Projet de Travail d'Été 2025

Constantin KEUKY

ESILV

28 juillet 2025

Contexte

À la suite du jury d'appel du 24 juillet 2025, l'année scolaire de Constantin KEUKY a été validée sous condition de réalisation d'un travail d'été à rendre avant le **29 août 2025**. Ce projet vise à consolider les compétences en **programmation orientée objet (POO)** en **C++**, appliquées à une base de données réelle.

Objectif du projet

L'objectif est d'analyser des données musicales issues de la plateforme Spotify, à travers un programme C++ structuré autour de classes. Le projet devra permettre de :

- Concevoir une architecture modulaire en POO,
- Implémenter manuellement des calculs statistiques (sans bibliothèques),
- Explorer, modéliser et interpréter les données à l'aide d'outils statistiques fondamentaux et intermédiaires.

Données utilisées

La base provient du site Kaggle :
`https://www.kaggle.com/datasets/meeratif/spotify-most-streamed-artists-of-all-time`

Elle contient les colonnes suivantes :

— **Artist** : nom de l'artiste.

— **Streams** : nombre total de lectures (streams) sur Spotify.

— **Daily** : moyenne quotidienne de streams.

— **As lead** : nombre de streams en tant qu'artiste principal.

— **Solo** : nombre de streams pour les projets solos.

— **As feature** : nombre de streams en tant qu'artiste invité sur des collaborations.

Chaque ligne représente un · e artiste. Ce jeu de données permet d'explorer des dynamiques de popularité, de performance individuelle et de collaboration.

Contraintes techniques

— Le projet doit être réalisé exclusivement en **langage C++**.

— L'usage de bibliothèques externes (comme Boost, Armadillo, etc.) est **interdit**.

— Tous les calculs statistiques devront être codés manuellement (moyenne, variance, régression...).

— Le fichier CSV devra être traité à l'aide de `ifstream` ou de classes créées spécifiquement pour parser les données.

— L'architecture du projet devra respecter les principes fondamentaux de la **programmation orientée objet**, tels que l'encapsulation, la modularité, et la réutilisabilité.

— Le programme devra proposer un **menu interactif** permettant à l'utilisateur de sélectionner les opérations statistiques à exécuter (moyenne, tests, régression, etc.). Les résultats devront être affichés à l'écran.

— Une option devra également permettre de **sauvegarder les résultats statistiques** dans un fichier de sortie (format texte).

Pour toute question sur la conception orientée objet en C++, l'étudiant peut se référer au projet de POO C++ présenté au cours de l'année, qui expose les bonnes pratiques attendues (structures de classes, fichiers séparés, méthodes d'accès, etc.).

Analyses statistiques attendues

Le projet devra inclure plusieurs des traitements suivants :

1. **Statistiques descriptives** :
   — Moyenne, médiane et mode (ex : des streams ou des streams quotidiens),
   — Valeurs minimale et maximale,
   — Écart-type, variance, amplitude.
2. **Classements et distributions** :

— Top 10 des artistes selon différentes métriques (streams totaux, solo, lead...),
— Répartition du poids des streams solos par rapport aux collaborations,
— Artistes avec le plus grand écart entre « as lead » et « as feature ».

3. **Probabilités élémentaires** :
   — Probabilité de tirer au hasard un · e artiste faisant partie du top 10 selon les streams,
   — Probabilité qu'un · e artiste ait plus de 70% de ses streams en solo,
   — Probabilité conditionnelle (ex : si un artiste a plus de 100M de streams, quelle est la proba qu'il soit dans le top 10 daily ?).
4. **Estimation statistique** :
   — Estimation ponctuelle de la moyenne des streams solo,
   — Intervalle de confiance pour la moyenne ou la proportion (ex : proportion d'artistes ayant plus de X millions de streams en feature),
   — Estimation de la proportion d'artistes « très actifs » selon un critère choisi.
5. **Tests statistiques simples** :
   — Test de moyenne (comparaison entre artistes solo et feature),
   — Test de proportion (ex : proportion d'artistes ayant plus de 1M de daily streams),
   — Test d'écart entre deux groupes (selon genre musical si disponible).
6. **Régression linéaire simple (Modèle Gaussien)** :
   — Régression entre le nombre de streams totaux et le nombre de streams solo ou « as lead »,
   — Hypothèse : relation linéaire + bruit gaussien (erreurs centrées, indépendantes),
   — Affichage de la droite d'ajustement et analyse des résidus.
7. **Corrélations et interprétation** :
   — Analyse des liens entre solo et feature : les artistes les plus collaboratifs sont-ils aussi les plus populaires ?
   — Existe-t-il des profils typiques selon la répartition des types de streams ?

Toutes les formules statistiques devront être codées manuellement en C++, sans bibliothèques externes. Le rapport devra expliciter les hypothèses, justifier les méthodes utilisées, et interpréter les résultats obtenus.

Livrables attendus

1. Un **rapport écrit** (10 pages), incluant :

   - une description des classes C++ conçues,
   - la méthode de traitement du fichier CSV,
   - les calculs effectués (formules, justifications, résultats),
   - l'interprétation des analyses.
2. Le **code source complet et commenté**, structuré de manière propre.
3. Une **présentation orale de 20 minutes** à effectuer à la rentrée.

Modalités de suivi

- Point d'étape facultatif autour du **18 août 2025**.
- Date de rendu final : **29 août 2025**.

Encadrant : Daniel Wladdimiro
Contact : daniel.wladdimiro@devinci.fr