Projet Jeux Olympiques - Journal de bord

T. Burnel, N. Grim, M. Griveau, M. Mechentel

Janvier 2024

1 Introduction

Nous somme une équipe de datajournalistes chargée, à l'approche des Jeux Olympiques de Paris, d'étudier l'existence ou non de liens entre le succès d'une pays aux Jeux et sa richesse. Notre hypothèse de départ est la suivante : les pays développés gagnent significativement plus de médailles en raison de leur niveau d'investissement dans le domaine sportif.

À l'aune de l'examen et du traitement des données que nous avons à notre disposition, nous avons pour ambition de déterminer si la politique d'investissement dans les infrastructures sportives des pays a une quelconque influence sur leurs résultats aux épreuves des Jeux.

Notre travail sera divisé en deux temps. Nous mettrons en relief le nombre de médailles remportées par les pays lors de six éditions des Jeux Olympiques (1996, 2000, 2004, 2008, 2012, 2016) par rapport à leur population, leur Produit Intérieur Brut, au pourcentage de ce dernier investi dans le domaine sportif. Un second temps sera consacré à l'examen de la réussite de la France

et du Royaume-Uni aux épreuves de natation sur la base du nombre d'infrastructures relatives pour 100.000 habitants.

Le choix de ces pays repose sur divers points de convergence : leur nombre d'habitants, leur économie et leur qualité de pays organisateur des Jeux.

Le résultat du traitement des données constituera une base à la réalisation de datavisualisations et d'une application web exploitables pour l'écriture d'articles journalistiques sur le sujet des Jeux Olympiques.

2 Jeux de données

2.1 Jeux du premier *flow*

Notre premier *flow* concerne les médailles remportées par les pays lors des six éditions des Jeux Olympiques sur lesquelles nous avons décidé de travailler. En amont du traitement, nos jeux étaient au nombre de deux. Le premier présente le nombre de médailles remportées par sportif aux Jeux sur une période de 120 ans. [1] Le deuxième, le Produit Intérieur Brut des pays et son pourcentage investi dans le domaine sportif. [2]

Nous avons enrichi ces jeux grâce au résultat d'une requête SPARQL, saisie sur le service de Wikidata. Le jeu ainsi produit présente le nombre d'habitants de l'ensemble des pays répertoriés dans la base de Wikidata, sur

1. *120 years of Olympic history : athletes and results*. URL : `https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results` (visité le 17/01/2024).

2. *Download entire World Economic Outlook database, April 2019*. IMF. URL : `https://www.imf.org/en/Publications/WEO/weo-database/2023/October/download-entire-database` (visité le 17/01/2024).

une période de trente ans (1993 - 2023). [3]

Trois autres jeux ont dû faire leur entrée au cours du traitement pour pallier de nombreux manquements. Nous avons constaté qu'en dépit d'indiquer le pourcentage du Produit Intérieur Brut investi dans le domaine sportif, le jeu du Fonds Monétaire International ne contenait pas le PIB des pays. Nous avons introduit un jeu de la Banque mondiale [4] dans le *flow* – lequel contient le montant du PIB avec une profondeur chronologique suffisante.

Les deux autres jeux concernent un problème de coordonnées. Lorsque Dataiku identifie des données de type *country*, nous pensions pouvoir en récupérer les coordonnées – notamment afin de les exploiter pour les datavisualisations. L'extension *reverse geocoding plugin* aurait pu nous le permettre, sans succès. Nos suppositions portent sur notre version gratuite de Dataiku et de son nombre réduit de fonctionnalités.

Ne disposant d'aucun budget pour disposer d'une version payante, nous avons recherché un jeu contenant les coordonnées des pays et l'avons trouvé sur la plateforme GitHub. [5] Nonobstant, les longitudes et latitudes des pays ainsi récupérées se sont avérées être fautives le moment des datavisualisations venu. C'est pourquoi nous avons mis au point une deuxième SPARQL, capable de renvoyer les bonnes coordonnées de tous les pays présents dans la base de Wikidata. Le jeu ainsi récupéré correspond au dernier renfort dont nous avons eu besoin pour être en mesure de réaliser des datavisualisations.

3. *Wikidata Query Service*. URL : `https://w.wiki/8uTN` (visité le 27/01/2024).

4. *World Bank Open Data*. World Bank Open Data. URL : `https://data.worldbank.org` (visité le 20/01/2024).

5. Alexandru Paul Copil. *Countries codes and coordinates*. URL : `https://gist.github.com/cpl/3dc2d19137588d9ae202d67233715478` (visité le 25/01/2024).

2.1.1 Jeu de Kaggle

Le jeu est composé de 15 colonnes et 271.117 lignes. 7 colonnes fournissent des informations sur les athlètes (*ID*, *Name*, *Sex*, *Age*, *Height*, *Weight*, *Medal*) ; 2 colonnes sur leurs pays (*Team*, *NOC*) et 6 sur les éditions des Jeux Olympiques (*Games* ; *Year* ; *Season* ; *City* ; *Sport* ; *Event*).

Les données sont exprimées en langue anglaise. La casse des chaînes de caractère suit le schéma d'une majuscule à chaque nouveau mot – exception faite des unités de mesure dans la colonne *Event* (comprendre *Épreuve*) : le terme *metres* est toujours noté en bas-de-casse. Les données sur la taille et le poids des athlètes sont des entiers. 2 colonnes, *Games* et *Year* contiennent des données de date, exprimées en année. La colonne *Games* a ceci de particulier qu'elle agrège une donnée de date et une donnée textuelle. Les Jeux Olympiques d'été de 1992 sont notés « 1992 Summer ».

Le jeu ne présente pas de défauts majeurs sur les colonnes que nous comptons conserver lors du traitement. Un problème plus préoccupant concerne la dénomination des pays, parfois obsolète (*Soviet Union*, *West Germany*) ou impropres (*Japan-1*, *Switzerland-2*). Nous effectuerons les jointures sur le code NOC des pays (un code unique attribué par le *National Olympic Committee* – le Comité National Olympique) pour plus de fiabilité.

2.1.2 Jeu du FMI

Le jeu est composé de 42 colonnes pour 7.715 lignes. La colonne *Country Name* permet d'identifier les pays. Celle nommée *COFOG Function Name* correspond au secteur des investissements et a pour unique valeur *Expenditure on recreational & sporting services*. La colonne *Unit Name* précise l'échelle de

calcul des investissements (soit en pourcentage du PIB, *Percent of GDP*, soit en monnaie courante *Domestic currency*). Le montant des investissements est renseigné sur 30 colonnes (1993 - 2022).

Les 9 autres colonnes correspondent à des codes et des dénominations propres au FMI (*Country Code*, *COFOG Function Code*, *Sector Name*, *Sector Code*, *Unit Code*, *Attribute*, *Indicator Code*, *Global DSD Time Series Code*, *col_41*). Le potentiel de croisement avec d'autres jeux étant par la force des choses très limité, nous n'utiliserons pas ces données.

L'ensemble du jeu est rédigé en langue anglaise. Noms de pays, sigles, codes, acronymes et unités de mesure (notées « Cash », « Non Cash », « Gross », « Net ») à part, la casse des données textuelles implique la présence d'une seule majuscule en début de chaîne. Nous notons l'utilisation de caractères spéciaux, à l'instar de l'esperluette ou de la barre oblique.

Exception faite des *Country Code* et du code correspondant à la valeur « Domestic currency », l'intégralité des codes sont un mélange de caractères numériques et textuels en haut-de-casse (« S13112 » pour la colonne *Sector Code* ou « XDC_R_B1GQ » dans la colonne *Unit Code*).

Les noms de pays souffrent d'un problème d'harmonisation et d'actualité (*West Bank and Gaza*, *Russian Federation*, *China, P.R. : Mainland*).

Ce jeu est de loin le moins bien structuré et le plus fautif. Bien que nous ne comptions pas les conserver au cours du traitement, les trois dernières colonnes (*Indicator Code*, *Global DSD Time Series Code*, *col_41*) sont d'une opacité confondante et témoignent du problème majeur du jeu : énormément de données sont vides, nulles ou incompréhensibles (« GERS_G14_GDP_PT » pour la colonne) *Indicator Code*, « A|GB|S1311|W0|S1|G2M|_Z|_Z|GF0801|

XDC|_T|_X » pour la colonne *Global DSD Time Series Code*). également Au sein des colonnes correspondant au montant des investissements par année, se côtoient des lignes vides et des sigles (« NA », « NP », « AC »), sans justification ni harmonisation aucune. En somme, ce jeu représente un véritable enjeu de nettoyage et de croisement des données.

2.1.3 Jeu de Wikidata (population)

En préambule de l'analyse du jeu de Wikidata, revenons un instant sur l'élaboration de la requête nous ayant permis de l'obtenir. Celle-ci renvoie le nombre d'habitants de l'ensemble des pays présents dans la base de données de Wikidata, sur une période de trente ans (1993 - 2023) :

```
SELECT ?paysLabel ?population ?date ?cio
WHERE
{
 ?pays wdt:P31 wd:Q6256.
 ?pays wdt:P984 ?cio.
 ?pays p:P1082 ?populationStatement.
 ?populationStatement ps:P1082 ?population.
 ?populationStatement pq:P585 ?date.
 FILTER(YEAR(?date) >= (YEAR(NOW()) - 30)).
 SERVICE wikibase:label { bd:serviceParam wikibase:language
     "[AUTO_LANGUAGE],fr". }
}
ORDER BY ?paysLabel ?date
```

Être en mesure de requêter l'intégralité des pays a été la première étape

de la construction de notre requête. Le premier triplet permet de spécifier la nature – notée `wdt:P31` – de notre variable inconnue `?pays` en la faisant correspondre à l'objet `country` – noté `wd:Q6256` :

```
?pays wdt:P31 wd:Q6256.
```

Le deuxième triplet constitue un ajout tardif, destiné à faciliter les jointures avec les autres jeux de données. Comme nous travaillons sur les Jeux Olympiques, les pays peuvent être non seulement identifiés par les codes basés sur la norme ISO 3166 mais aussi par les codes du CIO (Comité International Olympique). Nous avons donc récupéré cette donnée grâce à la propriété Wikidata correspondante – notée `wdt:P984` :

```
?pays wdt:P984 ?cio.
```

La deuxième partie de notre requête doit renvoyer le nombre d'habitants par pays en prenant en compte une dimension chronologique. La complexité de cette demande requiert un parcours de graphique en quatre temps.

Pour ce faire, nous avons consulté la liste de préfixes [6] de Wikidata afin de créer une nouvelle variable, `?populationStatement`. Le troisième triplet a recours à la classe `population` – notée `p:P1082` – et signifie que la variable `?populationStatement` a pour valeur la population des pays :

```
?pays p:P1082 ?populationStatement.
```

L'obtention du nombre d'habitants a nécessité le recours au préfixe `ps`, lequel permet de récupérer la valeur de la propriété relative à la population :

6. *Wikidata prefixes (E49) - Wikidata*. URL : `https://www.wikidata.org/wiki/EntitySchema:E49` (visité le 20/01/2024).

```
?populationStatement ps:P1082 ?population.
```

Enfin, nous avons rédigé le dernier triplet sur la base du préfixe `pq` pour attribuer la valeur chronologique à la variable `?date`. Un filtre y a été appliqué pour exprimer les limites extrêmes de notre période, soit `NOW` pour 2023 et `-30` pour 1993 :

```
?populationStatement pq:P585 ?date.
FILTER(YEAR(?date) >= (YEAR(NOW()) - 30)).
```

La première et dernière ligne permettent de cadrer l'affichage des résultats. Lorsque la requête s'exécute, Wikidata renvoie le nom de chaque pays (`?paysLabel`) par ordre alphabétique, le nombre d'habitants (`?population`) et l'année correspondante (`?date`) par ordre croissant :

```
SELECT ?paysLabel ?population ?date
ORDER BY ?paysLabel ?date
```

Nous nous sommes heurtés à plusieurs obstacles avant de rendre la requête fonctionnelle. Il nous a fallu un certain temps avant de comprendre la nécessité d'un parcours de graphique en quatre temps et du recours à une variable telle que `?paysStatement`. À cet égard, la documentation sur l'utilisation des propriétés `population` [7] et `date` [8] a été d'un grand secours.

Nous avons également pris en exemple la requête *Population in Europe after 1960.*[9] La consulter a été l'occasion d'une meilleure compréhension des

7. *Population.* URL : `https://www.wikidata.org/wiki/Property:P1082` (visité le 20/01/2024).
8. *Point in time.* URL : `https://www.wikidata.org/wiki/Property:P585` (visité le 20/01/2024).
9. *Wikidata Query Service.* URL : `https://w.wiki/8uHH` (visité le 20/01/2024).

propriétés Wikidata ainsi qu'une porte ouverte à la lecture de la documentation et à l'appréhension des requêtes en deux temps (collecte des propriétés d'une classe puis requêtage en fonction de notre besoin).

La rédaction de cette première requête a été d'une grande aide pour les autres recours à Wikidata qui ont émaillé notre chaîne de traitement.

Le jeu ainsi généré est composé de 13 colonnes et 3.737 lignes. Les lignes de la colonne, *paysLabel.xml :lang* sont composées d'une même et unique donnée : « fr ». Il s'agit de langue dans laquelle les noms des pays sont renvoyés.

Sur les 12 colonnes restantes, 6 fonctionnent selon une logique de duo et 6 autres selon une logique de trio. Pour chaque donnée, le jeu indique sont *type*, sa *value* et, pour les trio, son *datatype*. Ainsi, la donnée « cio » fait l'objet de 2 colonnes : *cio.type*, *cio.value* (*idem* pour les données « isoCode » et « paysLabel ») et la donnée « date » fait l'objet de 3 colonnes : *date.datatype*, *date.type*, *date.value* (*idem* pour la donnée « population »).

Bien que notre requête ne précise vouloir recueillir ni le *type* ni le *datatype* des données, ces colonnes sont malgré tout présentes dans le jeu de sortie. Les données textuelles sont exprimées en anglais (mis à part les noms de pays) et sont en bas-de-casse – exception faite des sigles et des noms de pays. Les dates sont exprimées selon le schéma YYYY-MM-DDThh:mm:ssZ. L'intégralité des *types* correspondent à « literal ». Les colonnes *datatype* ont pour valeur des liens renvoyant à des schémas XML du W3C.

Le jeu est d'une propreté exemplaire, les données sont harmonisées et correctement typées, ce qui nous sera d'une grande aide lors des jointures.

2.1.4 Jeu de la Banque mondiale

Le jeu se compose de 68 colonnes pour 533 lignes. Chaque ligne pleine est en réalité séparée de la suivante par une ligne vide. De surcroît, la dernière colonne est intégralement vide et ne porte que le sommaire nom de *col_67*. Sur l'ensemble du jeu, 63 colonnes correspondent aux années pour lesquelles le PIB des pays est renseigné (1960- 2022). 2 autres colonnes donnent des informations sur les pays : leur nom, leur code (*Country Name*, *Country Code*) et les 2 restantes, sur la donnée économique – en l'occurrence le PIB (*Indicator Name*, *Indicator Code*), en dollars courant.

Les données textuelles sont en langue anglaise. Les noms des pays portent une majuscule initiale. Les codes sont exclusivement en haut-de-casse. Les données numériques sont exprimées en décimaux.

Notons que certaines entrées de la colonne *Country Name* témoignent d'une certaine originalité dans la mesure où ils ne sont pas références à des pays nommément explicites (« Heavily indebted poor countries (HIPC) », « IBRD only », « Low & middle income », etc.). Ces excentricités soulignent l'importance des codes des pays pour effectuer les jointures et faire l'économie des entrées impropres. Enfin, les premières décennies souffrent d'un cruel manque de données mais plus nous avançons dans le temps, plus le jeu est complet sur l'ensemble des pays.

2.1.5 Jeu de GitHub

Le jeu se compose de 6 colonnes et 263 lignes. 2 de ces colonnes correspondent à la latitude et longitude (*Latitude (average)*, *Longitude (average)*) des pays dont le nom nous est donné dans la colonne *Country*. Les trois

autres colonnes (*Alpha-2 code*, *Alpha-3 code*, *Numeric code*) correspondent à des codes de pays prévus par la norme ISO 3166-1.

Les données textuelles sont exprimées en langue anglaise. Nonobstant, nous comptons nous appuyer sur les codes des pays pour réaliser les jonctions. Le jeu ne présente aucune donnée manquante.

2.1.6 Jeu de Wikidata (coordonnées)

Revenons derechef sur le détail de notre requête SPARQL, rédigée pour nous permettre de récupérer les coordonnée des pays recensés dans la base de données de Wikidata :

```
SELECT DISTINCT ?paysLabel ?geopoint ?noc
WHERE
{
 ?pays wdt:P31 wd:Q6256.
 ?pays wdt:P984 ?noc.
 ?pays wdt:P625 ?geopoint
 SERVICE wikibase:label { bd:serviceParam wikibase:language
     "[AUTO_LANGUAGE],en". }
}
```

Nous indiquons vouloir récupérer les données selon les colonnes suivantes : le nom du pays (noté `?paysLabel`), leurs coordonnées (notées `?geopoint`) et leur code CIO (noté `?noc`) :

```
SELECT DISTINCT ?paysLabel ?geopoint ?noc
```

Le premier triplet est *stricto sensu* le même que celui de notre première

requête Wikidata sur les populations (*cf. supra* p. 6).

Il permet de signifier que notre variable inconnue `?pays` est de nature (notée `wdt:P31`) « pays » (noté `wd:Q6256`) :

```
?pays wdt:P31 wd:Q6256.
```

Le deuxième triplet est également tout à fait similaire à son homonyme de la requête sur les populations (*cf. supra* p. 6). Il assigne à la variable `?noc` les codes des pays prévus par le Comité Olympique International grâce à la propriété relative notée `wdt:P984` :

```
?pays wdt:P984 ?noc.
```

Enfin, le dernier triplet récupère les coordonnées géographiques des pays grâce à la propriété correspondante, notée `wdt:P625` :

```
?pays wdt:P625 ?geopoint
```

Il en ressort un jeu composé de 8 colonnes et 185 lignes. Il présente la même logique de duo et de trio que notre autre jeu Wikidata (*cf. supra* p. 9) : les données « noc » et « paysLabel » font l'objet de 2 colonnes et les coordonnées, « geopoint », font l'objet de trois colonnes. La longitude et la latitude sont regroupées au sein de la même colonne, *geopoint.value* et sont présentées comme suit : « Point(<longitude> <latitude>) ». L'Islande a ainsi pour coordonnées « Point(-19.0 65.0) ».

Cette requête ayant été saisie dans l'urgence de devoir déboguer la réalisation des datavisualisations et la rédaction du présent journal de bord à moins d'une semaine de la date butoir, les noms de pays sont exprimés en

langue anglaise et les résultats n'ont pas été ordonnés par ordre alphabétique à l'instar de la requête sur la population. Nonobstant, la forme des données qui nous préoccupent – en l'occurrence, les codes NOC et les coordonnées – ne dépend pas de la langue pourvu que l'alphabet soit le même.

2.2 Jeux du second *flow*

Notre deuxième *flow* repose sur une comparaison entre la France et le Royaume-Uni à l'endroit de la natation. En amont du traitement, nos jeux étaient au nombre de cinq : trois pour le Royaume-Uni, deux pour la France.

Sur le Royaume-Uni, un premier jeu résulte d'une requête SPARQL, laquelle génère un jeu listant le nom des régions du pays, leur nombre d'habitants et leurs coordonnées. Les deux autres jeux proviennent du même site,[10] l'un situe les infrastructures sportives par rapport à leur région, l'autre les décrit d'un point de vue davantage technique.

Sur la France, l'un des deux jeux est également une requête SPARQL, renvoyant là aussi le nom des régions du pays, leur nombre d'habitants et leurs coordonnées. Le second jeu, fourni par le ministère de l'Éducation nationale, de la Jeunesse, des Sports et des Jeux Olympiques et Paralympiques[11] présente les infrastructures sportives et les relie aux régions françaises.

Au cours du traitement, nous avons réalisé qu'il nous fallait relier tôt ou tard ces divers jeux aux médailles remportées par les deux pays. C'est pourquoi nous avons fait usage d'une version du jeu de Kaggle issu de notre

10. *Active Places Power*. URL : `https://www.activeplacespower.com/` (visité le 20/01/2024).
11. *Data ES - Base de données*. URL : `https://equipements.sports.gouv.fr/explore/dataset/data-es/information/` (visité le 17/01/2024).

premier *flow*, où se trouvent les données nécessaires.

2.2.1 Jeu de Wikidata (régions de France)

Cette requête interroge la base de données de Wikidata selon le prisme des régions. Nous voulons ici afficher leur nom (`?RegionLabel`), leur nombre d'habitants (`?population`) et leurs coordonnées (`?geopoint`) :

```
SELECT ?RegionLabel ?population ?geopoint
WHERE
{
 ?Region wdt:P31 wd:Q36784.
 ?Region wdt:P1082 ?population.
 ?Region wdt:P625 ?geopoint.
 SERVICE wikibase:label { bd:serviceParam wikibase:language
     "[AUTO_LANGUAGE],fr". }
}
ORDER BY ?Region
```

Le premier triplet nous permet de faire correspondre la variable `?Region` à l'objet « région de France » noté `wd:Q48091` , derechef grâce à la propriété `wdt:P31` nous permettant de préciser la nature de notre sujet :

```
 ?Region wdt:P31 wd:Q36784.
```

Le deuxième triplet assigne à la variable `?population` la valeur de la propriété `wdt:P1082` , c'est-à-dire la population – ici des régions :

```
 ?Region wdt:P1082 ?population.
```

Selon la même logique, le dernier triplet récupère les coordonnées des régions grâce à la variable `?geopoint` et à la propriété « coordonnées géographiques » notée `wdt:P625` :

```
?Region wdt:P625 ?geopoint.
```

Et nous trions les résultats selon l'ordre alphabétique du nom des régions :

```
ORDER BY ?Region
```

Le jeu ainsi généré est composé de 9 colonnes et 18 lignes. Nous n'expliciterons pas de nouveau les logiques d'atomisation des données en deux ou trois colonnes. Le jeu présente le nom des régions (*RegionLabel.type*), leur nombre d'habitants (*population.value*) et leurs coordonnées (*geopoint.value*) et ce en français.

2.2.2 Jeu du ministère

Le jeu est composé de 104 colonnes et 143.192 lignes. Par souci de concision, nous ne nous attarderons que sur les colonnes que nous comptons conserver au cours du traitement. Ces dernières sont au nombre de quatre : *Type d'équipement sportif*, *Longitude (WGS84)*, *Latitude (WGS84)* et *Région Nom*.

La colonne *Type d'équipement sportif* représente un véritable enjeu de nettoyage. La nomenclature est très libre, les dénominations sont d'apparence toutes en bas-de-casse avec une majuscule initiale mais plusieurs exceptions perturbent ce schéma (« Parcours Acrobatique en Hauteur/Site d'accrobranche », « Structure Artificielle d'Escalade »).

De surcroît,nombre d'entrées font l'objet de multi-nommage avec par l'in-

termédiaire de la barre oblique (« Multisports/City-stades », « Salles polyvalentes / des fêtes / non spécialisées », « Piste d'aérodrome / d'aéroport », etc.) et d'autres sont accompagnées de précisions entre parenthèses : « Salle multisports (gymnase) », « Aire mixte (décollage et atterissage) », etc.

Les noms de régions sont saisies en bas-de-casse avec conservation des majuscules initiales. Enfin, la longitude et la latitude se basent sur le système *WGS 84 – World Geodetic System 1984*, un système géodésique mondial comprenant, entre autres, un modèle pour l'expression de coordonnées.

2.2.3 Jeu de Wikidata (régions d'Angleterre)

Cette requête est l'identique stricte de celle sur les régions de France (*cf. supra* p. 14). La description des triplets sera d'autant plus brève. Nous voulons afficher le nom des régions d'Angleterre (`?RegionLabel`), leur nombre d'habitants (`?population`) et leurs coordonnées (`?geopoint`) :

```
SELECT DISTINCT ?RegionLabel ?population ?geopoint
WHERE
{
 ?Region wdt:P31 wd:Q48091.
 ?Region wdt:P1082 ?population.
 ?Region wdt:P625 ?geopoint.
 SERVICE wikibase:label { bd:serviceParam wikibase:language
     "[AUTO_LANGUAGE],en". }
}
```

Le premier triplet fait correspondre la variable `?Region` à l'objet « région d'Angleterre » (`wd:Q48091`), grâce à la propriété « région » (`wdt:P31`) :

```
?Region wdt:P31 wd:Q48091.
```

Le deuxième triplet assigne à la variable `?population` la population (`wdt:P1082`) de notre sujet, c'est-à-dire les régions (`?Region`) :

```
?Region wdt:P1082 ?population.
```

Le dernier triplet récupère les coordonnées (`wdt:P625`) des régions (`?Region`) et les assigne à la variable `?geopoint` :

```
?Region wdt:P625 ?geopoint.
```

Il en ressort un jeu composé de 9 colonnes et 36 lignes. Nous n'expliciterons pas de nouveau les logiques d'atomisation des données en deux ou trois colonnes. Le jeu présente strictement les mêmes colonnes que le jeu résultant de la requête sur les régions de France : le nom des régions (*RegionLabel.type*), leur nombre d'habitants (*population.value*) et leurs coordonnées (*geopoint.value*) et ce en langue anglaise.

2.2.4 Jeux de *Sport England*

Le premier jeu provenant de *Sport England*, nommé *Geographics* est composé de 25 colonnes et 118.655 lignes. Comme son nom le suggère, les données présentées sont relatives aux infrastructures sportives d'Angleterre. Chacune possède un identifiant unique, répertorié dans la colonne *FACILITYID* et est rattachée à une région, grâce à la colonne *Region Name*. Leur nom est en anglais et en bas-de-casse, avec conservation des majuscules initiales. Les 2 autres colonnes qui survivront au traitement sont *Latitude* et *Longitude*.

Les 21 colonnes restantes fournissent une plus grande quantité de données sur la situation géographique et administrative des infrastructures – selon un schéma soit d'identifiants (*SiteID*), soit de code (*Output Area Code*), soit de colonnes en duo avec d'une part les noms (*Ward Name*, *Local Authority Name*, etc.), d'autre part les codes (*Ward Code*, *Local Authority Code*, etc.)

Le système de nommage et de dénomination ne laisse pas place à l'opacité. Certaines colonnes manquent cruellement de données (*Metro Name* ; *Core City Name* ; *LDP Code* ; *LDP Name*) mais nous ne comptons pas les conserver le moment du traitement venu.

Le deuxième jeux, nommé *SwimmingPool*, est composé de 24 colonnes et 6.474 lignes. Exception faite de la première colonne, *FacilityID*, l'intégralité des données sont à propos de détails techniques et matériels sur les infrastructures (*Length* ; *Maximum Depth* ; *Movable Floor* ; *Seating*, etc.).

Toutes les données sont de type numérique – soit des entiers, soit des décimaux. Certaines colonnes (*Designated Accessible Provision*, *Designated Accessible Provision - Covered*, *Designated Accessible Provision - Uncovered*, *Seating - Covered*, *Seating - Uncovered*, *Standing*, *Standing - Covered*, *Standing - Uncovered*) ont la singularité de faire se côtoyer des entrées vides et des entrées égales à zéro. Cependant, seule la colonne *FacilityID* nous importe afin de réaliser une jointure avec le jeu *Geographics*.

2.2.5 Jeu de Kaggle

Cette version du jeu *Kaggle* issu de notre premier *flow* est composée de 26 colonnes et 780 lignes. 2 colonnes, *NOC* et *Sport* correspondent respecti-

vement aux codes CIO des pays et à l'intitulé du sport (en langue anglaise et bas-de-casse, avec conservation des majuscules initiales) ayant fait l'objet d'au moins épreuve aux Jeux Olympiques.

Les 24 autres colonnes correspondent aux six éditions des Jeux selon le schéma suivant : chaque année fait l'objet de 4 colonnes. 3 pour les types de médailles (bronze, argent, or) et 1 pour la somme de l'ensemble des médailles. Ainsi, les colonnes de l'année 2008 sont nommées comme suit : *2008 - Bronze*, *2008 - Silver*, *2008 - Gold*, *2008 - Médailles*.

3 Traitement des données

3.1 Objectif du traitement

À l'aune de notre approche, notre premier *flow* aura pour ambition de faire ressortir les données suivantes : les codes des pays attribués par le *National Olympic Committee* – que nous nous permettrons d'abréger « code NOC » – le nombre de médailles remportées aux éditions six des Jeux Olympiques et le pourcentage d'investissement des pays dans le domaine sportif par rapport à leur PIB et à leur population.

3.2 Chaîne de traitement du premier *flow*

3.2.1 Préparation du jeu de la Banque Mondiale

La première recette appliquée au jeu est une préparation. Elle consiste en la suppression de toutes les lignes vides – soit une ligne sur deux, rappelons-le – et en la conservation de seulement 7 colonnes sur 67 :

— *Country Code* ;

— Les éditions des Jeux de 1996 à 2016.

Les codes NOC seront fondamentaux pour réaliser de futures jointures et les années 1996, 2000, 2004, 2008, 2012 et 2016 correspondent aux limites chronologiques de notre approche.

Deux autres étapes composent la recette : le renommage des 7 colonnes restantes (ainsi, *Country Code* et renommée *NOC* et les colonnes des années sont renommées selon le même motif : *1996* devient *1996 - GDP* pour *Gross Domestic Product*, c'est-à-dire le Produit Intérieur Brut.) et le passage de toutes les valeurs décimales en entiers, selon une logique d'arrondissement.

3.2.2 Préparation du jeu de Kaggle

Ce jeu a fait l'objet de trois recettes. La première, une préparation, nous a permis de conserver uniquement 3 colonnes sur 15 :

— *NOC* ;

— *Medal* ;

— *Year*.

Nous avons également épuré la colonne *Year* pour ne conserver que les années des Jeux relatives à notre sujet. Les entrées de la colonnes *Medal* correspondant à des valeurs nulles (exemplairement *N/A*) ont été supprimées.

L'étape suivante consiste en la réalisation d'un pivot : les années présentes dans les lignes sont devenues des colonnes dont les lignes ont pris pour valeur les entrées de la colonne *Medal*. 6 nouvelles colonnes ont donc été créées, une par édition des Jeux Olympiques, entre 1996 et 2016.

Enfin, nous avons compté les occurrences des trois types de médailles (bronze, argent, or) et créé des colonnes de sorties pour chaque type pour chaque édition. Par exemple, la colonne *1996* a donné lieu à la création des colonnes *1996 - Bronze* ; *1996 - Silver* ; *1996 - Gold*.

La deuxième recette est un *group*. Le *dataset* présente les médailles reportées par les athlètes, or nous n'avons conservé aucun donnée à leur endroit. Nonobstant, la répartition des médailles reposait toujours sur cette logique.

Nous avons donc réalisé un groupement pour réunir les doublons et compter les occurrences de chaque type de médaille remportée pour les six éditions des Jeux. Admettons qu'aux Jeux de 2016, un athlète français ait remporté une médaille de bronze et un autre deux. La recette a pour effet d'afficher le chiffre 3 pour l'entrée « France » sur la colonne *2016 - Bronze*.

La troisième et dernière recette en amont des jointures est également une préparation. Elle nous a permis, pour chacune des six éditions, de créer une nouvelle colonne présentant la somme de toutes les médailles remportées par les pays. En guise d'exemple, nous savons qu'à l'édition de 2008, l'Australie a remporté un total de 110 médailles.

3.2.3 Préparation du jeu de Wikidata

L'enrichissement *via* les données de Wikidata a été soumis a deux recettes. Une préparation où nous avons supprimé 8 colonnes – lesquelles représentaient un considérable bruit documentaire dans la mesure où notre requête SPARQL ne les sollicitaient pas (exemplairement à propos des « types » de données : *population.type*, *date.type*, *date.datatype*).

Trois colonnes ont été renommées : *cio.value* est devenu *NOC* ; *isoCode.value*

est devenu *Pays* et *paysLabel.value* est devenu *Nom - Pays*.

Nous avons modifié le format des dates : sur l'année, le jour, le mois et l'heure, nous n'avons conservé que l'année dans une colonne de sortie avant de supprimer la colonne initiale. Les années des six éditions mises à part, toutes les autres ont été supprimées et un nouveau pivot a été réalisé : les années en ligne sont passées en colonne et ont pris pour valeur le nombre d'habitants par pays. Ce pivot a permis la suppression des colonnes *Année* et *population.value*. Enfin, les colonnes des années ont été renommées selon un motif commun : *1996* est devenu *1996 - Pop*.

La dernière étape de cette recette consiste en la suppression de trois valeurs vides présentes dans la colonne *NOC*.

À l'instar du jeu de Kaggle, nous avons appliqué une recette *group* au jeu de Wikidata. Le groupe s'est effectué sur les colonnes *NOC*, *Pays* et *Nom - Pays* afin de regrouper le nombre d'habitants par année sur une même ligne.

3.2.4 Préparation du jeu de GitHub

Le jeu de GitHub n'a nécessité qu'une maigre recette de préparation : nous avons supprimé les colonnes *Alpha-2 Code* et *Numeric Code* pour ne conserver que la colonne *Alpha-3 Code* pour faciliter les jointures.

Nous avons également supprimé les entrées que Dataiku ne reconnaissait pas comme pays pour enfin renommer deux colonnes : *Country* est devenu *Pays* et *Alpha-3 Code* est devenu *Code*.

3.2.5 Préparation du jeu du FMI

À la faveur de la première recette, une préparation, nous avons supprimé de la colonne *Unit Name* les lignes où la valeur était *Domestic currency* pour ne conserver que le pourcentage du Produit Intérieur Brut (*Percent of GDP*) investi par les pays dans le domaine du sport.

Nous avons uniquement conservé 7 colonnes : *Country Name* ainsi que les dates des six éditions des Jeux. La dernière étape de la préparation consistait en la suppression de lignes vides pour la colonne *2016*. À partir de cette étape, Dataiku se figeait et nous laissait dans l'impossibilité d'ajouter de nouvelles étapes. Nous avons donc dû ajouter une autre recette de préparation pour que le *run* fonctionne et reprendre le traitement.

Dans cette deuxième préparation, les 6 colonnes correspondant aux éditions des Jeux selon le motif suivant : *1996* devient *1996 - FMI*.

Il nous a fallu manuellement modifier 13 valeurs dans la colonne *Country Name* pour une meilleure reconnaissance des pays par Dataiku. Par exemple : *Russian Federation*, *Czech Rep.* ne sont pas reconnus, tandis que *Russia* et *Czech Republic* le sont. Ces modifications ont donné lieu à la création d'une colonne de sortie *Pays* car il nous était impossible de modifier les noms des pays directement dans la colonne d'origine. Nous avons naturellement supprimé la colonne *Country Name*, une fois ces modifications effectuées.

Nous avons rencontré un problème de valeurs aberrantes dans les valeurs indiquées pour les investissements par rapport au PIB. La nature du problème repose sur des conversions mathématiques fautives : les pour mille et pour cent ont subi la même opération. Nous avons donc entrepris de corriger ces données par l'intermédiaire d'une recette Python que voici :

```
import dataiku
import pandas as pd
from dataiku import pandasutils as pdu

FMI_prepared2 = dataiku.Dataset("FMI_prepared2")
FMI_prepared2_df = FMI_prepared2.get_dataframe()

colonnes_a_traiter = ['1996 - FMI', '2000 - FMI', '2004 - FMI',
    '2008 - FMI', '2012 - FMI', '2016 - FMI']

for colonne in colonnes_a_traiter:
    condition = FMI_prepared2_df[colonne] > 0.1
    FMI_prepared2_df.loc[condition, colonne] /= 10

FMI_prepared2_df.drop_duplicates(subset=['Pays'], keep='first',
    inplace=True)

FMI_Python = dataiku.Dataset("FMI_Python")
FMI_Python.write_with_schema(FMI_prepared2_df)
```

Le code ci-dessus intervient sur les colonnes des six éditions des Jeux :

```
colonnes_a_traiter = ['1996 - FMI', '2000 - FMI', '2004 - FMI',
    '2008 - FMI', '2012 - FMI', '2016 - FMI']
```

Et modifie leur contenu grâce à une boucle `for` : si les colonnes contiennent des valeurs supérieures à `0.1`, ces dernières sont divisées par `10` pour obtenir la juste valeur d'investissement :

```
for colonne in colonnes_a_traiter:
    condition = FMI_prepared2_df[colonne] > 0.1
    FMI_prepared2_df.loc[condition, colonne] /= 10
```

La bonne valeur des données est ainsi rétablie, et l'ensemble de nos jeux sont prêts pour les jointures.

3.2.6 Première jointure

La première jointure concerne les jeux préparés de Kaggle et de Wikidata, sur la colonne *NOC*. Il en résulte la création du jeu *Med-Pop*, pour « Médailles-Population ». La jointure s'est réalisée sans suppression et n'a pas occasionné une perte significative d'informations.

3.2.7 Deuxième jointure

La deuxième jointure regroupe les jeux *Med-Pop* et *GDP* (jeu de la Banque mondiale) – également sur la colonne *NOC*, sans suppression ni perte significative d'informations. Le jeu *Med-Pop-GDP* a ainsi été créé.

3.2.8 Troisième jointure

La troisième jointure est un croisement du jeu *lien_pays* (issu de GitHub) et du jeu *FMI*, sur la colonne *Pays* du jeu du FMI. Tous les pays reconnus dans le ce jeu-ci ont été enrichis par les données du jeu de GitHub, soit par les données de longitude, de latitude et par un code similaire au code NOC. Le jeu *FMI-Liens* a ainsi été créé.

3.2.9 Quatrième jointure

Enfin, la quatrième jointure rassemble les jeux *Med-Pop-GDP* et *FMI-Liens* sur les colonnes *Pays* (du jeu *Med-Pop-GDP*) et *Code* (du jeu *FMI-Liens*). Dans la mesure où les médailles remportées constituent les données les plus importantes, la jointure a été réalisée de telle sorte à ce que les pertes touchent le jeu *FMI-Liens* et non pas *Med-Pop-GDP*.

3.2.10 Préparation du jeu final

Notre dernier jeu alors créé, *Reunion*, a subi une recette de préparation afin de nettoyer les données une dernière fois. Les données sur la population ont été arrondies, les colonnes dont l'entête est en français ont été renommées pour harmoniser la nomenclature de notre *dataset* de sortie. Enfin, les colonnes ont été organisées de la manière suivante : les deux premières colonnes sont *NOC* et *Country - Name* puis chaque édition des Jeux est présentée comme suit :

1996 - Bronze	1996 - Silver	1996 - Gold	1996 - Medals	1996 - Pop	1996 - GDP	1996 - Percent invested
bigint	bigint	bigint	bigint	double	bigint	double
Integer	Integer	Integer	Integer	Decimal	Integer	Decimal

Et les deux dernières colonnes, *Latitde (average)* et *Longitude (average)* présentent les coordonnées géographiques des pays.

3.3 Chaîne de traitement du second *flow*

SPARQL England :

Suppression de toutes les colonnes sauf 3 : geopoint.value ; RegionLabel.value ; population.value car c'est aussi du bruit documentaire.

Modification des valeurs de geopoint.value : from Point(0.41 52.24) 0.41 52.24.

Conservation d'une seule valeur de population par région + groupement pour avoir une seule ligne par région.

Renommage de certains noms de région pour faciliter la jointure

SWIMMINGPOOLS : conservation de la seule colonne *FacilityID* pour faire la jointure partant du principe que 1 ID = 1 installation

GEOGRAPHICS : conservation de FacilityID, de Region Name et de Latitude et Longitude

1re jointure : SWIMMINGPOOLS et GEOGRAPHICS. Une inner join propre sur les ID, aucun souci. Donne le jeu *InfraEnglandeJoin1*

2e jointure : SPARQL et InfraEnglandeJoin1. Jointure sur le nom de la région

Autre recette : group sur population.value avec copie de la colonne population pour la conserver.

Ensuite : produit en croix pour aboutir au nombre de piscine pour 100.000 habitants avec arrondissement pour cette colonne

POUR LA FRANCE

SPARQL, même opération. SPARQL a extrait toutes les régions de France mais nous n'avons gardé que celles de la France métropolitaine

Jeu de la France : conservation des colonnes qui nous intéressent (type_equipement, longitude, latitude, regionLabel, dans type équipement : remplacement de toutes les données qui nous intéressent en « piscine centre aquatique »

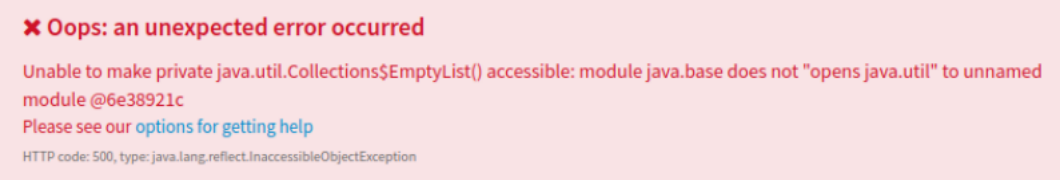
JOINTURE : inner join sur le nom de la region et le labelRegion

JOINTURE : inner join aussi sur le nom de la région

Regroupement : jointure sur type_equipement sans garder la longitude et la latitude

3.3.1 Semaine du 22 janvier 2024

Mauvaise surprise : alors que nous voulions produire un autre flow afin de remplir la seconde partie de notre objectif et produire des datavisualisations comparant la France et le Royaume-Uni, un bogue manifestement causé par une installation relative à Javascript perturbe l'affichage sur Dataiku. Probablement : problèmes de compatibilité.



ll est probable que cette erreur soit apparue après l'installation d'une version de JDK supérieure à celle prise en charge par Dataiku. Il aurait ainsi fallu rétrograder notre version de JDK, mais nous craignons que cette manipulation entraîne des difficultés dans d'autres projets que nous menons. Nous ne pouvons dès lors nous appuyer que sur un seul ordinateur, qui possède une version inférieure de JDK, ce qui a causé d'importants soucis organisationnels à la fin du mois de janvier.

3.4 Chaîne de traitement du second *flow*

4 Visualisation des données

4.1 Présentation des visualisations

4.2 Analyse

Qu'apprend-on en regardant les visualisations ? Quels sont les biais ?

5 Sitographie

120 years of Olympic history : athletes and results. URL : https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results (visité le 17/01/2024).

Active Places Power. URL : https://www.activeplacespower.com/ (visité le 20/01/2024).

Data ES - Base de données. URL : https://equipements.sports.gouv.fr/explore/dataset/data-es/information/ (visité le 17/01/2024).

Download entire World Economic Outlook database, April 2019. IMF. URL : https://www.imf.org/en/Publications/WEO/weo-database/2023/October/download-entire-database (visité le 17/01/2024).

Paul Copil, Alexandru. *Countries codes and coordinates*. URL : https://gist.github.com/cpl/3dc2d19137588d9ae202d67233715478 (visité le 25/01/2024).

Point in time. URL : https://www.wikidata.org/wiki/Property:P585 (visité le 20/01/2024).

Population. URL : https://www.wikidata.org/wiki/Property:P1082 (visité le 20/01/2024).

Wikidata prefixes (E49) - Wikidata. URL : https://www.wikidata.org/wiki/EntitySchema:E49 (visité le 20/01/2024).

Wikidata Query Service. URL : https://w.wiki/8uTN (visité le 27/01/2024).

Wikidata Query Service. URL : https://w.wiki/8uHH (visité le 20/01/2024).

World Bank Open Data. World Bank Open Data. URL : https://data.worldbank.org (visité le 20/01/2024).

Table des matières