

# Analyse du Dataset : *120 Years of Olympic History*

## Introduction :

Ce projet explore l'évolution des Jeux Olympiques modernes, été et hiver, de 1896 à 2016 à partir d'un dataset de plus de 271 000 enregistrements issus de Kaggle.

L'objectif est de comprendre les grandes tendances (participation, performances, parité femmes-hommes, palmarès par pays) et mettre en lumière l'impact des événements historiques sur cette compétition mondiale.

Pour y parvenir, les données brutes ont été nettoyées et enrichies avec Python (pandas, numpy) et SQL (SQLite), puis analysées et visualisées dans Power BI.

Le travail a inclus la gestion des valeurs manquantes, la traduction partielle en français, la création d'une clé unique, et la suppression des doublons pour garantir la fiabilité de l'analyse.

Les visualisations obtenues permettent de suivre l'évolution du nombre d'athlètes, de pays participants, la progression de la présence féminine, et mettre en évidence les nations les plus titrées dans l'histoire des JO.

## Table des matières

Analyse du Dataset : <i>120 Years of Olympic History</i> .....	1
Introduction .....	1
Contexte & objectifs d'analyse : .....	2
Sources .....	2
Objectifs techniques : .....	3
Structuration des données : .....	3
Outils et technologies utilisés : .....	3
Excel : .....	3
Python : .....	4
SQL .....	7
Power BI .....	9
Analyse : .....	10
Aperçu des résultats (insights majeurs, graphiques, capture Power BI) : .....	<b>Erreur ! Signet non défini.</b>
.....	<b>Erreur ! Signet non défini.</b>
Instructions pour reproduire le projet (comment exécuter le code) : .....	<b>Erreur ! Signet non défini.</b>
Crédits / remerciements (sources, inspirations) .....	<b>Erreur ! Signet non défini.</b>

## Contexte & objectifs d'analyse :

### 1. Contexte

Passionné par le sport et notamment par les Jeux olympiques que j'ai toujours suivi assidument, j'ai décidé de m'intéresser de plus près aux résultats, aux pays participants, aux athlètes, à la place de la femme au sein des JO.

Cette compétition est une « **vitrine** » mondiale à travers laquelle nous retrouvons l'Histoire, que ce soit en termes de géopolitique mondiale, de l'évolution de la place de la femme à travers les années. Nous allons mettre en lumière les grandes tendances de l'histoire olympique.


J'ai choisi ce dataset parmi tant d'autres car il apportait des données (disciplines, l'âge, la taille, le poids), que je souhaite utiliser dans une prochaine mise à jour (**WORK IN PROGRESS**).

### 2. Objectifs

- Montrer l'évolution du nombre d'athlète en fonction des années.
- Montrer l'évolution du nombre de pays participant en fonction des années
- Montrer l'évolution de la place de la femme aux JO
- Mettre en valeur les pays ayant gagné le plus de médailles d'or (JO d'été ou d'hiver)
- Les pays ayant gagné le plus de médailles à travers le monde entre 1896 et 2016.

## Sources :

 **Nom** : 120 years of Olympic history: athletes and results

 **Kaggle** : <https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results>

### Fichiers principaux :

- **athlete\_events.csv** → données brutes des athlètes, épreuves, médailles
- **noc\_regions.csv** → table de correspondance des pays (NOC → nom pays + région)

### Content

1. **ID** - Unique number for each athlete
2. **Name** - Athlete's name
3. **Sex** - M or F
4. **Age** - Integer
5. **Height** - In centimeters
6. **Weight** - In kilograms
7. **Team** - Team name
8. **NOC** - National Olympic Committee 3-letter code
9. **Games** - Year and season
10. **Year** - Integer
11. **Season** - Summer or Winter
12. **City** - Host city
13. **Sport** - Sport

14. **Event** - Event

15. **Medal** - Gold, Silver, Bronze, or NA

Citation de l'auteur du dataset à « [...] récupéré les données sur [www.sports-reference.com](http://www.sports-reference.com), qui héberge une base de données détaillée sur l'histoire olympique, développée et maintenue par un groupe indépendant de statisticiens olympiques ».

## Objectifs techniques :

Nous allons nettoyer un fichier CSV contenant plus de 270 000 individus & 15 colonnes.

- Traduire une partie des données en français.
- Compléter ou retirer les valeurs manquantes
- Vérifier qu'il n'y a pas de doublons dans ce dataset
- Définir les variables que nous souhaitons utiliser

## Structuration des données :

Voici la structure des colonnes, après traduction.

1. **ID** – Nombre unique donnée à pour un athlète
2. **Nom** – Nom des athlètes
3. **Sexe** – homme ou femme
4. **Age** – nombre entier
5. **Taille** – en centimètres
6. **Poids** – en kg
7. **Pays** – Nom du pays représenté
8. **NOC** – Code donné par le comité international olympique
9. **Jeux** – Année et Saison
10. **Année** – nombre entier
11. **Saison** – Été ou hiver
12. **Lieu** – Pays hôte
13. **Sport** – nom du sport
14. **Discipline** – nom de la discipline
15. **Médaille** – or, argent, bronze ou (NA) pour les athètes n'ayant pas reçu de médaille

## Outils et technologies utilisés :

### Excel :

Le document comprend un nombre de ligne raisonnable pour pouvoir l'ouvrir sur Excel afin d'avoir un grand aperçu du document.

Observation du fichier *athlete\_event.csv* :

- Le document comprend 271116 lignes et 15 colonnes
- 135571 athlètes
  - Vérifier qu'il n'y a pas d'erreur entre ID & nom de l'athlète
  - Vérifier qu'il n'y a pas de nom indique pour des athlètes différents (homonymes)
- Tout est en anglais, si je souhaite une lecture en français, je dois traduire une partie ou tout le document.

- La colonne Sexe est entièrement remplie
- Age : certaines données sont manquantes et d'autres sont à vérifier car cela va de 10 à 97 ans (vérifier qu'il n'y a pas d'outlier)
- Taille : certaines données sont manquantes et d'autres sont à vérifier car cela va de 127 à 226 cm (vérifier qu'il n'y a pas d'outlier)
- Poids : Idem valeur allant de 25 à 213 kg et penser à ne laisser qu'un chiffre après la virgule (Vérifier qu'il n'y a pas d'outlier)
- Certains noms d'équipe, renommé pays dans mon tableau sont particulier. Jointure importante à faire avec le document [noc\\_regions.csv](#)
- Vérifier qu'il n'y a pas d'erreur entre les colonnes Jeux et Année.
- Ville d'accueil : pas d'erreur notable
- Sport : pas d'erreur notable (prévoir de traduire en français)
- Discipline : pas d'erreur notable (décider si traduction ou non)
- Médaille : pas d'erreur notable mais remplacer NA par un « - » pour qu'il n'y ait pas d'erreur en suite.

Observation [noc\\_regions.csv](#) :

1. 3 colonnes et 230 NOC (abréviation crée par le CIO pour les pays)
2. Certains NOC crée ont évolués car les pays se sont divisés, regroupés, etc.. (Ex : Yougoslavie).
3. Certains pays sont représentés par différents NOC (ex : Allemagne pays qui à la suite des guerres mondiales a eu plusieurs, appellation, ici 4 NOC différents pour représenter l'Allemagne)
4. Liste non exhaustive des abréviations à vérifier : (AHO, ANZ, BOH, CRT, NBO, NFL, ROT, SCG, SKN...)
5. Liste des pays à vérifier car contient différents NOC : Yemen (3) ; Russie (3) ; Serbie (3) ; Vietnam (2) ; etc...
6. UNK abréviation attribué pour des pays inconnus (vérifier et voir si suppression à faire)

## Python :

Script à retrouver sur mon Github : « [JO - Athletes - 1896 – 2016.py](#) »

Chargement du CSV athlete\_events.csv : 271 116 lignes et 15 colonnes avec des valeurs manquantes dans plusieurs colonnes soit le même constat que sur Excel.

### 1. Traduction des colonnes en français

Comme prévu j'ai traduit le nom des colonnes pour une lecture française.

```
JO.columns = ["ID", "Nom", "Sexe", 'Age', 'Taille', 'Poids', 'Pays', 'NOC', 'Jeux',
              'Année', 'Saison', 'Lieu', 'Sport', 'Discipline', 'Medailles']
```

### 2. Vérification des noms des athlètes

Nous avons 713 homonymes, nous ne savons pas si l'ID est bien rempli.  
Nous devons donc créer notre propre clé unique.

### 3. Valeurs manquantes

Nous vérifions le pourcentage de valeurs manquantes pour nos 4 colonnes problématiques :

```
colonnes_cibles = ['Age', 'Taille', 'Poids', 'Medailles']
for col in colonnes_cibles:
    pourcentage = JO[col].isnull().mean() * 100
    print(f"{col} : {pourcentage:.2f}% de valeurs manquantes")
```

Nous obtenons :

- Age : 3.49% de valeurs manquantes
- Taille : 22.19% de valeurs manquantes
- Poids : 23.19% de valeurs manquantes
- Médailles : 85.33% de valeurs manquantes

Valeur manquantes - Médailles :

```
# Pour les médailles nous remplaçons les valeurs manquantes par un simple tiret (-)
JO['Médailles'] = JO['Médailles'].fillna('-')
print(JO["Médailles"].unique())
```

Valeur manquantes - Ages :

Nous allons réduire le nombre de valeurs manquantes pour l'âge en complétant les éléments manquant par : **la moyenne d'âge par discipline.**

```
age_sport = JO.groupby('Discipline')['Age'].transform('mean')
JO['Age'] = JO['Age'].fillna(age_sport)
```

Après cette étape, il nous reste 142 valeurs manquantes et les valeurs vont de 10 ans à 97 ans.

Après vérifications sur le site internet officiel des Jeux Olympiques <https://www.olympics.com/fr> :

- Le plus jeune athlète ayant participé à des JO avait bien 10 ans, c'était en 1896.
- En revanche nous trouvons que l'athlète le plus âgé ayant participé à des JO avait 72.

Je lance une recherche python pour comprendre pourquoi nous avons des valeurs au-dessus de cet âge-là :

```
print(JO.sort_values(by='Age', ascending=False)[['Nom', 'Age', 'Sport']].head(5))
```

Nous découvrons qu'il exista des compétitions artistiques dans les Jeux Olympiques modernes, sur une idée de Pierre de Coubertin. Ils ont eu lieu entre 1912 à 1948 et des médailles étaient décernées pour des œuvres d'art ayant un lien avec le sport dans 5 catégories : architecture, littérature, musique, peinture, sculpture.

Mais ces compétitions ont été abandonnées car les artistes devenaient professionnels tandis que les athlètes olympiques se devaient d'être amateurs selon les critères olympiques en vigueur.

Je décide donc de supprimer les lignes correspond à ces disciplines pour mon analyse et mes visualisations. Mais cela fait partie de l'histoire des Jeux Olympiques Moderne.

```
JO = JO[JO['Sport'] != 'Art Competitions']
print(JO.sort_values(by='Age', ascending=False)[['Nom', 'Age', 'Sport']].head(5))
```

À la suite de la suppression de ces données, nous retrouvons bien que le participant le plus âgés avait 72ans.

## Valeur manquantes – Poids & Taille :

Nous arrondissons le poids et la taille à une décimale :

```
JO['Taille'] = JO['Taille'].round(1)
JO['Poids'] = JO['Poids'].round(1)
```

Après avoir vérifié sur internet tous les *Outlier*, toutes les données semblent correctes (à +/- 4 / 5 unités en fonction des données trouvées).

J'ai volontairement choisi une approche réaliste du nettoyage en gardant un maximum de données tout en assurant une cohérence statistique.

L'imputation par discipline permet de préserver les différences sportives sans généraliser excessivement les profils physiques. Je décide donc de créer des moyennes de poids et de tailles par discipline (et non pas par sport) pour les valeurs manquantes.

```
JO['Taille'] = JO['Taille'].fillna(JO.groupby('Discipline')['Taille'].transform('mean'))
JO['Poids'] = JO['Poids'].fillna(JO.groupby('Discipline')['Poids'].transform('mean'))
```

Après ces étapes nous obtenons :

- Age : 0.05% de valeurs manquantes
- Taille : 0.48% de valeurs manquantes
- Poids : 1.47% de valeurs manquantes

Nous avons donc créé des valeurs fictives tout en s'approchant de la réalité pour une grande majorité des valeurs manquantes et nous avons contrôlé les valeurs extrêmes.

Nous découvrons également que certaines données sont manquantes pour certaines disciplines à une certaine date (Gymnastics Men's Team All-Around, Free System ; Sailing Mixed 7 metres etc...).

Mais je décide de garder toutes les données comme telle, les pourcentages de valeurs manquantes étant très faible.

## Traduction nom colonne - Sports :

Comme j'ai décidé de traduire le nom des Colonnes, je traduis également le nom des sports dans Python.

## Autres traductions :

Pour le moment je ne traduis pas les 736 disciplines différentes

Et je ne traduis pas non plus les pays car je veux que mon fichier NOC soit praticable avec ma table olympics\_clean.csv

## Création d'une colonne pour mentionner les données complètes en fonction des athlètes :

```
#Nous allons créer une colonne pour identifier les lignes où les données sont complètes
JO['Donnees_completes'] = (~JO['Age'].isnull()) & (~JO['Taille'].isnull()) &
(~JO['Poids'].isnull())
print(JO['Donnees_completes'].value_counts())
```

## Création d'une clé unique : ID\_Année\_Discipline\_Sexe

Comme nous ne sommes pas sûr que l'ID soit bien rempli et que nous avons des homonymes nous créons une clé unique comme ceci :

```
J0["cle"] = (  
    J0["ID"].astype(str) + "_" +  
    J0["Année"].astype(str) + "_" +  
    J0["Discipline"] + "_" +  
    J0["Sexe"]  
)
```

Export final : Export final vers olympics\_clean.csv (267 538 lignes) :

```
J0.to_csv(r"C:\Users\Sebas\Desktop\DA - Projet - Dataset J0\olympics_clean.csv",  
index=False)
```

## SQL

Importations des 2 tables CSV *olympics\_clean.csv* et *noc\_regions.csv* sur SQLiteStudio : création des colonnes et leurs types, puis importations des tables.

Vérifions tout d'abord si nous avons des doublons dans la clé créée :

```
SELECT  
COUNT(*) AS total_lignes,  
COUNT(DISTINCT CLE) AS distinct_cle,  
(COUNT(*) - COUNT(DISTINCT CLE)) AS nb_doublons  
FROM olympics_clean;
```

- Résultats : total lignes= 267538, distinct\_cle = 267393 ; nb\_doublons=145

Visualisation des clés problématiques :

```
SELECT CLE, COUNT(*) AS n  
FROM olympics_clean  
GROUP BY CLE  
HAVING COUNT(*) > 1  
ORDER BY n DESC, CLE;
```

- Résultat : 139 lignes chargées max 4 répétitions de la même clé

Je décide de supprimer les lignes correspondantes aux doublons :

```
WITH d AS (  
    SELECT rowid,  
           ROW_NUMBER() OVER (PARTITION BY CLE ORDER BY rowid) AS rn  
    FROM olympics_clean  
)  
DELETE FROM olympics_clean  
WHERE rowid IN (SELECT rowid FROM d WHERE rn > 1);  
  
SELECT COUNT(*) total, COUNT(DISTINCT CLE) distinct_cle  
FROM olympics_clean;
```

- J'ai décidé de supprimer les 145 doublons, il nous reste **267393 lignes**.

Maintenant on fige les données et nous créons quelques index pour fixer nos clés primaires :

```
PRAGMA foreign_keys = ON;
CREATE UNIQUE INDEX IF NOT EXISTS idx_olympics_pk ON olympics_clean(CLE);
CREATE UNIQUE INDEX IF NOT EXISTS idx_noc_pk ON noc(NOC);
CREATE INDEX IF NOT EXISTS idx_olympics_noc ON olympics_clean(NOC);
```

Avant d'effectuer la jointure entre nos 2 tables nous vérifions que nous n'avons pas d'erreur :

- I. On vérifie que nous n'avons pas de valeurs NOC orphelines du coté de notre table NOC :

```
SELECT o.NOC, COUNT(*) AS freq
FROM olympics_clean o
LEFT JOIN noc n USING(NOC)
WHERE n.NOC IS NULL
GROUP BY o.NOC
ORDER BY freq DESC;
```

➤ SGP = 349

Nous avons une erreur, SGP ne correspond à aucun NOC de notre table *olympics\_clean*. N  
Il se trouve que SGP = Singapour et il est écrit SIN dans notre table principale. Je décide d'attribuer le code CIO (NOC) : SIN pour Singapour.

```
UPDATE olympics_clean
SET NOC = 'SIN'
WHERE NOC = 'SGP';
```

- II. Valeurs manquantes critiques

```
SELECT
    SUM(NOC IS NULL OR NOC='') AS noc_vides,
    SUM("Année" IS NULL) AS annee_vides,
    SUM(Sport IS NULL OR Sport='') AS sport_vides
FROM olympics_clean;
```

Notre table ne comporte pas d'éléments manquants pour les colonnes : NOC, Année et Sport. Nous pouvons effectuer la jointure entre nos 2 tables.

- III. Jointure :

```
CREATE TABLE olympics_enriched AS
SELECT
    o.*,
    n.pays AS NOC_Pays,
    n.notes AS NOC_Region
FROM v_olympics o
LEFT JOIN noc n USING (NOC);

CREATE INDEX IF NOT EXISTS idx_enriched_noc ON olympics_enriched(NOC);
CREATE INDEX IF NOT EXISTS idx_enriched_annee ON olympics_enriched(Année);
CREATE INDEX IF NOT EXISTS idx_enriched_sport ON olympics_enriched(Sport);
```

Nous vérifions maintenant que la jointure a fonctionné sans perdre de valeurs :

```
SELECT COUNT(*) FROM olympics_enriched;
```

➤ Nous avons 267393 lignes. La jointure a fonctionné.

```
SELECT COUNT(*) AS lignes_diff
FROM olympics_enriched e
LEFT JOIN olympics_clean c
ON e.CLE = c.CLE
WHERE c.CLE IS NULL;
```



- Aucune CLE NULL.

```
SELECT DISTINCT e.NOC  
FROM olympics_enriched e  
LEFT JOIN noc n ON e.NOC = n.NOC  
WHERE n.NOC IS NULL;
```

- Aucun NOC NULL.

La jointure a fonctionné, nous avons donc notre table définitive. Nous pouvons passer sur Power BI Desktop

## Power BI Desktop

### Power Query

Voici les étapes réalisées avec Power Query :

```
#Source = Csv.Document(File.Contents("C:\Users\Sebas\Desktop\DA - Projet – Dataset JO\olympics_VD.csv"),[Delimiter=";",  
Columns=19, Encoding=65001, QuoteStyle=QuoteStyle.None]),
```

```
#"En-têtes promus" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
```

```
#"Valeur remplacée" = Table.ReplaceValue("#Type modifié", "Summer", "Eté", Replacer.ReplaceText, {"Saison"}),
```

```
#"Valeur remplacée1" = Table.ReplaceValue("#Valeur remplacée", "Winter", "Hiver", Replacer.ReplaceText, {"Saison"}),
```

Maintenant la traduction de tous les pays en français car ils apparaîtront dans mes visuels (étape que j'aurais dû faire en amont sur Python). Qlq exemples :

```
#"Valeur remplacée2" = Table.ReplaceValue("#Valeur remplacée1", "Denmark", "Danemark", Replacer.ReplaceText, {"Pays"}),
```

```
#"Valeur remplacée3" = Table.ReplaceValue("#Lignes triées", "Albania", "Albanie", Replacer.ReplaceText, {"NOC_Pays"}),
```

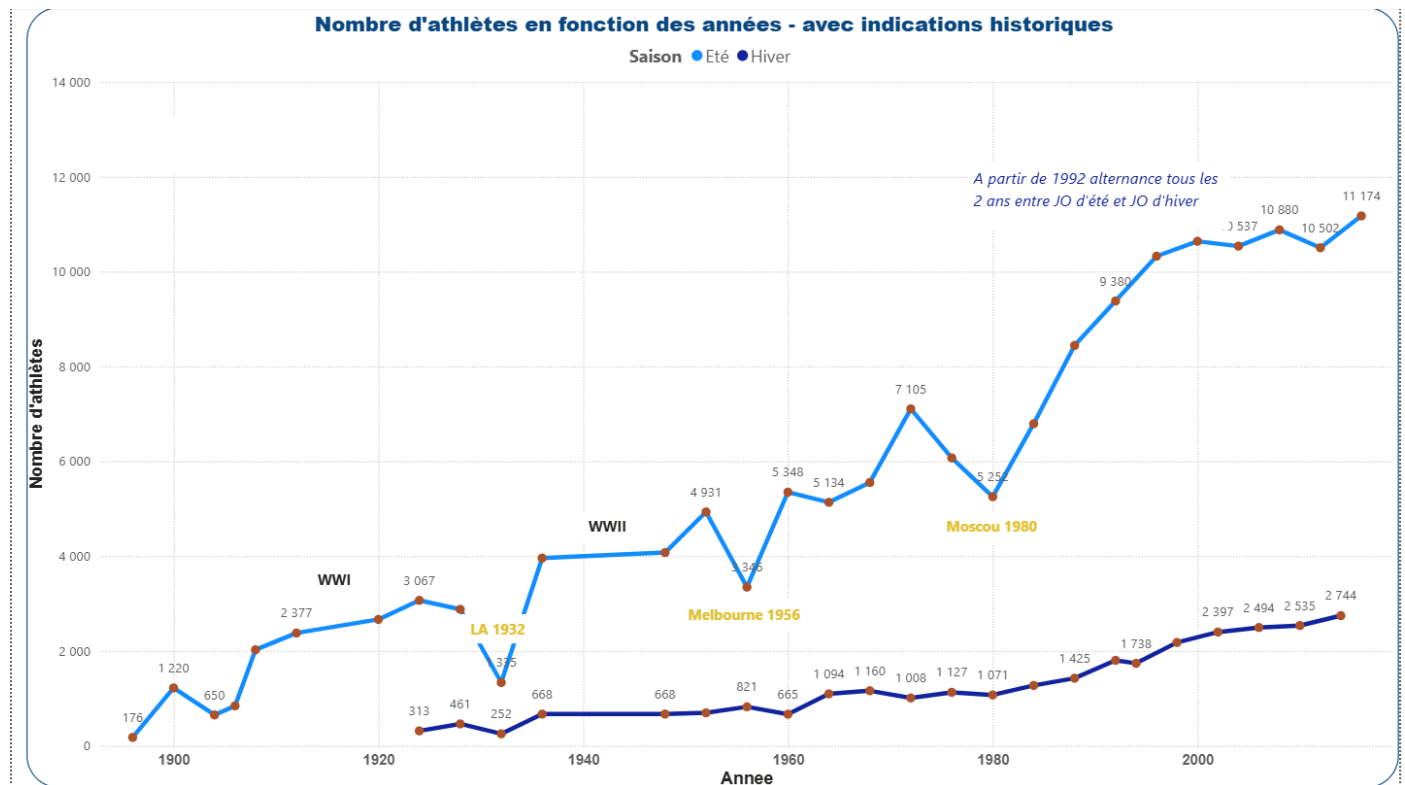
```
#"Valeur remplacée4" = Table.ReplaceValue("#Valeur remplacée3", "Algeria", "Algérie", Replacer.ReplaceText, {"NOC_Pays"}),
```

```
.  
.  
.
```

```
#"Valeur remplacée134" = Table.ReplaceValue("#Valeur remplacée133", "Zambia", "Zambie", Replacer.ReplaceText, {"NOC_Pays"}),
```

# Analyse (aperçu des résultats : insights majeurs, graphiques, capture PowerBI) :

## 1) Nombre d'athlètes en fonction des années :



Nous constatons deux longues périodes sans aucun Jeux la première entre 1912 & 1920 et la deuxième entre 1936 & 1948. Cela correspond aux périodes des 2 guerres mondiales.

Nous constatons également un changement à partir des années 2000 : Avant 1992, les jeux d'hiver et d'été avaient lieu la même année puis tous les 4 ans. Après 1992, une alternance tous les deux ans entre JO d'hiver et ceux d'été a été mis en place.

De plus, j'ai mis en évidence (jaunes) quelques points, ceux où la diminution d'athlètes est manifeste.

1. **L.A., 1932** : Beaucoup moins de participant car ces Jeux ont lieu loin de l'Europe, pendant la Grande Dépression. De nombreux athlètes n'ont pas pu se permettre de s'y rendre.
2. **Melbourne, 1956** : Plusieurs nations ont boycotté les JO cette année là comme :
  - L'Irak, l'Égypte et le Liban : en raison de l'implication de la France et de la Grande-Bretagne dans la crise du canal de Suez.
  - Les Pays-Bas, l'Espagne, la Suisse et le Cambodge n'ont pas participé en raison de la répression de la révolution hongroise par l'Union soviétique.
  - La Chine n'a pas participé car le Comité International Olympique a reconnu Taïwan.
3. **Montréal, 1976** : 25 nations, principalement africaines, ont boycotté les JO en réponse à la politique d'apartheid en Afrique du Sud.
4. **Moscou, 1980** : 66 pays ont boycotté les JO en réponse à l'invasion soviétique de l'Afghanistan

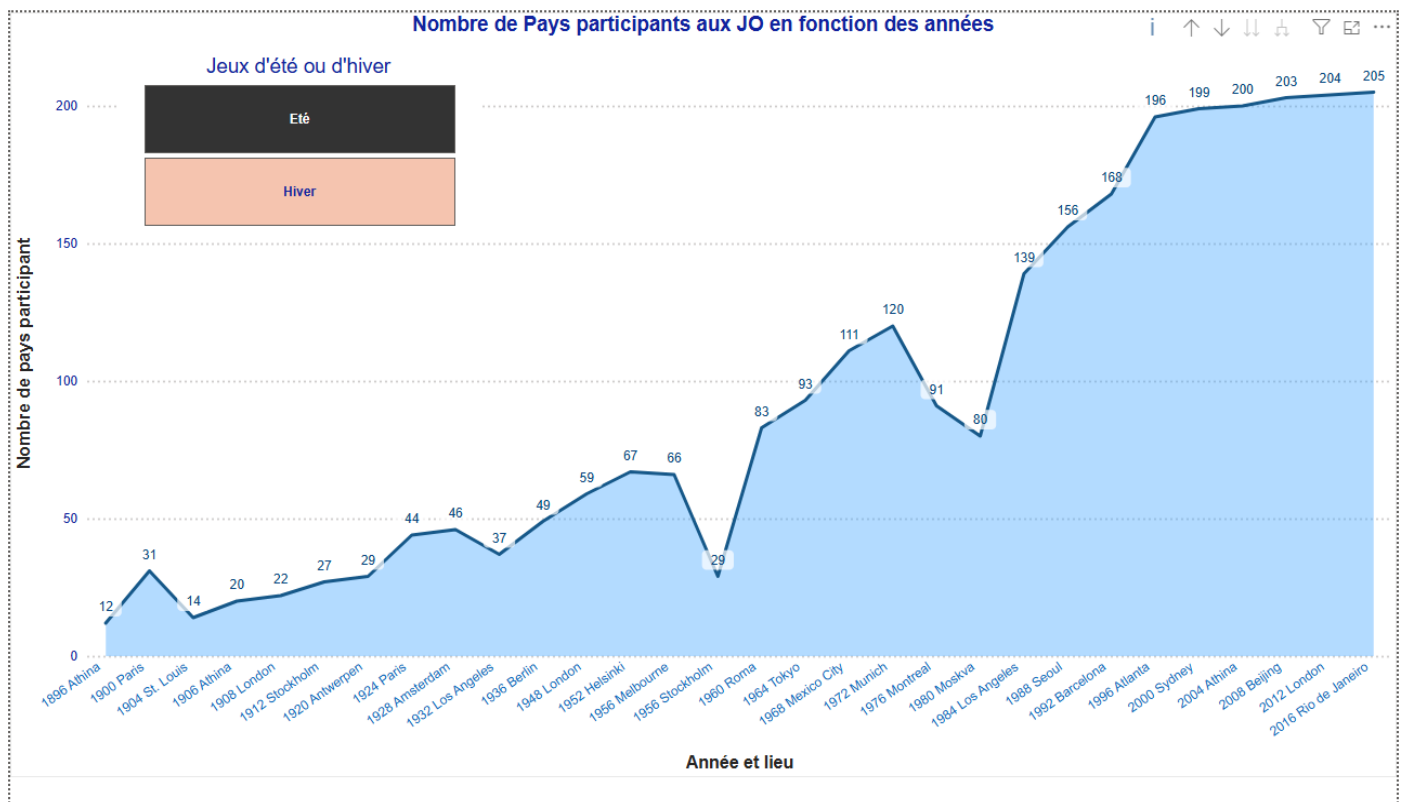
Autrement nous constatons une augmentation très nette du nombre de participants aux JO d'été et moindre sur les JO d'hiver.

Mais si nous prenons l'exemple entre les JO de 1960 et ceux de 2016 (été) / 2014 (hiver) :

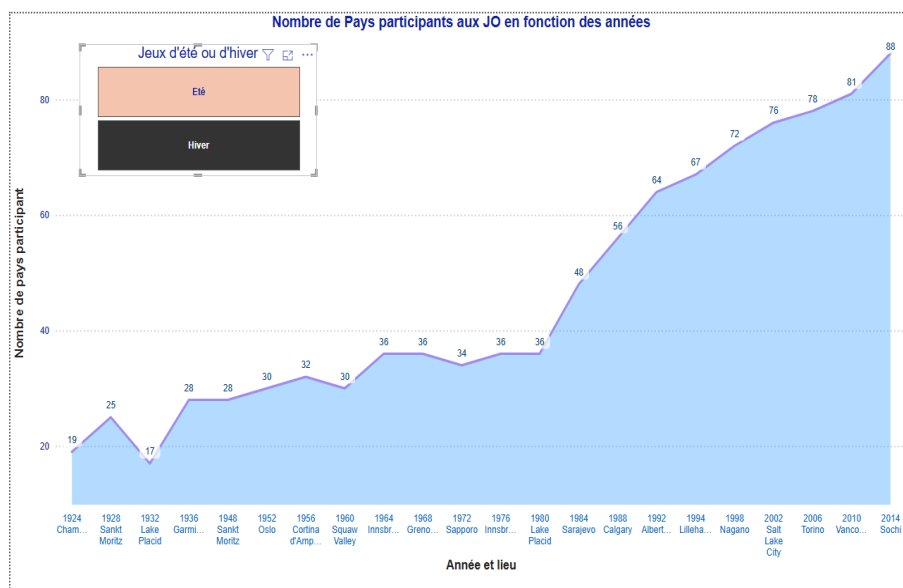
- Pour les JO d'été le nombre de participant à doublé
- Pour les JO d'hiver le nombre à quadruplé

On remarque également que depuis les années 2000 le nombre de participant est plutôt stagnant avec une hausse légère du nombre d'athlète.

## 2) Nombre de pays participants en fonction des années :



Concernant le nombre de pays participants aux JO nous retrouvons les mêmes données constatées dans l'analyse précédente.



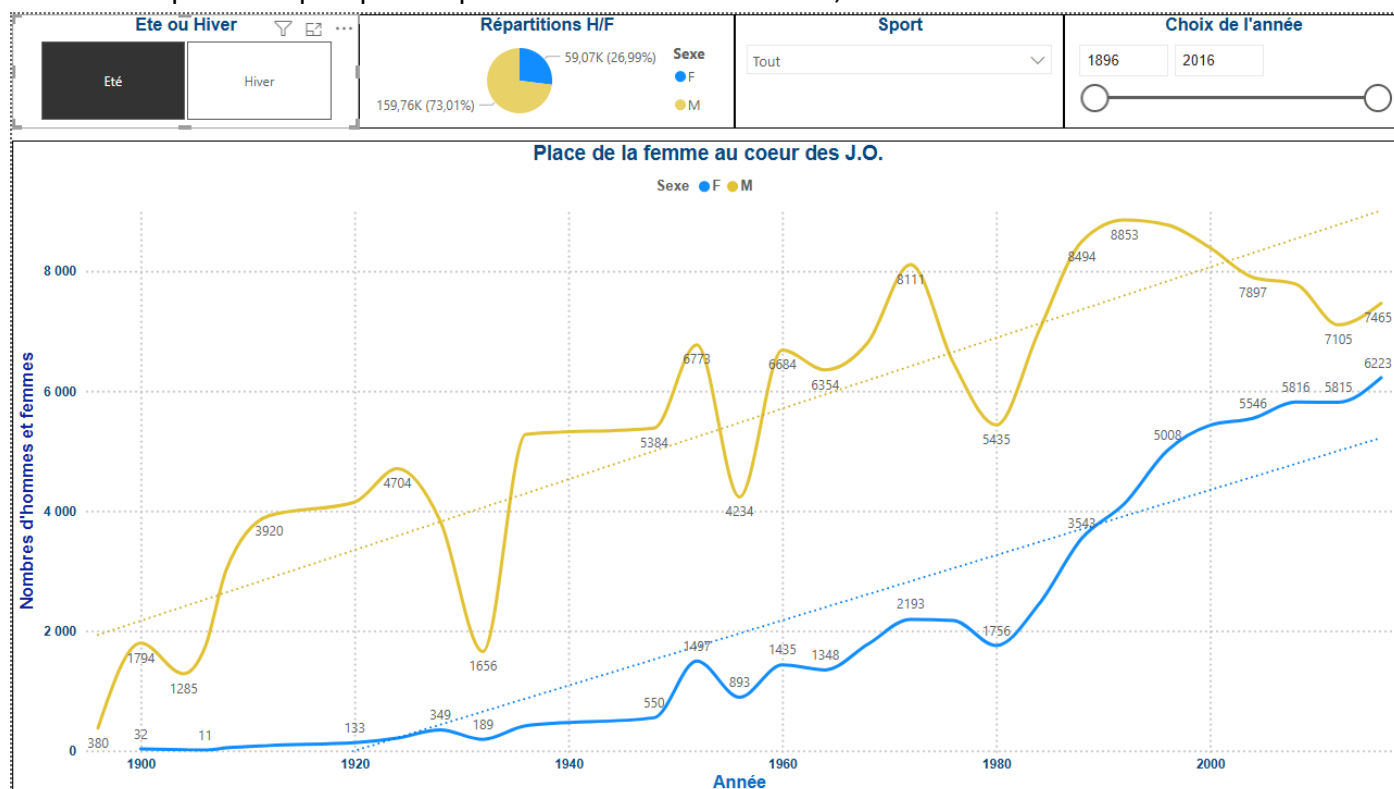
Ici j'ai choisi délibérément de pouvoir filtrer entre JO d'été et d'hiver car nous ne retrouvons pas les mêmes fluctuations en raison des différents boycotts.

Il semble que les JO d'hiver ont été moins impacté par le contexte mondial (cela est également du au faite que les nations représentées aux JO d'hiver sont moins nombreuses).

Nous voyons également que les JO d'hiver ont débuté en 1924.

### 3) Evolution de la place de la femme au cœur des Jeux Olympiques :

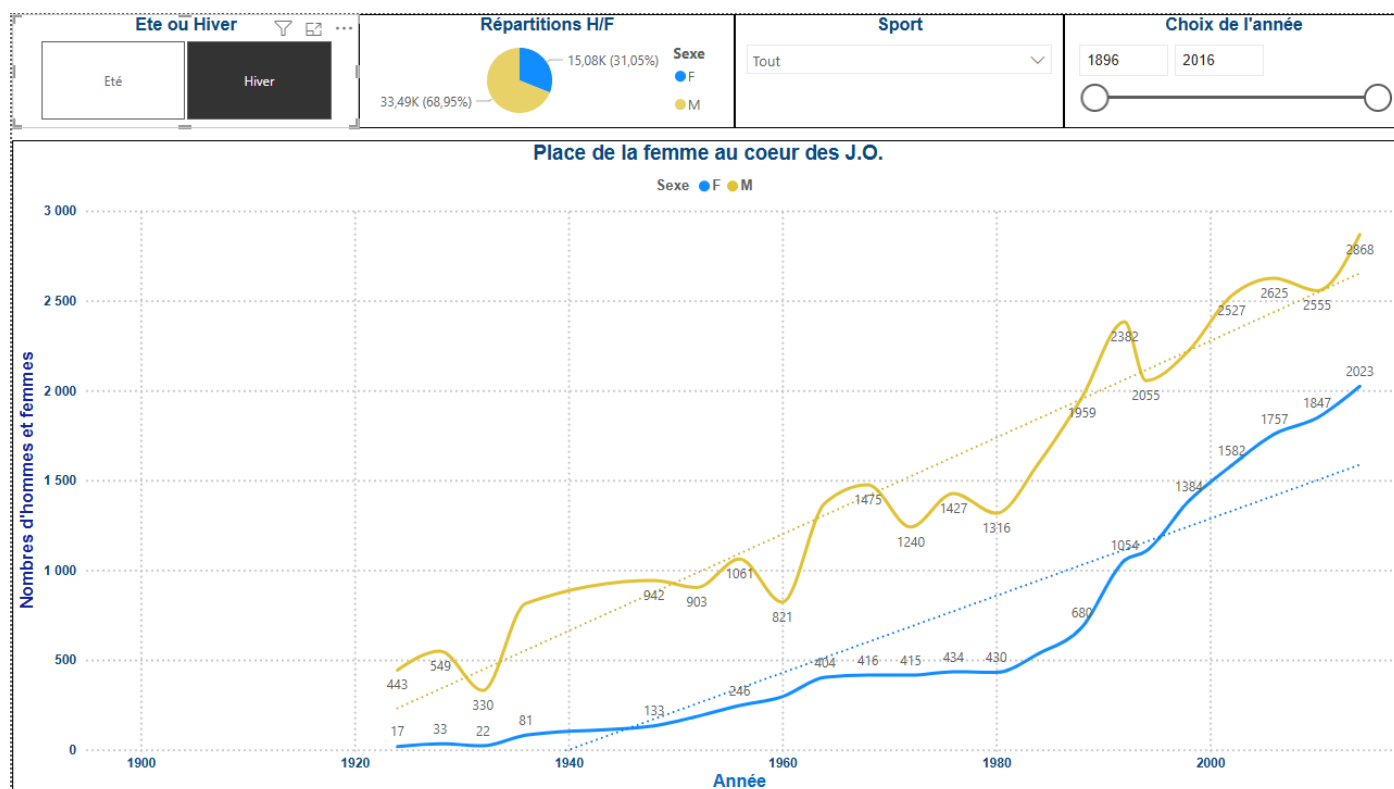
Les femmes prennent part pour la première fois aux JO en 1900, à Paris.



Nous remarquons qu'elles sont très peu nombreuses jusqu'en 1920 et l'évolution prend beaucoup de temps. Cela est dû en partie au fait que Pierre de Coubertin, réputé pour être misogyne, était en 1900 contre la participation des femmes aux JO. Et que la bataille a duré près de 90 ans.

Car ce n'est seulement qu'en 1991, que toute nouvelle discipline aux JO doit obligatoirement comporter des épreuves féminines. Depuis cette date les femmes représentent en moyenne 40% des athlètes.

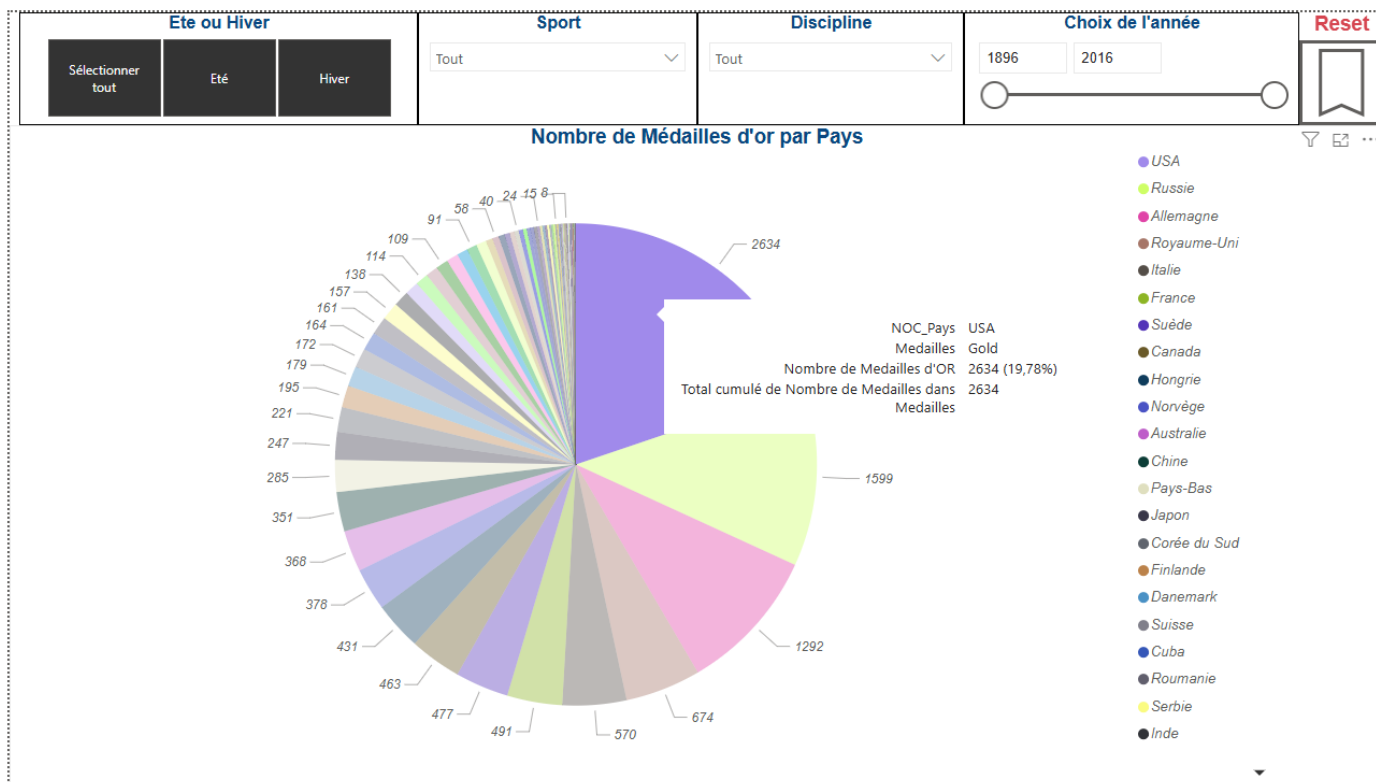
Les courbes et tendances sont assez équivalentes entre les JO d'Hiver et ceux d'été.



#### 4) Nombre de médaille d'OR remportées par pays :

Sans grandes surprises les Etats-Unis ont remporté 19.78% des médailles d'OR tout JO confondu.

Dans ce tableau de bord, j'ai choisi de créer des filtres de sport & disciplines, afin de pouvoir regarder plus en profondeur les résultats à travers les années des médaillés d'OR. Nous pouvons également sélectionner des périodes ou des années, et voir en fonction des JO d'Hiver et d'Eté.



WORK IN PROGRESS : Je souhaite prochainement créer un graphique montrant le nombre de Médaillé d'OR par rapport au nombre d'habitant par Pays. Cela ce serait un très bon indicateur pour voir et déterminer les capacités sportives de chaque pays.

## 5) Tableau total de médailles / pays

Dans ce tableau de bord, j'ai choisi de reproduire les mêmes filtres que dans le précédent. Tout en affichant un tableau pertinent sur la gauche du nombre total de médaille remporté par Pays. De plus j'ai ajouté une carte interactive distinguant tous les pays ayant déjà reçu une médaille depuis la création des Jeux Olympiques Modernes.

Pour cela j'ai créé des index de ma table initiale pour n'afficher que les pays ayant reçu des médailles et en effectuer la somme par médaille.

