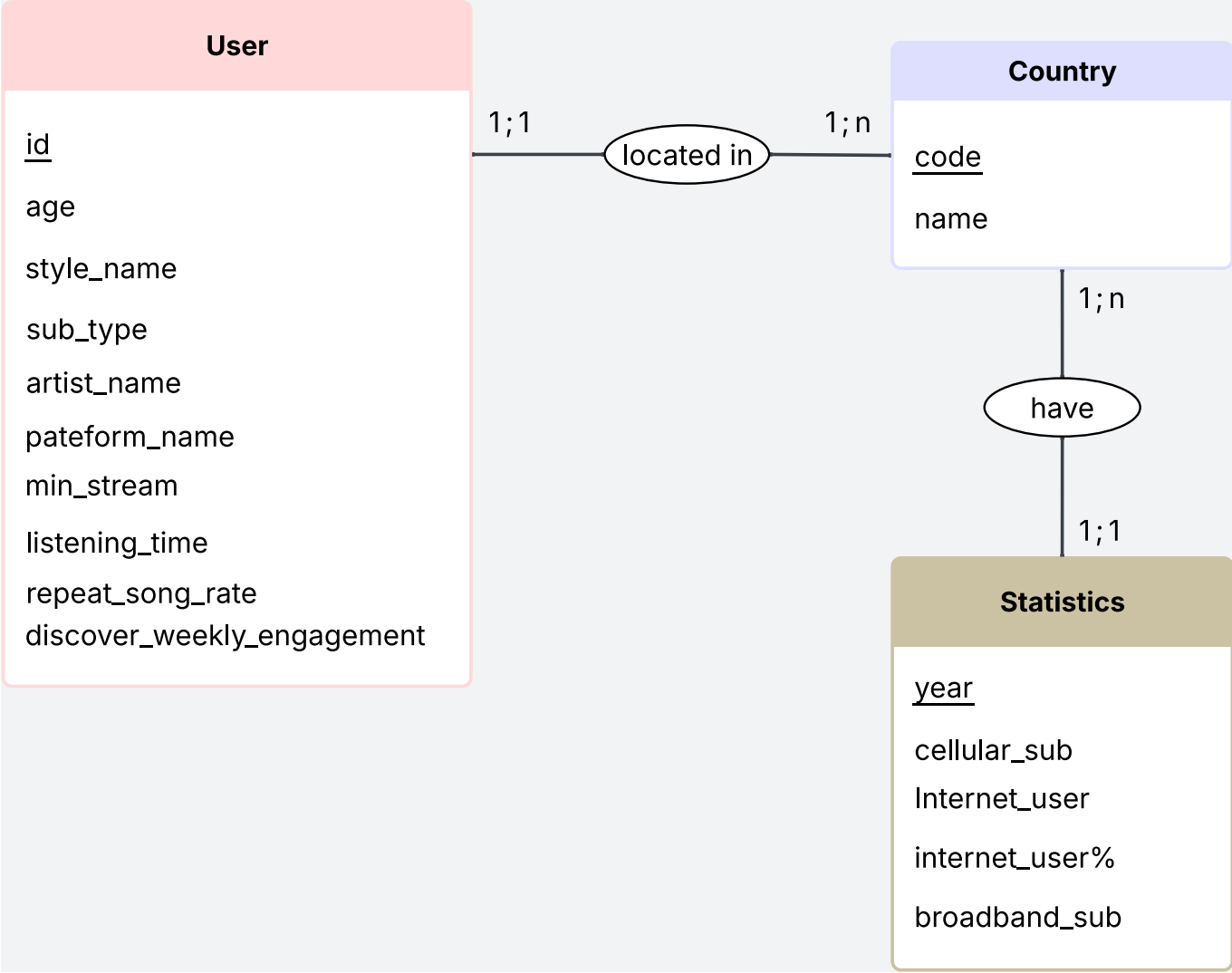
- **Le nombre d'utilisateurs d'Internet** : Il offre des données sur le nombre total d'internautes dans différents pays et régions à travers le monde.
- **Les années d'enquête** : Chaque entrée dans le jeu de données correspond à une année spécifique, permettant d'observer l'évolution de l'utilisation d'Internet au fil des années.
- **Les régions géographiques** : Les utilisateurs sont ventilés par région (Asie, Europe, Afrique, Amériques, etc.), permettant de comparer l'utilisation d'Internet selon les zones géographiques.
- **Les pays** : Il fournit des données détaillées pour chaque pays, permettant une analyse plus fine de la pénétration d'Internet à l'échelle nationale.
- **Les données démographiques** : Le jeu de données peut inclure des informations démographiques telles que l'âge, le sexe, et d'autres caractéristiques des utilisateurs d'Internet.

Types de données présents dans ce jeu de données :

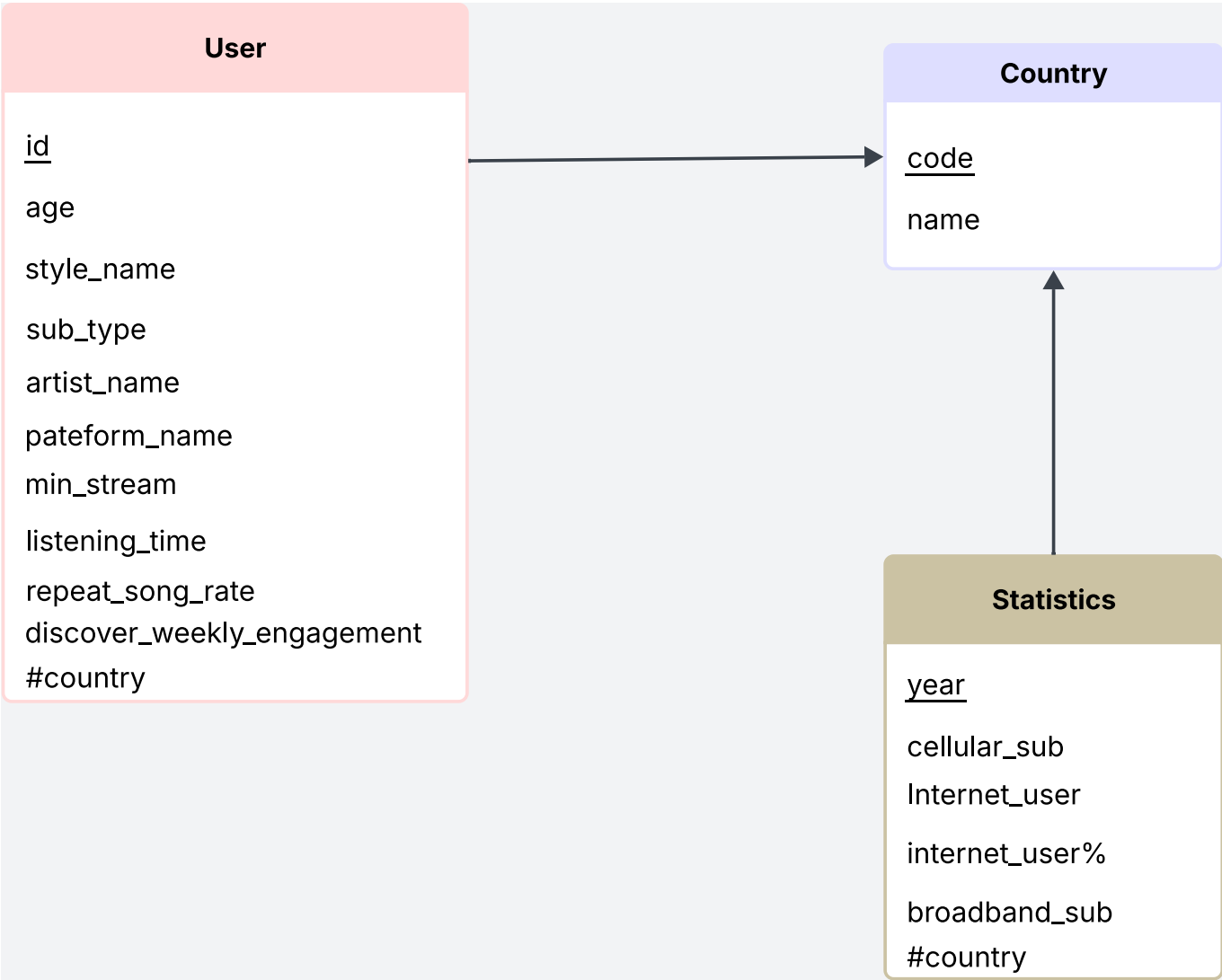
- **Données numériques** :
 - Nombre total d'utilisateurs d'Internet par pays, région et année.
 - Croissance en pourcentage du nombre d'utilisateurs d'Internet au fil des années.
 - **Données catégorielles** :
 - Régions géographiques (Asie, Europe, Afrique, etc.).
 - Pays, permettant d'analyser l'adoption d'Internet par pays.
 - Groupes démographiques, par exemple les tranches d'âge ou le sexe des utilisateurs.
 - **Données temporelles** :
 - Années spécifiques d'enquête, permettant d'analyser les tendances de croissance d'Internet dans le temps.
 - **Données textuelles** :
 - Noms des pays, des régions et des groupes démographiques (pour les tranches d'âge, les genres, etc.).
-

Présentation du MCD et du MLD

MCD



MLD

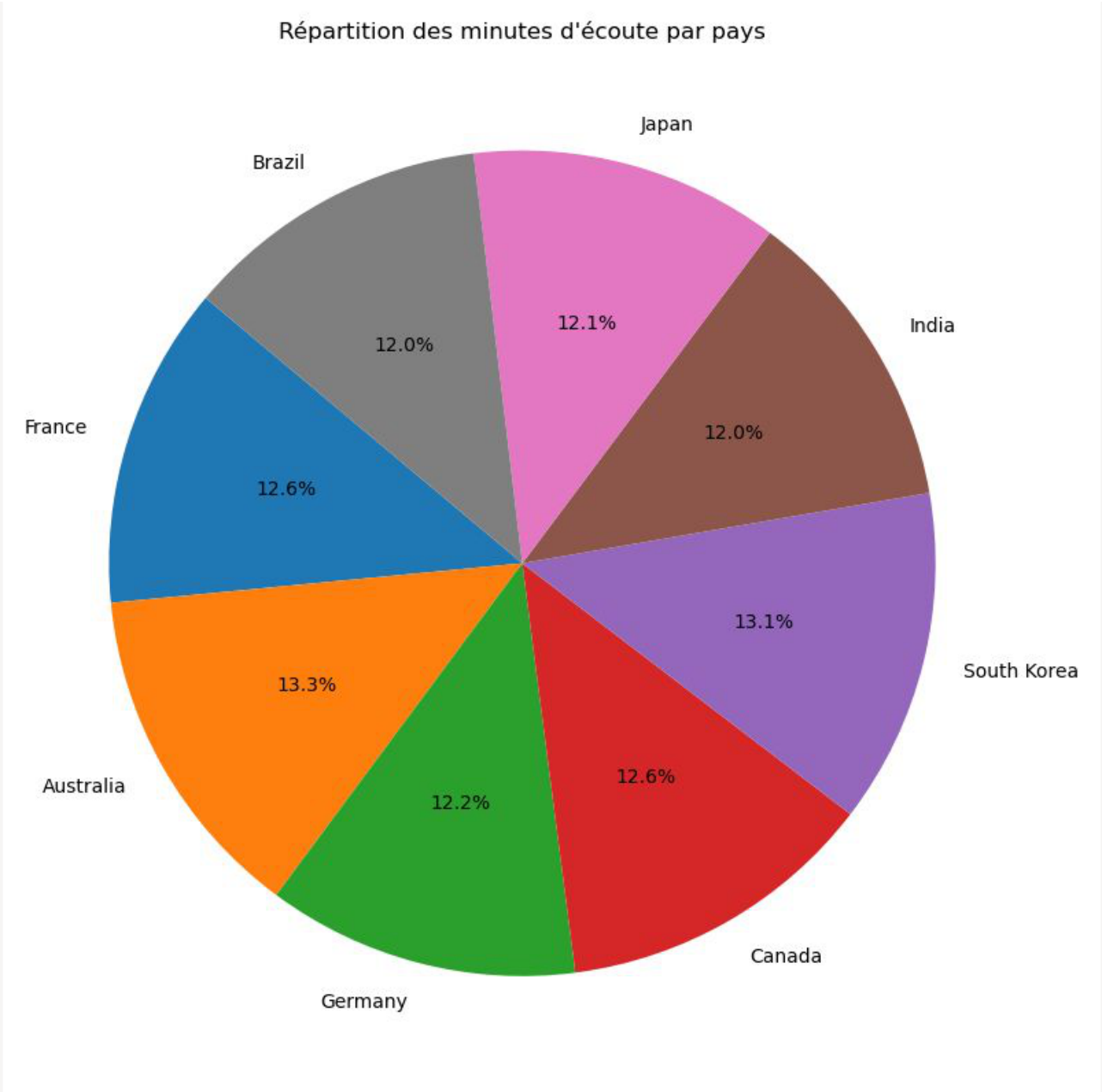


Analyse des données:

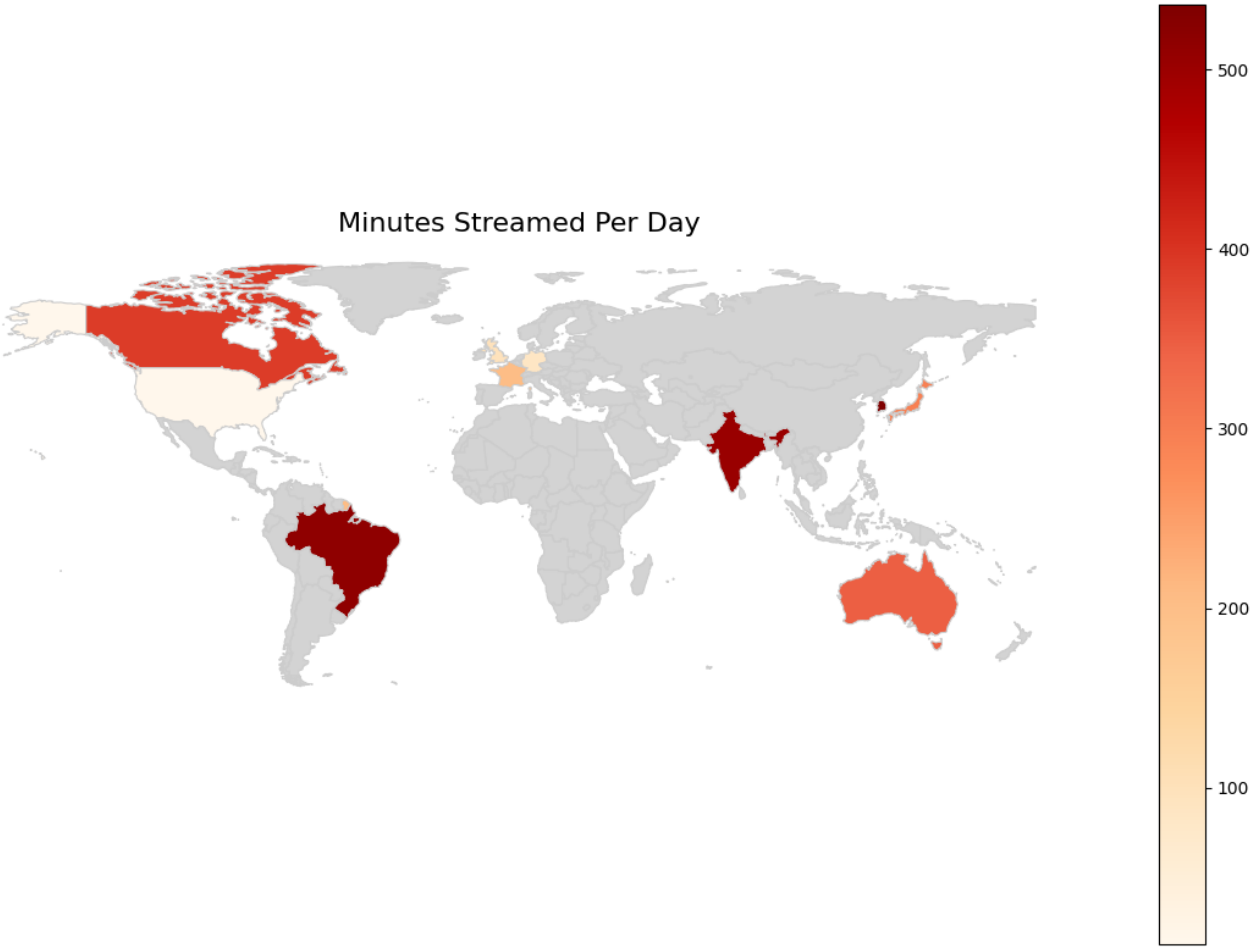
Répartition de la consommation musicale par région

Schémas :

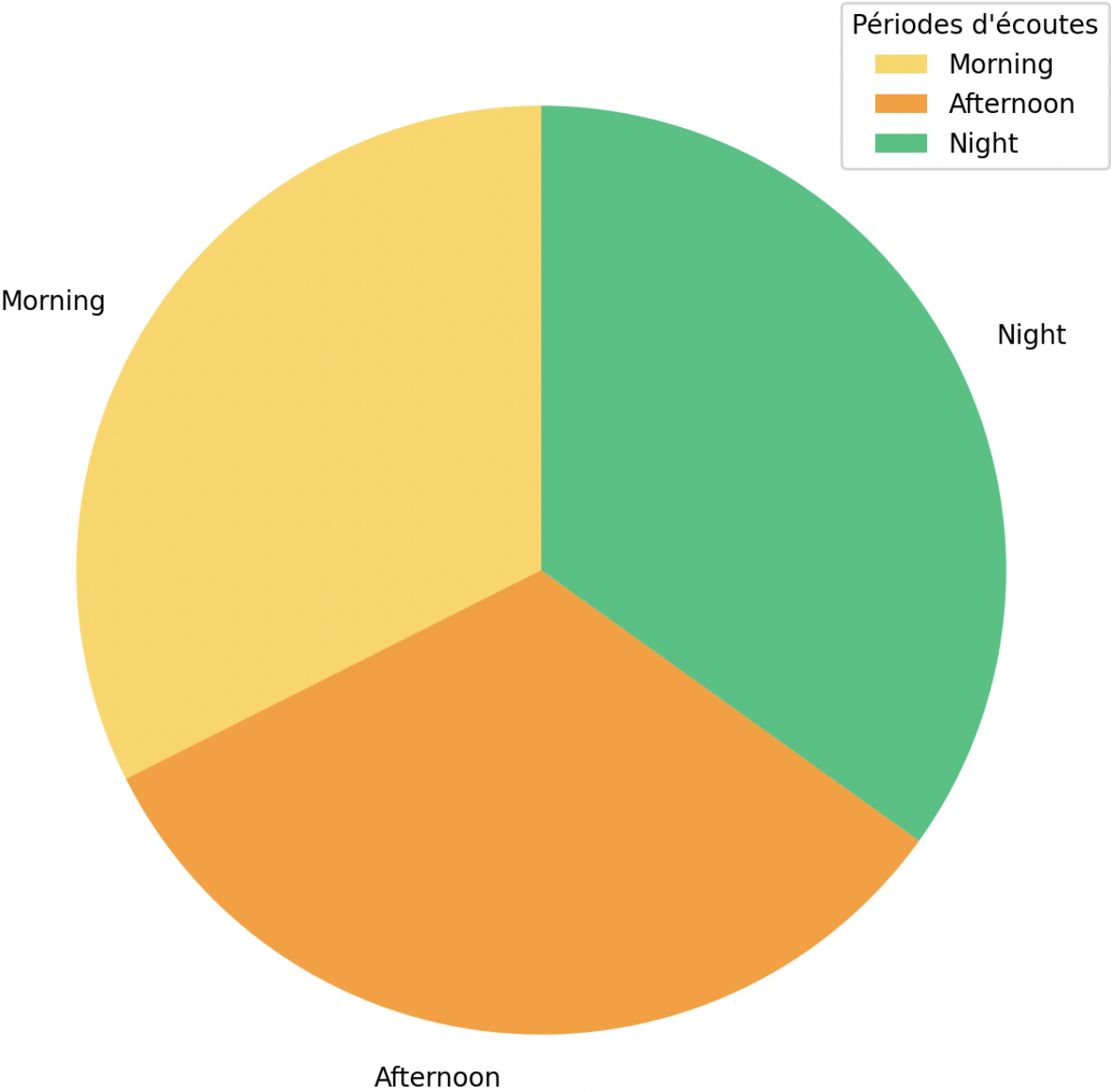
- graphique 1



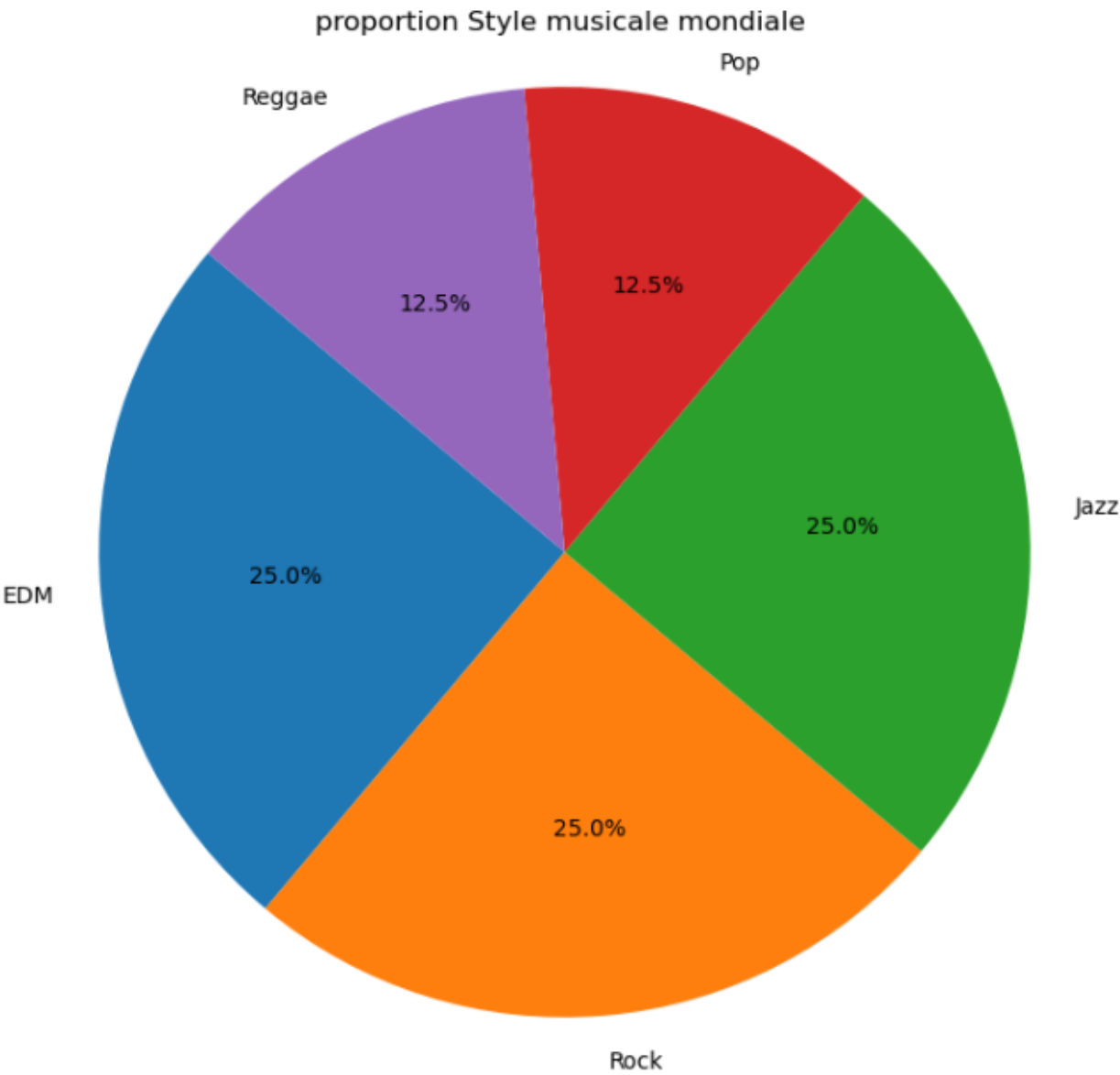
-graphique 2



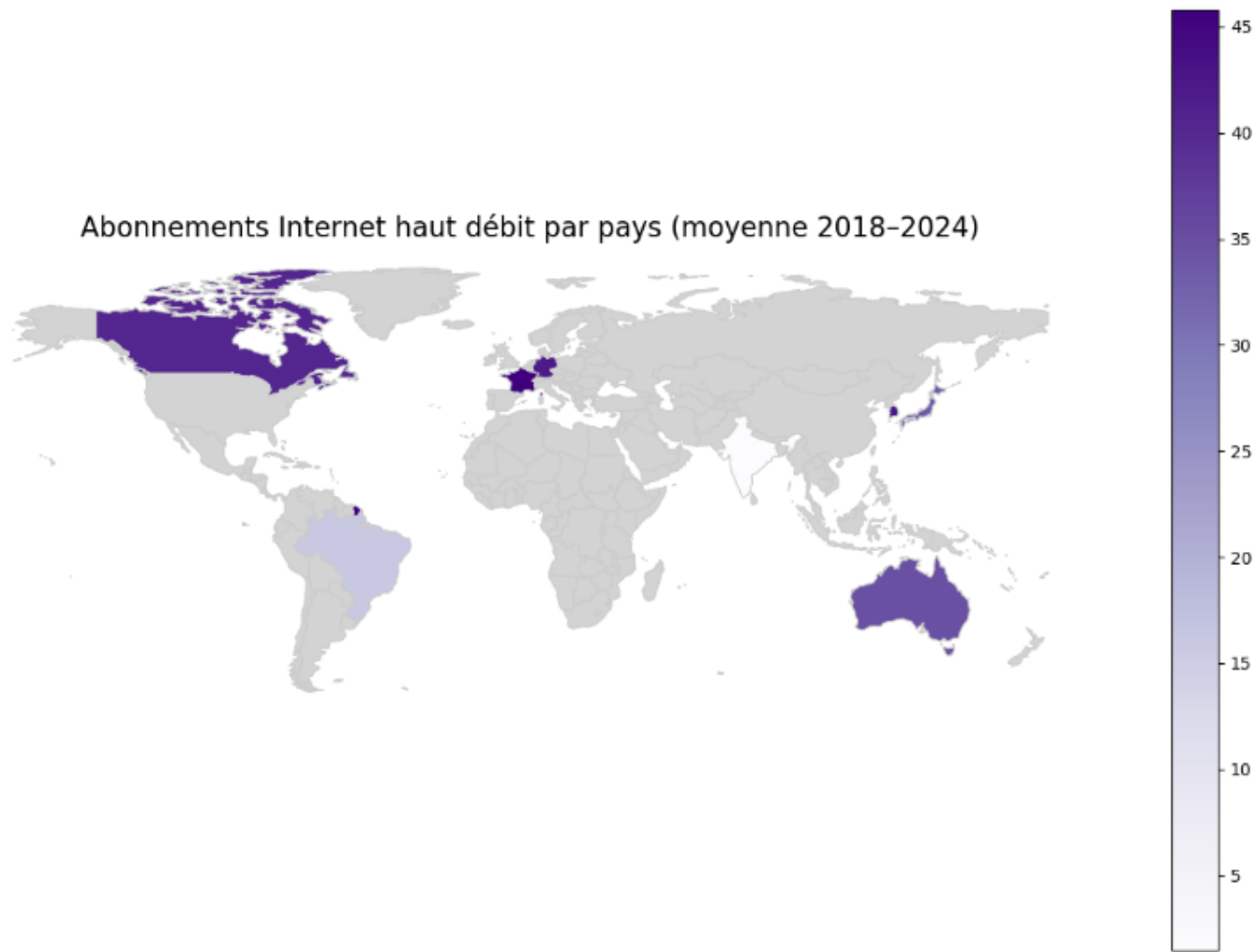
-graphique 3



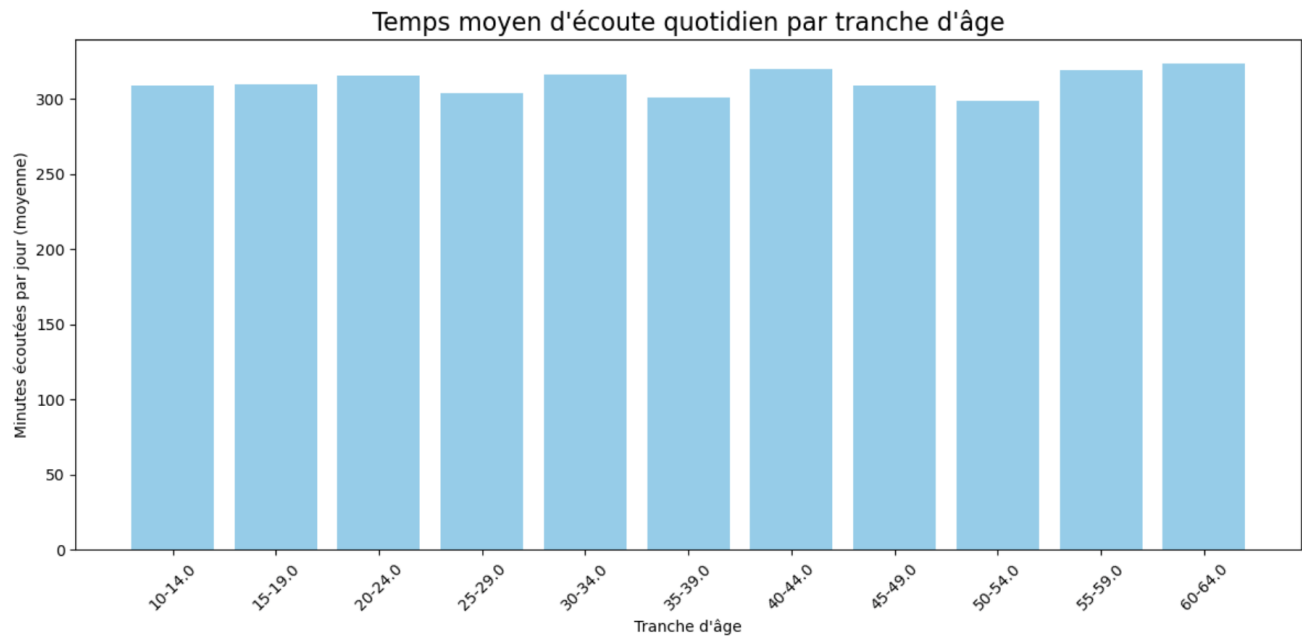
-graphique 4



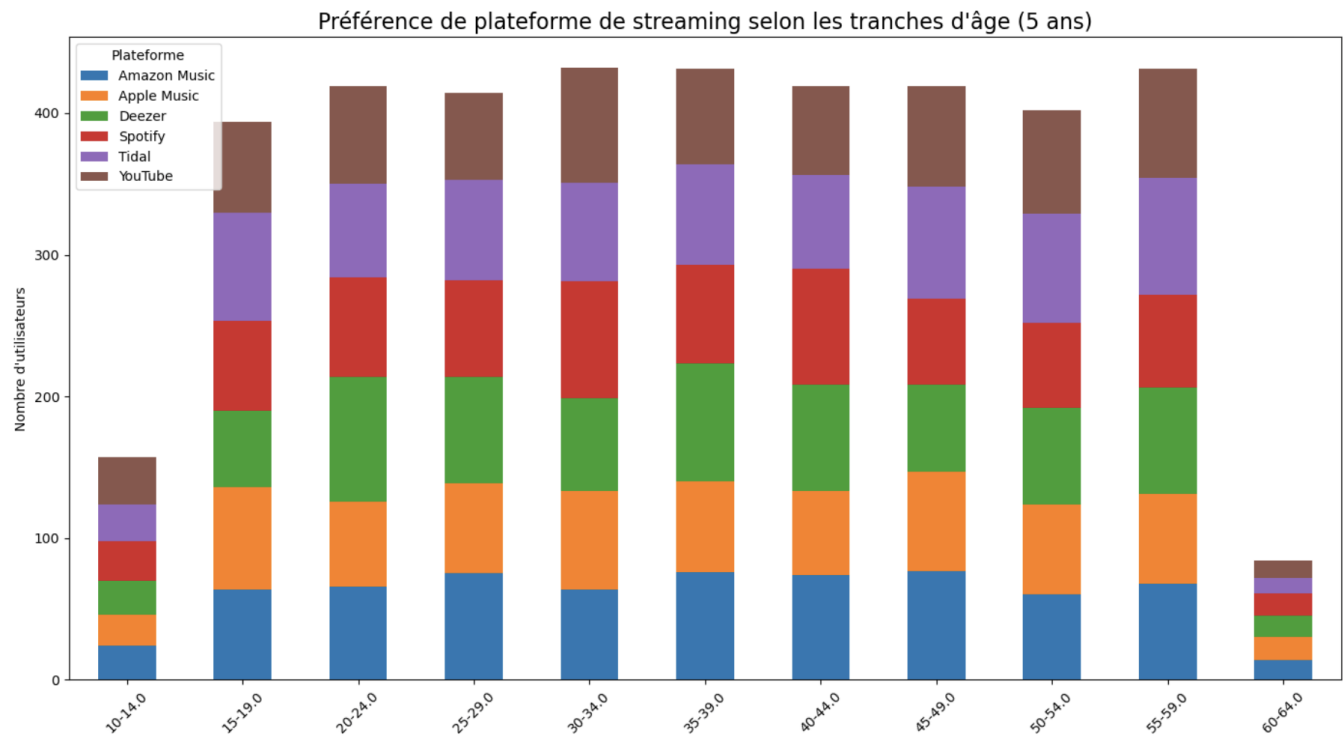
-graphique 5



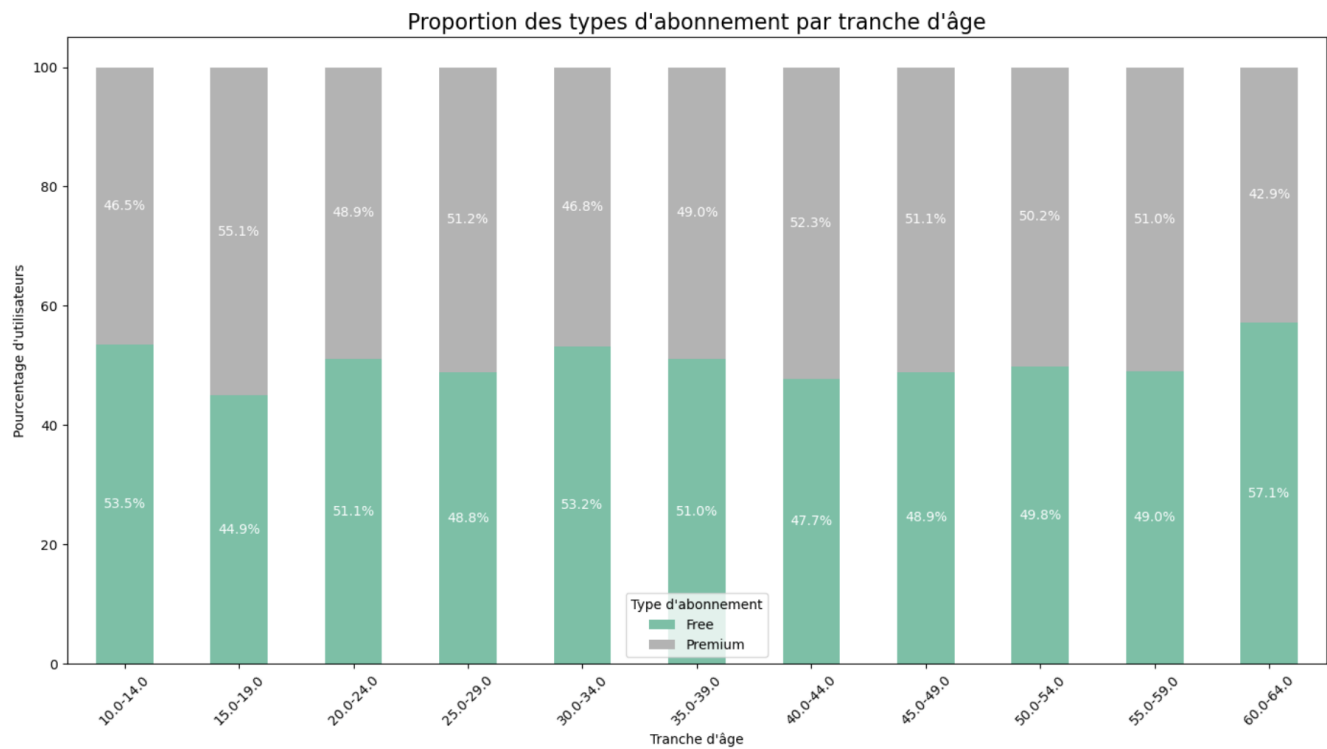
-graphique 6



-graphique 7



-graphique 8



Les graphiques fournis montrent des tendances significatives dans la consommation mondiale de streaming musical.

- Le premier **graphique circulaire** (voir graphique 1) présente la répartition des minutes d'écoute par pays, révélant que la France, le Japon, l'Allemagne et l'Inde représentent les marchés les plus actifs en termes de temps d'écoute. Il est cependant intéressant de relever que cette répartition est très homogène et qu'aucun pays ne se démarque particulièrement
- La première **heatmap** (voir graphique 2) confirme cette tendance tout en ajoutant une dimension géographique. On observe que certains pays comme l'Inde et le Brésil présentent une consommation

quotidienne particulièrement élevée, dépassant 500 minutes streamées par jour en moyenne. Le Canada suit également cette tendance avec une forte consommation. Cette répartition suggère que les pays émergents constituent actuellement les zones de plus forte croissance pour le streaming musical. On peut aussi noter que ce graphique est moins nuancé que le précédent

- Le deuxième **graphique circulaire** (voir graphique 3) représente les périodes de la journée comptabilisant le plus d'écoute (Matin, Après-midi, Nuit). Nous pouvons observer que les écoutes sont uniformes entre les trois périodes, avec une légère augmentation pour la nuit
- Le troisième **graphique circulaire** (voir graphique 4) représente quel style musical est le plus écouté dans le monde en proportion. Les styles dominants sont l'EDM, le Jazz ainsi que le Rock.
- la deuxième **heatmap** (voir graphique 5) représente l'accès à internet dans le monde. On constate que malgré leur appétence à utiliser les plateformes de streaming, des pays tels que l'Inde ou le Brésil ne sont pas les plus favorisés dans leur accès à internet. Cela traduit également une certaine pauvreté et précarité dans ces derniers pays.
- Ce graphique (voir graphique 6) présente le temps moyen d'écoute quotidien en minutes selon les tranches d'âge. Les résultats montrent une répartition relativement homogène : aucune tranche d'âge ne se démarque significativement en termes de consommation. Cette régularité suggère que l'engagement envers les plateformes de streaming est stable quel que soit l'âge, ce qui ouvre la voie à une stratégie tarifaire et fonctionnelle transversale, sans nécessité de différenciation marquée. Cependant, une offre modulable en fonction d'autres critères (type d'abonnement, plateforme préférée, genre musical) pourrait mieux capter les préférences individuelles au sein de chaque groupe.
- Ce graphique (voir graphique 7) en barres empilées met en évidence les préférences de plateformes de streaming selon les tranches d'âge. Il permet à OCTAVE d'identifier les segments démographiques sur lesquels les concurrents sont déjà très présents, et ceux qui sont sous-exploités. Cela ouvre la voie à une stratégie de différenciation ciblée par âge, avec un positionnement ajusté selon la génération (Z, milléniaux, etc.).
- Ici (voir graphique 8), on observe les préférences entre les types d'abonnement (gratuit ou premium) selon les tranches d'âge. Les données montrent une répartition relativement équilibrée, mais avec des préférences marquées pour certaines catégories d'abonnement chez les clients les plus âgés (60-64) ainsi que les plus jeunes (15-19). Ces données peuvent guider la conception d'offres adaptées : tarifs étudiants, formules familiales ou encore essais gratuits ciblés. Afin d'attirer des clients non premium vers une offre qui leur conviendrait.

Implications pour le projet OCTAVE

Cette analyse géographique a plusieurs implications importantes pour notre projet de plateforme de streaming musical :

Ciblage géographique stratégique :

- Nos efforts de lancement devraient se concentrer sur les marchés à forte consommation comme l'Inde, le Brésil et le Canada où les utilisateurs passent déjà beaucoup de temps sur les plateformes concurrentes.
- Adaptation culturelle : La forte présence de marchés asiatiques (Japon, Inde) et sud-américains (Brésil) dans les données suggère la nécessité d'une approche diversifiée en termes de catalogue musical et d'interface utilisateur.

- Potentiel inexploité : Certains pays européens comme la France montrent une forte présence dans le graphique circulaire mais une intensité moindre sur la carte thermique, ce qui pourrait indiquer un grand nombre d'utilisateurs avec des sessions d'écoute plus courtes c'est donc une opportunité pour accroître l'engagement.

Opportunités et recommandations

Sur la base de notre analyse, nous identifions plusieurs opportunités stratégiques pour le projet OCTAVE :

- Développement sur mesure par région : Créer des expériences utilisateur adaptées aux habitudes d'écoute spécifiques à chaque région. Par exemple, proposer des fonctionnalités de téléchargement hors-ligne dans les régions où la connectivité Internet peut être instable.
- Partenariats locaux : Établir des collaborations avec des artistes et labels locaux dans les marchés clés comme l'Inde et le Brésil pour développer un catalogue adapté aux préférences culturelles.
- Modèle d'abonnement flexible : Proposer différentes options tarifaires selon les caractéristiques économiques des marchés ciblés, avec des offres attractives pour les marchés émergents à fort potentiel (localisé pour la plus grande part dans des zones moins favorisées économiquement).
- Traduction adaptée : On sait grâce aux études des graphiques précédents que les publics visés seraient l'Inde, le Brésil, le Canada et plusieurs pays européens. Il est donc impératif que l'application soit traduite dans ces langues et puisse proposer des publicités adaptées pour la langue du pays.
- Chaque période de la journée est partiellement uniformément répartie au niveau de la densité d'écoute ce qui va nous permettre de vendre des spots publicitaires au même prix peu importe le moment de sa diffusion.
- Via les styles musicaux les plus écoutés que nous avons identifiés nous serons plus enclin à faire des collaborations commerciales avec des artistes appartenant à ces derniers pour que leur portée d'influence soit la plus importante possible.
- Comme le véhicule la dernière heatmap, certains de nos pays cibles ont des difficultés d'accès à Internet. Il faudra donc y proposer une option d'écoute hors connexion via le téléchargement des musiques et adapter les tarifs aux moyens des populations.

Conclusion

Cette analyse de marché démontre clairement le potentiel d'une nouvelle plateforme de streaming musical comme OCTAVE. La consommation de musique en streaming continue de croître à l'échelle mondiale, avec des disparités régionales significatives qui représentent autant d'opportunités d'adaptation et de différenciation. Les marchés émergents comme l'Inde et le Brésil, où la consommation quotidienne est particulièrement élevée, constituent des cibles prioritaires pour notre développement. Parallèlement, les marchés matures comme le Japon, la France et l'Allemagne offrent un potentiel important en termes de volume d'utilisateurs. OCTAVE devra se démarquer de concurrents établis comme SPOTIFY et DEEZER en proposant une expérience utilisateur personnalisée, un catalogue musical adapté aux préférences régionales et un modèle économique flexible qui répond aux spécificités de chaque marché.

Script python utilisé :

Création de la base de donnée

```
import pandas as pd
from sqlalchemy import create_engine, text, exc

df2 = pd.read_csv("Final.csv", encoding="latin-1").drop_duplicates()
df2 = df2.rename(columns={"Unnamed: 0": "index", "Entity": "Country"})
df2 = df2[df2['Year'].between(2018, 2024)]

df3 = pd.read_csv("GMLP.csv", encoding="latin-1").drop_duplicates()

df8 = df3.merge(df2, on="Country")
df8.to_csv("merged_result.csv", index=False)

df_user = df8[["User_ID", "Age", "Most Played Artist", "Minutes Streamed
Per Day", "Subscription Type", "Top Genre", "Streaming Platform",
"Code"]].drop_duplicates(subset=["User_ID"]).rename(columns={
    "User_ID": "user_id",
    "Age": "age",
    "Most Played Artist": "most_played_artist",
    "Minutes Streamed Per Day": "minutes_streamed_per_day",
    "Subscription Type": "subscription_type",
    "Top Genre": "top_genre",
    "Streaming Platform": "streaming_platform",
    "Code": "country_code"
})

df_country = df8[["Code", "Country"]].drop_duplicates().rename(columns={
    "Code": "code",
    "Country": "name"
})

df_statistics = df8[["Year", "Code", "Cellular Subscription", "No. of
Internet Users", "Internet Users(%)", "Broadband
Subscription"]].drop_duplicates().rename(columns={
    "Year": "year",
    "Code": "country_code",
    "Cellular Subscription": "cellular_subscription",
    "No. of Internet Users": "internet_users",
    "Internet Users(%)": "internet_users_pct",
    "Broadband Subscription": "broadband_subscription"
})

engine = create_engine("postgresql://nom:achanger@londres/dbnom")
```

```
try:
    with engine.begin() as conn:
        conn.execute(text("DROP TABLE IF EXISTS users, country,
statistics;"))

        conn.execute(text("""
            CREATE TABLE country (
                code VARCHAR(10) PRIMARY KEY,
                name VARCHAR(100) NOT NULL
            );
        """))

        conn.execute(text("""
            CREATE TABLE users (
                user_id VARCHAR(10) PRIMARY KEY,
                age INTEGER NOT NULL,
                most_played_artist VARCHAR(100),
                minutes_streamed_per_day INTEGER,
                subscription_type VARCHAR(50),
                top_genre VARCHAR(50),
                streaming_platform VARCHAR(50),
                country_code VARCHAR(10),
                FOREIGN KEY (country_code) REFERENCES
countries(country_code)
            );
        """))

        conn.execute(text("""
            CREATE TABLE statistics (
                id SERIAL PRIMARY KEY,
                year INTEGER NOT NULL,
                country_code VARCHAR(10) NOT NULL,
                cellular_subscription NUMERIC,
                internet_users INTEGER,
                internet_users_pct NUMERIC(5,2),
                broadband_subscription NUMERIC,
                FOREIGN KEY (country_code) REFERENCES
countries(country_code)
            );
        """))

        df_country.to_sql('country', engine, if_exists='append',
index=False)
        df_user.to_sql('users', engine, if_exists='append', index=False)
        df_statistics.to_sql('statistics', engine, if_exists='append',
index=False)

        print("Tables créées et données chargées avec succès.")

except exc.SQLAlchemyError as e:
    print("Une erreur est survenue lors de la connexion ou l'insertion
:", e)
```

Creation des graphiques

fichier 1 :

```
import pandas as pnd
import statsMaker as sm

dico_pays={}

for row in df.itertuples():
    if row.pays in dico_pays:
        dico_pays[row.pays] += row.Minute_jour
    else:
        dico_pays[row.pays] = row.Minute_jour

l_Pays = []
l_minutes = []

for pays,minutes in dico_pays.items():
    l_Pays.append(pays)
    l_minutes.append(minutes)

sm.pieChart(l_minutes,l_Pays,"Minutes d'écoutes par pays")

world = gpd.read_file("world.geojson")

df2 = pd.read_csv('music-streaming-stats.csv',sep=',')
df3 = pd.read_csv('Final.csv',sep=',')

data = {
    "NAME": df2["Country"],
    "value": df2["Minutes Streamed Per Day"]
}
df = pd.DataFrame(data)
df = df.drop_duplicates(subset=["NAME"])

world = world.rename(columns={"ADMIN": "country"})

merged = world.merge(df, left_on="NAME", right_on="NAME", how="left")

fig, ax = plt.subplots(figsize=(15, 10))

merged.plot(
    column="value",
    cmap="OrRd", # 'hot', 'YlGnBu', etc.
    linewidth=0.8,
    edgecolor="0.8",
    legend=True,
    missing_kwds={
```



```

        "color": "lightgrey", # Color for countries with no data
        "label": "No Data"
    },
    ax=ax
)

plt.title("Minutes Streamed Per Day", fontsize=16)
plt.axis("off")
plt.show()

```

fichier 2 :

```

import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import create_engine, text, exc
import geopandas as gpd

df2 = pd.read_csv("Final.csv", encoding="latin-1").drop_duplicates()
df2 = df2.rename(columns={"Unnamed: 0": "index", "Entity": "Country"})
df2 = df2[df2['Year'].between(2018, 2024)]

df3 = pd.read_csv("GMLP.csv", encoding="latin-1").drop_duplicates()

df8 = df3.merge(df2, on="Country")
df8.to_csv("merged_result.csv", index=False)

df_user = df8[["User_ID", "Age", "Most Played Artist", "Minutes Streamed
Per Day", "Subscription Type", "Top Genre", "Streaming Platform",
"Code"]].drop_duplicates(subset=["User_ID"]).rename(columns={
    "User_ID": "user_id",
    "Age": "age",
    "Most Played Artist": "most_played_artist",
    "Minutes Streamed Per Day": "minutes_streamed_per_day",
    "Subscription Type": "subscription_type",
    "Top Genre": "top_genre",
    "Streaming Platform": "streaming_platform",
    "Code": "country_code"
})

df_country = df8[["Code", "Country"]].drop_duplicates().rename(columns={
    "Code": "code",
    "Country": "name"
})

df_statistics = df8[["Year", "Code", "Cellular Subscription", "No. of
Internet Users", "Internet Users(%)", "Broadband
Subscription"]].drop_duplicates().rename(columns={
    "Year": "year",
    "Code": "country",
    "Cellular Subscription": "cellular_subscription",

```

```

        "No. of Internet Users": "internet_users",
        "Internet Users(%)": "internet_users_pct",
        "Broadband Subscription": "broadband_subscription"
    })

# Connexion PostgreSQL
engine = create_engine("postgresql://etveau:achanger@londres/dbetveau")

try:
    with engine.connect() as con:
        graph1 = pd.read_sql(text("""
            SELECT c.name, u1.top_genre
            FROM country c
            JOIN users u1 ON u1.country = c.code
            WHERE (
                SELECT COUNT(*)
                FROM users u2
                WHERE u2.country = c.code
                AND u2.top_genre = u1.top_genre
            ) >= ALL (
                SELECT COUNT(*)
                FROM users u3
                WHERE u3.country = c.code
                GROUP BY u3.top_genre
            )
            GROUP BY c.name, u1.top_genre;

        """), con=con)

        style_counts = graph1['top_genre'].value_counts()

        plt.figure(figsize=(8, 8))
        plt.pie(style_counts, labels=style_counts.index, autopct='%1.1f%%',
startangle=140)
        plt.title("proportion Style musicale mondiale ")
        plt.axis('equal')
        plt.show()

except exc.SQLAlchemyError as e:
    print(f"Erreur SQLAlchemy : {e}")

world = gpd.read_file("world.geojson")
world = world.rename(columns={"NAME": "country"})
try:
    with engine.connect() as con:
        df_broadband = pd.read_sql(text("""
            SELECT c.name AS country, AVG(s.broadband_subscription) AS
value
            FROM statistics s
            JOIN country c ON s.country = c.code
            GROUP BY c.name;

        """), con=con)

```

```

merged = world.merge(df_broadband, on="country", how="left")

fig, ax = plt.subplots(figsize=(15, 10))
merged.plot(
    column="value",
    cmap="Purples",
    linewidth=0.8,
    edgecolor="0.8",
    legend=True,
    missing_kwds={
        "color": "lightgrey",
        "label": "Aucune donnée"
    },
    ax=ax
)

plt.title("Abonnements Internet haut débit par pays (moyenne 2018-
2024)", fontsize=16)
plt.axis("off")
plt.show()

except exc.SQLAlchemyError as e:
    print(f"Erreur SQLAlchemy : {e}")

try:
    with engine.connect() as con:
        df_age_platform = pd.read_sql(text("""
            SELECT
                FLOOR(u.age / 5) * 5 AS age_group_start,
                u.streaming_platform,
                COUNT(*) AS count
            FROM users u
            WHERE u.age IS NOT NULL AND u.age >= 0
            GROUP BY age_group_start, u.streaming_platform
            ORDER BY age_group_start, streaming_platform;
        """), con=con)

        df_age_platform["age_group"] =
df_age_platform["age_group_start"].astype(int).astype(str) + "-" +
(df_age_platform["age_group_start"] + 4).astype(str)

        pivot_df = df_age_platform.pivot(index="age_group",
columns="streaming_platform", values="count").fillna(0)

        pivot_df.plot(kind="bar", stacked=True, figsize=(14, 8))
        plt.title("Préférence de plateforme de streaming selon les tranches
d'âge (5 ans)", fontsize=16)
        plt.xlabel("Tranche d'âge")
        plt.ylabel("Nombre d'utilisateurs")
        plt.xticks(rotation=45)
        plt.legend(title="Plateforme")
        plt.tight_layout()
        plt.show()

```

```

except exc.SQLAlchemyError as e:
    print(f"Erreur SQLAlchemy (plateformes par tranche d'âge) : {e}")
try:
    with engine.connect() as con:

        df_avg_minutes = pd.read_sql(text("""
            SELECT
                FLOOR(u.age / 5) * 5 AS age_group_start,
                AVG(u.minutes_streamed_per_day) AS avg_minutes
            FROM users u
            WHERE u.age IS NOT NULL AND u.minutes_streamed_per_day IS NOT
NULL
            GROUP BY age_group_start
            ORDER BY age_group_start;
        """), con=con)

        df_avg_minutes["age_group"] =
df_avg_minutes["age_group_start"].astype(int).astype(str) + "-" +
(df_avg_minutes["age_group_start"] + 4).astype(str)

        plt.figure(figsize=(12, 6))
        plt.bar(df_avg_minutes["age_group"], df_avg_minutes["avg_minutes"],
color="skyblue")
        plt.title("Temps moyen d'écoute quotidien par tranche d'âge",
fontsize=16)
        plt.xlabel("Tranche d'âge")
        plt.ylabel("Minutes écoutées par jour (moyenne)")
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()

except exc.SQLAlchemyError as e:
    print(f"Erreur SQLAlchemy (moyenne minutes d'écoute) : {e}")

try:
    with engine.connect() as con:
        df_subscription = pd.read_sql(text("""
            SELECT
                FLOOR(u.age / 5) * 5 AS age_group_start,
                u.subscription_type,
                COUNT(*) AS count
            FROM users u
            WHERE u.age IS NOT NULL AND u.subscription_type IS NOT NULL
            GROUP BY age_group_start, u.subscription_type
            ORDER BY age_group_start, u.subscription_type;
        """), con=con)

        df_subscription["age_group"] =
df_subscription["age_group_start"].astype(str) + "-" +
(df_subscription["age_group_start"] + 4).astype(str)

```

```
    pivot = df_subscription.pivot(index="age_group",
columns="subscription_type", values="count").fillna(0)
    pivot_pct = pivot.div(pivot.sum(axis=1), axis=0) * 100 # en
pourcentage

    ax = pivot_pct.plot(kind="bar", stacked=True, figsize=(14, 8),
colormap="Set2")

    for p in ax.patches:
        height = p.get_height()
        width = p.get_width()
        x = p.get_x() + width / 2
        y = p.get_y() + height / 2
        percentage = f'{height:.1f}%'
        ax.text(x, y, percentage, ha='center', va='center', fontsize=10,
color='white')

    plt.title("Proportion des types d'abonnement par tranche d'âge",
fontsize=16)
    plt.xlabel("Tranche d'âge")
    plt.ylabel("Pourcentage d'utilisateurs")
    plt.legend(title="Type d'abonnement")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

except exc.SQLAlchemyError as e:
    print(f"Erreur SQLAlchemy (proportions des abonnements) : {e}")
```