

```
In [1]: # A décommenter à la première exécution si jupyter se plaint de ne pas trouver oracledb
#!pip install --upgrade oracledb
```

```
In [2]: # Compléter ici les imports dont vous avez besoin, ne pas modifier ceux déjà présents
import getpass
from os import getenv
import pandas as pd
import oracledb
import warnings
import matplotlib.pyplot as plt
import numpy as np
```

```
In [3]: # Nécessaire pour éviter les problèmes de session
class Connexion(object):
    def __init__(self, login, password):
        self.conn = oracledb.connect(
            user=login,
            password=password,
            host="oracle.iut-orsay.fr",
            port=1521,
            sid="etudom",
        )
        self.conn.autocommit = False

    def __enter__(self):
        self.conn.autocommit = False
        return self.conn

    def __exit__(self, *args):
        self.conn.close()
```

```
In [4]: # La fonction ci-dessous est à utiliser pour exécuter une requête et stocker les résultats dans un dataframe Pandas sans afficher d'
# Vous pouvez vous en inspirer pour créer vos propres fonctions.
def requete_vers_dataframe(connexion_data, requete, valeurs = None):
    with Connexion(login=connexion_data['login'], password=connexion_data['password']) as connexion:
        warnings.simplefilter(action='ignore', category=UserWarning)
        if valeurs is not None:
            df = pd.read_sql(requete, connexion, params=valeurs)
        else:
            df = pd.read_sql(requete, connexion)
```

```
warnings.simplefilter("always")
return df
```

```
In [5]: # Saisir ci-dessous l'édition des JO qui vous a été attribuée. Cela correspond au LibelleHote dans la table HOTE de la base de données
# Par exemple EDITION_JO = "2020 Summer Olympics"
EDITION_JO = "1948 Winter Olympics"
# Saisir ci-dessous le login court de la base utilisée pour votre carnet
SCHEMA = "MKULIKO"
```

```
In [6]: # On demande à l'utilisateur son login et mot de passe pour pouvoir accéder à la base de données
if getenv("DB_LOGIN") is None:
    login = input("Login : ")
else:
    login = getenv("DB_LOGIN")
if getenv("DB_PASS") is None:
    password = getpass.getpass("Mot de passe : ")
else:
    password = getenv("DB_PASS")
conn = {'login': login, 'password': password}
```

Analyse statistique des JO

Maël Alleaume, Matthieu Kulikowski-Bontemps

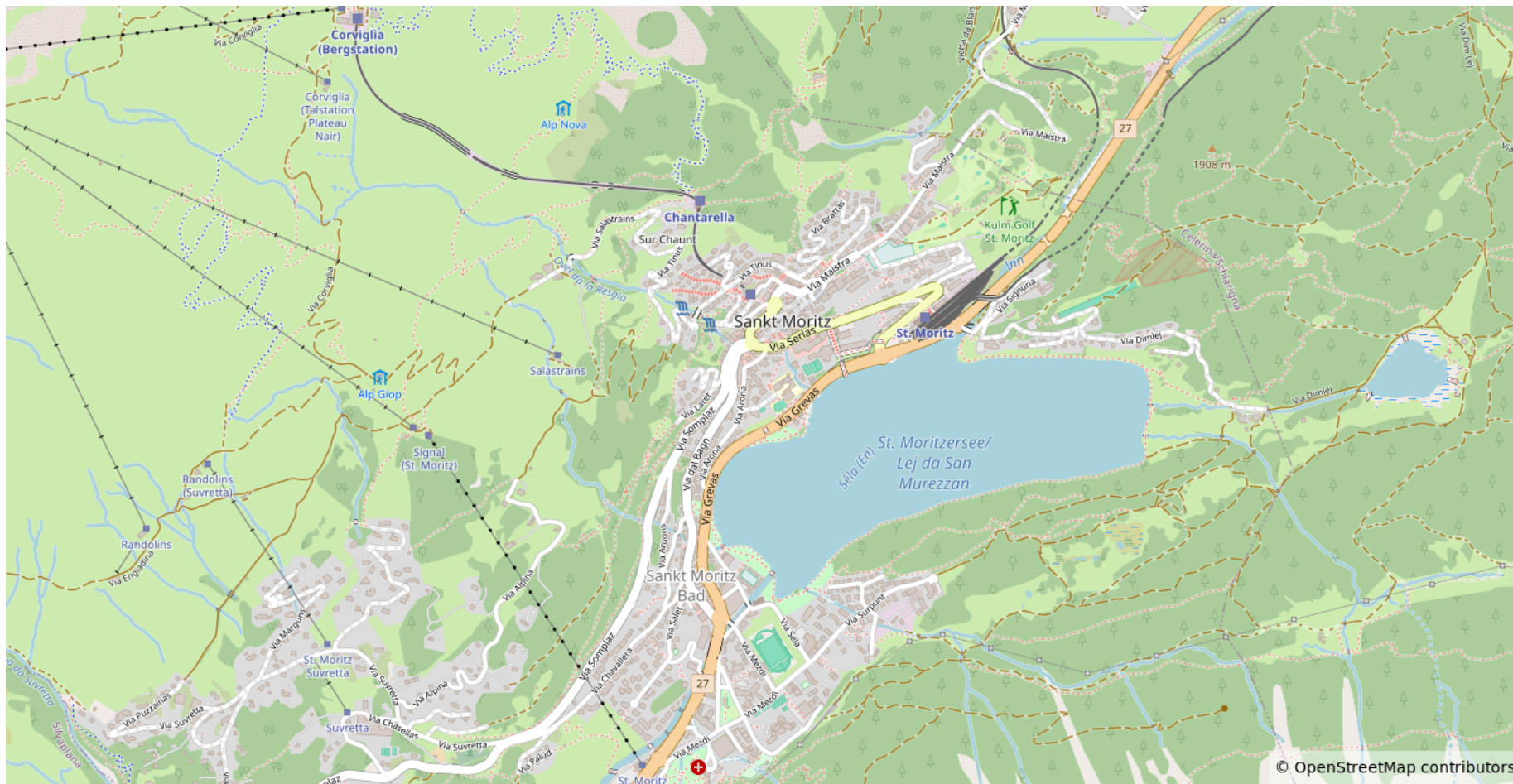
Présentation générale de l'édition des JO

Ces jeux Olympiques d'hiver de 1948 sont la cinquième édition des JO d'hiver.

Ces JO se passent principalement dans la ville de Sankt Moritz (Saint-Moritz) en Suisse mais quelques épreuves se déroulent dans la commune Suisse de Celerina.

Ces JO se déroulent du 30 janvier 1948 (date de la cérémonie d'ouverture) au 8 février 1948 (date de cérémonie de fermeture).

Ce n'est pas la première fois que la Suisse organise des JO puisqu'elle a déjà organisé la deuxième édition des JO d'hiver en 1928 dans la ville de Sankt Moritz.



[Lien grande carte](#)

Liste des sites des épreuves :

Nom	Sports	Lien carte
Corviglia	Ski alpin	Lien carte
Cresta Run	Bobsleigh / Luge / Skeleton	Lien carte
Französische Kirche	Pentathlon d'hiver	Lien carte
Kurhaus	Pentathlon d'hiver	Lien carte

Nom	Sports	Lien carte
Olympia Bobrun	Bobsleigh	Lien carte
Olympia-Eisstadion Badrutts Park	Patinage artistique / Hockey sur glace / Patinage de vitesse / Cérémonie d'ouverture/de fermeture	Lien carte
Olympiaschanze	Combiné Nordique	Lien carte
Palace-Hotel, St. Moritz	Hockey sur glace / Pentathlon d'hiver	Lien carte
Piz Nair	Ski alpin	Lien carte
Salastrains	Pentathlon d'hiver	Lien carte
Skistadion	Ski de fond / Patrouille de ski militaire / Patrouille de ski militaire / Combiné Nordique / Pentathlon d'hiver	Lien carte
Suvretta	Ski alpin	Lien carte
Suvretta-Haus	Hockey sur glace	Lien carte

Liste des épreuves :

- Ski alpin
- Patinage artistique
- Skeleton
- Bobsleigh
- Hockey sur glace
- Saut à ski
- Ski de fond
- Combiné Nordique
- Patinage de vitesse
- Patrouille de ski militaire
- Pentathlon d'hiver

Listes sports/catégories : voir requête SQL

Liste des nouveaux sports :

- Pentathlon d'hiver (seul jeux Olympiques à avoir ce sport)

Répartition par genre (épreuves H, F, mixtes...) : voir requête SQL

Liste des épreuves

```
In [7]: requete = f"""
SELECT DISTINCT NomDiscipline AS NomEpreuve
FROM {SCHEMA}.EVENEMENT E
JOIN {SCHEMA}.HOTE H ON H.idHote = E.idHote
JOIN {SCHEMA}.DISCIPLINE D ON D.CodeDiscipline = E.CodeDiscipline
WHERE libelleHote = (:libelle)
ORDER BY NomDiscipline
"""

data = requete_vers_dataframe(conn, requete, {"libelle":EDITION_JO})
print(data)
```

```
      NOMEPREUVE
0      Alpine Skiing
1      Bobsleigh
2  Cross Country Skiing
3      Figure Skating
4      Ice Hockey
5  Military Ski Patrol
6      Nordic Combined
7      Skeleton
8      Ski Jumping
9      Speed Skating
10     Winter Pentathlon
```

Liste des sports/catégories :

```
In [8]: requete = f"""
SELECT NomDiscipline, NomEvenement
FROM {SCHEMA}.EVENEMENT E
JOIN {SCHEMA}.HOTE H ON H.idHote = E.idHote
JOIN {SCHEMA}.DISCIPLINE D ON D.CodeDiscipline = E.CodeDiscipline
WHERE libelleHote = (:libelle)
ORDER BY NomDiscipline, NomEvenement DESC
"""
```

```
data = requete_vers_dataframe(conn, requete, {"libelle":EDITION_J0})
print(data)
```

	NOMDISCIPLINE	NOMEVENEMENT
0	Alpine Skiing	Slalom, Women
1	Alpine Skiing	Slalom, Men
2	Alpine Skiing	Downhill, Women
3	Alpine Skiing	Downhill, Men
4	Alpine Skiing	Combined, Women
5	Alpine Skiing	Combined, Men
6	Bobsleigh	Two, Men
7	Bobsleigh	Four, Men
8	Cross Country Skiing	50 kilometres, Men
9	Cross Country Skiing	4 x 10 kilometres Relay, Men
10	Cross Country Skiing	18 kilometres, Men
11	Figure Skating	Singles, Women
12	Figure Skating	Singles, Men
13	Figure Skating	Pairs, Mixed
14	Ice Hockey	Ice Hockey, Men
15	Military Ski Patrol	Military Ski Patrol, Men
16	Nordic Combined	Individual, Men
17	Skeleton	Skeleton, Men
18	Ski Jumping	Large Hill, Individual, Men
19	Speed Skating	500 metres, Men
20	Speed Skating	5,000 metres, Men
21	Speed Skating	10,000 metres, Men
22	Speed Skating	1,500 metres, Men
23	Winter Pentathlon	Winter Pentathlon, Men

Répartition par genre (épreuves H, F, mixtes...) :

```
In [9]: #Repartition epreuve mixte/homme/femme :
requete = f"""
SELECT
    SUM(CASE WHEN nomEvenement LIKE '%Men%' THEN 1 ELSE 0 END) as nb_epreuve_homme,
    SUM(CASE WHEN nomEvenement LIKE '%Women%' THEN 1 ELSE 0 END) as nb_epreuve_femme,
    SUM(CASE WHEN nomEvenement LIKE '%Mixed%' THEN 1 ELSE 0 END) as nb_epreuve_mixte,
    COUNT(*) as nb_total_epreuve
FROM {SCHEMA}.discipline d
INNER JOIN {SCHEMA}.EVENEMENT E ON d.CodeDiscipline = E.CodeDiscipline
INNER JOIN {SCHEMA}.HOTE H ON H.IdHote = E.IdHote
```

```

WHERE LibelleHote = '1948 Winter Olympics'
"""

nb_epreuve = requete_vers_dataframe(conn, requete).iloc[0]

counts = [nb_epreuve['NB_EPREUVE_HOMME'], nb_epreuve['NB_EPREUVE_FEMME'], nb_epreuve['NB_EPREUVE_MIXTE']]
labels = ['Epreuves hommes', 'Epreuves femmes', 'Epreuves mixtes']

colors = ['#3498db', '#e91e63', '#AAAAAA']

plt.figure(figsize=(8, 8))

plt.pie(
    counts,
    labels = labels,
    autopct='%1.1f%%',
    startangle=0,
    colors = colors
)

plt.title(f"Répartition des événements par genre (Total : {nb_epreuve['NB_TOTAL_EPREUVE']} épreuves)", fontsize=14)

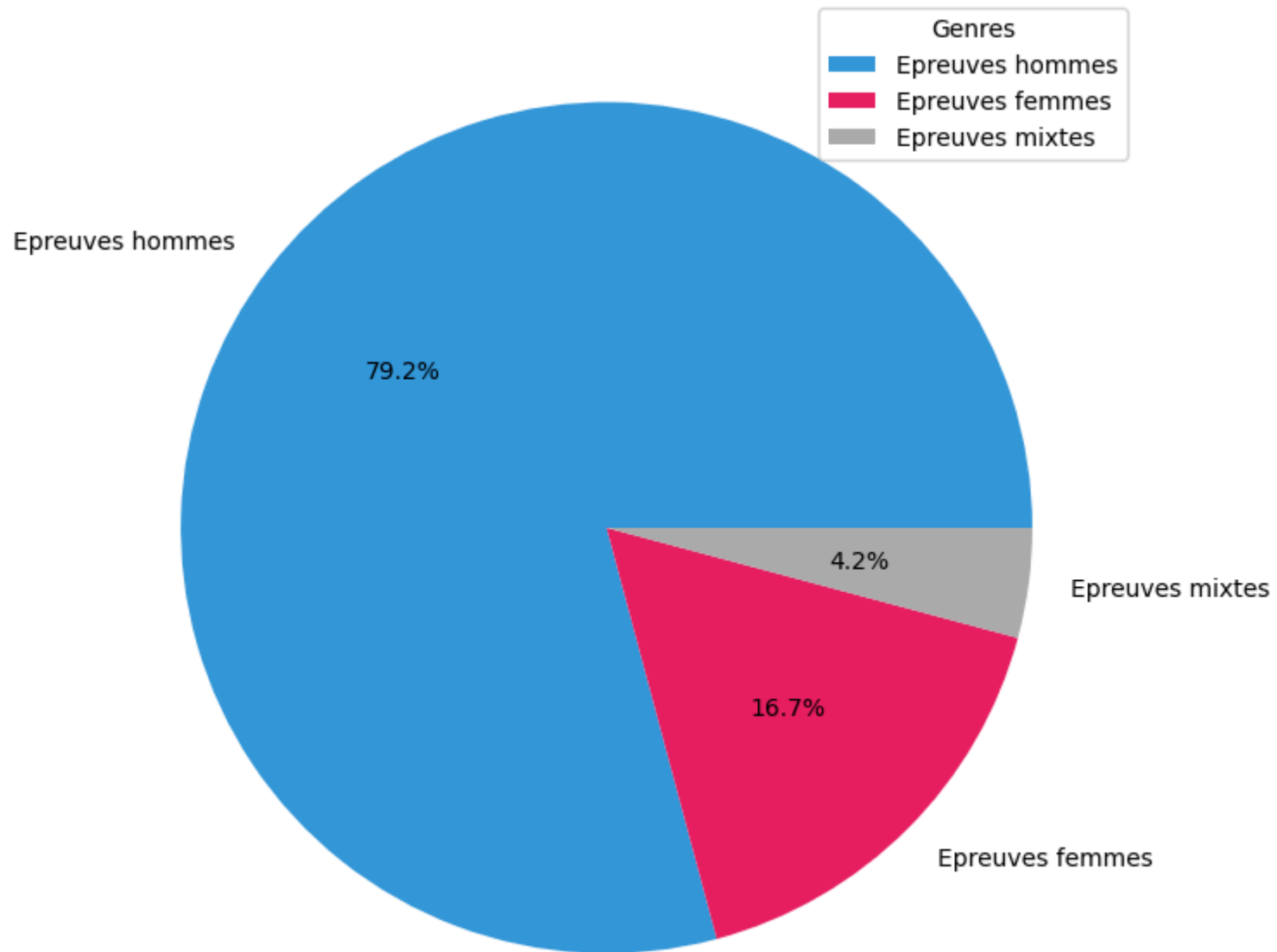
plt.legend(labels, loc="upper right", title="Genres")

plt.show

```

Out[9]: <function matplotlib.pyplot.show(close=None, block=None)>

Répartition des événements par genre (Total : 24 épreuves)



Comparaison avec les éditions précédentes

Classement des nations

```
In [10]: limite_inferieure = int(input("Afficher les résultats à partir de la ligne: "))
limite_superieure = int(input("Afficher les résultats jusqu'à la ligne (-1 pour toutes les lignes): "))

data_classement_nations = requete_vers_dataframe(conn, f"""
SELECT
    n.codeNOC AS NOC,
    n.nomNOC AS "Nom du NOC",

    NVL(i_total.nb_total, 0) + NVL(e_total.nb_total, 0) AS "Nombre de médailles",
    NVL(i_gold.nb_gold, 0) + NVL(e_gold.nb_gold, 0) AS "Médailles d'or",
    NVL(i_silver.nb_silver, 0) + NVL(e_silver.nb_silver, 0) AS "Médailles d'argent",
    NVL(i_bronze.nb_bronze, 0) + NVL(e_bronze.nb_bronze, 0) AS "Médailles de bronze"

FROM {SCHEMA}.NOC n

LEFT JOIN (
    SELECT
        pi.NOC,
        COUNT(*) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    WHERE pi.medaille IS NOT NULL
    GROUP BY pi.NOC
) i_total
ON i_total.NOC = n.codeNOC

LEFT JOIN (
    SELECT
        pi.NOC,
        COUNT(*) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    WHERE pi.medaille = 'Gold'
    GROUP BY pi.NOC
) i_gold
ON i_gold.NOC = n.codeNOC
```

```
LEFT JOIN (  
  SELECT  
    pi.NOC,  
    COUNT(*) AS nb_silver  
  FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi  
  WHERE pi.medaille = 'Silver'  
  GROUP BY pi.NOC  
) i_silver  
  ON i_silver.NOC = n.codeNOC
```

```
LEFT JOIN (  
  SELECT  
    pi.NOC,  
    COUNT(*) AS nb_bronze  
  FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi  
  WHERE pi.medaille = 'Bronze'  
  GROUP BY pi.NOC  
) i_bronze  
  ON i_bronze.NOC = n.codeNOC
```

```
LEFT JOIN (  
  SELECT  
    e.NOC,  
    COUNT(*) AS nb_total  
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe  
  JOIN {SCHEMA}.EQUIPE e  
    ON e.idEquipe = pe.idEquipe  
  WHERE pe.medaille IS NOT NULL  
  GROUP BY e.NOC  
) e_total  
  ON e_total.NOC = n.codeNOC
```

```
LEFT JOIN (  
  SELECT  
    e.NOC,  
    COUNT(*) AS nb_gold  
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe  
  JOIN {SCHEMA}.EQUIPE e  
    ON e.idEquipe = pe.idEquipe  
  WHERE pe.medaille = 'Gold'  
  GROUP BY e.NOC  
) e_gold
```

```

        ON e_gold.NOC = n.codeNOC

LEFT JOIN (
    SELECT
        e.NOC,
        COUNT(*) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    WHERE pe.medaille = 'Silver'
    GROUP BY e.NOC
) e_silver
    ON e_silver.NOC = n.codeNOC

LEFT JOIN (
    SELECT
        e.NOC,
        COUNT(*) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    WHERE pe.medaille = 'Bronze'
    GROUP BY e.NOC
) e_bronze
    ON e_bronze.NOC = n.codeNOC

ORDER BY "Nombre de médailles" DESC
""")

if limite_superieure == -1:
    limite_superieure = data_classement_nations.shape[0]

display(data_classement_nations.iloc[limite_inferieure:limite_superieure])

display(requete_vers_dataframe(conn, f"""
SELECT
    NVL(i_total.nb_total, 0) + NVL(e_total.nb_total, 0) AS "Total médailles",
    NVL(i_gold.nb_gold, 0) + NVL(e_gold.nb_gold, 0) AS "Total Or",
    NVL(i_silver.nb_silver, 0) + NVL(e_silver.nb_silver, 0) AS "Total Argent",
    NVL(i_bronze.nb_bronze, 0) + NVL(e_bronze.nb_bronze, 0) AS "Total Bronze"
FROM
    (

```

```

        SELECT COUNT(*) AS nb_total
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        WHERE pi.medaille IS NOT NULL
    ) i_total,
    (
        SELECT COUNT(*) AS nb_gold
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        WHERE pi.medaille = 'Gold'
    ) i_gold,
    (
        SELECT COUNT(*) AS nb_silver
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        WHERE pi.medaille = 'Silver'
    ) i_silver,
    (
        SELECT COUNT(*) AS nb_bronze
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        WHERE pi.medaille = 'Bronze'
    ) i_bronze,
    (
        SELECT COUNT(*) AS nb_total
        FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
        JOIN {SCHEMA}.EQUIPE e
            ON e.idEquipe = pe.idEquipe
        WHERE pe.medaille IS NOT NULL
    ) e_total,
    (
        SELECT COUNT(*) AS nb_gold
        FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
        JOIN {SCHEMA}.EQUIPE e
            ON e.idEquipe = pe.idEquipe
        WHERE pe.medaille = 'Gold'
    ) e_gold,
    (
        SELECT COUNT(*) AS nb_silver
        FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
        JOIN {SCHEMA}.EQUIPE e
            ON e.idEquipe = pe.idEquipe
        WHERE pe.medaille = 'Silver'
    ) e_silver,
    (
        SELECT COUNT(*) AS nb_bronze

```

```
FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
WHERE pe.medaille = 'Bronze'
) e_bronze
"""))
```

	NOC	Nom du NOC	Nombre de médailles	Médailles d'or	Médailles d'argent	Médailles de bronze
0	USA	United States	2983	1180	963	840
1	URS	Soviet Union	1204	473	376	355
2	GER	Germany	1076	348	370	358
3	GBR	Great Britain	961	304	326	331
4	FRA	France	910	271	298	341
5	ITA	Italy	772	263	238	271
6	CHN	People's Republic of China	713	285	231	197
7	SWE	Sweden	683	214	228	241
8	JPN	Japan	575	186	178	211
9	NOR	Norway	567	207	187	173
10	AUS	Australia	560	168	177	215
11	RUS	Russian Federation	551	194	169	188
12	CAN	Canada	549	147	183	219
13	HUN	Hungary	524	184	157	183
14	GDR	East Germany	519	192	165	162
15	FIN	Finland	485	149	151	185
16	NED	Netherlands	473	150	154	169
17	SUI	Switzerland	377	118	129	130
18	KOR	Republic of Korea	366	129	121	116
19	AUT	Austria	352	93	125	134
20	POL	Poland	327	81	97	149
21	ROU	Romania	309	90	97	122
22	FRG	West Germany	243	67	82	94

	NOC	Nom du NOC	Nombre de médailles	Médailles d'or	Médailles d'argent	Médailles de bronze
23	CUB	Cuba	235	84	69	82
24	BUL	Bulgaria	230	55	90	85

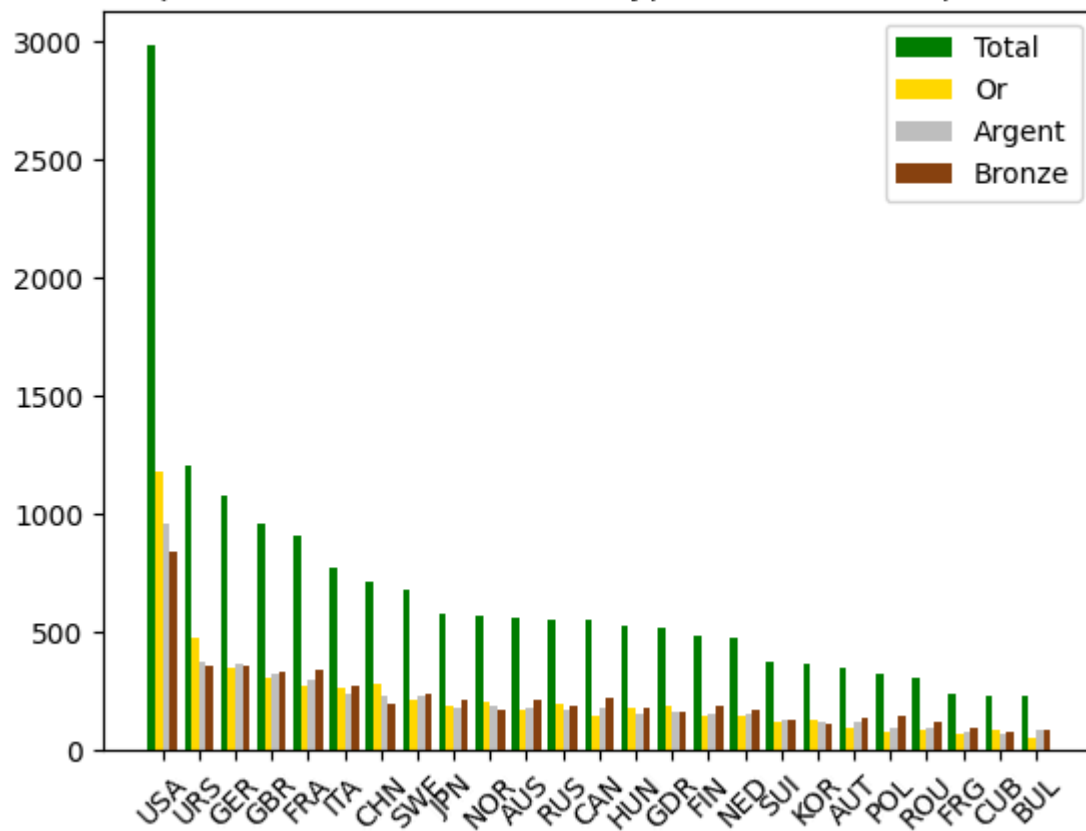
	Total médailles	Total Or	Total Argent	Total Bronze
0	20389	6663	6636	7090

Diagramme à barres (max de 25 lignes pour un résultat lisible)

```
In [11]: array_classement_medailles = data_classement_nations["Nombre de médailles"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_classement_or = data_classement_nations["Médailles d'or"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_classement_argent = data_classement_nations["Médailles d'argent"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_classement_bronze = data_classement_nations["Médailles de bronze"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_classement_noc = data_classement_nations["NOC"].iloc[limite_inferieure:limite_superieure].to_numpy()

x = np.arange(len(array_classement_noc))
width = 0.2
plt.bar(x - 1.5*width, array_classement_medailles, width, label='Total', color='green')
plt.bar(x - 0.5*width, array_classement_or, width, label='Or', color='gold')
plt.bar(x + 0.5*width, array_classement_argent, width, label='Argent', color='silver')
plt.bar(x + 1.5*width, array_classement_bronze, width, label='Bronze', color='saddlebrown')
plt.title("Répartition du nombre et des types de médailles par NOC")
plt.xticks(x, array_classement_noc, rotation=45)
plt.legend()
plt.show()
```

Répartition du nombre et des types de médailles par NOC



On remarque que les USA dominent largement le classement. Ils possèdent quasiment autant de médailles d'or qu'en possédait l'URSS au total, et près de 15% de toutes les médailles obtenues. De plus, les médailles sont principalement réparties sur les pays les plus développés; on ne remarque pas d'apparition de pays en voie de développement dans le haut de la liste, ce qui pourrait impliquer un lien de causalité entre ces deux facteurs. En comparant le haut et le bas du classement, on peut remarquer que plus un pays possède de médailles, plus le total de chaque type remporté par ce dernier tend à s'uniformiser (c'est-à-dire qu'il n'y a pas ou peu de différence entre le total de chaque type pour un pays donné), alors qu'au contraire, moins un pays en possède, plus le total de chaque type est hétérogène. Avec la conclusion précédente, on en tire que les pays développés remportent régulièrement des médailles, peu importe le type, alors qu'à l'inverse, les pays en développement ou sous-développés n'ont que des médailles à titre exceptionnel, et leur distribution selon leur type est peu régulière en conséquence.

Répartition par genre et pays

Note: si une équipe possède des membres des deux genres, la médaille est comptée pour chaque genre, donc les valeurs peuvent différer de l'analyse précédente

```
In [12]: limite_inferieure = int(input("Afficher les résultats à partir de la ligne: "))
limite_superieure = int(input("Afficher les résultats jusqu'à la ligne (-1 pour toutes les lignes): "))
data_classement_genre = requete_vers_dataframe(conn, f"""
SELECT
    n.codeNOC AS NOC,
    n.nomNOC AS "Nom du NOC",

    NVL(hm.nb_total, 0) + NVL(hme.nb_total, 0) AS "Médailles (Homme)",
    NVL(hmg.nb_gold, 0) + NVL(hmeg.nb_gold, 0) AS "Or (Homme)",
    NVL(hms.nb_silver, 0) + NVL(hmes.nb_silver, 0) AS "Argent (Homme)",
    NVL(hmb.nb_bronze, 0) + NVL(hmeb.nb_bronze, 0) AS "Bronze (Homme)",

    NVL(fm.nb_total, 0) + NVL(fme.nb_total, 0) AS "Médailles (Femme)",
    NVL(fmg.nb_gold, 0) + NVL(fmeg.nb_gold, 0) AS "Or (Femme)",
    NVL(fms.nb_silver, 0) + NVL(fmes.nb_silver, 0) AS "Argent (Femme)",
    NVL(fmb.nb_bronze, 0) + NVL(fmeb.nb_bronze, 0) AS "Bronze (Femme)"

FROM {SCHEMA}.NOC n

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille IS NOT NULL
        AND a.genre = 'Male'
    GROUP BY pi.NOC
) hm ON hm.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Gold'
        AND a.genre = 'Male'
    GROUP BY pi.NOC
) hmg ON hmg.NOC = n.codeNOC
```

```

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Silver'
        AND a.genre = 'Male'
    GROUP BY pi.NOC
) hms ON hms.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Bronze'
        AND a.genre = 'Male'
    GROUP BY pi.NOC
) hmb ON hmb.NOC = n.codeNOC

LEFT JOIN (
    SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille IS NOT NULL
        AND a.genre = 'Male'
    GROUP BY e.NOC
) hme ON hme.NOC = n.codeNOC

LEFT JOIN (
    SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe

```

```

JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = ce.idAthlete
WHERE pe.medaille = 'Gold'
    AND a.genre = 'Male'
GROUP BY e.NOC
) hmeg ON hmeg.NOC = n.codeNOC

LEFT JOIN (
    SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Silver'
        AND a.genre = 'Male'
    GROUP BY e.NOC
) hmes ON hmes.NOC = n.codeNOC

LEFT JOIN (
    SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Bronze'
        AND a.genre = 'Male'
    GROUP BY e.NOC
) hmeb ON hmeb.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille IS NOT NULL
        AND a.genre = 'Female'

```

```

        GROUP BY pi.NOC
    ) fm ON fm.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Gold'
        AND a.genre = 'Female'
    GROUP BY pi.NOC
) fmg ON fmg.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Silver'
        AND a.genre = 'Female'
    GROUP BY pi.NOC
) fms ON fms.NOC = n.codeNOC

LEFT JOIN (
    SELECT pi.NOC, COUNT(*) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Bronze'
        AND a.genre = 'Female'
    GROUP BY pi.NOC
) fmb ON fmb.NOC = n.codeNOC

LEFT JOIN (
    SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete

```

```

WHERE pe.medaille IS NOT NULL
      AND a.genre = 'Female'
GROUP BY e.NOC
) fme ON fme.NOC = n.codeNOC

LEFT JOIN (
  SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_gold
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
  JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
  JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
    ON ce.idEquipe = e.idEquipe
  JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = ce.idAthlete
  WHERE pe.medaille = 'Gold'
      AND a.genre = 'Female'
  GROUP BY e.NOC
) fmeg ON fmeg.NOC = n.codeNOC

LEFT JOIN (
  SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_silver
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
  JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
  JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
    ON ce.idEquipe = e.idEquipe
  JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = ce.idAthlete
  WHERE pe.medaille = 'Silver'
      AND a.genre = 'Female'
  GROUP BY e.NOC
) fmes ON fmes.NOC = n.codeNOC

LEFT JOIN (
  SELECT e.NOC, COUNT(DISTINCT e.idEquipe) AS nb_bronze
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
  JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
  JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
    ON ce.idEquipe = e.idEquipe
  JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = ce.idAthlete

```

```

        WHERE pe.medaille = 'Bronze'
              AND a.genre = 'Female'
        GROUP BY e.NOC
    ) fmeb ON fmeb.NOC = n.codeNOC

    ORDER BY "Médailles (Homme)" DESC
""")
if limite_superieure == -1:
    limite_superieure = data_classement_nations.shape[0]
display(data_classement_genre.iloc[limite_inferieure:limite_superieure])
display(requete_vers_dataframe(conn, f"""
SELECT
    hm.nb_total + hme.nb_total AS "Total médailles (Homme)",
    hmg.nb_gold + hmeg.nb_gold AS "Total Or (Homme)",
    hms.nb_silver + hmes.nb_silver AS "Total Argent (Homme)",
    hmb.nb_bronze + hmeb.nb_bronze AS "Total Bronze (Homme)",

    fm.nb_total + fme.nb_total AS "Total médailles (Femme)",
    fmg.nb_gold + fmeg.nb_gold AS "Total Or (Femme)",
    fms.nb_silver + fmes.nb_silver AS "Total Argent (Femme)",
    fmb.nb_bronze + fmeb.nb_bronze AS "Total Bronze (Femme)"
FROM
    (
        SELECT COUNT(*) AS nb_total
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        JOIN {SCHEMA}.ATHLETE a
            ON a.idAthlete = pi.idAthlete
        WHERE pi.medaille IS NOT NULL
              AND a.genre = 'Male'
    ) hm,
    (
        SELECT COUNT(*) AS nb_gold
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        JOIN {SCHEMA}.ATHLETE a
            ON a.idAthlete = pi.idAthlete
        WHERE pi.medaille = 'Gold'
              AND a.genre = 'Male'
    ) hmg,
    (
        SELECT COUNT(*) AS nb_silver
        FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
        JOIN {SCHEMA}.ATHLETE a

```

```

        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Silver'
        AND a.genre = 'Male'
) hms,
(
    SELECT COUNT(*) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Bronze'
        AND a.genre = 'Male'
) hmb,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille IS NOT NULL
        AND a.genre = 'Male'
) hme,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Gold'
        AND a.genre = 'Male'
) hmeg,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce

```

```

        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Silver'
        AND a.genre = 'Male'
) hmes,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Bronze'
        AND a.genre = 'Male'
) hmeb,

(
    SELECT COUNT(*) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille IS NOT NULL
        AND a.genre = 'Female'
) fmf,
(
    SELECT COUNT(*) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Gold'
        AND a.genre = 'Female'
) fmg,
(
    SELECT COUNT(*) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Silver'
        AND a.genre = 'Female'
)

```

```

) fms,
(
    SELECT COUNT(*) AS nb_bronze
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = pi.idAthlete
    WHERE pi.medaille = 'Bronze'
        AND a.genre = 'Female'
) fmb,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_total
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille IS NOT NULL
        AND a.genre = 'Female'
) fme,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_gold
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE pe.medaille = 'Gold'
        AND a.genre = 'Female'
) fmeg,
(
    SELECT COUNT(DISTINCT e.idEquipe) AS nb_silver
    FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
    JOIN {SCHEMA}.EQUIPE e
        ON e.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
        ON ce.idEquipe = e.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete

```

```
WHERE pe.medaille = 'Silver'
      AND a.genre = 'Female'
) fmes,
(
  SELECT COUNT(DISTINCT e.idEquipe) AS nb_bronze
  FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
  JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
  JOIN {SCHEMA}.COMPOSITION_EQUIPE ce
    ON ce.idEquipe = e.idEquipe
  JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = ce.idAthlete
  WHERE pe.medaille = 'Bronze'
      AND a.genre = 'Female'
) fmeb
""")
```

	NOC	Nom du NOC	Médailles (Homme)	Or (Homme)	Argent (Homme)	Bronze (Homme)	Médailles (Femme)	Or (Femme)	Argent (Femme)	Bronze (Femme)
0	USA	United States	2124	844	690	590	907	347	291	269
1	URS	Soviet Union	872	351	279	242	355	134	105	116
2	FRA	France	739	227	235	277	190	53	68	69
3	GBR	Great Britain	732	244	250	238	260	71	89	100
4	GER	Germany	709	236	228	245	392	126	146	120
5	ITA	Italy	617	219	188	210	163	46	53	64
6	SWE	Sweden	572	181	187	204	117	34	43	40
7	NOR	Norway	467	174	155	138	109	35	36	38
8	FIN	Finland	423	136	130	157	64	14	22	28
9	JPN	Japan	409	133	129	147	170	54	50	66
10	HUN	Hungary	406	139	124	143	125	45	35	45
11	AUS	Australia	334	94	110	130	232	76	69	87
12	RUS	Russian Federation	326	117	85	124	243	86	91	66
13	CAN	Canada	325	88	110	127	244	65	79	100
14	CHN	People's Republic of China	310	127	96	87	426	167	143	116
15	SUI	Switzerland	294	90	105	99	95	31	31	33
16	GDR	East Germany	278	98	81	99	245	94	85	66
17	NED	Netherlands	258	73	89	96	221	77	69	75
18	AUT	Austria	252	68	88	96	110	28	41	41
19	POL	Poland	246	65	69	112	82	17	28	37
20	KOR	Republic of Korea	236	74	82	80	134	58	40	36
21	CUB	Cuba	180	71	50	59	55	13	19	23

	NOC	Nom du NOC	Médailles (Homme)	Or (Homme)	Argent (Homme)	Bronze (Homme)	Médailles (Femme)	Or (Femme)	Argent (Femme)	Bronze (Femme)
22	FRG	West Germany	178	47	61	70	71	23	22	26
23	DEN	Denmark	169	37	69	63	49	11	15	23
24	BUL	Bulgaria	161	40	62	59	69	15	28	26
		Total médailles (Homme)	Total Or (Homme)	Total Argent (Homme)	Total Bronze (Homme)	Total médailles (Femme)	Total Or (Femme)	Total Argent (Femme)	Total Bronze (Femme)	
0		14378	4699	4667	5012	6348	2077	2087	2184	

Diagramme à barres (max de 25 lignes pour un résultat lisible)

```
In [13]: array_h_or = data_classement_genre["Or (Homme)"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_h_argent = data_classement_genre["Argent (Homme)"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_h_bronze = data_classement_genre["Bronze (Homme)"].iloc[limite_inferieure:limite_superieure].to_numpy()

array_f_or = data_classement_genre["Or (Femme)"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_f_argent = data_classement_genre["Argent (Femme)"].iloc[limite_inferieure:limite_superieure].to_numpy()
array_f_bronze = data_classement_genre["Bronze (Femme)"].iloc[limite_inferieure:limite_superieure].to_numpy()

array_noc = data_classement_genre["NOC"].iloc[limite_inferieure:limite_superieure].to_numpy()

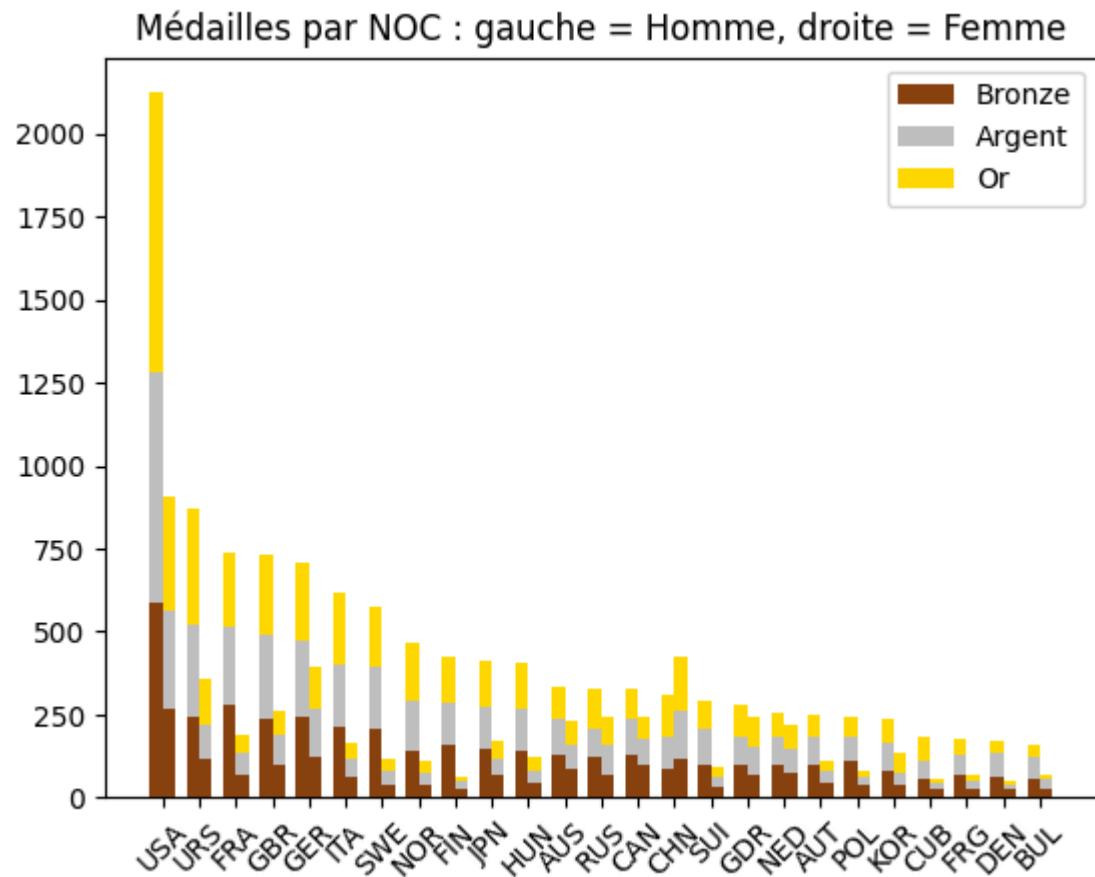
x = np.arange(len(array_noc))
width = 0.35

plt.bar(x - width/2, array_h_bronze, width, color='saddlebrown', label='Bronze')
plt.bar(x - width/2, array_h_argent, width, bottom=array_h_bronze, color='silver', label='Argent')
plt.bar(x - width/2, array_h_or, width, bottom=array_h_bronze + array_h_argent, color='gold', label='Or')

plt.bar(x + width/2, array_f_bronze, width, color='saddlebrown')
plt.bar(x + width/2, array_f_argent, width, bottom=array_f_bronze, color='silver')
plt.bar(x + width/2, array_f_or, width, bottom=array_f_bronze + array_f_argent, color='gold')

plt.xticks(x, array_noc, rotation=45)
plt.title("Médailles par NOC : gauche = Homme, droite = Femme")
```

```
plt.legend()  
plt.show()
```



Parmi les 25 premiers pays:

- Hommes:
 - Somme de toutes les médailles: 11617
 - Moyenne: $11617/25 \approx 464,68$ médailles par pays
 - Médiane: 326 (valeur du milieu de la série)
 - Quartile 25%: 252 (même raisonnement que ci-dessus)
 - Quartile 75%: 572
 - Minimum: 161

- Maximum: 2124
- Écart-type: $\approx 400,17$
- Femmes:
 - Somme de toutes les médailles: 5128
 - Moyenne: $5128/25 \approx 205,12$ médailles par pays
 - Médiane: 163
 - Quartile 25%: 95
 - Quartile 75%: 244
 - Minimum: 49
 - Maximum: 907
 - Écart-type: $\approx 180,13$
- Comparaison:
 - Total des médailles: $11617 + 5128 = 16745$
 - Part d'hommes: $11617/16745 \approx 69,4\%$
 - Part de femmes: $5128/16745 \approx 30,6\%$
 - Écart homme/femme: $11617 - 5128 = 6489$
 - Rapport homme/femme: $5128/11617 \approx 0,441$

Les femmes représentent donc 30,6% des médailles remportées, et constituent l'équivalent de 44,1% du volume masculin

Les hommes remportent globalement plus de médailles que les femmes dans les pays du top 25. Cela se voit d'abord par la moyenne : 464,68 chez les hommes contre 205,12 chez les femmes. La médiane confirme ce décalage : 326 pour les hommes contre 163 pour les femmes.

Autrement dit, pour presque n'importe quel pays, le total masculin est environ deux fois plus élevé que le total féminin. La dispersion est aussi beaucoup plus forte chez les hommes. L'écart-type masculin vaut 400,17, contre 180,13 chez les femmes. Les hommes présentent donc des écarts plus marqués entre pays. Les quartiles vont dans le même sens : la moitié centrale des valeurs masculines est comprise entre 252 et 572, alors que pour les femmes elle se situe entre 95 et 244. Enfin, les extrêmes sont nettement plus élevés chez les hommes : de 161 à 2124, contre 49 à 907 chez les femmes. On peut donc conclure que, dans cet ensemble, les résultats masculins sont plus élevés en niveau et plus dispersés, tandis que les résultats féminins sont plus faibles et plus resserrés.

Exemple particulier: la Chine

Il y a 310 médailles chez les hommes et 426 médailles chez les femmes en Chine. La part féminine y est de $426/736 \approx 57,9\%$. C'est le seul pays de cet échantillon qui s'oppose à la tendance générale et où les femmes y dépassent les hommes. Cela pourrait s'expliquer par des politiques spécifiques à la Chine, car les femmes démontrant un talent sportif sont encouragées à en faire leur carrière, plutôt que rejetées à cause de leur genre autre part.

Évolution à long terme du Top 3

On considère le top 3 comme étant les États-Unis (USA), l'Allemagne (GER) et la Grande Bretagne (GBR) car l'URSS n'existait pas à l'époque de ces jeux. Les éditions de 1940 et 1944 ne sont pas représentées dans les sections suivantes car annulées à cause de la guerre.

```
In [14]: data_top3_evolution = requete_vers_dataframe(conn, f"""
WITH hivers AS (
    SELECT h.idHote, h.libelleHote
    FROM {SCHEMA}.HOTE h
    WHERE (h.idHote <= 33 AND h.libelleHote LIKE '% Winter %')
),
nocs AS (
    SELECT n.codeNOC, n.nomNOC
    FROM {SCHEMA}.NOC n
    WHERE n.codeNOC IN ('USA', 'GBR', 'GER')
),
medailles AS (
    SELECT
        pi.NOC,
        ev.idHote,
        pi.medaille
    FROM {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    JOIN {SCHEMA}.EVENEMENT ev
        ON ev.idEvenement = pi.idEvent
    JOIN hivers h
        ON h.idHote = ev.idHote
    WHERE pi.medaille IS NOT NULL

    UNION ALL

    SELECT
        e.NOC,
        ev.idHote,
        pe.medaille
```

```

FROM {SCHEMA}.PARTICIPATION_EQUIPE pe
JOIN {SCHEMA}.EQUIPE e
    ON e.idEquipe = pe.idEquipe
JOIN {SCHEMA}.EVENEMENT ev
    ON ev.idEvenement = pe.idEvenement
JOIN hivers h
    ON h.idHote = ev.idHote
WHERE pe.medaille IS NOT NULL
),
medailles_agregees AS (
    SELECT
        m.NOC,
        m.idHote,
        COUNT(*) AS nb_total,
        SUM(CASE WHEN m.medaille = 'Gold' THEN 1 ELSE 0 END) AS nb_gold,
        SUM(CASE WHEN m.medaille = 'Silver' THEN 1 ELSE 0 END) AS nb_silver,
        SUM(CASE WHEN m.medaille = 'Bronze' THEN 1 ELSE 0 END) AS nb_bronze
    FROM {SCHEMA}.medailles m
    GROUP BY m.NOC, m.idHote
)
SELECT
    n.codeNOC AS NOC,
    n.nomNOC AS "Nom du NOC",
    h.libelleHote AS "Jeux d'hiver",

    NVL(ma.nb_total, 0) AS "Nombre de médailles",
    NVL(ma.nb_gold, 0) AS "Médailles d'or",
    NVL(ma.nb_silver, 0) AS "Médailles d'argent",
    NVL(ma.nb_bronze, 0) AS "Médailles de bronze"

FROM nocs n
CROSS JOIN hivers h
LEFT JOIN medailles_agregees ma
    ON ma.NOC = n.codeNOC
    AND ma.idHote = h.idHote

ORDER BY
    n.codeNOC DESC,
    h.idHote
)
)
display(data_top3_evolution.style)

```

	NOC	Nom du NOC	Jeux d'hiver	Nombre de médailles	Médailles d'or	Médailles d'argent	Médailles de bronze
0	USA	United States	1924 Winter Olympics	4	1	2	1
1	USA	United States	1928 Winter Olympics	6	2	2	2
2	USA	United States	1932 Winter Olympics	12	6	4	2
3	USA	United States	1936 Winter Olympics	4	1	0	3
4	USA	United States	1948 Winter Olympics	9	3	4	2
5	GER	Germany	1924 Winter Olympics	0	0	0	0
6	GER	Germany	1928 Winter Olympics	1	0	0	1
7	GER	Germany	1932 Winter Olympics	2	0	0	2
8	GER	Germany	1936 Winter Olympics	6	3	3	0
9	GER	Germany	1948 Winter Olympics	0	0	0	0
10	GBR	Great Britain	1924 Winter Olympics	4	1	1	2
11	GBR	Great Britain	1928 Winter Olympics	1	0	0	1
12	GBR	Great Britain	1932 Winter Olympics	0	0	0	0
13	GBR	Great Britain	1936 Winter Olympics	3	1	1	1
14	GBR	Great Britain	1948 Winter Olympics	2	0	0	2

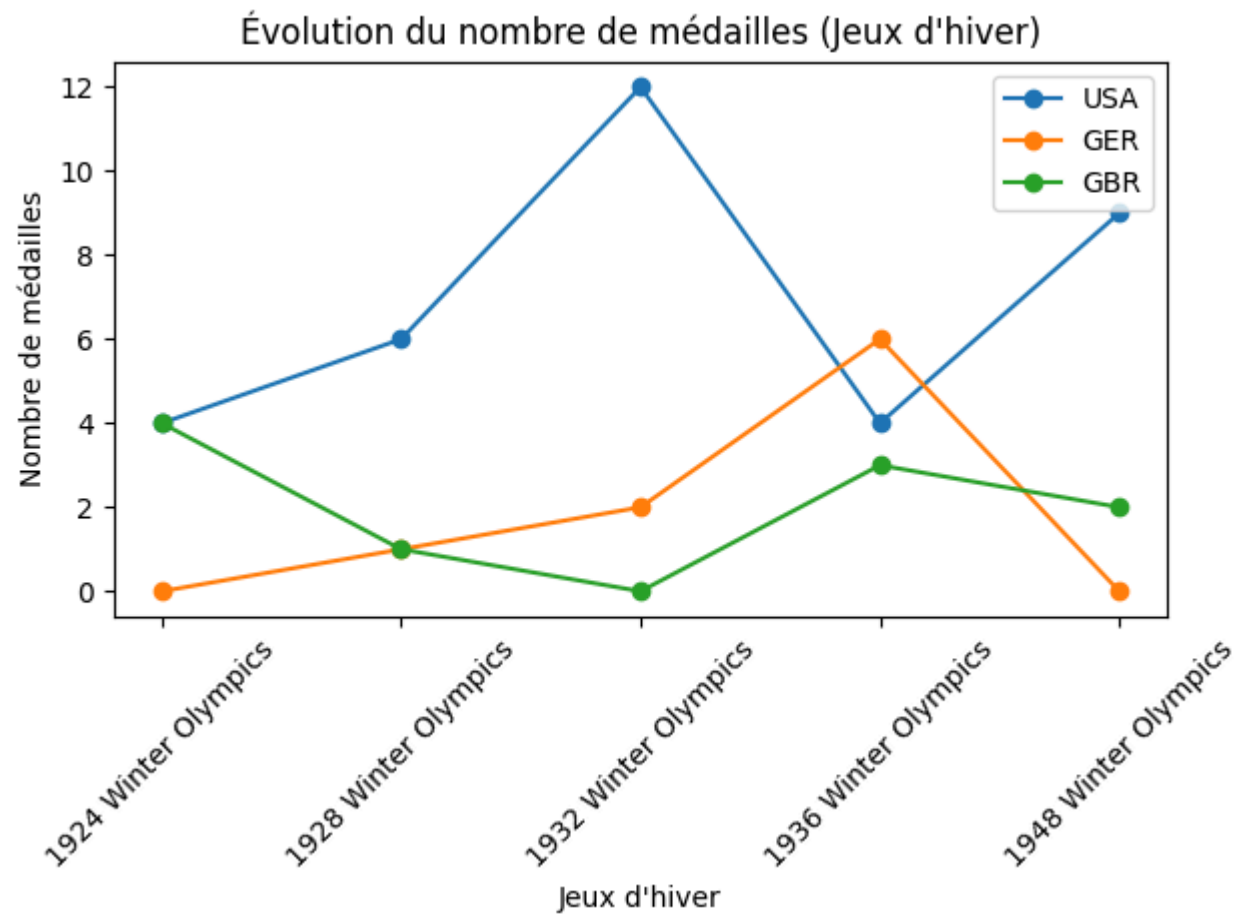
```
In [15]: plt.figure()

for noc in data_top3_evolution['NOC'].unique():
    subset = data_top3_evolution[data_top3_evolution['NOC'] == noc]
    plt.plot(subset["Jeux d'hiver"],
             subset["Nombre de médailles"],
             marker='o',
             label=noc)

plt.xticks(rotation=45)
plt.title("Évolution du nombre de médailles (Jeux d'hiver)")
plt.xlabel("Jeux d'hiver")
plt.ylabel("Nombre de médailles")
```

```
plt.legend()

plt.tight_layout()
plt.show()
```



```
In [16]: pivot = data_top3_evolution.pivot(index="NOC",
      columns="Jeux d'hiver",
      values="Nombre de médailles").fillna(0)

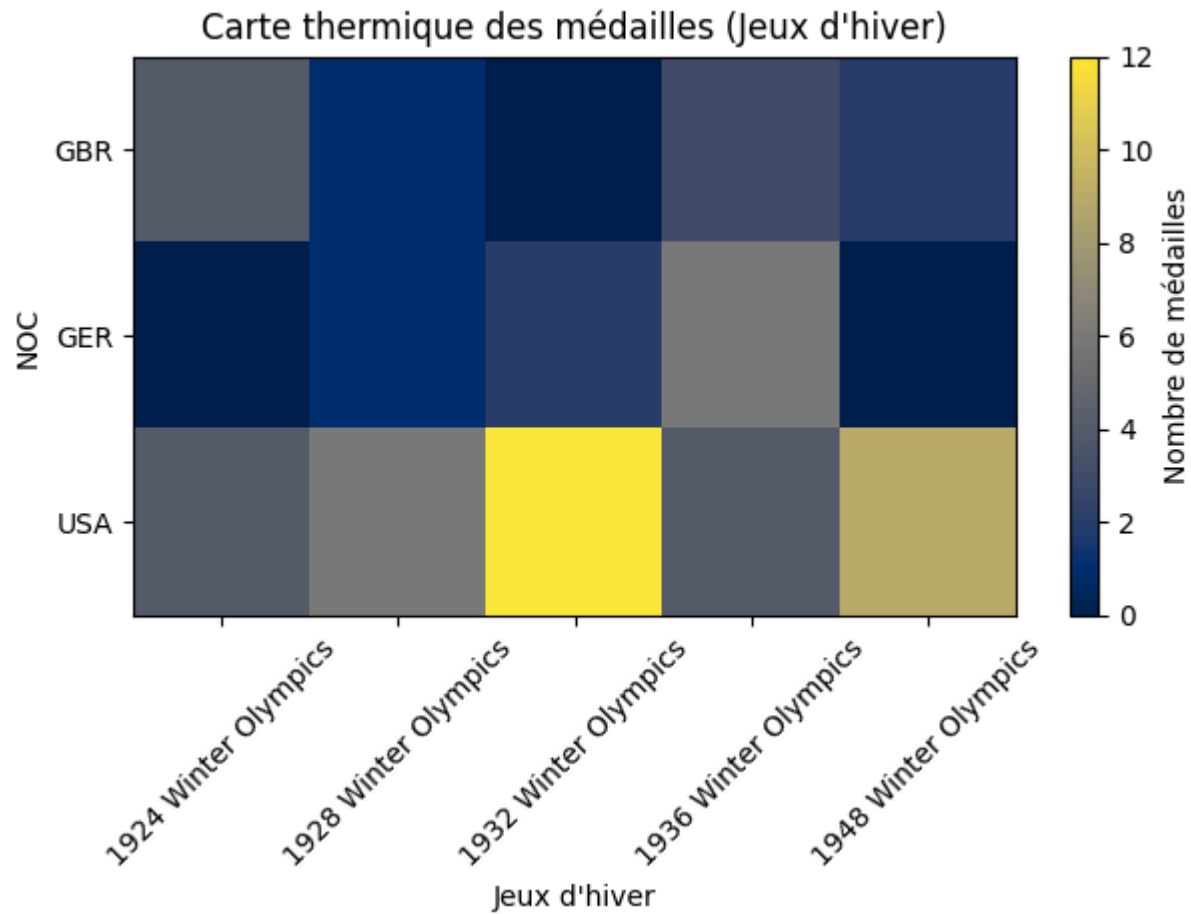
plt.figure()
plt.imshow(pivot.values, aspect='auto', cmap='cividis')

plt.xticks(range(len(pivot.columns)), pivot.columns, rotation=45)
plt.yticks(range(len(pivot.index)), pivot.index)
```

```
plt.title("Carte thermique des médailles (Jeux d'hiver)")
plt.xlabel("Jeux d'hiver")
plt.ylabel("NOC")

plt.colorbar(label="Nombre de médailles")

plt.tight_layout()
plt.show()
```



Les États-Unis sont le pays le plus performant, mais surtout le plus capable de convertir une hausse de volume en domination réelle : en 1932, leur total explose et s'accompagne d'une forte proportion d'or, ce qui suggère une capacité à transformer les finales en victoires. À l'inverse,

leur baisse de 1936 montre une dépendance à certaines éditions particulièrement favorables, ce qui nuance l'idée d'une suprématie totalement stable.

L'Allemagne suit une logique différente : sa progression jusqu'en 1936 est nette, mais elle repose sur une montée brusque et concentrée dans le temps. De plus, son pic de 1936 repose entièrement sur l'or et l'argent, sans bronze, ce qui indique une polarisation des résultats : soit les athlètes allemands se placent très haut, soit ils ne montent pas sur le podium.

La Grande-Bretagne, en revanche, présente le profil le moins compétitif au sens strict : ses médailles sont relativement peu nombreuses et souvent peu prestigieuses, car elles se convertissent rarement en or. Cela suggère une présence ponctuelle dans certaines disciplines, mais sans profondeur suffisante pour soutenir une performance d'ensemble.

Moyenne d'âge des athlètes

```
In [17]: data_moyenne_age = requete_vers_dataframe(conn, f"""
WITH participants AS (
    SELECT DISTINCT
        h.idHote,
        h.libelleHote,
        a.idAthlete,
        TRUNC(
            MONTHS_BETWEEN(
                TO_DATE('01-02-' || h.anneeHote, 'DD-MM-YYYY'),
                a.dateNaissance
            ) / 12
        ) AS age
    FROM {SCHEMA}.HOTE h
    JOIN {SCHEMA}.EVENEMENT e ON e.idHote = h.idHote
    JOIN {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi ON pi.idEvent = e.idEvenement
    JOIN {SCHEMA}.ATHLETE a ON a.idAthlete = pi.idAthlete
    WHERE h.idHote <= 33
        AND h.libelleHote LIKE '% Winter %'
        AND a.dateNaissance IS NOT NULL

    UNION

    SELECT DISTINCT
        h.idHote,
        h.libelleHote,
        a.idAthlete,
```

```

        TRUNC(
            MONTHS_BETWEEN(
                TO_DATE('01-02-' || h.anneeHote, 'DD-MM-YYYY'),
                a.dateNaissance
            ) / 12
        ) AS age
FROM {SCHEMA}.HOTE h
JOIN {SCHEMA}.EVENEMENT e ON e.idHote = h.idHote
JOIN {SCHEMA}.PARTICIPATION_EQUIPE pe ON pe.idEvenement = e.idEvenement
JOIN {SCHEMA}.COMPOSITION_EQUIPE ce ON ce.idEquipe = pe.idEquipe
JOIN {SCHEMA}.ATHLETE a ON a.idAthlete = ce.idAthlete
WHERE h.idHote <= 33
      AND h.libelleHote LIKE '% Winter %'
      AND a.dateNaissance IS NOT NULL
)
SELECT
    libelleHote AS "Jeux d'hiver",
    ROUND(AVG(age), 2) AS "Âge moyen"
FROM participants
GROUP BY libelleHote
ORDER BY libelleHote
""")
display(data_moyenne_age)

```

	Jeux d'hiver	Âge moyen
0	1924 Winter Olympics	28.26
1	1928 Winter Olympics	26.75
2	1932 Winter Olympics	28.10
3	1936 Winter Olympics	25.09
4	1948 Winter Olympics	26.63

```

In [18]: data_moyenne_age = data_moyenne_age.sort_values(by="Jeux d'hiver").reset_index(drop=True)

x_numeric = np.arange(len(data_moyenne_age))
y = data_moyenne_age["Âge moyen"].values

coeffs = np.polyfit(x_numeric, y, 1)

```

```
trend = np.poly1d(coeffs)

plt.figure()

plt.plot(x_numeric, y, marker='o', label="Âge moyen")

plt.plot(x_numeric, trend(x_numeric), linestyle='--', label="Tendance")

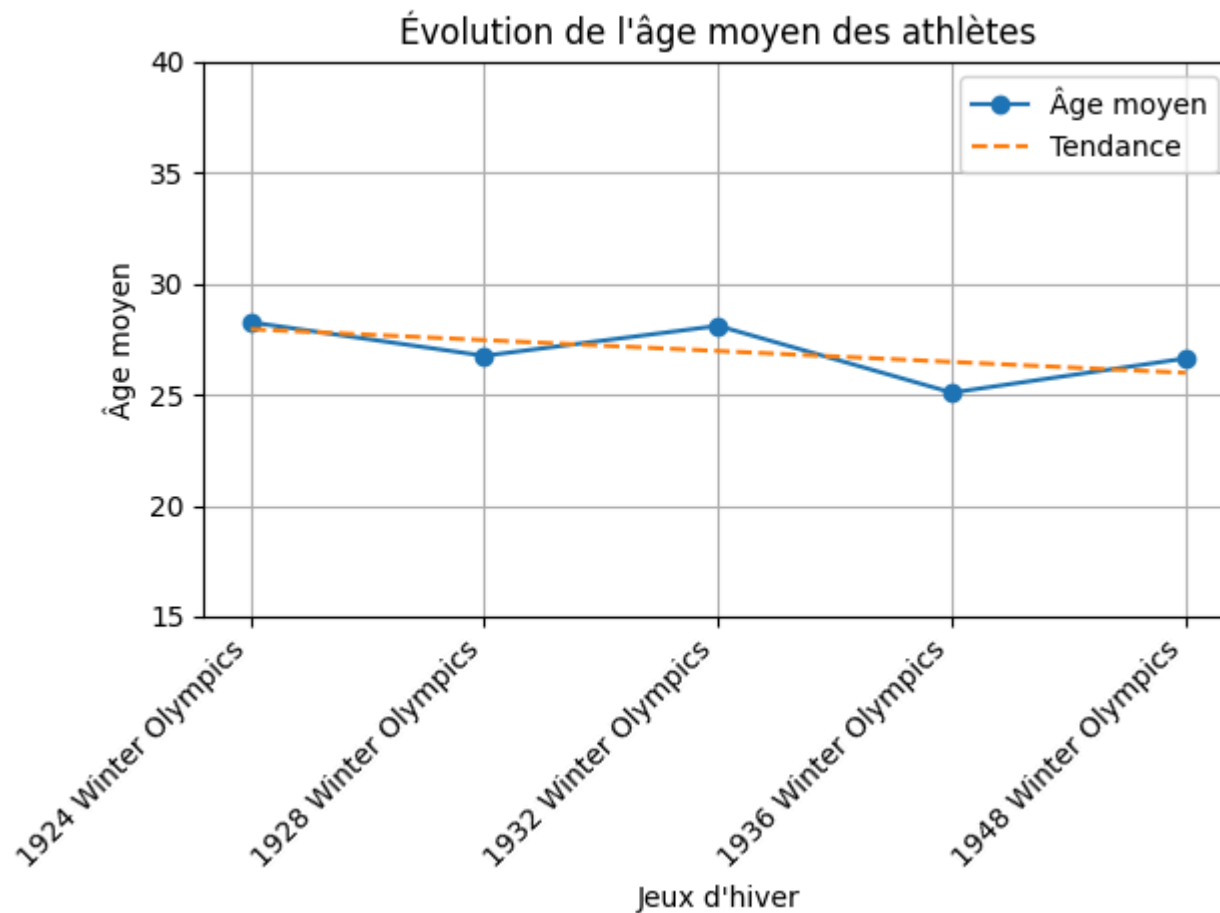
plt.xticks(x_numeric, data_moyenne_age["Jeux d'hiver"], rotation=45, ha='right')

plt.xlabel("Jeux d'hiver")
plt.ylabel("Âge moyen")
plt.title("Évolution de l'âge moyen des athlètes")

plt.ylim(15, 40)

plt.legend()
plt.grid()

plt.tight_layout()
plt.show()
```



Le graphique met en évidence une relative stabilité de l'âge moyen des athlètes lors des premières éditions des Jeux olympiques d'hiver, avec des valeurs oscillant globalement entre 25 et 28 ans. On observe un léger pic autour de 1932, suivi d'une baisse marquée en 1936, puis d'une remontée en 1948.

La droite de régression présente une pente légèrement négative, suggérant une très faible tendance à la diminution de l'âge moyen sur la période, bien que cette évolution reste modérée. On peut en conclure que l'âge moyen a tendance à diminuer au fil des années.

Féminisation

```
In [19]: data_feminisation = requete_vers_dataframe(conn, f"""
        WITH hivers AS (
```

```

SELECT h.idHote, h.libelleHote
FROM {SCHEMA}.HOTE h
WHERE h.idHote <= 33
      AND h.libelleHote LIKE '% Winter %'
),
evenements_feminins AS (
  SELECT
    h.idHote,
    h.libelleHote,
    COUNT(DISTINCT e.idEvenement) AS nb_evenements_feminins
  FROM hivers h
  JOIN {SCHEMA}.EVENEMENT e
    ON e.idHote = h.idHote
  WHERE e.nomEvenement LIKE '% Women'
  GROUP BY h.idHote, h.libelleHote
),
participantes AS (
  SELECT DISTINCT
    h.idHote,
    h.libelleHote,
    a.idAthlete
  FROM hivers h
  JOIN {SCHEMA}.EVENEMENT e
    ON e.idHote = h.idHote
  JOIN {SCHEMA}.PARTICIPATION_INDIVIDUELLE pi
    ON pi.idEvent = e.idEvenement
  JOIN {SCHEMA}.ATHLETE a
    ON a.idAthlete = pi.idAthlete
  WHERE a.genre = 'Female'

  UNION

  SELECT DISTINCT
    h.idHote,
    h.libelleHote,
    a.idAthlete
  FROM hivers h
  JOIN {SCHEMA}.EVENEMENT e
    ON e.idHote = h.idHote
  JOIN {SCHEMA}.PARTICIPATION_EQUIPE pe
    ON pe.idEvenement = e.idEvenement
  JOIN {SCHEMA}.COMPOSITION_EQUIPE ce

```

```

        ON ce.idEquipe = pe.idEquipe
    JOIN {SCHEMA}.ATHLETE a
        ON a.idAthlete = ce.idAthlete
    WHERE a.genre = 'Female'
),
femmes_athletes AS (
    SELECT
        idHote,
        libelleHote,
        COUNT(DISTINCT idAthlete) AS nb_femmes_athletes
    FROM participantes
    GROUP BY idHote, libelleHote
)
SELECT
    h.libelleHote AS "Jeux d'hiver",
    NVL(ef.nb_evenements_feminins, 0) AS "Nombre d'évènements féminins",
    NVL(fa.nb_femmes_athletes, 0) AS "Nombre de femmes athlètes"
FROM hivers h
LEFT JOIN evenements_feminins ef
    ON ef.idHote = h.idHote
LEFT JOIN femmes_athletes fa
    ON fa.idHote = h.idHote
ORDER BY "Jeux d'hiver"
)"""
display(data_feminisation)

```

	Jeux d'hiver	Nombre d'évènements féminins	Nombre de femmes athlètes
0	1924 Winter Olympics	1	16
1	1928 Winter Olympics	1	28
2	1932 Winter Olympics	4	34
3	1936 Winter Olympics	4	115
4	1948 Winter Olympics	4	80

```
In [20]: data_feminisation = data_feminisation.sort_values(by="Jeux d'hiver").reset_index(drop=True)
```

```

x = np.arange(len(data_feminisation))
labels = data_feminisation["Jeux d'hiver"]

```

```

def tracer_serie(colonne, ylabel, titre):
    y = data_feminisation[colonne].values

    coeffs = np.polyfit(x, y, 1)
    trend = np.poly1d(coeffs)

    plt.figure()

    plt.plot(x, y, marker='o', label=ylabel)

    plt.plot(x, trend(x), linestyle='--', label="Tendance")

    plt.xticks(x, labels, rotation=45, ha='right')
    plt.xlabel("Jeux d'hiver")
    plt.ylabel(ylabel)
    plt.title(titre)
    plt.ylim(bottom=0)
    plt.grid()
    plt.legend()

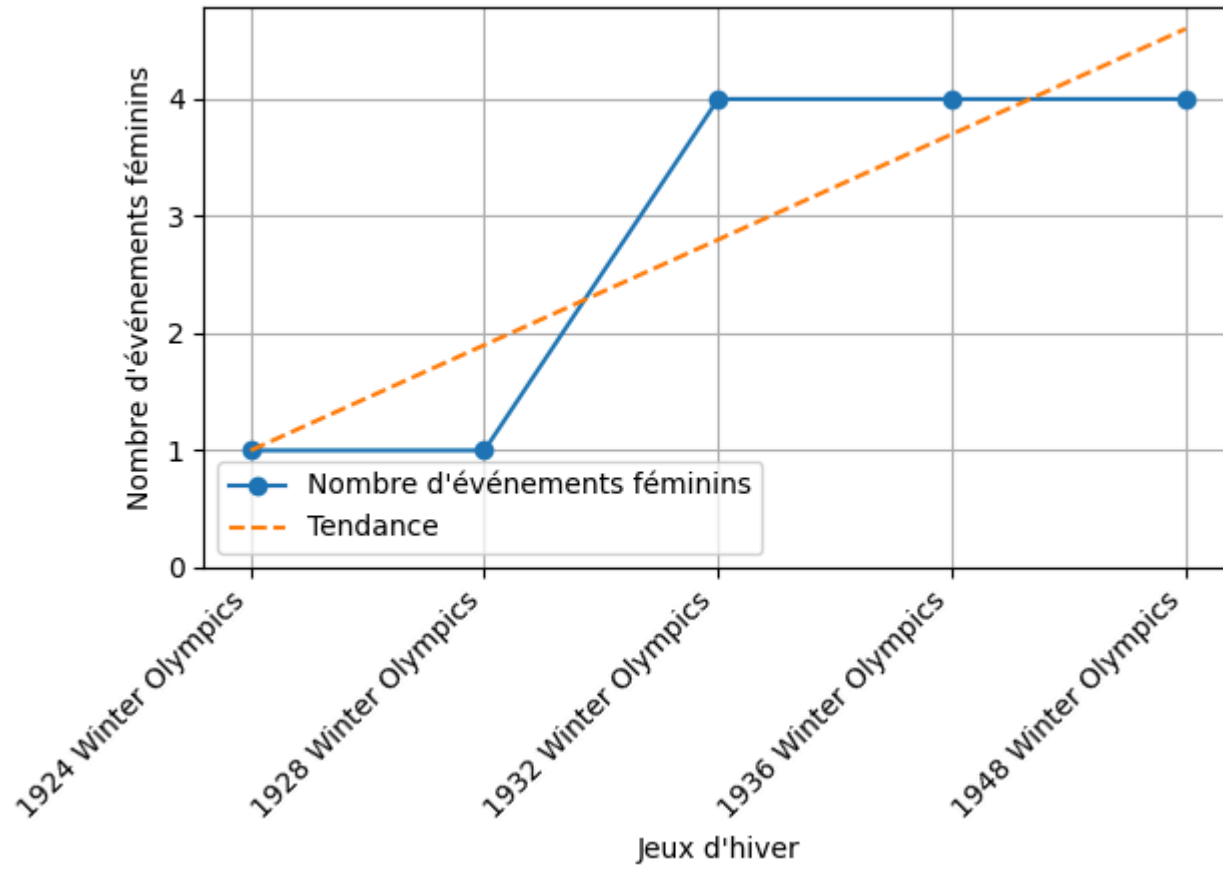
    plt.tight_layout()
    plt.show()

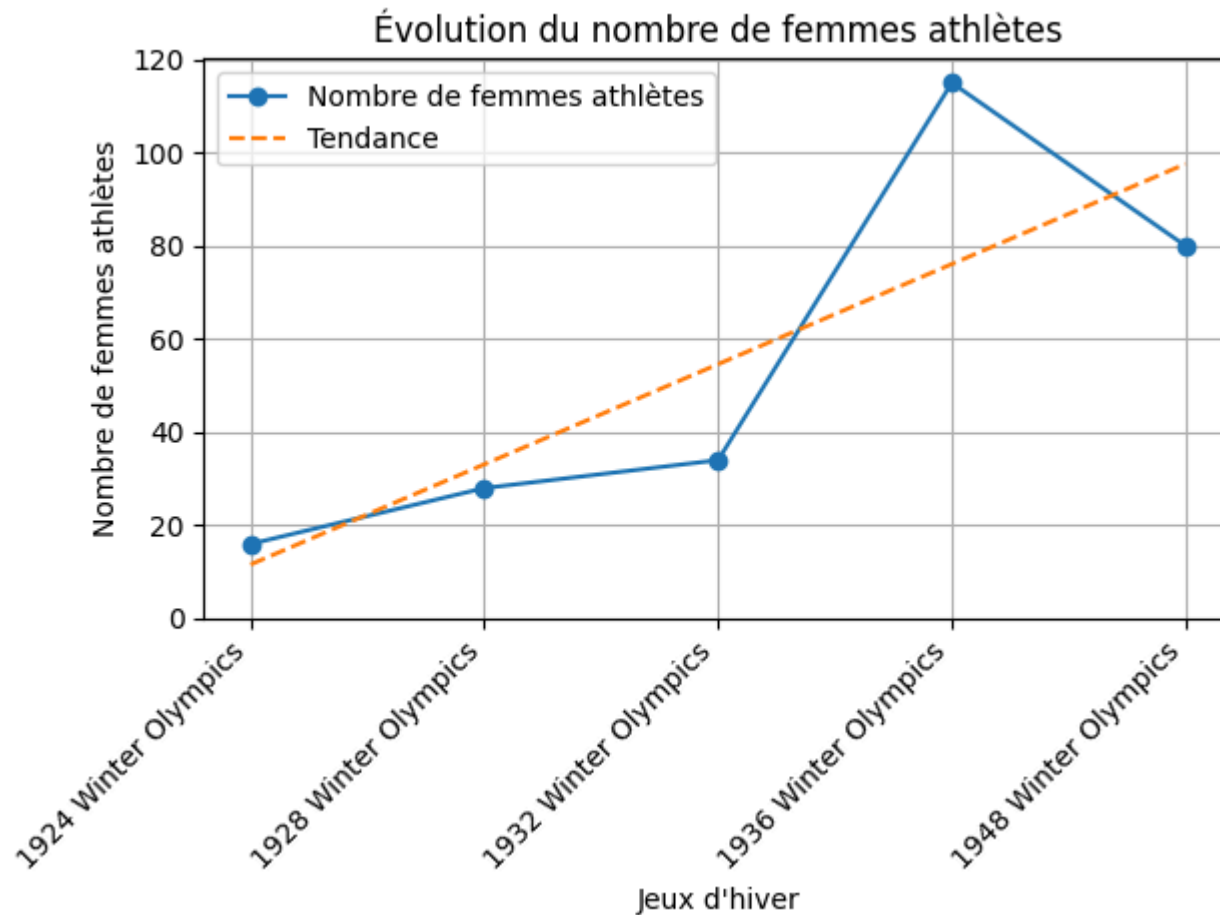
tracer_serie(
    "Nombre d'évènements féminins",
    "Nombre d'évènements féminins",
    "Évolution du nombre d'évènements féminins"
)

tracer_serie(
    "Nombre de femmes athlètes",
    "Nombre de femmes athlètes",
    "Évolution du nombre de femmes athlètes"
)

```

Évolution du nombre d'événements féminins





L'analyse des deux graphiques met en évidence une féminisation progressive mais non linéaire sur la période étudiée. Le nombre d'événements féminins reste d'abord très limité et stable entre 1924 et 1928 (une unique épreuve), avant de connaître une augmentation nette en 1932, puis de se stabiliser autour de 4 épreuves. La droite de tendance confirme ainsi une progression globale de l'offre sportive féminine, bien que celle-ci se fasse par paliers plutôt que de manière continue.

En parallèle, le nombre de femmes athlètes suit une évolution plus marquée mais aussi plus irrégulière. Après une croissance modérée entre 1924 et 1932, on observe une hausse très importante en 1936, traduisant une participation féminine beaucoup plus forte, suivie d'un léger recul en 1948. Malgré cette fluctuation, la tendance générale reste nettement croissante.

Le décalage entre les deux courbes montre qu'alors que le nombre d'événements féminins se stabilise à partir de 1932, la participation féminine continue d'augmenter fortement, ce qui suggère que la féminisation ne repose pas uniquement sur l'ajout d'épreuves, mais aussi sur une

intensification de la participation au sein des disciplines existantes.

Performance de l'équipe de votre choix

Nous avons choisi d'étudier la nation du Canada (NOC : CAN)

Evolution au cours des éditions précédentes (nombre d'athlètes, nombre de médaillés/médailles, etc.) :

```
In [21]: # Nombre d'athletes par edition

requete_nom_jeux = f"""
SELECT LibelleHote
FROM {SCHEMA}.HOTE
WHERE LibelleHote LIKE '%winter%'
AND AnneeHote <= 1948
ORDER BY LibelleHote
"""

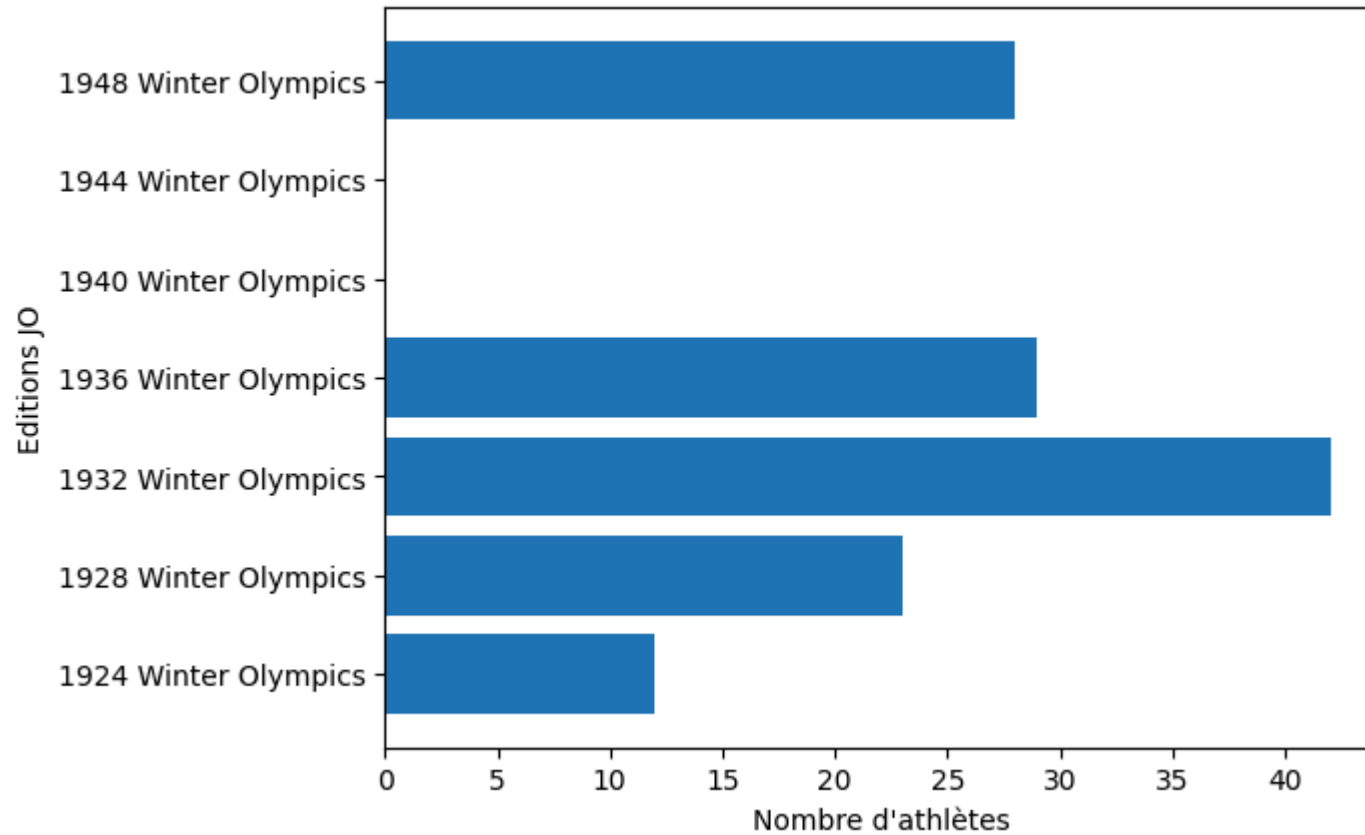
colonne = 'LIBELLEHOTE'

data = requete_vers_dataframe(conn, requete_nom_jeux)

nb_athletes = [12, 23, 42, 29, 0, 0, 28]

plt.barh(data[colonne], nb_athletes)
plt.title("Nombre d'athlètes par édition pour le Canada")
plt.ylabel('Editions JO')
plt.xlabel("Nombre d'athlètes")
plt.show()
```

Nombre d'athlètes par édition pour le Canada



Les éditions de 1940 et 1944 n'ont pas eu lieu à cause de la 2nd Guerre Mondiale

```
In [22]: requete_nom_jeux = f"""
SELECT LibelleHote
FROM {SCHEMA}.HOTE
WHERE LibelleHote LIKE '%winter%'
AND AnneeHote <= 1948
ORDER BY LibelleHote
"""

colonne = 'LIBELLEHOTE'

data = requete_vers_dataframe(conn, requete_nom_jeux)
```

```
data = data[colonne]

nb_or = [1, 1, 1, 0, 0, 0, 2]
nb_argent = [0, 0, 1, 1, 0, 0, 0]
nb_bronze = [0, 0, 5, 0, 0, 0, 1]
nb_total = [1, 1, 7, 1, 0, 0, 3]

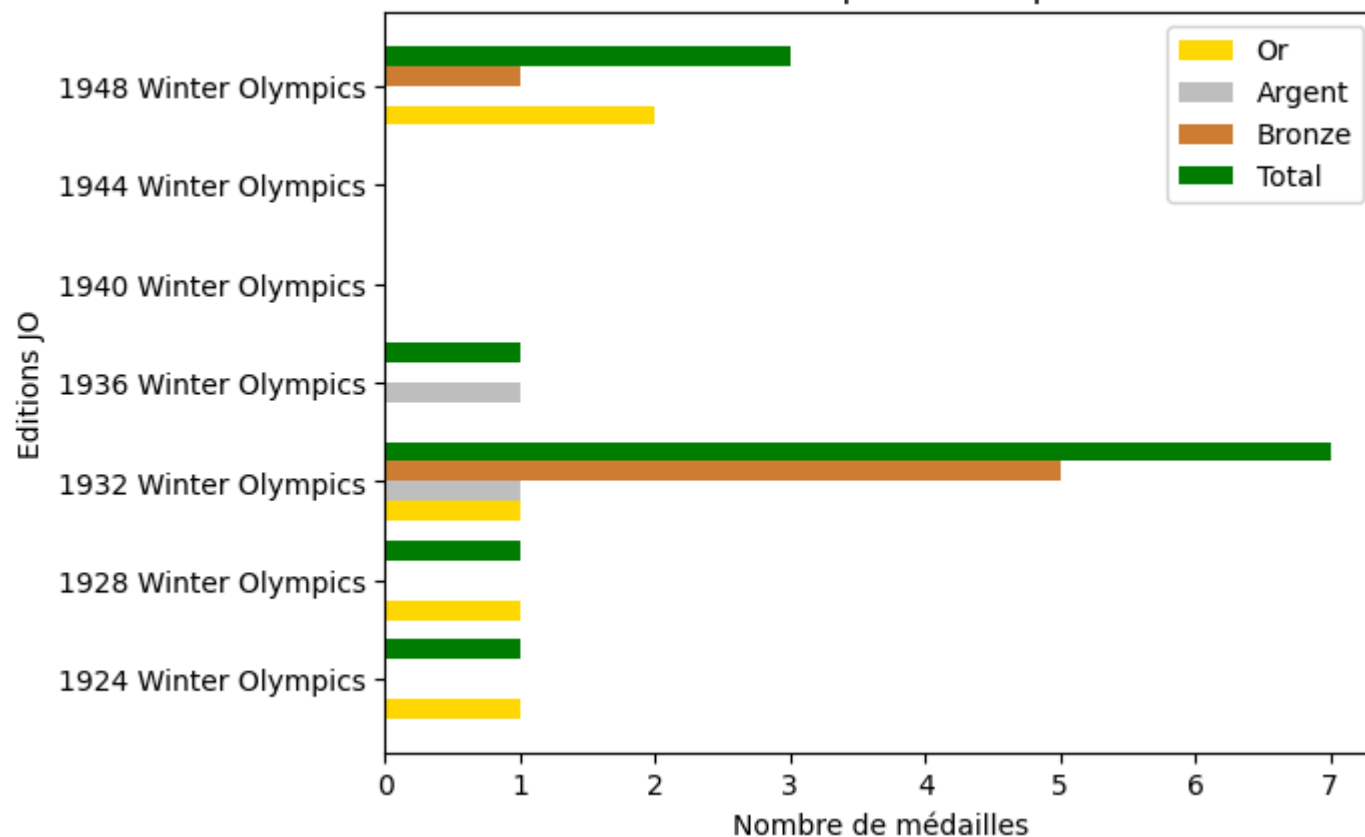
y = np.arange(len(data))
h = 0.2

plt.barh(y - 1.5*h, nb_or, height=h, label="Or", color="#FFD700")
plt.barh(y - 0.5*h, nb_argent, height=h, label="Argent", color="#C0C0C0")
plt.barh(y + 0.5*h, nb_bronze, height=h, label="Bronze", color="#CD7F32")
plt.barh(y + 1.5*h, nb_total, height=h, label="Total", color="#008000")

plt.yticks(y, data)
plt.xlabel("Nombre de médailles")
plt.ylabel("Editions JO")
plt.title("Nombre de médailles par édition pour le Canada")
plt.legend()

plt.show()
```

Nombre de médailles par édition pour le Canada



Les éditions de 1940 et 1944 n'ont pas eu lieu à cause de la 2nd Guerre Mondiale

Estimation de la performance du Canada à la prochaine édition des JO d'Hiver

Edition	Nombre medailles	Nombre athletes	Pourcentage nombre de médailles/nombre d'athletes
1924	1	12	0.083333333
1928	1	23	0.043478261
1932	7	42	0.166666667
1936	1	29	0.034482759
1948	3	28	0.107142857

(1940 et 1944 absent à cause de la 2nd Guerre Mondiale)

On peut voir que la moyenne du pourcentage du nombre de médailles par rapport au nombre d'athlètes est égal à 8%.

On peut aussi noter que la prochaine édition (1952 Winter Olympics) se passera en Italie et que le Canada a environ 30 athlètes par éditions lorsque les Jeux Olympiques se passent en Europe.

On peut donc estimer que le Canada aura 30-40 participants et qu'ils gagneront entre $(30(8/100)) = 2.4$ médailles et $(40(8/100)) = 3.2$ médailles

Réalisation du PDF

Pour réaliser le PDF, nous trouvions que l'affichage HTML du notebook sur JupyterLab était plus joli que le rendu PDF par défaut, donc nous l'avons exporté dans ce format. Ensuite, nous avons simplement "imprimé" cette page web en un fichier PDF à l'aide du navigateur.