Projet de traitement des données

Travail sur la base de données des Jeux Olympiques

Tual Groix
Khalid Jerrari
Gabriel Lecointre
Teodora Moldovan

7 mai 2025

Sommaire

1 Introduction

Présentation de l'évènement : les Jeux Olympiques modernes.

En juin 1894 à Paris a lieu une semaine de congrès sur le sport qui se conclut par la création du Comité International des Jeux Olympiques (CIJO, actuel CIO) avec des membres de 13 nations (toutes européennes à l'exception des États-Unis) et la décision de ce même comité de ressusciter les jeux olympiques de l'Antiquité. La première édition des Jeux Olympiques modernes se tient à Athènes en Grèce en 1896 puis une nouvelle édition tous les quatre ans sauf années de guerre mondiale depuis (éditions de 1916, de 1940 et de 1944 annulées) avec de plus en plus de délégations nationales représentant maintenant la quasi totalité des pays de la planète. Ces Jeux sont qualifiés de Jeux d'été. À partir de 1924, se tiennent les années olympiques des Jeux d'hiver pour les sports se concourant sur neige ou sur glace. Ces Jeux d'hiver se disputent en 1994 deux ans après la précédente édition et ont lieu depuis les années paires sans Jeux d'été.

Chaque édition se tient dans une ville différente, choisie par le Comité International Olympique (CIO), et inclut des dizaines de disciplines sportives variées, allant de l'athlétisme à la natation, en passant par la gymnastique, le judo ou encore le basketball. Les Jeux Olympiques représentent non seulement un évènement sportif de haut niveau, mais aussi un évènement culturel et diplomatique majeur, favorisant les échanges entre les peuples.

2 Présentation du jeu de données

Le jeu de données analysé concerne les participations d'athlètes aux différentes éditions des Jeux Olympiques modernes, qu'il s'agisse des Jeux d'été ou d'hiver.

Les données sont regroupées dans deux tableaux, un tableau sommaire associant aux nations les codes utilisés par le CIO pour chacune d'elles et un deuxième tableau plus conséquent, comprenant 271 116 enregistrements répartis sur 15 variables, représentant la participation d'un athlète à au moins une épreuve olympique.

Le tableau de données compile un historique des résultats pour toutes les éditions des Jeux Olympiques (été et hiver) depuis les premiers Jeux modernes jusqu'aux Jeux d'été de 2016. Sont inclus dans le tableau de données les résultats des Jeux Olympiques d'Athènes de 1906 (actuellement non reconnus comme officiels par le CIO), ceux des compétitions artistiques aux Jeux Olympiques existantes de 1912 à 1948 et les médailles honorifiques en aéronautisme et en alpinisme entre 1924 et 1932 (mais absents des tableaux de médailles publiés en ligne par le Mouvement olympique pour ces mêmes Jeux).

Chaque ligne est structuré de la manière suivante :

— une ligne par sportif et par épreuve, ainsi un sportif participant à plusieurs éditions des Jeux Olympiques, par exemple à 4 épreuves à chacun des deux Jeux où il a concouru sera représenté par 8 lignes distinctes ;

— sur chaque ligne, il y a un identifiant unique pour chaque sportif (en cas d'homonymie), le nom du sportif, son sexe, son âge (si connu), son poids (si connu), sa taille (si connue), son « équipe »[1], le code CIO de sa délégation, l'édition des Jeux Olympiques concernée, l'année, la saison (hiver ou été), la ville d'accueil des Jeux, le sport, l'épreuve et enfin dans la dernière colonne la couleur de la médaille (vide si le sportif a participé sans terminer sur le podium).

Voici un extrait de la base de données :

1. C'est « équipe » vraiment au sens large, sont par exemple parfois mentionnés à la place des délégations, le nom des bateaux, des équipages ou des clubs dans les épreuves de voiles ou d'aviron ou le nom des chevaux dans les épreuves d'équitation ; on retrouve aussi le nom de la délégation suivi d'un numéro dans des épreuves sportives où il peut y avoir plusieurs équipes de la même nationalité. Devant la difficulté de traitement de ces informations très variées et devant le fait que cela fait doublon avec la colonne voisine avec le code de la délégation, nous avons décidé d'écarter cette donnée de nos analyses.

TABLE 1 – Extrait des 5 premières observations de la base de données

ID	1	2	3	4	5
Name	A Dijiang	A Lamusi	Gunnar Nielsen Aaby	Lindenau Aabye	Christine Jacoba Aaftink
Sex	M	M	M	M	F
Age	24	23	24	34	21
Height	180	170			185
Weight	80	60			82
Team	China	China	Denmark	Sweden	Netherlands
NOC	CHN	CHN	DEN	DEN	NED
Games	1992 Summer	2012 Summer	1920 Summer	1900 Summer	1988 Winter
Year	1992	2012	1920	1900	1988
Season	Summer	Summer	Summer	Summer	Winter
City	Barcelona	London	Antwerpen	Paris	Calgary
Sport	Basketball	Judo	Football	Tug-Of-War	Speed Skating
Event	Men's Basketball	Extra-Lightweight	Football Men's Football	Men's Tug-Of-War	Women's 500 metres
Medal				Gold	

Type des variables et valeurs manquantes

TABLE 2 – Type des variables, valeurs manquantes et valeurs uniques

Colonne	Type	Valeurs non nulles	Valeurs manquantes	Valeurs uniques
ID	int64	271 116	0	135 571
Name	object	271 116	0	134 732
Sex	object	271 116	0	2
Age	float64	261 642	9 474	74
Height	float64	210 945	60 171	95
Weight	float64	208 241	62 875	220
Team	object	271 116	0	1 184
NOC	object	271 116	0	230
Games	object	271 116	0	51
Year	int64	271 116	0	35
Season	object	271 116	0	2
City	object	271 116	0	42
Sport	object	271 116	0	66
Event	object	271 116	0	765
Medal	object	39 783	231 333	3

Les valeurs uniques représentent le nombre de modalités pour les variables qualitative ou de valeur différentes pour les variables quantitatives.

Structure du jeu de données

Le tableau comporte les colonnes suivantes :

— **ID** : identifiant unique attribué à chaque enregistrement ;

— **Name** : nom complet de l'athlète ;

— **Sex** : sexe de l'athlète (M pour masculin, F pour féminin) ;

— **Age** : âge de l'athlète lors de l'événement ;

— **Height** : taille (en centimètres) ;
— **Weight** : poids (en kilogrammes) ;
— **Team** : équipe ou pays[1] ;
— **NOC** : code du Comité National Olympique ;
— **Games** : nom complet de l'édition des Jeux (année + saison) ;
— **Year** : année des Jeux ;
— **Season** : saison des Jeux (Summer ou Winter) ;
— **City** : ville hôte ;
— **Sport** : discipline sportive ;
— **Event** : épreuve précise ;
— **Medal** : médaille obtenue (Gold, Silver, Bronze ou NA si aucune).

3 Organisation du code en différents modules

Le projet est conçu de manière modulaire et structurée, avec plusieurs répertoires clairement définis, chacun correspondant à une tâche spécifique du traitement de données. À la racine du dossier principal, nommé `code_traitement_donnees/` se trouvent quatre répertoires principaux : `donnees/`, `programmes/`, `src/` et `output/`.

— Le répertoire `donnees/` contient deux fichiers de données brutes au format CSV, conservés dans leur état original, conformément aux exigences du projet : `athlete_events.csv` et `noc_regions.csv`. Ces fichiers regroupent les informations relatives aux Jeux olympiques et servent de base aux différents traitements et analyses réalisés dans les scripts du projet.

— Le répertoire `programmes/` contient deux sous-répertoires : `classification/` et `questions/` :

  — Le répertoire `classification/` contient plusieurs modules, chacun dédié à une étape spécifique du processus de classification par K-Means.

    — Le module `main.py` sert de point d'entrée principal pour exécuter l'ensemble du processus de classification, en orchestrant l'utilisation des autres modules.

    — Le fichier `distance_euclidienne.py` calcule la distance euclidienne entre chaque individu et les barycentres des clusters à chaque itération.

    — Le module `initialisation_kmean.py` s'occupe de l'initialisation aléatoire des barycentres, en sélectionnant des individus du jeu de données comme points de départ pour les clusters.

    — Ensuite, `renormalisation.py` applique une normalisation des données à l'aide de la méthode du z-score, ce qui assure que toutes les variables sont comparables et contribuent de manière équitable à l'algorithme.

    — Les données sont ensuite réduites à une dimension plus faible grâce au module `reduction_dimension.py` qui utilise l'Analyse en Composantes Principales (PCA) pour préserver la variance tout en simplifiant la structure des données, facilitant ainsi leur visualisation et le travail de l'algorithme K-Means.

    — Pour déterminer le nombre optimal de clusters, le module `critere_coude.py` utilise la méthode du coude, un aspect essentiel pour évaluer la qualité de la classification..

    — Enfin, le module `test_kmeans.py` permet de comparer l'implémentation personnalisée de K-Means avec celle de scikit-learn, afin de valider la précision et la cohérence des résultats obtenus par l'algorithme.

  Ces différents modules collaborent pour effectuer une série d'étapes, incluant le prétraitement, la normalisation, la réduction de dimension, l'application de l'algorithme K-Means, et la validation des résultats. L'ensemble garantit la pertinence et la fiabilité des classifications produites.

  — Le sous-répertoire `questions/` contient plusieurs sous-dossiers organisés en fonction des types d'analyse en deux sous-répertoires :

    — `questions_basepython/`

    — `questions_pandapython/`

Ces sous-répertoires regroupent des scripts Python destinés à traiter les questions d'analyse de données, en utilisant respectivement la programmation de base et la bibliothèque Pandas. Ces scripts abordent différentes analyses, telles que le calcul des statistiques, la manipulation des données et la visualisation des résultats.

— Le répertoire `src/` contient le code source principal de l'application, incluant le fichier main.py, ainsi que deux sous-répertoires :

  — `test/`

  — `interface_graphique`

  Ces sous-répertoires organisent le code en modules spécifiques, chacun ayant une fonction bien définie, ce qui permet de structurer le projet de manière claire et modulaire.

  Le fichier main.py dans ce répertoire constitue le point d'entrée principal du programme, où l'exécution du projet est initiée. Ce script lance une interface graphique permettant à l'utilisateur de sélectionner parmi 12 questions, et d'afficher les réponses correspondantes directement sur l'interface.
  Utilisant la bibliothèque Tkinter, l'application crée une interface interactive qui permet à l'utilisateur de choisir une question via un menu déroulant. En fonction de la question sélectionnée, l'application ouvre l'interface associée en appelant des fonctions spécifiques à chaque question. L'interface comporte également un label qui affiche la question choisie, ainsi qu'un bouton "Quitter" pour fermer l'application. Ce programme est conçu de manière extensible, avec des interfaces dédiées à chaque question, facilitant ainsi une navigation fluide entre les différentes sections. L'utilisation de Tkinter et de fonctions modulaires pour chaque question permet une organisation claire du code, tout en offrant une expérience utilisateur intuitive et agréable.

— Le sous-répertoire `tests/` contient des scripts de test permettant de comparer les performances de différentes solutions mises en œuvre.
  Ces tests comparent notamment les temps d'exécution d'un même code, une fois implémenté en Python pur (sans bibliothèques externes) et une autre fois en utilisant le package Pandas. Les scripts visent à comparer les temps d'exécution de deux fonctions qui accomplissent des tâches similaires mais adoptent des approches différentes : l'une utilisant du code Python standard, l'autre utilisant la bibliothèque Pandas. Les tests se concentrent sur deux paires de fonctions, correspondant à deux questions distinctes : `question1_basepython` vs `question1_pandapython` et `question2_basepython` vs `question2_pandapython`. Pour chaque question, le script compare les temps d'exécution des versions basepython (une version simple utilisant uniquement Python) et pandapython (une version tirant parti de Pandas).

Les programmes présents dans ce répertoire sont principalement axés sur l'interface graphique et l'analyse des données. Chaque question de l'application dispose d'une interface interactive développée avec la bibliothèque Tkinter, offrant à l'utilisateur la possibilité de choisir des options, d'entrer des données et de consulter les résultats.

Chaque question est traitée par un module séparé, garantissant ainsi une structure modulaire et facilement extensible. L'application importe les données nécessaires depuis des fichiers CSV, puis applique des traitements spécifiques à chaque question en s'appuyant sur les programmes du modèle `questions_pandapython` (tels que le comptage des médailles par pays ou l'analyse des performances par genre) pour générer des résultats sous forme de tableaux ou autres formats appropriés. En outre, elle inclut des fonctionnalités permettant à l'utilisateur de sauvegarder les résultats sous forme de fichiers CSV. En somme, chaque programme est conçu pour traiter une question spécifique tout en offrant une interface claire et intuitive, facilitant l'interaction avec les données.

Les données générées par les programmes seront enregistrées dans le répertoire `output/`.

4 Présentation des questions posées, des réponses apportées et des méthodes utilisées pour obtenir ces réponses

4.1 Questions imposées et réponses

4.1.1 Question 1 : déterminer le nombre de médailles remportées par Michael Phelps. Nom complet : Michael Fred Phelps, II

Afin de répondre à cette question, nous avons commencé par examiner la structure des noms dans la base de données. Dans ce jeu de données, le nom de chaque athlète est enregistré sous forme d'une concaténation du prénom et du nom, ce qui peut poser des problèmes de cohérence (variations de casse, de caractère de séparation, d'espacement ou de ponctuation). Nous avons donc vérifié la présence de variantes du nom Michael Fred dans l'ensemble des données. L'exploration des données a montré qu'un seul athlète correspondait exactement au nom Michael Fred Phelps, II, ce qui limite à la fois le risque d'oubli d'une de ses participations et celui de confusion avec un éventuel homonyme.

Pour renforcer cette vérification, nous avons également pris en compte deux autres variables disponibles dans la base : l'âge et la nationalité (NOC). Ces éléments nous ont permis de confirmer que toutes les occurrences renvoyaient bien au même individu. En l'absence d'identifiant unique (comme un numéro de sécurité sociale), cette combinaison d'attributs fait office de clé d'identification approximative.

Nous avons ensuite isolé les variables pertinentes pour l'analyse, à savoir : "Year", "Name", "Event" et "Medal". Si seules les colonnes "Name" et "Medal" suffisent à comptabiliser les médailles, conserver l'année et l'épreuve permet d'effectuer des vérifications complémentaires, notamment pour identifier d'éventuels doublons. En effet, un athlète ne peut recevoir qu'une seule médaille par épreuve lors d'une édition donnée des Jeux Olympiques.

Une vérification des modalités de la variable "Medal" a confirmé la cohérence des données :

— Gold pour une médaille d'or
— Silver pour une médaille d'argent
— Bronze pour une médaille de bronze
— NA indique qu'aucune médaille n'a été remportée

Nous avons filtré toutes les lignes correspondant à Michael Fred Phelps, II et compté les occurrences pour chacune des trois modalités de médaille (en excluant les NA). Le total obtenu est de 28 médailles.

Pour valider notre résultat, nous avons consulté la fiche officielle de Michael Phelps sur le site du Comité International Olympique : `https://www.olympics.com/en/athletes/michael-phelps-ii`

Notre analyse corresponde parfaitement aux données officielles : Michael Phelps a remporté 23 médailles d'or, 3 d'argent et 2 de bronze, soit 28 médailles olympiques au total.

4.1.2 Question 2 : Trouver des bornes inférieure et supérieure pour le nombre de médailles par nation aux Jeux Olympiques de 2016

Afin de répondre à cette question, nous avons formulé une démarche analytique visant à encadrer le nombre de médailles remportées par chaque nation lors des Jeux Olympiques de 2016 à l'aide de deux bornes : une borne inférieure et une borne supérieure. L'objectif était de refléter la réalité des résultats en tenant compte des spécificités du format olympique, notamment les épreuves collectives où plusieurs athlètes peuvent recevoir une médaille pour une seule victoire d'équipe. Pour ce faire, nous avons d'abord exploré le jeu de données `athlete_events.csv`, dans lequel chaque ligne correspond à la participation d'un athlète à une épreuve donnée. Les variables essentielles à cette analyse sont : *Team, Year, Event et Medal*.

Nous avons développé une fonction `compter_medailles_par_pays`, paramétrable par année et par nation(s), afin de systématiser cette analyse. Pour répondre à la question, il suffit de filtrer les données par l'année 2016.

La borne supérieure repose sur le total brut des médailles attribuées dans les données, c'est-à-dire en comptant chaque ligne où figure une médaille, y compris les doublons induits par les médailles d'équipe (ex. : tous les membres d'un relais reçoivent chacun une médaille). À l'inverse, la borne inférieure vise à estimer le nombre minimal de médailles effectivement remises en éliminant les doublons par triplet (Team, Event, Medal). Cette méthode considère donc qu'une équipe ne peut recevoir qu'une seule médaille par épreuve et type de médaille, ce qui permet de corriger les surestimations dans le cas des sports collectifs.

Une fois les totaux par nation établis, nous avons renommé et organisé les colonnes pour distinguer clairement les résultats selon les deux bornes : `Gold_sup`, `Silver_sup`, `Bronze_sup`, `Total_sup` pour la borne supérieure et `Gold_inf`, `Silver_inf`, `Bronze_inf`, `Total_inf` pour la borne inférieure. Les résultats ont été triés en fonction de la borne supérieure afin de refléter l'ordre usuel des classements olympiques, puis exportés dans un fichier CSV dont le nom est généré dynamiquement en fonction de l'année et des pays analysés (ex. : `resultats_medailles_2016_all_countries.csv`).

Cette méthode fournit ainsi une vision nuancée et précise des performances olympiques par pays, tout en laissant la possibilité de généraliser l'analyse à d'autres années ou groupes de nations. En complément, le fichier exporté peut être utilisé pour créer des visualisations ou alimenter d'autres rapports. Cette démarche répond directement à la question posée tout en fournissant un outil flexible pour des analyses futures.

À l'issue de notre traitement sur les données des Jeux Olympiques de 2016, nous avons obtenu des bornes inférieure et supérieure pour le nombre de médailles remportées par les principales nations. Ces bornes permettent d'encadrer le nombre réel de médailles physiques obtenues par pays, en tenant compte des spécificités des épreuves collectives (où plusieurs athlètes peuvent recevoir une médaille pour une même victoire). Par exemple, pour les nations ayant remporté le plus de médailles, telles que les États-Unis, la borne inférieure est de 117 médailles (sans compter les doublons liés aux épreuves collectives), tandis que la borne supérieure atteint 256 médailles (en comptabilisant chaque médaille attribuée à un athlète, même en cas de doublons pour les épreuves par équipe). L'écart important entre ces deux valeurs met en évidence l'impact majeur des épreuves collectives dans leur total. De même, pour l'Allemagne, le nombre de médailles passe de 41 à 157 selon la méthode de comptage, ce qui reflète une forte présence dans les disciplines collectives. La Grande-Bretagne (67 à 145), la Russie (55 à 113) et la Chine (68 à 109) présentent également des écarts significatifs, bien que moins marqués. Ces résultats soulignent que le simple total brut des médailles peut être trompeur si l'on ne distingue pas les performances individuelles des contributions collectives. Ainsi, en fournissant à la fois une estimation minimale (borne inférieure) et maximale (borne supérieure), notre approche permet d'obtenir une évaluation plus juste et transparente des performances olympiques nationales.

Le graphique généré par le programme Python, sous forme d'histogramme, illustre ces écarts pour les cinq pays ayant obtenu le plus de médailles en 2016. Il permet de visualiser non seulement le classement des nations, mais aussi l'ampleur possible de la variation dans leurs résultats, en comparant les bornes minimale et maximale. Cette représentation offre une lecture plus équilibrée et met en lumière les incertitudes liées à la nature des épreuves, tout en confirmant la robustesse globale des classements.

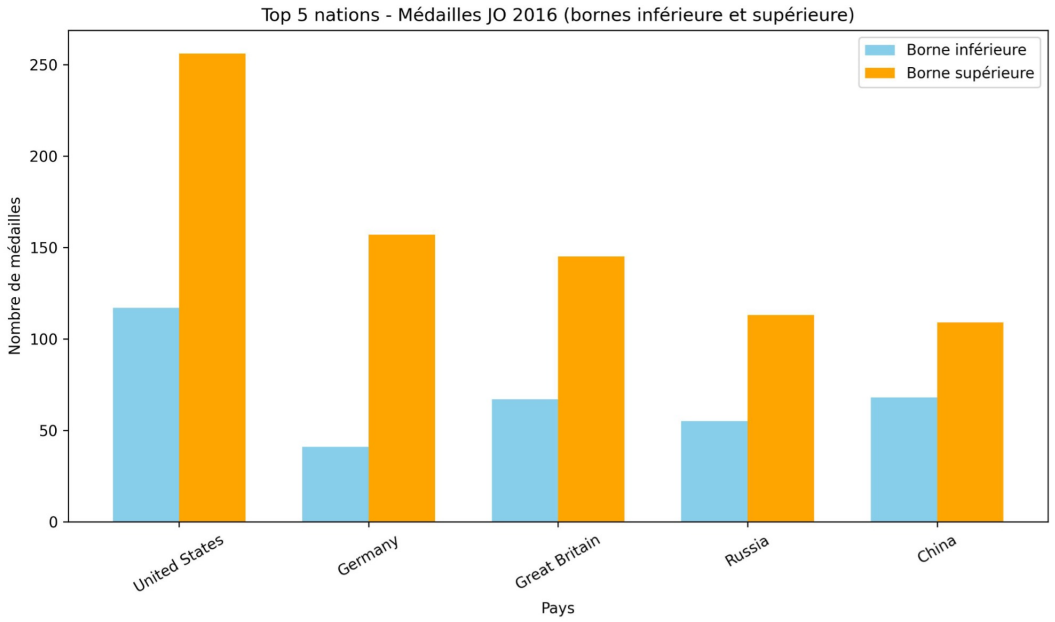

FIGURE 1 – Les bornes inférieures et supérieures des cinq premières nations médaillées aux JO 2016

4.2 Questions sélectionnées et réponses

4.2.1 Question 3 : Combien d'athlètes ont remporté des médailles aux Jeux olympiques d'hiver et d'été ? Combien en ont remporté la même année ?

Avant 1924, années de création des JO d'hiver, les disciplines d'hiver étaient incluses dans les JO d'été. C'était le cas pour l'année 1920. En conséquence, la comparaison commence à partir de 1924.

Nous avons répondu à cette question à la fois avec pandas et en Python pur. L'approche pandas utilise la bibliothèque pandas pour la manipulation des données, qui fournit des structures de données et des fonctions adaptées à l'analyse des données. L'approche purement Python utilise uniquement les fonctionnalités intégrées de Python (listes, dictionnaires, ensembles) et le module csv pour les opérations sur les fichiers.

Algorithme pour répondre à la question

1. **Chargement des données**
   - **Pandas** : charge les données directement dans un `DataFrame`.
   - **Python pur** : lit manuellement le fichier CSV ligne par ligne en utilisant le module `csv`.
2. **Traitement des données**
   A. **Filtrer la base de données pour sélectionner uniquement les participants médaillés**
      - **Pandas** : utilise l'indexation booléenne avec `donnees[donnees["Medal"].notnull()]` pour filtrer les lignes avec des médailles.
      - **Python pur** : utilise une compréhension de liste pour filtrer les lignes où la colonne `Medal` n'est pas `None` ou `"NA"`.

   B. **Grouper les données par ID de l'athlète**
      - **Pandas** : utilise la fonction `groupby()` pour créer des groupes par ID d'athlète.
      - **Python pur** : crée manuellement un dictionnaire pour regrouper les données par ID.
3. **Identifier les athlètes ayant remporté des médailles à la fois aux JO d'été et aux JO d'hiver**
   - **Avec Pandas** : pour chaque athlète, on vérifie s'il a participé à plus d'une saison olympique (été et hiver). Si un nom de fichier Excel est fourni, la fonction exporte ces données dans ce fichier.
   - **En Python pur** :
     - On crée un dictionnaire `athletes_par_id` où chaque clé est un ID, et chaque valeur est une liste des lignes correspondantes.
     - On extrait les saisons uniques (position 10 dans chaque ligne).
     - On collecte les sports pratiqués par saison (position 12) et on les stocke dans des ensembles pour comparer.
     - On vérifie que l'athlète a participé à plus d'une saison et que les sports d'été et d'hiver n'ont aucune intersection.
   - La fonction retourne une liste de dictionnaires contenant les informations complètes sur les athlètes ayant participé aux deux types de Jeux dans des disciplines différentes.

Comparaison temps d'exécutions :

- Le temps d'exécution du script en Python pur est de : 32.1542 s
- Le temps d'exécution du script en Pandas est de : 1.8703 s

Sans affichage des résultats (avant le print) :

- Le temps d'exécution du script en Python pur est de : 23.9564 s
- Le temps d'exécution du script en Pandas est de : 1.6330 s

Contrairement aux attentes, le temps d'exécution en utilisant la bibliothèque `Pandas` est plus long qu'avec `Python` pur.

Résultat obtenu (comme affiché dans la console) :

Athlètes ayant remporté des médailles dans différentes saisons

— **ID : 31167**
Nom : Edward Patrick Francis “Eddie” Eagan
Années : [[1920], [1932]]

— **ID : 50859**
Nom : Clara Hughes
Années : [[1996, 1996], [2002, 2006, 2006, 2010]]

— **ID : 102862**
Nom : Christa Rothenburger-Luding
Années : [[1988], [1984, 1988, 1988, 1992]]

— **ID : 119489**
Nom : Jacob Tullin Thams
Années : [[1936], [1924]]

— **ID : 130626**
Nom : Lauryn Chenet Williams
Années : [[2004, 2012], [2014]]

Athlètes ayant remporté des médailles dans différentes saisons la même année

— **ID : 102862**
Nom : Christa Rothenburger-Luding
Années : [[1988], [1984, 1988, 1988, 1992]]
Années communes : [1988]

4.2.2 Question 4 : Quelle a été la performance détaillée de la Roumanie lors des JO de 1984 ?

Le code en Python sert à analyser et corriger les données de médailles des Jeux Olympiques, puis à produire un classement des pays en fonction des médailles pour une année donnée (1984 pour répondre à la question).

La fonction `correct_medal_counts` a pour objectif de corriger les anomalies dans les données de médailles lorsqu'il y a plus de 6 médailles attribuées pour un même événement (ce qui peut arriver à cause d'ex-aequo ou dans les épreuves collecrtives).

Afin de corriger le nombre de médailles, nous avons considéré les deux situations suivantes :

— si le nombre total de médailles (or, argent, bronze) attribuées est supérieur ou égal à 6, nous considértons qu'il s'agit d'une épreuve collective (tous les athlètes de l'équipe reçoit une médaille). Dans ce cas, on garde uniquement la première occurrence de chaque type de médaille (or, argent, bronze).

— si le nombre total de médailles (or, argent, bronze) attribuées est inférieur à 6, on garde toutes les lignes. Dans ce cas on considère que l'épreuve est individuelle, avec éventuellement des cas ex-aequo.

Toutes les lignes corrigées sont concaténées en un seul DataFrame.

La fonction `classement_jo` a pour objectif de créer un classement des pays pour une année donnée des Jeux Olympiques d'été, en tenant compte du nombre de médailles obtenues.

Les données sont filtrées pour garder seulement les lignes correspondant à l'année donnée et aux Jeux d'été. Les doublons sont corrigés en utilisant la fonction `correct_medal_counts`.

Pour compter le nombre de médailles par pays, on crée un tableau croisé (`pivot_table`) qui compte les médailles par type (or, argent, bronze) pour chaque pays.

Le programme retourne le classement suivant (affichage dans l'interface) pour 1984 :
USA : Or : 83, Argent : 61, Bronze : 30, Total : 174 médailles
Romania : Or : 20, Argent : 16, Bronze : 17, Total : 53 médailles
Germany : Or : 17, Argent : 19, Bronze : 23, Total : 59 médailles
China : Or : 15, Argent : 8, Bronze : 9, Total : 32 médailles
Italy : Or : 14, Argent : 6, Bronze : 12, Total : 32 médailles
Canada : Or : 10, Argent : 18, Bronze : 16, Total : 44 médailles

Japan : Or : 10, Argent : 8, Bronze : 14, Total : 32 médailles
New Zealand : Or : 8, Argent : 1, Bronze : 2, Total : 11 médailles
Serbia : Or : 7, Argent : 4, Bronze : 7, Total : 18 médailles
South Korea : Or : 6, Argent : 6, Bronze : 7, Total : 19 médailles

Le résultat obtenu correspond au classement du CIO `https://www.olympics.com/fr/olympic-games/los-angeles-1984/medals` : en 1984 la Roumanie a terminé deuxième au classement des médailles avec 20 médailles d'or, 16 d'argent et 17 de bronze.

4.2.3 Question 5 : Quels ont été les résultats spécifiques de Nadia Comăneci lors des JO de 1976 ?

On peut répondre de manière assez simple à cette question, sans implémenter une fonnction. L'utilisateur est demandé d'entrer le nom d'un sportif (ou une partie du nom) et l'année des Jeux Olympiques.

A partir de ces paramètres d'entrée, on convertit le nom en minuscules pour faire une recherche insensible à la casse, et on fait une recherche partielle, c'est à dire on cherche toutes les lignes où le nom du sportif contient le texte saisi, peu importe la casse.

On sauvegarde les résultats dans un fichier Excel et on affiche à l'écran le résultat obtenu :
Entrez le nom du sportif pour afficher ses résultats : Nadia Elena
Entrez l'année des Jeux Olympiques : 1976

Résultats pour Nadia Elena Comneci (-Conner) en 1976 :

- **Épreuve :** Gymnastics Women's Individual All-Around
  Médaille : Gold
- **Épreuve :** Gymnastics Women's Team All-Around
  Médaille : Silver
- **Épreuve :** Gymnastics Women's Floor Exercise
  Médaille : Bronze
- **Épreuve :** Gymnastics Women's Horse Vault
  Médaille : `nan`
- **Épreuve :** Gymnastics Women's Uneven Bars
  Médaille : Gold
- **Épreuve :** Gymnastics Women's Balance Beam
  Médaille : Gold

Le résultat obtenu correspond à la performance de l'athlète affichée sur le site du CIO : `https://www.olympics.com/fr/athletes/nadia-comaneci`.

4.2.4 Question 6a : combien de sportifs ont changé de délégation (hors délégation neutre) ?

On commence par ne conserver que les variables utiles (ID, Year, NOC, Sport) et exclure les athlètes neutres. On compte ensuite le nombre de modalités de délégation (NOC) par sportif, et ensuite on ne conserve que les sportifs ayant concouru pour plus d'une délégation. Le résultat est ensuite exporté avec les résultats des questions 6b et 6c dans un fichier csv. Avec pandas, la réponse à ces trois questions est faite par le biais d'une fonction. Avec python de base, le programme n'est pas dans une fonction. Le nombre de sportifs ayant participé à plusieurs délégations est de 1537.

4.2.5 Question 6b : dans combien de sports se sont faits ces changements de délégation ?

A la suite de la question 6a (dans le même programme), on sélectionne les sports pour lesquels ont concouru les sportifs qui ont changé de délégation puis on compte les différents sports. Le résultat est ensuite exporté avec les résultats des questions 6b et 6c dans un fichier csv. Avec pandas, la réponse à ces trois questions est faite par le biais d'une fonction. Avec python de base, le programme n'est pas dans une fonction. Le nombre total de modalités de sports concernées par ces changements de délégation est de 47.

4.2.6 Question 6c : dans quel sport y a-t-il eu le le plus de sportifs qui ont changé de délégation ?

A la suite des questions 6a et 6b (dans le même programme), on sélectionne le sport avec le plus de sportifs différents qui ont changé de délégation ainsi que le nombre de sportifs correspondant. Le résultat est ensuite exporté avec les résultats des questions 6a et 6b dans un fichier csv. Avec pandas, la réponse à ces trois questions est faite par le biais d'une fonction. Avec python de base, le programme n'est pas dans une fonction. Le sport avec le plus de sportifs changeant de délégation est Athletics (athlétisme) avec 862 sportifs différents concernés.

4.2.7 Question 7 : comment la taille et le poids des sportifs varient-ils selon les sports et le sexe ?

Vigilance à avoir sur les données :

— la taille et le poids sont parmi les informations de la base de données avec le plus de données manquantes. Soit 22 % et 23 % de lignes avec une information vide pour respectivement la taille et le poids. Plus les éditions des Jeux sont anciennes, plus ces données sont manquantes. On a décidé ici de ne traiter que les données existantes sans estimer les valeurs des données manquantes ;

— attention à la pertinence de certains résultats dans le tableau final qu'affiche le programme informatique car certains sports ne figurant plus au menu des Jeux Olympiques les plus récents disposent que de peu d'éléments pour effectuer les calculs. Néanmoins le tableau final des résultats donne une bonne idée globale des corpulences moyennes par sexe et sport.

Choix effectués :

— les mêmes choix ont été faits pour la taille et le poids mais les deux variables ont été traitées séparément ;

— calculer pour chaque sport et chaque sexe une moyenne de la taille, respectivement du poids si l'on a au moins l'information de la taille, respectivement du poids pour au moins un sportif de ce sport et de ce sexe ;

— un même sportif peut participer à plusieurs sports, à plusieurs épreuves dans le même sport et à plusieurs éditions des Jeux : pour un même sportif, les données n'ont été conservées qu'une fois par sport et par Jeux. Ainsi, les informations concernant ce sportif ne seront comptabilisées qu'à une reprise et non quatre fois si un nageur participe à quatre courses différentes lors de la même édition des Jeux Olympiques ; par contre, elles seront comptabilisées une deuxième fois s'il participe à des épreuves de natation aussi aux Jeux suivants ou une deuxième fois pour un autre sport s'il participe aussi à un autre sport comme le water-polo que cela soit aux mêmes Jeux ou à des Jeux différents.

Algorithme pour répondre à la question :

1. Avant le chargement des données, on importe `pandas` et `os` et on écrit une option sur `pandas` pour afficher l'entièreté des résultats dans le terminal (pour que ne s'affichent pas uniquement les premières et dernières lignes des tableaux finaux demandés).
2. On charge les données que l'on enregistre dans un tableau utilisant `pandas`.
3. Dans le tableau de données, on supprime toutes les colonnes inutiles pour notre calcul et pour chaque sportif pour un sport et une année donnés, on ne garde qu'une seule ligne.
4. On réalise enfin une moyenne sur un `groupby` effectué sur le sexe et le sport et on trie ces moyennes par ordre croissant.
5. On affiche dans le terminal et on enregistre dans des fichiers externes les tableaux de résultats.

Résultats :

Voici un aperçu partiel des résultats pour le premier tableau, celui de la taille ; cela donne la taille moyenne (en centimètres) pour un sport et un sexe donnés par ordre croissant (les espaces entre les colonnes sont ici représentés par des points) :

```
.................................Height
Sport.....................Sex............
Gymnastics................F....155.992259
Art Competitions..........F....160.000000
Weightlifting.............F....160.467391
Figure Skating............F....160.636476
```

```
Diving.....................F....161.278447
Trampolining...............F....161.733333
Wrestling..................F....163.865132
```
pour le début du tableau puis pour la fin du tableau :
```
Tennis.....................M....184.673854
Water Polo.................M....186.801739
Rowing.....................M....186.959108
Handball...................M....188.778373
Volleyball.................M....193.265660
Beach Volleyball...........M....193.290909
Basketball.................M....194.872624
```

De même pour le poids (en kilogrammes) :
```
.................................Weight
Sport.....................Sex...........
Gymnastics.................F....47.705338
Rhythmic Gymnastics........F....48.760976
Figure Skating.............F....49.883941
Ski Jumping................F....52.615385
Trampolining...............F....52.893333
Diving.....................F....53.651226
Triathlon..................F....54.724138
```
pour le début du tableau puis pour la fin du tableau :
```
Water Polo.................M....87.706172
Handball...................M....89.387914
Beach Volleyball...........M....89.512821
Bobsleigh..................M....90.264045
Rugby Sevens...............M....91.006623
Basketball.................M....91.683529
Tug-Of-War.................M....95.615385
```

Il n'y a pas de source en ligne ayant effectué ce même travail mais le résultat semble cohérent par rapport aux deux sexes et aux capacités physiques demandées par chaque discipline. Les premières valeurs des deux tableaux sont essentiellement les représentantes féminines dans des sports requérant de la souplesse (gymnastique et gymnastique rythmique, patinage artistique, plongeon, trampoline), de la légèreté (saut à ski) ou ayant des catégories de poids (haltérophilie ou lutte) ; on peut écarter les compétitions artistiques qui n'est pas une compétition sportive et qui, de plus, comportent trop peu de données pour avoir une valeur significative. De la même façon, les dernières valeurs sont essentiellement les représentants masculins dans des sports requérant force (tir à la corde, aviron, water-polo, rugby, bobsleigh) ou grande taille (basket, volley et beach-volley) ou les deux.

4.2.8 Question 8 : comment évolue le nombre de sportifs par sport au fil des différentes éditions ?

Vigilance à avoir sur les données :

— à part les deux ou trois premières éditions des Jeux Olympiques (1896, 1900, 1904), les données sur quel sportif a participé à quelles épreuves sont connues dans leur quasi totalité. Il peut donc y avoir des résultats qui diffèrent de quelques unités ou un peu plus pour un sport donné sur les premières éditions.

Choix effectués :

— un même sportif peut participer à plusieurs sports, à plusieurs épreuves dans le même sport et à plusieurs éditions des Jeux : un même sportif n'est compté qu'une fois maximum par sport et par Jeux. Ainsi, un sportif ne sera comptabilisé qu'à une reprise et non à quatre s'il participe à quatre courses différentes en natation lors de la même édition des Jeux Olympiques ; par contre, il sera comptabilisé une autre fois s'il participe à des épreuves de natation aux Jeux suivants ou une deuxième fois pour un autre sport s'il participe aussi à un autre sport comme le water-polo que cela soit aux mêmes Jeux ou à des Jeux différents ;

— pour une meilleure visualisation des résultats, en plus d'un tableau final, j'affiche un graphique avec un certain nombre de sports ;

— cet affichage graphique est scindé en trois parties pour montrer les 24 sports les plus pratiqués dans l'histoire des Jeux ; les regrouper sur un seul et même graphique était trop fouilli, j'ai ainsi mis les 8 premiers sports sur un graphique, les 8 suivants sur un deuxième ; et les 8 encore suivants sur un troisième ;

— l'affichage avec un nuage de points a été préféré à des segments reliant chaque point pour deux raisons : la première, c'était que 1906 (Jeux intercalaires, cf. la présentation de la base de données) ou les Jeux annulés en 1916, 1940 et 1944 (cf. la même présentation) étaient mal gérés, ainsi que l'espacement exceptionnel de seulement deux ans entre les Jeux d'hiver de 1992 et de 1994 ; la deuxième, c'était pour les sports qui disparaissaient un temps du programme olympique pour réapparaître plusieurs éditions après, les années sans ce sport étaient aussi mal gérées (par exemple, le hand, absent entre 1936 et 1972, avait le point de 1936 relié par un segment unique au point de 1972).

Algorithme pour répondre à la question :

1. Avant le chargement des données, on importe `pandas`, `os` et `matplotlib.pyplot` (ce dernier pour l'affichage graphique) et on écrit une option sur `pandas` pour afficher l'entièreté des résultats dans le terminal (pour que ne s'affichent pas uniquement les premières et dernières lignes des tableaux finaux demandés).
2. On charge les données que l'on enregistre dans un tableau utilisant `pandas`.
3. Dans le tableau de données, on supprime toutes les colonnes inutiles pour notre calcul et pour chaque sportif pour un sport et une année donnés, on ne garde qu'une seule ligne.
4. On compte le nombre de lignes et donc de sportifs pour un sport et une année donnés. On classe par ordre décroissant.
5. On affiche dans le terminal et on enregistre dans des fichiers externes les tableaux de résultats. On somme également le nombre de sportifs cumulé (un sportif peut ainsi être compté plusieurs fois s'il a participé à plusieurs Jeux) pour chaque sport. On affiche un graphique avec les 8 premiers sports (selon cette somme), puis les 8 suivants et encore un troisième graphique avec les 8 encore suivants.

Résultats :

Voici un aperçu partiel des résultats affichés dans le tableau :

```
Sport.....................Year........
Aeronautics...............1936.......1
Alpinism..................1936.......2
.........................1932.......2
Basque Pelota.............1900.......2
Roque.....................1904.......4
Weightlifting.............1904.......5
Diving....................1904.......5
```

pour le début du tableau puis pour la fin du tableau :

```
Rowing....................1976.....593
.........................1988.....593
.........................1996.....608
.........................1992.....627
Swimming..................1988.....633
.........................1992.....641
Athletics.................1924.....659
.........................1928.....706
.........................1956.....720
.........................1948.....745
Swimming..................1996.....762
Athletics.................1936.....776
Swimming..................2012.....931
.........................2004.....937
.........................2016.....942
.........................2000.....954
```

```
Athletics.................1980.....960
..........................1952.....963
..........................1976....1006
..........................1960....1016
..........................1964....1018
Swimming..................2008....1022
Athletics.................1968....1029
..........................1984....1280
..........................1972....1330
..........................1988....1618
..........................1992....1726
..........................2004....1995
..........................2008....2056
..........................1996....2057
..........................2012....2079
..........................2000....2137
..........................2016....2269
```

Notez que quand une ligne est vide de toute mention de disciplines sportives, c'est simplement la dernière discipline sportive qui est répétée avec une nouvelle année, le dernier chiffre correspondant au nombre de sportifs pour l'année indiquée. Les premières lignes correspondent soit à des premières éditions des Jeux (avec des participants manquants ou très peu de participants) soit à des médailles honorifiques (aéronautisme et alpinisme).

On peut retrouver ces données par exemple dans :

— https://en.wikipedia.org/wiki/Aeronautics at the 1936 Summer Olympics ;

— https://en.wikipedia.org/wiki/Alpinism at the Olympic Games.

Les sports qui accueillent les plus de participants sont l'aviron, la natation et l'athlétisme, et généralement dans chaque sport, il y a une tendance à un accroissement du nombre de participants (il y a quand même des exceptions avec l'aviron qui a atteint son maximum dans les années 1990). On retrouve les chiffres de particpants dans les pages Wikipédia des disciplines concernées pour une année précise. Il y a quand même quelques unités de différence qui s'explique par des inscrits qui n'ont pas participé le jour de la compétition. Exemple avec l'athlétisme en 2016 : https://fr.wikipedia.org/wiki/Athlétisme aux Jeux olympiques d'été de 2016 qui comptabilise 2283 athlètes contre 2269 ici, les 14 unités de différence correspondent à des DNS (*did not start*) le jour de l'épreuve comme Bailey et Filimone sur le 100m ou Francis et Mokamba Nyang'au sur le 200.

Pour les graphiques qui s'affichent, on obtient par exemple ici le premier graphique :

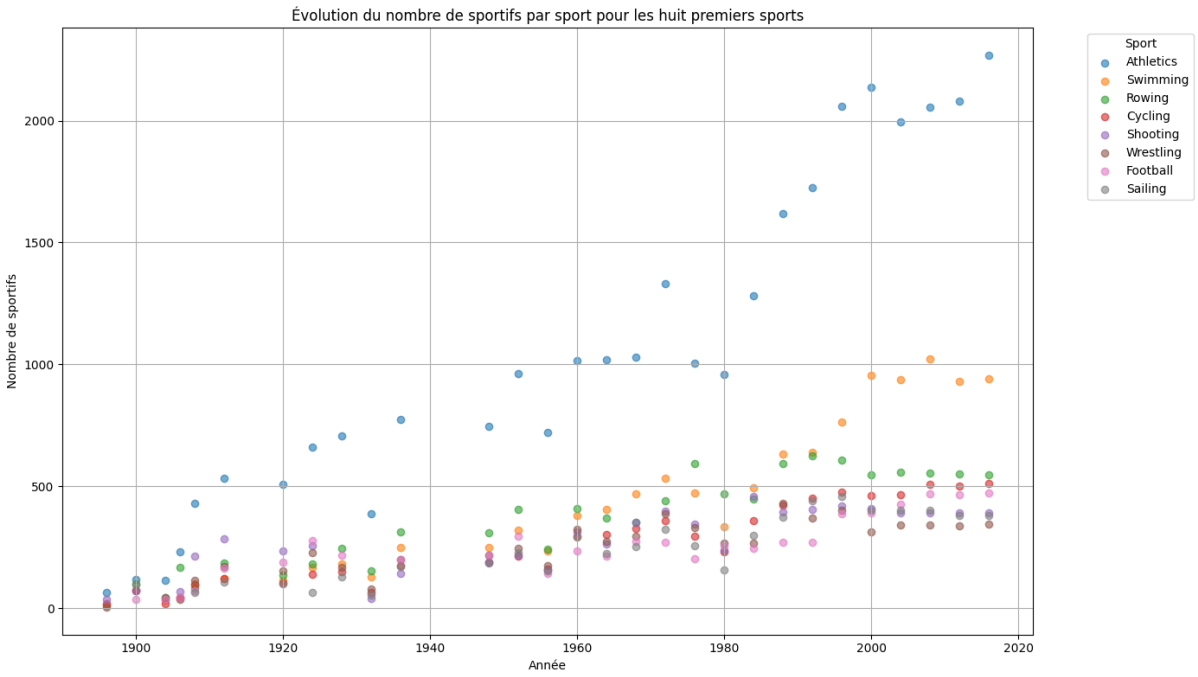

FIGURE 2 – Évolution du nombre de sportifs dans les huit sports avec le plus de représentants aux Jeux Olympiques modernes de 1896 à 2016

4.2.9 Question 9 : pour chaque pays, donner le nombre d'éditions différentes dans laquelle au moins un membre de sa délégation a été médaillé.

Cette question a été traitée en `pandas` et en Python pur.

Vigilance à avoir sur les données et choix effectués :

— la base de données comprend les Jeux d'été de 1906 (actuellement non reconnus comme officiels), on choisit de garder les données de 1906 en les considérant comme officielles ;

— le tableau des codes des pays est imparfait car 1. SGP pour *Singapore* est manquant, 2. HKG pour Hong Kong a été affecté à *China* alors qu'il s'agit de deux délégations différentes, 3. WIF pour Indes occidentales a été affecté à *Trinidad* mais *Jamaica* est plus logique car plus peuplée et les médailles sont remportées par des sportifs jamaïquains, 4. GDR pour Allemagne de l'Est a été affecté à *Germany* or *East Germany* est plus logique car période où Allemagne de l'Ouest a aussi sa délégation. Ces 4 élements font l'objet de corrrections directement dans le corps du programme ;

— on regroupe les pays suivants (ou leur prolongement historique) qui ont eu des médailles sous des codes différents selon les périodes, le nom de pays renvoyé par le tableau est identique et les données peuvent être sommées : ANZ = Australasie (soit Australie + Nouvelle-Zélande) affecté à *Australia* ; BOH et TCH = Bohême et Tchécoslovaquie (soit République tchèque et Slovaquie) affecté à *Czech Republic* (entité principale démographique) ; YUG et SCG = Yougoslavie (soit 7 pays) et Serbie-et-Monténégro (soit Serbie et Monténégro) affecté à *Serbia* (entité démographique principale) ; URS et EUN = Union soviétique (soit 15 pays, Jeux de 1988 et avant) et Équipe unifiée (12 pays, Jeux de 1992) affecté à *Russia* (son entité centrale) ; FRG = Allemagne de l'Ouest affecté à *Germany* ; et enfin WIF = Indes occidentales affecté à *Jamaica*.

Algorithmes :

1. En `pandas`, avant le chargement des données, on importe `pandas` et `os` et on écrit une option sur `pandas` pour afficher l'entièreté des résultats dans le terminal (pour que ne s'affichent pas uniquement les premières et dernières lignes des tableaux finaux demandés). En Python pur, on ne fait qu'importer `csv` et `os`
2. On charge les données que l'on enregistre dans deux tableaux utilisant `pandas` dans le premier cas, on use d'un `os.path.join` pour les enregistrer dans le deuxième cas.

3. Dans le tableau de données `pandas`, on supprime toutes les colonnes inutiles pour notre calcul, ainsi que toutes les lignes sans médailles. Puis, on ne garde qu'une seule donnée par pays pour une édition donnée. On somme le nombre total de lignes pour chaque code de pays puis via un `groupby` avec la saison, on somme pour chaque code de pays, le nombre de Jeux d'été et le nombre de Jeux d'hiver. En Python pur, je n'effectue aucune suppression de données, je commence par regarder la concordance entre codes et pays avant de passer directement à l'étape suivante.
4. Dans les deux cas, on fait ensuite des correctifs comme indiqué dans la section précédente (sur Singapour, Hong Kong, les Indes occidentales et l'Allemagne de l'Est), on remplace les valeurs manquantes par des 0 (zéro) et pour une meilleure esthétiques, on transforme les flottants par des entiers.
5. En Python pur, j'effectue des boucles dans les données en ignorant les lignes sans médailles, je crée des listes total et pour été et hiver dès que je croise une médaille pour une édition pour un pays non encore médaillé. En `pandas`, je n'effectue rien à cette étape.
6. Dans les deux cas, on regroupe ensuite comme indiqué aussi dans la section précédente les pays selon leur prolongement historique.
7. En Python pur, on regarde ensuite la longueur de chaque liste pour déterminer le nombre de Jeux total et de chaque type pour chaque pays.
8. Dans les deux cas, on crée une ligne TOTAL qui calcule le nombre d'éditions des Jeux de chaque type.
9. Dans les deux cas, on classe enfin les pays par ordre décroissant puis à nombre égal par ordre alphabétique.
10. Dans les cas, on fait ensuite l'affichage, en `pandas`, l'affichage se fait d'un seul bloc ; en Python pur, ligne par ligne.

En `pandas`, le début du tableau :

```
.........................Nb JO..Nb JO E..Nb JO H
délégation.....................................
TOTAL.........................51.......29.......22
France........................50.......29.......21
USA...........................50.......28.......22
Canada........................49.......27.......22
Sweden........................49.......27.......22
Switzerland...................49.......28.......21
Austria.......................48.......26.......22
Finland.......................48.......26.......22
Norway........................47.......25.......22
Germany.......................45.......25.......20
Italy.........................45.......27.......18
```

En Python pur, le même début de tableau :

```
Délégation.....................| Nb JO | Nb JO E | Nb JO H
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
TOTAL.........................|....51 |......29 |......22
France........................|....50 |......29 |......21
USA...........................|....50 |......28 |......22
Canada........................|....49 |......27 |......22
Sweden........................|....49 |......27 |......22
Switzerland...................|....49 |......28 |......21
Austria.......................|....48 |......26 |......22
Finland.......................|....48 |......26 |......22
Norway........................|....47 |......25 |......22
Germany.......................|....45 |......25 |......20
Italy.........................|....45 |......27 |......18
```

Les résultats trouvés sont les mêmes avec les deux méthodes. La France a effectivement obtenu au moins une médaille à chaque édition des Jeux Olympiques sauf une fois à des Jeux d'hiver où le meilleur classement d'un Français a été quatrième (donc hors podium). Les États-Unis ont été médaillés à tous

les Jeux auxquels ils ont participé et ils ont participé à tous les Jeux à l'exception des Jeux d'été de Moscou en 1980 qu'ils ont boycottés.

Après avoir lancé plusieurs fois les deux programmes, on arrive à un temps de traitement entre 0,17 et 0,25 secondes en Python pur et entre 1,2 et 2,3 secondes en `pandas`.

5 Apprentissage automatique

5.1 Énoncé de la problématique

L'objectif de cette étude est de comprendre comment les caractéristiques physiques des athlètes (taille, poids, âge, etc.) influencent la formation des groupes d'épreuves sportives aux Jeux Olympiques, et d'analyser l'évolution de cette structuration au cours du dernier siècle.

La question centrale est donc la suivante : Existe-t-il des regroupements naturels d'épreuves sportives selon les caractéristiques physiques des athlètes, et ces regroupements ont-ils évolué historiquement ?

5.2 Les variables retenues

Pour répondre à notre problématique, nous considérons l'épreuve sportive comme unité statistique. Afin de construire notre base d'étude, nous procédons à une agrégation de la table initiale, dont l'unité est l'athlète, en regroupant les données par épreuve. Ci-dessous les variables que nous avions retenues lors de la définition de notre problèmatique :

Variables de la classification	**Variables de la base utilisée**
Individu statistique	Event
nombre d'épreuves dans le sport associé	Sport
part des femmes participantes	Sex
nombre de participants	ID
nombre de pays participant	ID et NOC
âge moyen des participants	Age
écart type de l'âge des athlètes	Age
poids moyen des athlètes	Weight
écart type du poids des athlètes	Weight
taille moyen des athlètes	Height
écart type de la taille des athlètes	Height
année d'apparition de l'épreuve	Year
type d'épreuve (individuel, collectif)	exogène à la base de données

Table 3 – Correspondance entre variables de classification et variables de la base de données

Le k-means ne prenant en compte que les variables quantitatives, nous aurions pu convertir la variable « type d'épreuve » en attribuant la valeur 1 pour une épreuve collective et 0 pour une épreuve individuelle. Toutefois, une telle transformation aurait pu introduire un biais dans les résultats. Nous avons donc choisi d'exclure cette variable de l'analyse.

Avant de procéder à la classification, il est nécessaire de réduire la dimensionnalité des données afin d'obtenir une représentation en deux dimensions. Pour cela, nous appliquerons une analyse en composantes principales (ACP) dans le cas de variables quantitatives. Nous souhaitions évaluer l'impact d'une forte dispersion de certaines variables, notamment la taille, le poids et l'âge, sur la composition des clusters. L'idée initiale était d'intégrer les écarts types de ces variables. Cependant, cette approche n'est pas adéquate, car les variables et leurs écarts types présentent des corrélations quadratiques, ce qui empêche de les considérer comme indépendantes.

Finalement, nous avons retenu les variables suivantes :

— Nombre moyen de participants

— Part de femmes
— Âge moyen
— Poids moyen
— Taille moyenne

5.3 L'algorithme

Afin d'identifier des regroupements d'épreuves sportives fondés sur les caractéristiques physiques des athlètes, nous avons recours à une méthode de classification non supervisée. L'objectif est de former des groupes homogènes d'épreuves présentant des profils physiques similaires, sans connaissance préalable des catégories.

Nous utilisons l'algorithme k-means, qui est particulièrement adapté à ce type d'analyse. Il s'agit d'une méthode de partitionnement qui divise un ensemble d'observations en k groupes de manière à minimiser la variance intra-groupe.

L'algorithme k-means commence par la fixation d'un nombre de classes k, défini à l'avance par l'utilisateur. Ce paramètre correspond au nombre de groupes que l'on souhaite identifier dans les données. L'algorithme sélectionne ensuite aléatoirement k individus parmi l'ensemble des données, qui servent de barycentres initiaux. Ces barycentres représentent les centres provisoires des futurs groupes. Dans un premier temps, il calcule la distance euclidienne entre chaque individu et chacun des barycentres.

Le nombre de participants pourrait, en théorie, être utilisé pour pondérer la distance euclidienne entre épreuves lors de la classification, mais cela ne nous paraît pas pertinent au regard de notre problématique centrée sur les caractéristiques physiques des athlètes. Pour la distance euclidienne nous considérons que chaque individu statistique a le même poids. Chaque individu est alors affecté au groupe dont le barycentre est le plus proche.

Une fois tous les individus répartis, l'algorithme calcule un nouveau barycentre pour chaque groupe, correspondant à la moyenne des coordonnées des individus qui le composent. Le processus est ensuite itératif : on recalcule les distances entre les individus et les nouveaux barycentres, on réaffecte les individus en fonction de ces distances, puis on met à jour les barycentres. Ce cycle est répété jusqu'à atteindre la convergence, c'est-à-dire lorsque les barycentres n'évoluent plus de manière significative ou que la variation de l'inertie intra-classe entre deux itérations devient inférieure à un seuil fixé. En structurant les données autour de barycentres stables, l'algorithme k-means permet de constituer des groupes homogènes d'épreuves sportives, définis selon les caractéristiques physiques moyennes des athlètes, ce qui est parfaitement en lien avec notre problématique.

Afin de rendre les variables comparables entre elles, nous procédons à une normalisation des données en utilisant leur moyenne et leur écart-type. Dans le cadre de notre analyse, nous avons implémenté notre propre version de l'algorithme *K-Means*, nécessitant cette étape de normalisation pour garantir un traitement équitable des différentes variables.

Cette normalisation, ainsi que l'algorithme de regroupement, sont également pris en charge par la fonction `kmeans` du module `sklearn`. Une comparaison entre notre implémentation et celle de la bibliothèque a montré une concordance parfaite des résultats. Pour des raisons de performance et de robustesse, nous avons opté pour l'utilisation de la version de `sklearn`, plus rapide et optimisée.

Pour l'analyse de l'ensemble des Jeux Olympiques disponibles dans la base de données, nous avons utilisé notre propre implémentation, tandis que pour l'analyse par période, nous avons privilégié la version proposée par la bibliothèque `sklearn`, via la fonction `kmeans`.

Enfin, avant l'application de l'algorithme *K-Means*, nous réduisons la dimensionnalité des données à deux composantes principales à l'aide de l'Analyse en Composantes Principales (ACP), afin de faciliter la visualisation des clusters.

L'algorithme du code :

Algorithme de classification par k-moyennes (k-means)

Entrée :
Données numériques D (n individus, m variables)
Nombre de classes k (fixé par l'utilisateur)
Seuil de convergence ε (petite valeur)

Initialisation :
Choisir aléatoirement k individus dans D comme barycentres initiaux $C_1, C_2, \ldots, C_k$

Répéter :

1. Pour chaque individu x dans D :
   — Calculer la distance euclidienne entre x et chaque barycentre C_j
   — Affecter x au groupe G_j correspondant au barycentre le plus proche
2. Pour chaque groupe G_j :
   — Recalculer le barycentre C_j comme la moyenne des individus de G_j
3. Calculer l'inertie intra-classe (somme des distances de chaque individu à son barycentre)

Jusqu'à :

— Les barycentres ne changent presque plus (variation $< \varepsilon$)
— OU la diminution de l'inertie intra-classe entre deux itérations est inférieure à ε

Sortie :
Répartition finale des individus en k groupes
Coordonnées finales des barycentres

5.4 Solution pour l'ensemble des JO

Nous obtenons cette projection en deux dimensions de nos clusters :

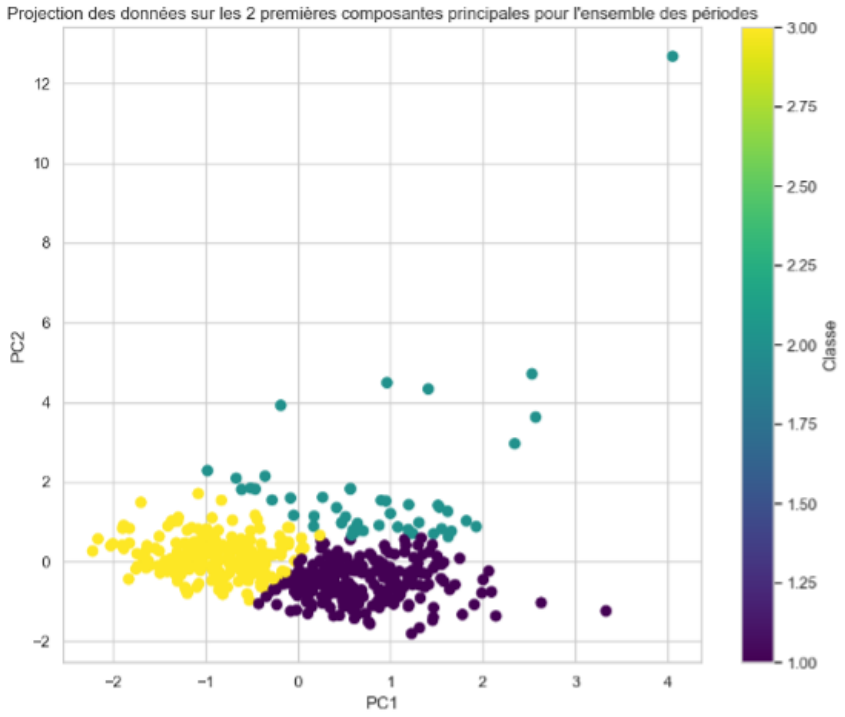

FIGURE 3 – Projection des données sur les deux premières dimension dans l'ACP

Pour l'ensemble des jeux olympiques depuis 1896, nous obtenons les résultats suivants :
Le critère du coude pour tous les jeux olympiques depuis 1896 nous donne 3 ou 4 classes (cf figure 4), mais prendre 4 classe réduit davantage l'inertie totale. Nous prendrons donc 3 classes.

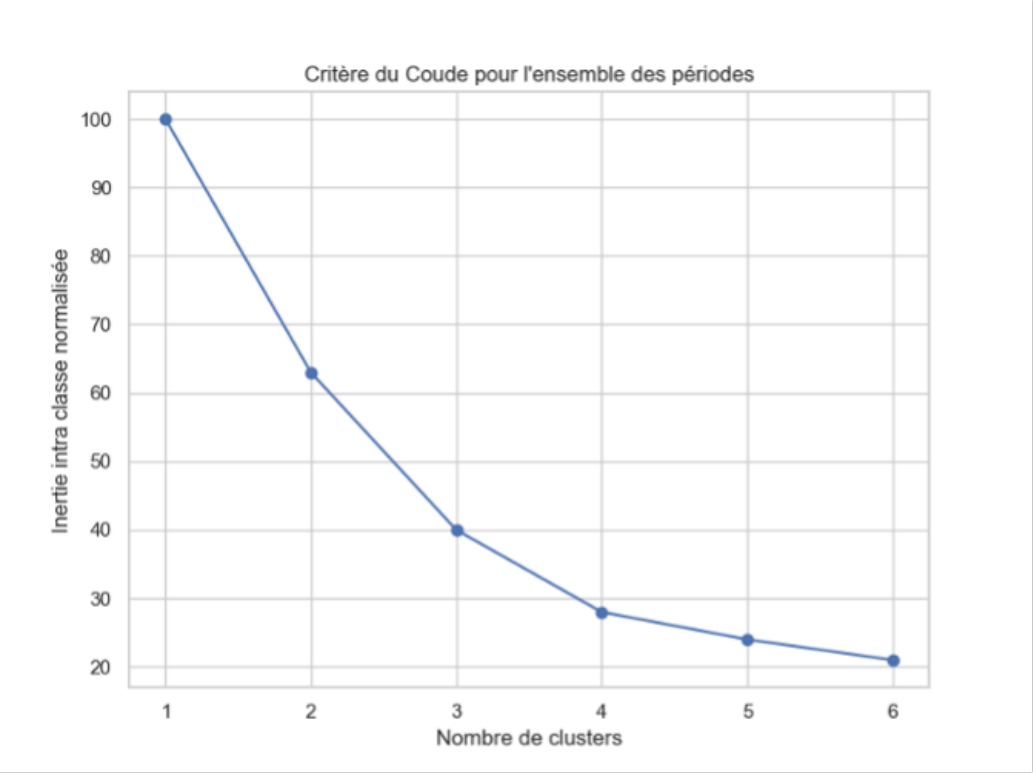

FIGURE 4 – L'inertie portée par chaque dimension dans l'ACP

Nous pouvons résumer les données dans un tableau récapitulatif présentant les moyennes et les écarts-types des variables utilisées pour la classification pour chaque cluster. Nous obtenons ainsi :

Pour les moyennes :

Cluster	Nb participants (%)	Part femmes	Âge moyen	Poids moyen	Taille moyenne
0	3,58	0,02	26,42	79,70	180,72
1	3,63	0,83	24,01	60,14	167,36
2	21,35	0,12	25,67	74,46	176,55

Pour les écarts-types :

Cluster	Nb participants	Part femmes	Âge moyen	Poids moyen	Taille moyenne
0	5,25	0,12	4,18	13,28	6,77
1	4,14	0,37	3,61	7,49	6,56
2	47,7	0,32	5,25	11,99	7,03

Nous déterminons les groupes suivants :

Typologie des clusters d'épreuves sportives olympiques depuis 1896

- **Cluster 0 :**
  - Épreuves sportives en moyenne majoritairement individuelles
  - Avec des athlètes principalement masculins
  - Plutôt de poids lourds
  - Et de grande taille
- **Cluster 1 :**
  - Épreuves sportives en moyenne majoritairement individuelles

- Avec des athlètes principalement féminins
- Plutôt de poids légers
- Et de petite taille

— **Cluster 2 :**

- Épreuves sportives en moyenne majoritairement collectives
- Avec des athlètes principalement masculins
- Plutôt de poids moyens
- Et de taille moyenne

5.5 Solution par périodes considérées

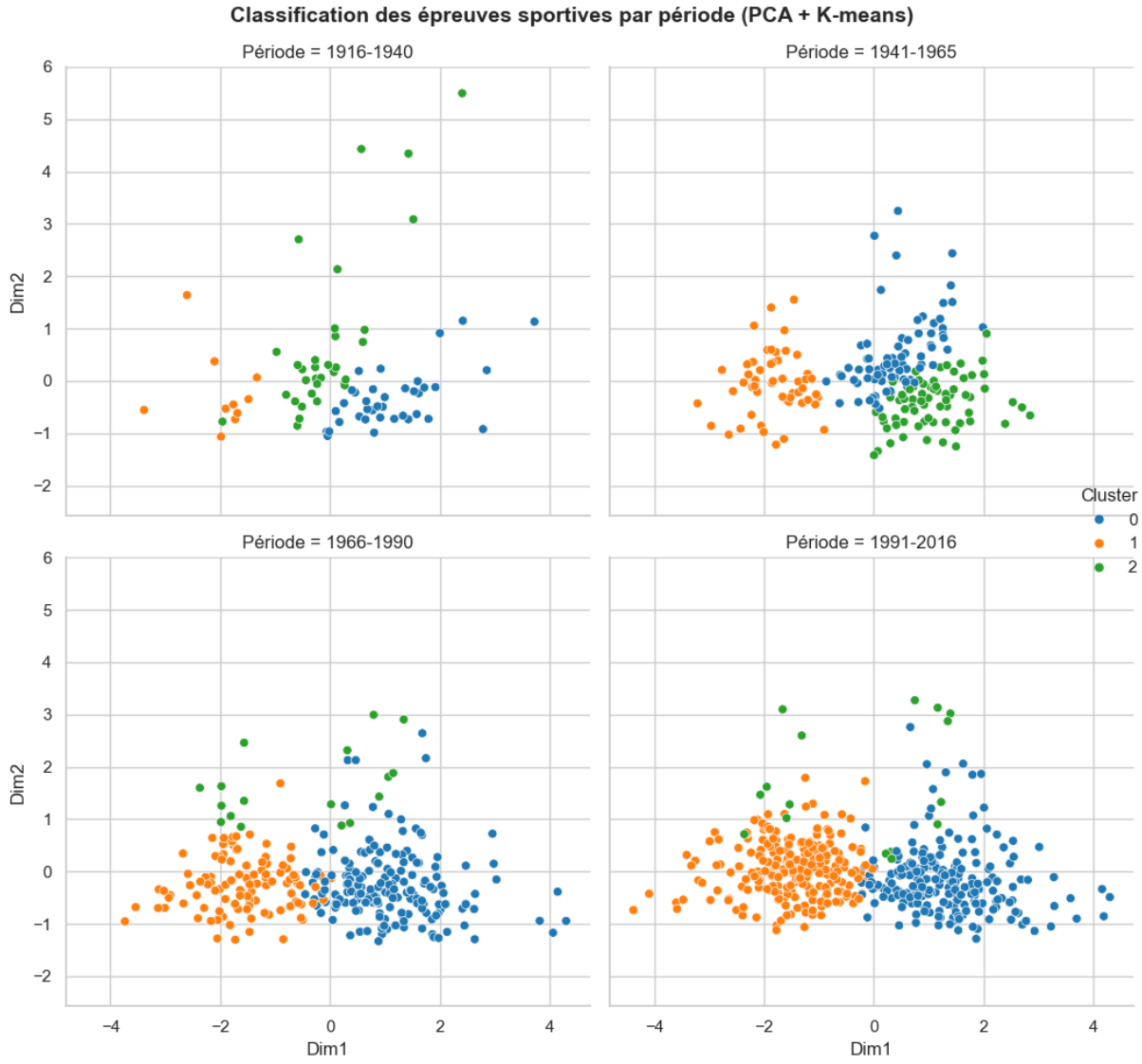

FIGURE 5 – Projection des données sur les deux premières dimensions dans l'ACP, pour les périodes considérées

Nous pouvons résumer les données dans un tableau récapitulatif présentant les moyennes et les écarts-types des variables utilisées pour la classification pour chaque cluster. Nous obtenons ainsi :

Pour les moyennes :
Analyse des clusters

Période	Cluster	Nb participants	Part femmes	Âge moyen	Poids moyen	Taille moyenne
1916-1940	0	4,54	0	24,77	82,01	181,12
	1	8,5	1	21,33	58,51	166,77
	2	31,06	0,02	28,86	66,16	170,62
1941-1965	0	6,92	0,02	28,57	70,45	174,25
	1	5,26	0,83	23,2	58,86	164,79
	2	3,96	0,01	24,43	82	181,14
1966-1990	0	2,98	0,01	25,59	79,57	180,94
	1	3,31	0,8	22,74	60,06	167,75
	2	49,88	0,47	23,84	66,76	173,02
1991-2016	0	2,78	0,03	26,64	82,51	182,64
	1	3,19	0,83	24,92	60,54	167,8
	2	45,25	0,45	25,84	68,37	173,51

Cluster 0 — Hommes grands et lourds
Profil : Athlètes très grands et lourds, exclusivement masculins.
Âge moyen : 24–26 ans
Part femmes : 0 ou très proche de 0
Poids : 79–82 kg (forts)
Taille : 180–182 cm (grands)

Le cluster 0 correspond à des sports de puissance et de gabarit imposant : *lancers, lutte, haltérophilie, judo.*

Cluster 1 — Femmes jeunes, petites et légères
Profil : Femmes très jeunes, taille et poids faibles.
Âge moyen : 21–25 ans
Part femmes : $\approx 100\,\%$
Poids : 58–60 kg
Taille : 166–168 cm

Le cluster 1 correspond à des sports féminins artistiques ou techniques, principalement *gymnastique, natation, patinage, ski alpin.*

Cluster 2 — Groupe mixte dominant, gabarit moyen
Profil : Taille et poids moyens, part féminine croissante selon la période.
Âge moyen : 23–29 ans
Part femmes : Passe de $\approx 0\,\%$ à $45\,\%$ entre 1916 et 2016
Poids : 66–68 kg
Taille : 170–173 cm

Le cluster 2 est le plus varié, mixte, et couvre des disciplines moins dépendantes d'un gabarit extrême, comme *tir, tennis, natation, sprint, ski, golf, snowboard*, etc.

6 Conclusion

Ce rapport présente les analyses menées à partir des données des Jeux Olympiques, couvrant un large spectre de thématiques allant des performances individuelles des athlètes aux statistiques agrégées par pays, jusqu'à l'application de méthodes d'apprentissage automatique pour regrouper les sportifs selon leurs caractéristiques communes. Les traitements s'appuient sur la puissance de bibliothèques spécialisées comme `pandas` pour manipuler efficacement de grands volumes de données, tout en intégrant des approches en Python natif afin de mieux comprendre certains mécanismes sous-jacents.

L'objectif principal était de produire des réponses précises et nuancées aux différentes questions posées, tout en mettant en évidence les défis spécifiques liés à l'analyse de données sportives. Les résultats obtenus soulignent l'importance de contextualiser les données, en particulier pour les disciplines collectives ou celles ayant connu des évolutions significatives dans le temps.

D'un point de vue technique, le projet a été conçu selon une architecture modulaire et extensible, facilitant la maintenance et l'ajout de nouvelles fonctionnalités. Une interface graphique interactive a été développée afin d'offrir une expérience utilisateur fluide et accessible, permettant une exploration intuitive des résultats et des visualisations.

Les comparaisons entre les traitements réalisés avec `pandas` et ceux réalisés en Python pur ont mis en lumière des arbitrages intéressants entre performance et souplesse. Si `pandas` est particulièrement performant pour le traitement massif de données, certaines opérations ciblées peuvent être plus efficaces en Python classique.

Ce travail met en évidence la richesse des informations contenues dans les bases de données sportives, et la pertinence d'une approche combinée mêlant outils analytiques, programmation, visualisation et apprentissage automatique. Il ouvre également la voie à des explorations futures, notamment sur l'évolution des performances.

Enfin, ce projet a constitué une expérience formatrice tant sur le plan technique que collaboratif. Le travail en équipe nous a permis d'expérimenter la répartition des tâches, la gestion des contributions multiples, la résolution de conflits et l'intégration progressive du code. Il nous a également confrontés à des défis personnels, en exigeant une montée en compétences et une réflexion continue sur les bonnes pratiques de développement. Cette démarche nous a confortés dans l'idée que la mise en œuvre concrète est un levier d'apprentissage très efficace, et que l'organisation du travail est aussi déterminante que le contenu technique. Nous avons pu mesurer notre maîtrise de Python, et apprécier la valeur ajoutée d'outils comme `pandas` et `matplotlib` dans un projet de data science appliquée.