

# ANALYSE DE L'ENQUÊTE PISA

---

L'investissement dans l'éducation est-il  
un facteur majeur de réussite scolaire?

Analyse au niveau mondial

---

Conception d'une base de données SQL  
Analyse  
Visualisation

---

Sarah STITOU  
Sarah.stito@gmail.com  
<https://www.linkedin.com/in/sarah-stitou-8a352918a/>

# ANALYSE DE L'ENQUÊTE PISA.

L'investissement dans l'éducation est-il un facteur majeur de réussite scolaire?

## TABLE DES MATIERES

### Introduction

#### 01 -

- 1.Présentation personnelle
- 2.Introduction -Présentation du projet
- 3. Mission
- 4.Langages et outils utilisés
- 5.Schéma Fonctionnel

### Construction de la base de données

#### 02 -

- 1. Récolte des données
- 2. Décryptage des données
- 3. Data processing (nettoyage et mise à disposition des données)
- 4. Modèle conceptuel des données et sécurité

### Requêtage dans la base de données

#### 03 -

- 1. Requêtes principales en SQL
- 2. Optimisation de la base de données :
  - Optimiser les types de données
  - Requêtes préparées
- 3.création d'un dossier de sauvegarde Dump

### Analyse et Visualisation des données

#### 04 -

- 1. Rappel des angles d'analyse
- 2. Analyse des données avec Jupyter Notebook.
- 3. Visualisation des données avec Python et Tableau
- 4.Création d'un site avec API Flask et réalisation d'un Dashboard en D3 sur site

# PRESENTATION PERSONNELLE.

# Hello,

Après des études universitaires de philosophie qui me destinaient à l'enseignement, j'ai pu exercer ce métier quelques années, d'abord en tant que tutrice privée puis comme enseignante dans un établissement privé.

La pédagogie et la réussite scolaire sont devenus pour moi une source de questionnement. Quelle est la bonne « formule » pour faire d'un novice un élève confirmé et compétent ?

Quels sont les différents facteurs de la réussite scolaire ?

Plusieurs facteurs contribuant à la réussite des élèves sont généralement avancés: des facteurs de moyens financiers (infrastructures, livres, formations des enseignants etc.), des facteurs pédagogiques (enseignants qualifiés, méthode d'enseignement, système scolaire etc.) et des facteurs de stabilité psychologique (sécurité des élèves, environnement familial sain, nutrition, sommeil etc.).

L'école dans laquelle j'ai pu suivre une vingtaine d'élèves de CE2 pendant 3 mois, m'avais proposé d'enseigner les mathématiques selon la méthode pédagogique empruntée à Singapour. Méthode suivie dans ce petit pays et qui avait donné des résultats très satisfaisants puisque Singapour arrivait en tête des classements internationaux pour l'enseignement des mathématiques dans les classements internationaux.

Depuis ce temps-là, j'ai continué à avoir une curiosité pour les méthodes pédagogiques de certains pays qui semblent avoir trouvé la bonne formule pour enseigner une discipline particulière à leurs élèves.

Par ailleurs, le destin a voulu que je fasse partie d'une des quelques classes sélectionnées en France pour répondre aux tests de l'enquête PISA (**Programme of International Student Assessment**) en 2003.

L'enquête PISA m'a semblé une bonne base pour proposer une analyse sur les facteurs de réussite scolaire.

Par ailleurs, la recherche de données fiables et comparables sur plusieurs pays (concernant des facteurs tels que le système éducatif ou les méthodes pédagogiques employées) n'ayant pas été fructueuse, j'ai alors décidé de diriger mon analyse sur le facteur financier : l'investissement financier des pays dans l'éducation (part du PIB).



# INTRODUCTION- PROJET

## 02

Depuis longtemps, des générations d'hommes et de femmes dans le monde entier ont pris soin d'éduquer les plus jeunes pour leur assurer un avenir meilleur. L'éducation présente de nombreux avantages, non seulement pour les individus concernés mais aussi pour la société dans son ensemble. Celle-ci permet de relier les individus autour de savoirs communs et les équipe d'outils théoriques et pratiques pour la gestion du réel.

S'il semble aujourd'hui que les gens soient plus que jamais en concurrence pour l'emploi, non seulement au niveau local mais aussi au niveau international, c'est parce que l'instruction scolaire et les attentes du marché de l'emploi se sont standardisées. L'éducation est devenue l'un des instruments les plus importants pour s'assurer une place de salarié. Pour les pays, elle est devenue un enjeu économique majeur.

Les pays rivalisent pour attirer les cerveaux du monde entier et conserver leur forces vives. Cette concurrence entre les nations n'est pas un concept nouveau et l'éducation n'est qu'un des nombreux paramètres pris en compte dans les comparaisons internationales.

Il existe certainement de nombreux facteurs à considérer pour comprendre ce qui détermine les performances scolaires. L'une des questions que nous posons est de savoir s'il existe un lien entre la richesse d'une nation, ses dépenses en matière d'éducation et les résultats des étudiants.

Essayons de répondre à cette question par l'analyse d'un ensemble de données open-source mis à disposition par l'OCDE (L'enquête PISA) en créant une base de données et en analysant et visualisant ces données au moyen du langage de programmation Python.



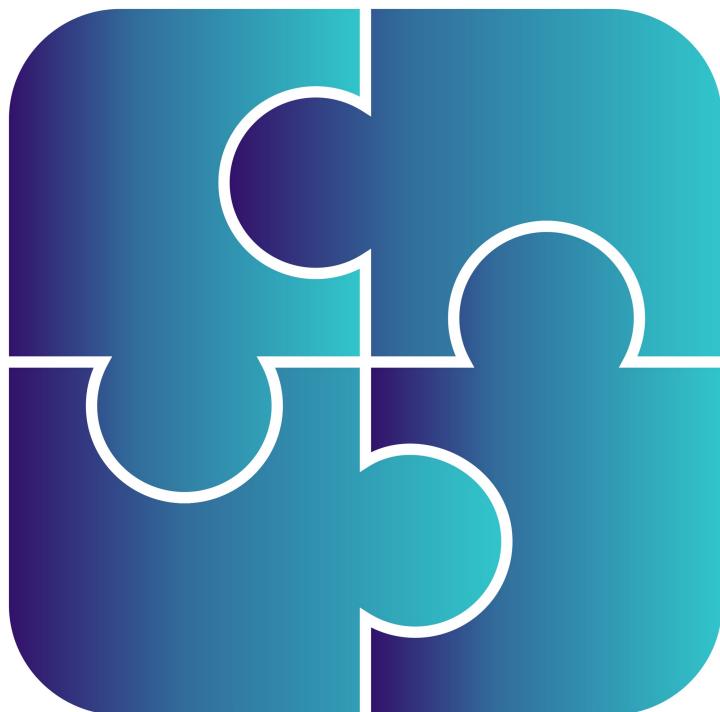
# MISSION

## 03

Ma mission en tant qu'analyste sera de regrouper toutes les données existantes concernant l'enquête PISA en une base de données afin d'offrir une vue comparative entre les pays participants à l'enquête et d'y ajouter les facteurs de réussite scolaire par pays et par année. Pour ce présent travail, je me concentrerai uniquement sur le facteur de l'investissement dans l'éducation.

Il s'agira d'étudier le rapport entre le niveau de performance scolaire à partir des tests de l'enquête PISA par pays (sur une période de 6 ans) et l'investissement des pays dans l'éducation (part du BIP en pourcentage). Le facteur matériel est-il un facteur déterminant dans la réussite scolaire ?

**Peut-on prédire le classement international PISA de l'année 2018 ?** Une fois l'année terminée, je pourrai de nouveau télécharger les données sur le site de l'OCDE pour vérifier le caractère décisif de ce facteur.



# 04

# LANGAGES ET OUTILS UTILISÉS

Nom outil	Utilisation	Raisons
<b>MySQL</b> <b>MySQL Workbench 8.0</b>	Conception de la base de données	Manager de base de données en langage SQL étudié en cours. Documentation abondante en cas de problème.
<b>DBschema</b>	Diagramme conceptuel	Gratuit, pour une durée limitée de 15 jours, inclus la possibilité de se connecter depuis plusieurs manager de base de données SQL et NoSQL (Postgre ou MongoDB par exemple)
<b>Microsoft Excel</b>	Lecture des données et exportation en fichier CSV	Préinstallé sur mon ordinateur Outil professionnel
<b>Jupyter Notebook</b>	Nettoyage des données, chargement dans la base de données	Inclus dans le package Anaconda Offre un environnement de test ergonomique et rapide
<b>SublimeText</b>	Éditeur de texte	Gratuit, tutoriels disponibles sur YouTube
<b>Python (Pandas, Numpy, Plotly)</b>	Nettoyage des données, chargement vers la base de données, Analyse, Visualisation des données	Langage de programmation le plus utilisé pour le Data Processing. Nettoyage et visualisation des données
<b>Tableau Public</b>	Visualisation des données	Gratuit, apprentissage en cours
<b>Trello</b>	Organisation du travail en cycles courts d'une semaine, Agile.	Gratuit, Outil ergonomique, Facile d'utilisation
<b>Flask</b>	API pour connecter la base de données à une interface graphique	Tutoriel sur le site : <a href="https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-iv-database">https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-iv-database</a>
<b>D3 : JavaScript</b>	Bibliothèque qui permet de faire des graphes et Dashboard dynamiques sur interface web.	Tutoriels accessibles sur Udemy.com notamment, technologie abordée lors d'un stage d'un mois en entreprise

—  
05

# SCHEMA FONCTIONNEL

Voici les étapes qui seront empruntées pour réaliser mon analyse:



02

---

# Construction de la base de données.

# 01 RECOLTE DES DONNÉES.

Les données de base pour ce projet proviennent du site de l'OCDE, les données sont en libre-service sous format CSV ou format Excel (enquête PISA + part du PIB investi par les pays de l'OCDE).

## PISA : Programme international de l'OCDE pour le suivi des acquis des élèves

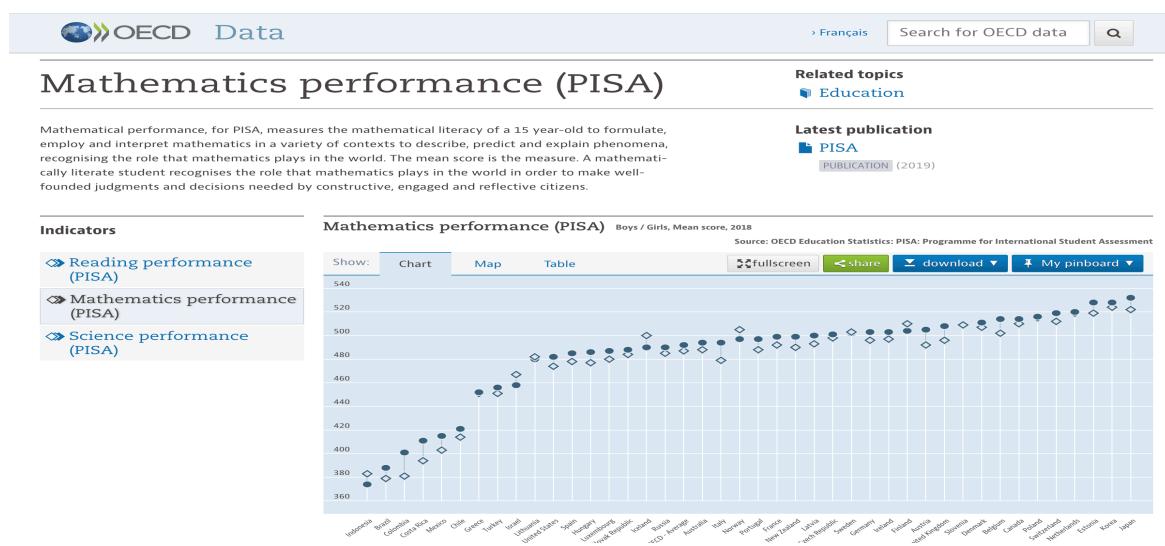
Le Programme international pour le suivi des acquis des élèves (PISA) est une évaluation internationale standardisée qui a été conjointement développée par les pays participants, et qui a été réalisée dans les écoles, à des élèves de 15 ans. Typiquement, dans chaque pays, 4 500 à 10 000 élèves participent aux test.

Vers la fin de l'éducation obligatoire, PISA évalue dans quelle mesure les élèves ont acquis certaines des connaissances et compétences essentielles à une pleine participation à la société. Dans tous les cycles, les domaines de la compréhension de l'écrit, de la culture mathématique et de la culture scientifique sont définis non pas seulement en termes d'assimilation du programme d'enseignement, mais en termes de connaissances et de compétences indispensables pour une vie d'adulte.

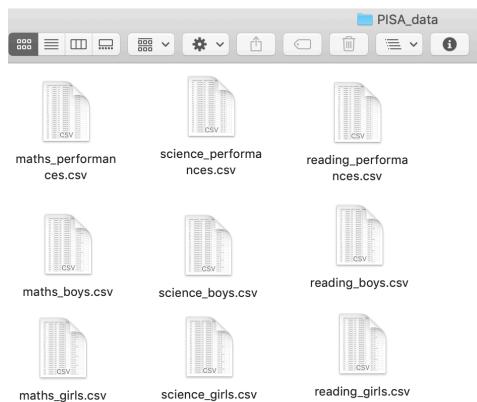
<http://www.oecd.org/pisa/>

Les données qui feront l'objet de cette étude sont téléchargeables sur l'onglet « Download » du site Internet, dont voici les lien URL et une capture d'écran ici :

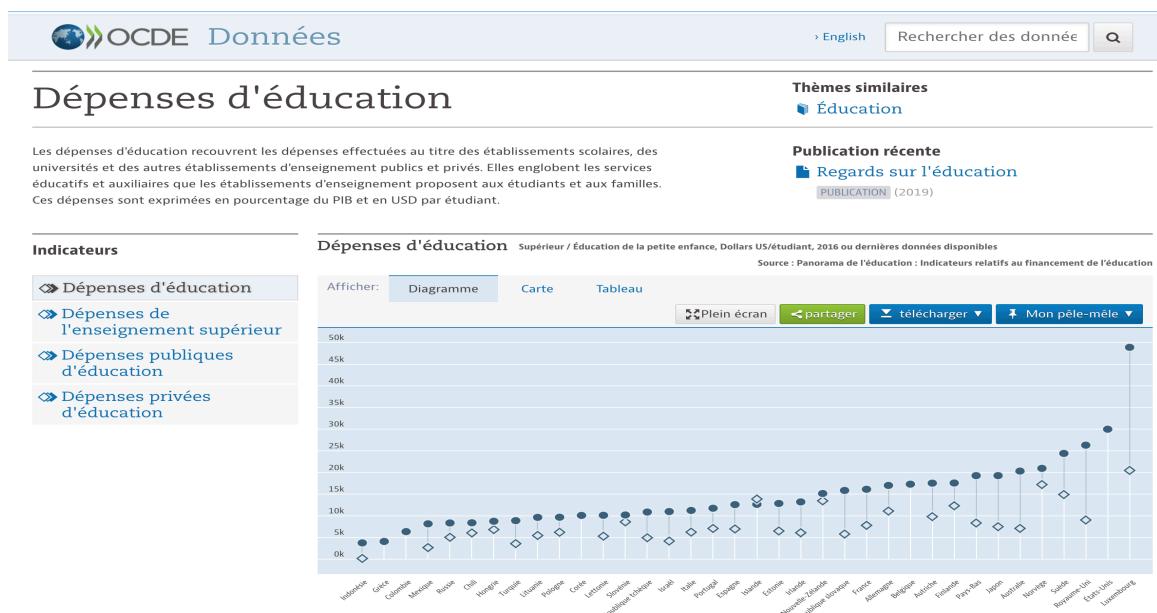
<https://data.oecd.org/pisa/mathematics-performance-pisa.htm - indicator-chart>



Sur ce site je télécharge 9 fichiers CSV (*Comma Separated Value*) correspondant aux résultats des enquêtes PISA par discipline (Mathématiques, Science et Lecture) de l'année 2006 à l'année 2015.



<https://data.oecd.org/fr/eduresource/depenses-d-education.htm>



Sur ce site je télécharge 2 fichiers CSV correspondant au budget investi par les états dans l'éducation primaire et secondaire (pourcentage du PIB du pays) allant de l'année 2006 à l'année 2015, avec une un taux de données satisfaisant entre 2012 et 2015.



Chaque document téléchargé est un fichier CSV que je vais ouvrir dans Jupiter Notebook afin de faciliter la lecture et permettre le travail de nettoyage des données. Pour ceci, j'utilise la fonction Pandas en Python : pd.read\_csv().

```

1 import os
2 import pandas as pd
3 import numpy as np

```

Import des données sur CSV

```

1 df1 = pd.read_csv('/Users/sarahsttit/Desktop/PROJET_CHEF_DOUEVRE/PISA_new_csv/math_performances.csv')

```

# 02 DECRYPTAGE DES DONNÉES.

## LECTURE DES DONNÉES BRUTES AVANT NETTOYAGE

## Fichiers CSV- Enquête PISA

## **Performances en mathématiques:**

## Lecture des données brutes :

1	df1.head(15)													
	Unnamed: 0	LOCATION	INDICATOR_x	SUBJECT_x	MEASURE_x	FREQUENCY_x	TIME	Value_x	Flag_Codes_x	INDICATOR_y	...	MEASURE_y	FREQUENCY_y	
0	0	AUS	PISAMATH	TOT	MEANSCORE		A 2003	524	NaN	PISAMATH	...	MEANSCORE	A	
1	1	AUS	PISAMATH	TOT	MEANSCORE		A 2006	520	NaN	PISAMATH	...	MEANSCORE	A	
2	2	AUS	PISAMATH	TOT	MEANSCORE		A 2009	514	NaN	PISAMATH	...	MEANSCORE	A	
3	3	AUS	PISAMATH	TOT	MEANSCORE		A 2012	504	NaN	PISAMATH	...	MEANSCORE	A	
4	4	AUS	PISAMATH	TOT	MEANSCORE		A 2015	494	NaN	PISAMATH	...	MEANSCORE	A	
5	5	AUT	PISAMATH	TOT	MEANSCORE		A 2003	506	NaN	PISAMATH	...	MEANSCORE	A	
6	6	AUT	PISAMATH	TOT	MEANSCORE		A 2006	505	NaN	PISAMATH	...	MEANSCORE	A	
7	7	AUT	PISAMATH	TOT	MEANSCORE		A 2012	506	NaN	PISAMATH	...	MEANSCORE	A	
8	8	AUT	PISAMATH	TOT	MEANSCORE		A 2015	497	NaN	PISAMATH	...	MEANSCORE	A	
9	9	BEL	PISAMATH	TOT	MEANSCORE		A 2003	529	NaN	PISAMATH	...	MEANSCORE	A	
10	10	BEL	PISAMATH	TOT	MEANSCORE		A 2006	520	NaN	PISAMATH	...	MEANSCORE	A	

## Fichiers CSV- dépenses d'éducation

## **Investissement dans l'éducation: part du PIB par pays**

#### **Investissement dans le primaire:**

1	df12 = pd.read_csv(' /Users/sarahstit/Desktop/PROJET_CHEF_DOUEVRE/data_invest/invest_primary.csv' )							
1	df12.head()							
<b>LOCATION INDICATOR SUBJECT MEASURE FREQUENCY TIME Value Flag Codes</b>								
0	AUS	EDUEXP	EARLYCHILDEDU	PC_GDP	A	1995	NaN	M
1	AUS	EDUEXP	EARLYCHILDEDU	PC_GDP	A	2000	0.090	NaN
2	AUS	EDUEXP	EARLYCHILDEDU	PC_GDP	A	2005	0.072	NaN
3	AUS	EDUEXP	EARLYCHILDEDU	PC_GDP	A	2008	0.086	NaN
4	AUS	EDUEXP	EARLYCHILDEDU	PC_GDP	A	2009	0.106	NaN

Tous les fichiers CSV issus du site de l'OCDE devront être nettoyés (renommer les colonnes, supprimer les colonnes inutiles, fusionner les *dataframes* des différents fichiers CSV, optimiser le types de données etc.) pour en faciliter la lecture et l'intégration dans la base de données.

## DECRYPTER LES DONNÉES FINALES APRES NETTOYAGE

Nomenclature des *dataframes* obtenus en fin de traitement pour l'importation dans la base de données :

Dataframe classement\_PISA :

Nom de la colonne	Datatype (Python)	Datatype (SQL)	Description
<b>Discipline</b>	Object	Varchar (15)	Le nom de la discipline sur laquelle ont été évalués les élèves (Mathématiques, Sciences ou Lecture)
<b>Genre</b>	Objet	Varchar (15)	Le type de population évaluée : Filles, Garçons, Globale (Filles et Garçons).
<b>Moyenne</b>	Float64	Float(12,0)	La valeur moyenne obtenue à un examen sur une population donnée, une année donnée.
<b>Id_Pays</b>	Int64	Smallint(6)	Index qui permet de renvoyer à un pays unique présent dans le <i>dataframe</i> puis la table : Table_Pays .
<b>Id_Année</b>	Int64	Smallint(6)	Index qui permet de renvoyer à une année unique présente dans le <i>dataframe</i> puis la table : Table_Année

Dataframe Investissements :

Nom de la colonne	Datatype (Python)	Datatype (SQL)	Description
<b>Primaire invest</b>	Float64	Float(12,0)	La valeur en pourcentage de la part du PIB investi dans l'éducation primaire, pour une année donnée.
<b>Secondaire invest</b>	Float64	Float(12,0)	La valeur en pourcentage de la part du PIB investi dans l'éducation secondaire, pour une année donnée.
<b>Id_Pays</b>	Int64	Smallint(6)	Index qui permet de renvoyer à un pays unique présent dans le <i>dataframe</i> de la table : Table_Pays .
<b>Id_Année</b>	Int64	Smallint(6)	Index qui permet de renvoyer à une année unique présente dans le <i>dataframe</i> de la table : Table_Année

03

# DATA PROCESSING : NETTOYAGE DES DONNÉES.

Tous les fichiers CSV issus de l'enquête PISA suivent le même processus de nettoyage (suppression de colonnes inutiles, renommer des colonnes, fusion de *dataframes*, changement de type de données).

Après nettoyage des données brutes j'obtiens un premier *dataframe* par discipline (mathématiques, Sciences et Lecture) sous la forme suivante :

```
1 df1.head()
```

Pays	Année	Moyenne Globale Maths	Moyenne Garçons Maths	Moyenne Filles Maths
0 AUS	2003	524	527.000	522.000
1 AUS	2006	520	527.000	513.000
2 AUS	2009	514	519.000	509.000
3 AUS	2012	504	510.115	497.821
4 AUS	2015	494	497.000	491.000

```
1 len(df1)
```

189

J'ai ensuite fusionné les trois *dataframes* par discipline (mathématiques, sciences et lecture) en un seul:

Fusionner les DATAFRAMES :

```
: 1 df4 =pd.merge(df1, df2, on=["Pays", "Année"])
```

```
: 1 df4
```

	Pays	Année	Moyenne Globale Maths	Moyenne Garçons Maths	Moyenne Filles Maths	Moyenne Globale Sci	Moyenne Garçons Sci	Moyenne Filles Sci
0	AUS	2006	520	527.000	513.000	527	527.000	527.000
1	AUS	2009	514	519.000	509.000	527	527.000	528.000
2	AUS	2012	504	510.115	497.821	521	523.728	519.124
3	AUS	2015	494	497.000	491.000	510	511.000	509.000
4	AUT	2006	505	517.000	494.000	511	515.000	507.000
...	...	...	...	...	...	...	...	...
151	HKG	2015	548	549.000	547.000	523	523.000	524.000
152	PER	2015	387	391.000	382.000	397	402.000	392.000
153	SGP	2015	564	564.000	564.000	556	559.000	552.000
154	TWN	2015	542	545.000	539.000	532	535.000	530.000
155	MAC	2015	544	540.000	548.000	529	525.000	532.000

156 rows x 8 columns

J'ai enfin obtenu le *dataframe* suivant en fin de processus :

```
1 PISA_dataframe
```

	Pays	Année	Moyenne Globale Maths	Moyenne Garçons Maths	Moyenne Filles Maths	Moyenne Globale Sci	Moyenne Garçons Sci	Moyenne Filles Sci	Moyenne Globale Lecture	Moyenne Garçons Lecture	Moyenne Filles Lecture
0	AUS	2006	520	527.000	513.000	527	527.000	527.000	513	495.00	532.000
1	AUS	2009	514	519.000	509.000	527	527.000	528.000	515	496.00	533.000
2	AUS	2012	504	510.115	497.821	521	523.728	519.124	512	495.09	529.542
3	AUS	2015	494	497.000	491.000	510	511.000	509.000	503	487.00	519.000
4	AUT	2006	505	517.000	494.000	511	515.000	507.000	490	468.00	513.000
...	...	...	...	...	...	...	...	...	...	...	...
148	HKG	2015	548	549.000	547.000	523	523.000	524.000	527	513.00	541.000
149	PER	2015	387	391.000	382.000	397	402.000	392.000	398	394.00	401.000
150	SGP	2015	564	564.000	564.000	556	559.000	552.000	535	525.00	546.000
151	TWN	2015	542	545.000	539.000	532	535.000	530.000	497	485.00	510.000
152	MAC	2015	544	540.000	548.000	529	525.000	532.000	509	493.00	525.000

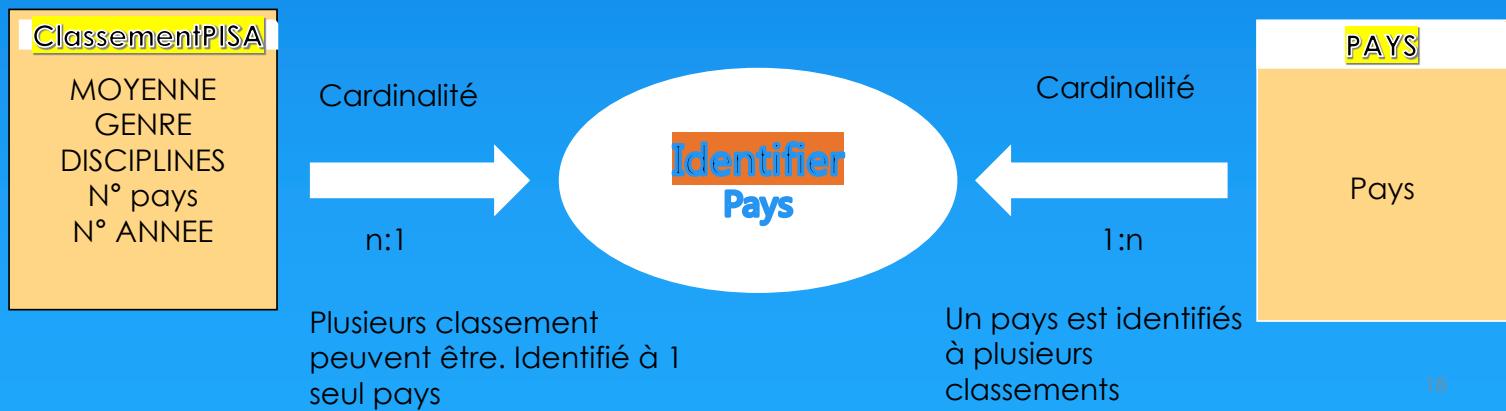
153 rows x 11 columns

```
1 len(PISA_dataframe)
```

## MODÈLE CONCEPTUEL DE DONNÉES

Schéma Entités -Relations - Associations

*Analyse conceptuelle des données (monde réel)*



## MODÈLE CONCEPTUEL DE DONNÉES

Schéma Entités -Relations - Associations

*Analyse conceptuelle des données (monde réel)*



Le modèle conceptuel de données a permis de dégager des relations « one to many » à mettre en évidence lors de la création de la base de données. Du **dataframe PISA\_dataframe**, j'ai pu dégager trois nouveaux **dataframes correspondant chacun à une table** de ma future base de données que j'enregistre en CSV: classement\_PISA - Table\_Pays - Table\_Année.

Les étapes suivantes sont dédiées à la construction de ces 3 *dataframes* :

## DATAFRAME PISA

Les fusions successives des différents *dataframes* qui ont permis la lecture de PISA\_dataframe ont fait perdre à celui-ci sa structure initiale qui comprenait peu de colonnes.

En langage Python, avec l'aide de loan, j'ai pu reconstituer un *dataframe* optimisé pour une insertion en base de données.

### Creation tableau PISA

```
In [6]: 1 Tableau_PISA = PISA_dataframe

In [ ]: 1 #Changer la structure du dataframe
2 columns_to_keep = ['Moyenne Globale Maths',
3                     'Moyenne Garçons Maths', 'Moyenne Filles Maths', 'Moyenne Globale Sci',
4                     'Moyenne Garçons Sci', 'Moyenne Filles Sci', 'Moyenne Globale Lecture',
5                     'Moyenne Garçons Lecture', 'Moyenne Filles Lecture']
6
7 mapper = {'Moyenne Globale Maths' : 'Math',
8           'Moyenne Garçons Maths' : 'Math', 'Moyenne Filles Maths' : 'Math', 'Moyenne Globale Sci' : 'Sciences',
9           'Moyenne Garçons Sci': 'Sciences', 'Moyenne Filles Sci': 'Sciences', 'Moyenne Globale Lecture': 'Lecture',
10          'Moyenne Garçons Lecture' : 'Lecture', 'Moyenne Filles Lecture': 'Lecture'}
11 #initialiser dataframe
12 Discipline_df = pd.DataFrame(columns=['Année', 'Pays', 'Discipline', 'Genre', 'Moyenne'])
13
14 for row_num in range(len(Tableau_PISA)): #row_num prendra toutes les valeurs des nombres de 0 à 152
15     row = Tableau_PISA.loc[row_num] #la variable row sera la ligne du dataframe à la position de row_num
16     for col in columns_to_keep: #pour chaque colonne qu'on veut garder
17         Discipline_df = Discipline_df.append({'Moyenne' : row[col], 'Discipline' : mapper[col], \
18                                         'Genre' : col.split(' ')[1], 'Année' : row.Année, 'Pays' : row.Pays}, \
19                                         ignore_index = True)
20     #ajouter une nouvelle ligne au dataframe discipline_df

In [8]: 1 classement_PISA = Discipline_df

In [9]: 1 # arrondir la 'moyenne globale PISA' à deux chiffre apres la virgule
2 decimals = 2
3 classement_PISA['Moyenne'] = classement_PISA['Moyenne'].apply(lambda x: round(x, decimals))
```

Nous obtenons le *dataframe* suivant:

```
1 classement_PISA
```

	Année	Pays	Discipline	Genre	Moyenne
0	2006	AUS	Math	Globale	520.0
1	2006	AUS	Math	Garçons	527.0
2	2006	AUS	Math	Filles	513.0
3	2006	AUS	Sciences	Globale	527.0
4	2006	AUS	Sciences	Garçons	527.0
...	...	...	...	...	...
1372	2015	MAC	Sciences	Garçons	525.0
1373	2015	MAC	Sciences	Filles	532.0
1374	2015	MAC	Lecture	Globale	509.0
1375	2015	MAC	Lecture	Garçons	493.0
1376	2015	MAC	Lecture	Filles	525.0

1377 rows × 5 columns

- La colonne **Année** comprend les valeurs annuelles des précédentes enquêtes PISA : **2009, 2012 et 2015**.
- La colonne **Pays** contient les valeurs des noms de pays partenaires de l'enquête PISA au format standard ISO 3.
- La colonne **Discipline** comprend les valeurs **Math, Sciences et Lecture**.
- La colonne **Genre** correspond à la population étudiée, elle comprend les valeurs **Filles, Garçons et Globale** (filles et garçons).
- La colonne **Moyenne** comprend les valeurs de moyenne obtenues par les élèves lors de l'enquête PISA.

## DATAFRAME PAYS

### Création d'un tableau Pays:

```
1 Table_Pays = classement_PISA['Pays'].unique()
2 Table_Pays = pd.DataFrame(Table_Pays)
3 Table_Pays.rename(columns={0: 'Pays'}, inplace=True)

1 #renommer le nom des colonnes
2 Table_Pays.rename(columns={'Pays': 'ISO3'}, inplace=True)

1 Table_Pays.head(15)

ISO3
0 AUS
1 AUT
2 BEL
3 CAN
4 CZE
5 DNK
6 FIN
7 FRA
8 DEU
9 GRC
10 HUN
11 ISL
12 IRL
13 ITA
14 JPN
```

J'ai ensuite scrapé deux sites, l'un contenant les informations de longitude, latitude et le code pays ISO 2 et l'autre le nom des pays et le code pays ISO 3 dont j'ai besoin pour opérer un « *merge* » sur mon dataframe. Il s'agira de faire fusionner le *dataframe* scrapé avec le *dataframe* Table\_Pays:

```
1 #page scrapée pour intégrer les latitudes et longitudes des pays:
2 coordonnées_pays = pd.read_csv('/Users/sarahstit/Desktop/PROJET CHEF DOUEVRE/PISA_new_csv/coordonnées_pays.csv')

1 coordonnées_pays.head(100)

   Pays_x ISO2 ISO3 Numeric  latitude  longitude      Pays_y
0 Afghanistan AF AFG        4  33.939110  67.709953 Afghanistan
1 Albania AL ALB        8  41.153332  20.168331    Albania
2 Algeria DZ DZA       12  28.033866  1.659626      Algeria
3 American Samoa AS ASM       16 -14.270972 -170.132217 American Samoa
4 Andorra AD AND       20  42.546245  1.601554      Andorra
... ...
95 Holy See (the) VA VAT      336  41.902916  12.453389 Vatican City
96 Honduras HN HND      340  15.199999 -86.241905      Honduras
97 Hong Kong HK HKG      344  22.396428  114.109497      Hong Kong
98 Hungary HU HUN      348  47.162494  19.503304      Hungary
99 Iceland IS ISL      352  64.963051 -19.020835      Iceland
100 rows × 7 columns
```

```
1 # faire un merge pour obtenir les coordonnées de géolocalisation:
2 Table_Pays = pd.merge(Table_Pays, coordonnées_pays, on=['ISO3'])
```

Nous obtenons la Table\_pays contenant 44 pays distincts. Les longitudes et latitudes serviront pour la visualisation des données. Je conserve la colonne ISO3 qui est un référentiel standard pour les pays:

```
1 Table_Pays

   ISO3  latitude  longitude      Pays
0 AUS -25.274398  133.775136 Australia
1 AUT  47.516231  14.550072 Austria
2 BEL  50.503887  4.469936 Belgium
3 CAN  56.130366 -106.346771 Canada
4 CZE  49.817492  15.472982 Czech Republic
5 DNK  56.633920  9.501785 Denmark
6 FIN  61.924110  25.748151 Finland
7 FRA  46.227638  2.213749 France
8 DEU  51.165691  10.451526 Germany
9 GRC  39.074208  21.824312 Greece
10 HUN 47.162494  19.503304 Hungary
11 ISL  64.963051 -19.020835 Iceland
12 IRL  53.412910 -8.243890 Ireland
13 ITA  41.871940  12.567380 Italy
14 JPN  36.204824  138.252924 Japan
15 KOR  35.907757  127.769222 South Korea
16 LUX  49.815273  6.129583 Luxembourg
```

## DATAFRAME ANNÉE

### Création un tableau Année:

```
: 1 #Créer un dataframe Année:  
2 Table_Année = classement_PISA['Année'].unique()  
3 Table_Année = pd.DataFrame(Table_Année)  
4 Table_Année.rename(columns={0: 'Année'}, inplace=True)
```

```
: 1 Table_Année
```

```
:  
: Année  
0 2006  
1 2009  
2 2012  
3 2015
```

## DATAFRAME INVESTISSEMENTS

Voici une part du traitement du *dataframe* et le résultat après nettoyage :

```
: 1 #suppression colonnes  
2 df13= df13.drop(['INDICATOR', 'SUBJECT', 'MEASURE', 'FREQUENCY', 'Flag Codes'], axis = 1)  
3 df13
```

```
:  
: LOCATION TIME Value  
0 AUS 1995 NaN  
1 AUS 2000 0.090  
2 AUS 2005 0.072  
3 AUS 2008 0.086  
4 AUS 2009 0.106  
... ... ...  
541 ZAF 2012 NaN  
542 ZAF 2013 NaN  
543 ZAF 2014 NaN  
544 ZAF 2015 NaN  
545 ZAF 2016 NaN
```

546 rows × 3 columns

```
: 1 #renommer le nom des colonnes  
2 df12.rename(columns={'LOCATION': 'Pays', 'TIME': 'Année', 'Value': 'Primaire'}, inplace=True)
```

```
: 1 df12.head()
```

```
:  
: Pays Année Primaire  
0 AUS 1995 NaN  
1 AUS 2000 0.090  
2 AUS 2005 0.072  
3 AUS 2008 0.086  
4 AUS 2009 0.106
```

```
: 1 #vérifier la présence de valeurs nulles  
2 print("Frequency count of missing values")  
3 df12.apply(lambda X:sum(X.isnull()))
```

Frequency count of missing values

### Fusionner le DATAFRAME :

```
: 1 Investissement_PIB =pd.merge(df12, df13, on=["Pays", "Année"])
```

```
: 1 Investissement_PIB
```

```
:  
: Pays Année Primaire Secondaire  
0 AUS 1995 NaN NaN  
1 AUS 2000 0.090 0.090  
2 AUS 2005 0.072 0.072  
3 AUS 2008 0.086 0.086  
4 AUS 2009 0.106 0.106  
... ... ... ...  
541 ZAF 2012 NaN NaN  
542 ZAF 2013 NaN NaN
```

# — MODÈLE

## 04 CONCEPTUEL DE DONNÉES (MCD).

Passons à une étape décisive de la création d'une base de données : Modélisation des tables.

Il s'agit du **schéma** de la future base de données Education\_PISA.

Pour construire mon modèle conceptuel de données, j'ai choisi l'outils DBschema:

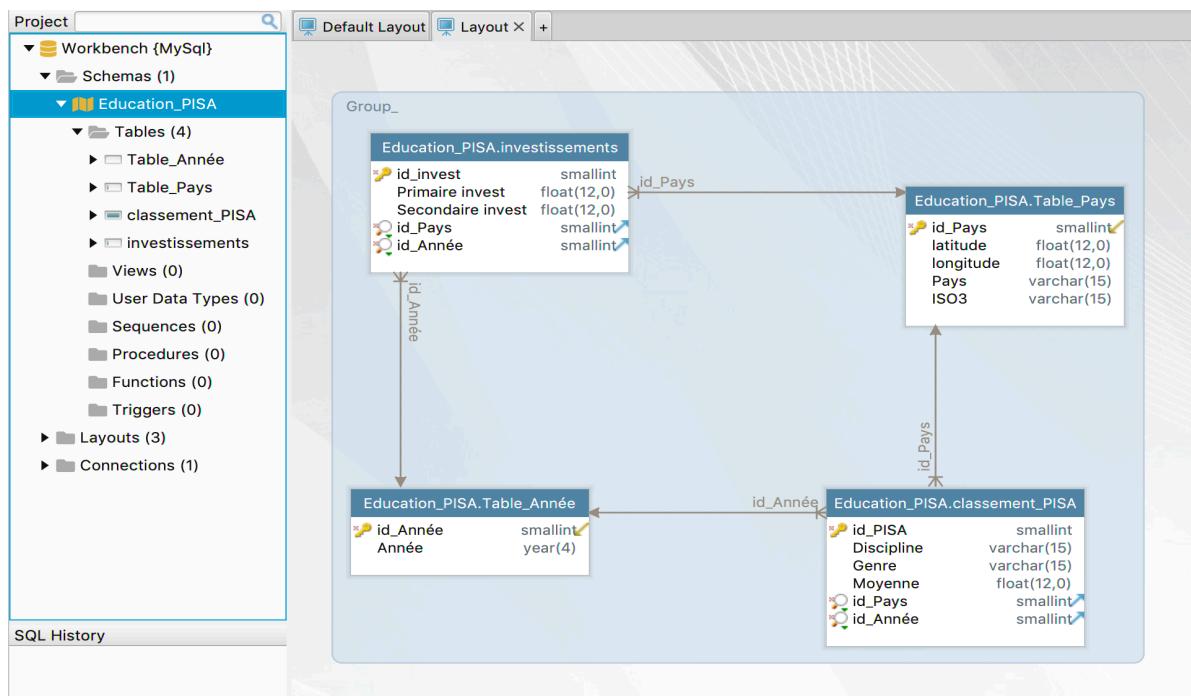


Figure - MCD de la Base de données Education\_PISA

### La création des tables

- ▶ Les **clés primaires** servent à **identifier une ligne** de manière unique
- ▶ Les **clés étrangères** permettent de gérer des **relations entre plusieurs tables**

### Relation un-à-plusieurs

**Un-à-plusieurs:** dans une relation un-à-plusieurs, une ligne peut être liée à plusieurs lignes. Un seul pays peut être lié à plusieurs années, recensant les différents classements.

## SÉCURITÉ DE LA BASE DE DONNÉES

Pour garantir la sécurité de ma base de données, je crée trois comptes « utilisateur » avec des fonctions précises : un compte administrateur **Admin\_PISA** qui reçoit tous les priviléges et deux comptes invités (**Guest1\_PISA**, **Guest2\_PISA**) avec des priviléges réduit d'accès et de consultation de la base. Tous avec un mot de passe différent de plus de 10 caractères<sup>1</sup> pour prévenir d'une attaque.

The screenshot shows the 'Users and Privileges' section in MySQL Workbench. A new user 'Admin\_PISA' is being created with 'localhost' as the 'From Host'. In the 'Administrative Roles' tab, all roles are selected: DBA, MaintenanceAdmin, ProcessAdmin, UserAdmin, SecurityAdmin, MonitorAdmin, DBManager, DBDesigner, ReplicationAdmin, BackupAdmin, and Custom. Under 'Global Privileges', all options are checked, including ALTER ROUTINE, CREATE, and GRANT OPTION. The 'Description' column for each role provides a brief explanation of its functions.

### SÉCURISATION DE LA BASE DE DONNÉES

The screenshot shows the 'Users and Privileges' section in MySQL Workbench. A new user 'Admin\_PISA' is being created with 'localhost' as the 'From Host'. To the right, a 'MySQL Connections' window shows a connection to 'Education\_PISA' on '127.0.0.1:3306'. A password entry dialog is open, prompting for a password for the service 'Mysql@127.0.0.1:3306' with the user 'Admin\_PISA'. The password field contains '\*\*\*\*\*'.

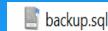
#### CREATION D' UN COMPTE ADMINISTRATEUR:

- Lui accordant tous les priviléges
- Avec un mot de passe protégé

# Creation d'un dump automatique via Workbench

# Creation manuelle d'un dump via le shell:

Path =  
cd /usr/local/mysql-8.0.18-macos10.14-x86\_64/bin



Save=  
mysqldump -u Admin\_PISA -p Education\_PISA >  
/Users/sarahstit/Desktop/PROJET CHEF DOEUVRE/Dump/backup.sql

# Restauration manuelle d'un dump via le shell:

mysqladmin -u Admin\_PISA -p create Education\_PISA

mysql -u Admin\_PISA -p Education\_PISA  
</Users/sarahstit/Desktop/PROJET CHEF  
DOEUVRE/Dump/backup.sql

# — OPTIMISATION ET

## 05 IMPORTATION DES

## DONNÉES.

Après avoir nettoyé les données et modélisé notre future base de données relationnelle, il est judicieux d'optimiser au maximum l'insertion des données dans la base.

Puisque la base sera composée de quatre tables liées entre elles par des clés étrangères, il faut créer des index qui permettront de relier les tables et supprimer les colonnes redondantes.

### DATAFRAME CLASSEMENT\_PISA

**Créer un index unique pour la colonne 'Pays':**

```
1 #Faire un mapping entre les valeurs de la colonne ISO3 et les index
2 pays = Table_Pays['ISO3'].to_dict()
3 pays = {v: k for k, v in pays.items()}
4

1 # Trouver les pays en trop dans le dataframe classement_PISA
2 set(classement_PISA.ISO3.unique())-set(Table_Pays.ISO3.unique())
{'OAVG'}

1 # supprimer les lignes qui ne sont pas présentent dans le dataframe
2 classement_PISA = classement_PISA[classement_PISA.ISO3 != 'OAVG']

1 # Pour chaque Pays dans la colonne ISO3 ajouter l'index correspondant à la liste vide id_Pays
2 id_Pays= []
3
4 for i in classement_PISA['ISO3']:
    id_Pays.append(pays[i])

1 # Inserer une nouvelle colonne id_Pays à partir de la liste id_Pays
2 classement_PISA['id_Pays'] = id_Pays
3 classement_PISA
4
```

**Créer un index unique pour la colonne 'Année':**

```
1 années = Table_Année['Année'].to_dict()
2 années = {v: k for k, v in années.items()}
3 années

{2006: 0, 2009: 1, 2012: 2, 2015: 3}

1 id_Année= []
2
3 for i in classement_PISA['Année']:
    id_Année.append(années[i])
4

3 classement_PISA
```

	Discipline	Genre	Moyenne	id_Pays	id_Année
0	Math	Globale	520.0	0	0
1	Math	Garçons	527.0	0	0
2	Math	Filles	513.0	0	0
3	Sciences	Globale	527.0	0	0
4	Sciences	Garçons	527.0	0	0
...	...	...	...	...	...
1372	Sciences	Garçons	525.0	43	3
1373	Sciences	Filles	532.0	43	3
1374	Lecture	Globale	509.0	43	3
1375	Lecture	Garçons	493.0	43	3
1376	Lecture	Filles	525.0	43	3

1359 rows × 5 columns

## DATAFRAME INVESTISSEMENTS

Exemple d'optimisation du *dataframe* :

Le *dataframe investissements* : une transformation des données d'investissement en pourcentage.

Sur le même modèle que le *dataframe* précédent, un id\_Pays et un id\_Année a été ajouté sur le *dataframe investissements*. Les colonnes Pays et Année ont été supprimées.

### Convertir les données en pourcentage:

```
1 Investissement_PIB['Primaire invest'] = Investissement_PIB['Primaire'] * 100  
1 Investissement_PIB['Secondaire invest'] = Investissement_PIB['Secondaire'] * 100  
1 #suppression des colonnes inutiles  
2 Investissement_PIB=Investissement_PIB.drop(['Primaire', 'Secondaire'], axis = 1)  
3 Investissement_PIB
```

	Pays	Année	Primaire invest	Secondaire invest
0	AUS	1995	NaN	NaN
1	AUS	2000	9.0	9.0
2	AUS	2005	7.2	7.2
3	AUS	2008	8.6	8.6
4	AUS	2009	10.6	10.6
...	...	...	...	...
541	ZAF	2012	NaN	NaN
542	ZAF	2013	NaN	NaN
543	ZAF	2014	NaN	NaN
544	ZAF	2015	NaN	NaN
545	ZAF	2016	NaN	NaN

546 rows × 4 columns

Il s'agit maintenant de créer un index unique à partir du *Dataframe Table\_Pays* et de la *Table\_Année* (sur les colonne ISO3 et Année):

## Créer un index unique pour la colonne 'Pays':

## Egaliser les colonnes pays des dataframes investissements et table\_pays:

comme les dataframes sont issus de csv différents, ils ne comptent pas le même nombre de pays ni les mêmes noms de pays. Il faut faire en sorte d'uniformiser les deux colonnes pour pouvoir créer des index communs (id).

```
In [721]: 1 pays = Table_Pays['ISO3'].to_dict()
2 pays = {v: k for k, v in pays.items()}
3 pays
```

```
Out[721]: {'AUS': 0,
'AUT': 1,
'BEL': 2,
'CAN': 3,
'CZE': 4,
'DNK': 5,
'FIN': 6,
'FRA': 7,
'DEU': 8,
'GRC': 9,
'HUN': 10,
'ISL': 11,
'IRL': 12,
'ITA': 13,
'JPN': 14,
```

```
In [722]: 1 invest = investissements['ISO3'].to_dict()
2 invest = {v: k for k, v in invest.items()}
3 invest
```

```
Out[722]: {'AUS': 11,
'AUT': 23,
'BEL': 35,
'CAN': 46,
'CZE': 58,
'DNK': 70,
'FIN': 82,
'FRA': 94,
'DEU': 103,
'GRC': 115,
'HUN': 127,
'ISL': 139,
'IRL': 151,
'ITA': 163,
'JPN': 175,
'KOR': 187,
'LUX': 199,
'MEX': 211,
'NLD': 223,
'NZL': 235,
'NOR': 247,
'POL': 259,
```

```
1 #Vérifier qu'il n'y a plus de valeurs non concordantes (soustraction des dictionnaires)
2 set(investissements.ISO3.unique())-set(Table_Pays.ISO3.unique())
```

```
{'ARG', 'CHN', 'CRI', 'IND', 'LTU', 'SAU', 'ZAF'}
```

```
1 # supprimer les lignes qui ne sont pas présentent dans le dataframe Table_pays
2 investissements = investissements[-investissements.ISO3.isin(['ARG', 'CHN', 'CRI', 'IND', 'LTU', 'SAU', 'ZAF'])]
```

```
1 #Vérifier qu'il n'y a plus de valeurs non concordantes
2 set(Table_Pays.ISO3.unique())-set(investissements.ISO3.unique())
```

```
{'HKG', 'MAC', 'PER', 'SGP', 'TWN'}
```

```
1 #Ajouter les lignes manquantes dans le dataframe investissements
2 #'HKG', 'MAC', 'PER', 'SGP', 'TWN'
```

```
1 #calculer la longueur de la colonne ISO3 dans le colonne investissement
2 len(Table_Pays.ISO3.unique())
```

```
44
```

```
1 # calculer la longueur de la colonne ISO3 dans le dataframe investissement
2 len(investissements.ISO3.unique())
```

```
44
```

Maintenant, les deux colonnes ISO3 des dataframes *Table\_Pays* et *investissements* sont bien identiques et comportent le même nombre de colonnes.

Primaire invest	Secondaire invest	id_Pays	id_Année
4	10.6	10.6	0
7	43.3	43.3	0
10	57.3	57.3	0
16	NaN	NaN	1
19	59.0	59.0	1
			2

```
1 #exporter le nouveau dataframe optimisé dans un format csv:
2 investissements.to_csv('investissements.csv')
```

## EXPLORATION DES DONNÉES DANS LA BASE MYSQL

Une fois les quatre *dataframes* optimisés, je crée une base de donnée sur Workbench au nom d' Education\_PISA, puis dans mon Notebook Jupyter je lance un programme d'exportation **par tables** des données vers ma base, dont voici les captures d'écran ici :

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view shows a database named 'Education\_PISA' selected. Under it, there are tables like 'classement\_PISA', 'investissements', 'Table\_Année', and 'Table\_Pays'. On the right, the main pane displays the SQL code for creating these objects:

```
1 • CREATE DATABASE IF NOT EXISTS Education_PISA ;
2 • USE Education_PISA;
3
4
5 • DROP TABLE IF EXISTS Table_Année;
6
7 • CREATE TABLE Table_Année (
8     id_Année int(15),
9     Année year(4)
10 );
11
12
13 • DROP TABLE IF EXISTS Table_Pays;
14
15 • CREATE TABLE Table_Pays (
16     id_Pays int(15),
17     Pays varchar(15),
18     Nom varchar(15),
19     latitude float(15),
20     longitude float(15)
21 );
22
```

### Connecter Pandas à la Base de données avec SQLAlchemy

#### Intégrer les données depuis Pandas à la la base de données SQL

The screenshot shows a Jupyter Notebook with several code cells:

- In [3]:

```
1 import pandas as pd
2 import numpy as np
3 import csv
```
- In [4]:

```
1 #if connecting to MySQL, installing PyMySQL
2 #pip install pymysql
3 import pymysql
```
- In [906]:

```
1 Table_Pays.head(10)
```
- Out[906]:

	ISO3	latitude	longitude	Pays
0	AUS	-25.274398	133.775136	Australia
1	AUT	47.516231	14.550072	Austria
2	BEL	50.503887	4.469936	Belgium
3	CAN	56.130366	-106.346771	Canada
4	CZE	49.817492	15.472962	Czech Republic
5	DNK	56.263920	9.501785	Denmark
6	FIN	61.924110	25.748151	Finland
7	FRA	46.227638	2.213749	France
8	DEU	51.165691	10.451526	Germany
9	GRC	39.074208	21.824312	Greece
- In [907]:

```
1 # importer le module
2 from sqlalchemy import create_engine
3
4 # creer sqlalchemy engine
5 engine = create_engine("mysql+pymysql://user:{pw}@localhost/{db}"
6                         .format(user="root",
7                                 pw="10132310SS",
8                                 db="Education_PISA"))
9 # inserer les données dans mysql
10 Table_Pays.to_sql('Table_Pays', con = engine, if_exists = 'replace', index=True)
```

# 03

---

## REQUÊTAGE DANS LA BASE DE DONNÉES.

# 01

---

## REQUÊTES PRINCIPALES.

Une fois la base de données générée je lance des requêtes SQL pour obtenir un calcul de la base.

Voici un exemple de requête, je voulais savoir quel était le TOP10 des pays les mieux classés en mathématiques en 2015 :

```
21      -- Classement TOP 10 des pays qui ont obtenu les meilleurs résultats en Mathématiques en 2015 sur toute la population étudiée
22
23 •  SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre
24   FROM classement_PISA
25   JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année)
26   JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays)
27 WHERE Table_Année.Année = 2015 AND classement_PISA.Genre = 'Globale' AND classement_PISA.Discipline= 'Math'
28 ORDER BY classement_PISA.Moyenne DESC
29 LIMIT 10;
```

Result 2

Pays	Année	Discipline	Moyenne	Genre
Singapore	2015	Math	564	Globale
Hong Kong	2015	Math	548	Globale
Macau	2015	Math	544	Globale
Taiwan	2015	Math	542	Globale
Japan	2015	Math	532	Globale
South Korea	2015	Math	524	Globale
Switzerland	2015	Math	521	Globale
Estonia	2015	Math	520	Globale
Canada	2015	Math	516	Globale
Netherlands	2015	Math	512	Globale

Voici un autre exemple de requête, je voulais savoir quel était le classement BOTTOM 10 des pays qui ont obtenus les moins bons résultats en mathématiques en 2015 :

```
42      -- Classement BOTTOM 10 des pays qui ont obtenu les résultats les plus faibles en Mathématiques en 2015 sur toute la population étudiée
43
44 •  SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre
45   FROM classement_PISA
46   JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année)
47   JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays)
48 WHERE Table_Année.Année = 2015 AND classement_PISA.Genre = 'Globale' AND classement_PISA.Discipline= 'Math'
49 ORDER BY classement_PISA.Moyenne ASC
50 LIMIT 10;
```

Result 3

Pays	Année	Discipline	Moyenne	Genre
Brazil	2015	Math	377	Globale
Indonesia	2015	Math	386	Globale
Peru	2015	Math	387	Globale
Colombia	2015	Math	390	Globale
Mexico	2015	Math	408	Globale
Turkey	2015	Math	420	Globale
Chile	2015	Math	423	Globale
Greece	2015	Math	454	Globale
United States	2015	Math	470	Globale
Israel	2015	Math	470	Globale

## — 02 OPTIMISATION DE LA BASE.

Optimiser la base de donnée, c'est réduire le temps de traitement des données. Pour ce faire nous pouvons :

- user d'alias dans la requête
- optimiser les types de données ou *datatypes* (volumétrie des données)
- créer des requêtes préparées (qui stockent des requêtes sous une variable)
- indexation des tables (déjà réalisée dans notre cas)

Connaître précisément la taille d'une base de données avant et après optimisation de la base peut être très intéressant pour un administrateur de base de données. J'utilise la requête suivante :

AVANT OPTIMISATION :

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing the 'Education\_PISA' schema selected. Under 'Tables', there are several tables listed: 'classement\_PISA', 'investissements', 'Table\_Année', 'Table\_Pays', 'Views', 'Stored Procedures', and 'Functions'. In the main pane, a SQL editor window displays the following query:

```
1 • | USE Education_PISA;
2
3 • | SELECT
4 |   table_schema AS NomBaseDeDonnees,
5 |   ROUND(SUM( data_length + index_length ) / 1024 / 1024, 2) AS BaseDonneesMo
6 | FROM information_schema.TABLES
7 | GROUP BY TABLE_SCHEMA;
```

Below the SQL editor is a 'Result Grid' window. It has two columns: 'NomBaseDeDonnees' and 'BaseDonneesMo'. The data is as follows:

NomBaseDeDonnees	BaseDonneesMo
mysql	2.36
information_schema	0.00
performance_schema	0.00
Education_PISA	0.30

APRES OPTIMISATION :

The screenshot shows the MySQL Workbench interface again, with the 'SCHEMAS' tree view and the same SQL query as before. The 'Result Grid' window now shows the following data:

NomBaseDeDonnees	BaseDonneesMo
Education_PISA	0.27
information_schema	0.00
mysql	2.36
performance_schema	0.00

The 'BaseDonneesMo' value for the 'mysql' schema has been reduced from 2.36 to 0.27, indicating a successful optimization.

## OPTIMISATION : les datatypes

Après avoir exécuté mon script de création de table, je lance un script pour modifier les *datatypes*, grâce à la commande MODIFY:

```
24 • DROP TABLE IF EXISTS classement_PISA;
25
26 • CREATE TABLE classement_PISA (
27     id_PISA int(15),
28     Discipline varchar(15),
29     Genre varchar(15),
30     Moyenne float(15),
31     id_Pays int(15),
32     id_Année int(15)
33 );
34
35
36 • DROP TABLE IF EXISTS investissements;
37
38 • CREATE TABLE investissements (
39     id_invest int(15),
40     Primaire_invest float(15),
41     Secondaire_invest float(15),
42     id_Pays int(15),
43     id_Année int(15)
44 );
45
1 • USE Education_PISA;
2
3 -- changer le type des colonnes dans table classement_PISA
4
5 • ALTER TABLE `classement_PISA`
6     CHANGE `index` `id_PISA` smallint,
7     MODIFY `Discipline` varchar(15),
8     MODIFY `Genre` varchar(15),
9     MODIFY `Moyenne` float4(15),
10    MODIFY `id_Pays` smallint(15),
11    MODIFY `id_Année` smallint(15);
12
13 -- changer le type des colonnes dans table investissements
14
15 • ALTER TABLE `investissements`
16     CHANGE `index` `id_invest` smallint,
17     MODIFY `Primaire invest` float4(15),
18     MODIFY `Secondaire invest` float4(15),
19     MODIFY `id_Pays` smallint(15),
```

## OPTIMISATION : Usage d'alias

```
19     -- Classement TOP 10 des pays qui ont obtenu les meilleurs résultats en Mathématiques en 2015 sur toute la population étudiée
20
21 • SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre
22     FROM classement_PISA
23     JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année)
24     JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays)
25     WHERE Table_Année.Année = 2015 AND classement_PISA.Genre = 'Globale' AND classement_PISA.Discipline= 'Math'
26     ORDER BY classement_PISA.Moyenne DESC
27     LIMIT 10;
28
29     -- Requête optimisée:
30
31 • SELECT Pays,Année,Discipline,Moyenne,Genre
32     FROM classement_PISA AS C
33     JOIN Table_Année AS A ON C.id_Année = A.id_Année
34     JOIN Table_Pays AS P ON C.id_Pays = P.id_Pays
35     WHERE A.Année = 2015 AND C.Genre = 'Globale' AND C.Discipline= 'Math'
36     ORDER BY C.Moyenne DESC
37     LIMIT 10;
38
```

## OPTIMISATION : les requêtes préparées

Voici un exemple de requête « JOIN », un type de jointure liant plusieurs tables entre-elles.

Cette commande retourne les colonnes pertinentes de ma base de données. Nous remarquons que la vitesse d'exécution de la requête préparée est bien plus courte que la même requête sans « préparation ».

```
1      -- Selection de quelques colonnes pertinentes dans chacunes des tables existantes.
2
3
4 •  SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre,investissements.`Primaire invest`,
5   FROM classement_PISA
6   JOIN investissements ON (classement_PISA.id_Pays=investissements.id_Pays AND classement_PISA.id_Année=investissements.id_Année)
7   JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année )
8   JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays);
9
10
11 --
12
13 •  SET @req = 'SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre,investissements.`Primaire invest`,
14   FROM classement_PISA
15   JOIN investissements ON (classement_PISA.id_Pays=investissements.id_Pays AND classement_PISA.id_Année=investissements.id_Année)
16   JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année )
17   JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays)';
18 •  PREPARE PISA_et_investissements
19   from @req;
20
21 •  EXECUTE PISA_et_investissements;
22
```

Action Output			
	Time	Action	Response
✓ 6	02:43:16	USE Education_PISA	0 row(s) affected
✓ 7	02:43:16	SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre,investissements.`Primaire invest`	981 row(s) returned
✓ 8	02:44:23	SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre,investissements.`Primaire invest`	981 row(s) returned
✓ 9	02:44:23	SET @req = 'SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,classement_PISA.Genre,investissements.`Primaire invest`'	0 row(s) affected
✓ 10	02:44:23	PREPARE PISA_et_investissements from @req	0 row(s) affected Statement prepared
✓ 11	02:44:23	EXECUTE PISA_et_investissements	981 row(s) returned
✓ 12	02:44:42	EXECUTE PISA_et_investissements	981 row(s) returned
✓ 13	02:44:57	EXECUTE PISA_et_investissements	981 row(s) returned

Une requête préparée est une **requête stockée en mémoire** elle permet une plus grande vitesse d'exécution et évite le gaspillage des ressources. Elle a aussi un intérêt dans la sécurisation des données, évitant ainsi toutes injections SQL.

La requête préparée se fait en trois temps :

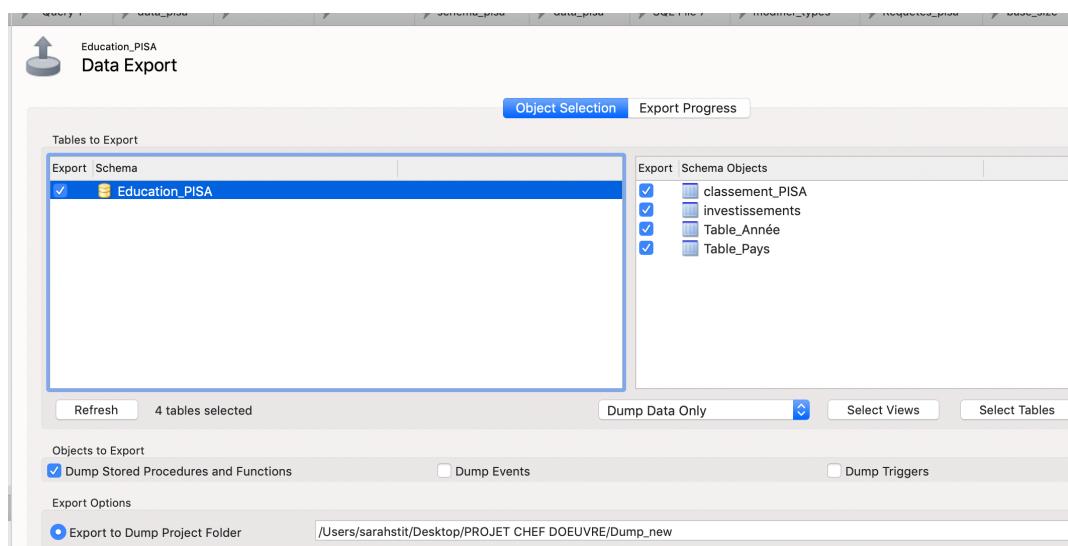
- La création de la variable « set @req = »
- Mise sous variable d'une requête SQL classique. PREPARE + le nom qu'on lui donne pour être appelée
- L'exécution de la requête préparée : EXECUTE + le nom de la requête préparée

Result Grid										
Pays	Année	Discipline	Moyenne	Genre	Primaire invest	Secondaire invest	latitude	longitude	ISO3	
Australia	2009	Math	509	Filles	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Sciences	527	Globale	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Sciences	527	Garçons	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Sciences	528	Filles	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Lecture	515	Globale	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Lecture	496	Garçons	10.6	10.6	-25.2744	133.775	AUS	
Australia	2009	Lecture	533	Filles	10.6	10.6	-25.2744	133.775	AUS	
Australia	2012	Math	504	Globale	43.3	43.3	-25.2744	133.775	AUS	
Australia	2012	Math	510.12	Garçons	43.3	43.3	-25.2744	133.775	AUS	

# — SAUVEGARDE DE 03 LA BASE.

## SAUVEGARDE DE LA BASE DE DONNÉES - DUMP

Parmi les tâches assignées à un administrateur, la sauvegarde et la restauration des bases de données occupent une place importante. Personne n'est à l'abri d'une perte de données. Je vais créer un fichier « Dump » pour sauvegarder la base de données depuis MySQL Workbench.



J'obtiens alors 8 fichiers correspondants aux schémas de chaque table et aux data de chaque table.

Voici un exemple du code de sauvegarde data pour la « table investissements » :

```
--  
18  --  
19  -- Dumping data for table `investissements`  
20  --  
21  
22 •  LOCK TABLES `investissements` WRITE;  
23 •  /*!40000 ALTER TABLE `investissements` DISABLE KEYS */;  
24 •  INSERT INTO `investissements` VALUES (0,10.6,10.6,0,1),(1,43.3,43.3,0,2),(2,57.3,  
25 •  /*!40000 ALTER TABLE `investissements` ENABLE KEYS */;  
26 •  UNLOCK TABLES;  
27 •  /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;  
28  
29 •  /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
30 •  /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
31 •  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
32 •  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
33 •  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
34 •  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
35 •  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
36
```

Dans mon dossier Dump contenant les fichiers de sauvegarde de la base de données Education\_PISA, je constitue trois fichiers appelés : schema.sql, data.sql et contraintes.sql

04

# ANALYSE ET VISUALISATION.

01

## RAPPEL DES ANGLES D'ANALYSE.

Il y a probablement un certain nombre de facteurs qui doivent être pris en compte lorsqu'on se questionne sur ce qui aide les élèves à se préparer à la poursuite d'études supérieures. L'une des questions que nous pouvons poser est de savoir s'il existe un lien entre la richesse d'une nation ou ses dépenses dans l'éducation et la performance des élèves.

Voici les angles que je souhaite aborder dans mon analyse comparative entre réussite scolaire et investissement:

- Le facteur financier a-t-il un impact déterminant sur les performances scolaires ?
- Les filles et les garçons sont-ils égaux face à la réussite scolaire dans tous les pays et les matières étudiés? Y a-t-il des disciplines « genrées » qui ont la préférences des filles ou des garçons ?
- Peut-on prédire la réussite scolaire par pays selon le critère du financement dans l'éducation ?

# 02 ANALYSE DES DONNÉES AVEC JUPYTER NOTEBOOK.

Voici certaines des bibliothèques Python que j'ai importé pour réaliser mon analyse visuelle :

```
1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 import plotly.offline as py
7 py.init_notebook_mode(connected=True)
8 import plotly.graph_objs as go
9 from plotly import tools
10 import plotly.figure_factory as ff
11
12 import warnings
```

Usage d'un programme d'interrogation de la base de données via une requête SQL :

```
In [51]: 1 # import le module
2 from sqlalchemy import create_engine
3
4 # créer un moteur sqlalchemy et connexion à ma base de donnée
5 engine = create_engine("mysql+pymysql://{}:{}@localhost/{}"
6                         .format(user="Admin_PISA",
7                                pw="10132310SS",
8                                db="Education_PISA"))

In [52]: 1 merge_sql = pd.read_sql('''
2     SELECT Table_Pays.Pays,Table_Année.Année,classement_PISA.Discipline,classement_PISA.Moyenne,
3     FROM classement_PISA
4     JOIN investissements ON (classement_PISA.id_Pays=investissements.id_Pays AND classement_PISA.
5     JOIN Table_Année ON (classement_PISA.id_Année=Table_Année.id_Année )
6     JOIN Table_Pays ON (classement_PISA.id_Pays=Table_Pays.id_Pays); ''',con = engine)
7

In [53]: 1 merge_sql
```

Pays	Année	Discipline	Moyenne	Genre	Primaire invest	Secondaire invest	latitude	longitude	ISO3
0 Australia	2009	Math	514.0	Globale	10.6	10.6	-25.2744	133.7750	AUS
1 Australia	2009	Math	519.0	Garçons	10.6	10.6	-25.2744	133.7750	AUS
2 Australia	2009	Math	509.0	Filles	10.6	10.6	-25.2744	133.7750	AUS
3 Australia	2009	Sciences	527.0	Globale	10.6	10.6	-25.2744	133.7750	AUS
4 Australia	2009	Sciences	527.0	Garçons	10.6	10.6	-25.2744	133.7750	AUS
...	...	...	...	...	...	...	...	...	...
976 Slovenia	2015	Sciences	510.0	Garçons	112.7	112.7	46.1512	14.9955	SVN
977 Slovenia	2015	Sciences	516.0	Filles	112.7	112.7	46.1512	14.9955	SVN

# PRÉ-ANALYSE DONNÉES PISA

Voici comment j'obtiens le classement Pisa par année sur la moyenne globale des trois disciplines évaluées :

## Analyse classement:

```
1 PISA_dataframe = pd.read_csv('/Users/sarahstit/Desktop/PROJET CHEF DOUEUVRE/PISA_new_csv/PISA_dataframe.csv')
2 #exporter les colonnes intéressantes
3 classement_PISA = PISA_dataframe[['Pays', 'Année','Moyenne Globale Maths','Moyenne Globale Sci','Moyenne Globale Lecture']]
4
5 classement_PISA
```

	Pays	Année	Moyenne Globale Maths	Moyenne Globale Sci	Moyenne Globale Lecture
0	AUS	2006	520	527	513
1	AUS	2009	514	527	515
2	AUS	2012	504	521	512
3	AUS	2015	494	510	503
4	AUT	2006	505	511	490
...	...	...	...	...	...
148	HKG	2015	548	523	527
149	PER	2015	387	397	398
150	SGP	2015	564	556	535
151	TWN	2015	542	532	497
152	MAC	2015	544	529	509

153 rows × 5 columns

## Créer une colonne "moyenne" de toutes les disciplines confondues pour déterminer le classement PISA par année:

```
1 # Créer une colonne moyenne
2 classement_PISA['Moyenne Globale PISA'] = classement_PISA[['Moyenne Globale Maths','Moyenne Globale Sci','Moyenne Globale Lecture']].mean(axis=1)
3
4 # arrondir la 'moyenne globale PISA' à deux chiffre apres la virgule
5 decimals = 2
6 classement_PISA['Moyenne Globale PISA'] = classement_PISA['Moyenne Globale PISA'].apply(lambda x: round(x, decimals))
7
8 #regrouper le dataframe par Année et par Pays afin de faire un classement par Année
9 classement_PISA = classement_PISA.groupby(['Année','Pays']).mean()
10
11 # classer la colonne 'Moyenne Globale PISA' par ordre descroissant par année de classement
12 classement_PISA = classement_PISA.sort_values(by=['Année','Moyenne Globale PISA'], ascending=False)
13
14 classement_PISA
```

	Année	Pays	Moyenne Globale Maths	Moyenne Globale Sci	Moyenne Globale Lecture	Moyenne Globale PISA
		SGP	564	556	535	551.67
		HKG	548	523	527	532.67
	2015	JPN	532	538	516	528.67
		MAC	544	529	509	527.33
		EST	520	534	519	524.33
		...	...	...	...	...
		TUR	424	424	447	431.67
		CHL	411	438	442	430.33
	2006	MEX	406	410	410	408.67
		IDN	391	393	393	392.33
		BRA	370	390	393	384.33

153 rows × 4 columns

## ANALYSE DONNÉES de la base Education\_PISA

Montrer la moyenne générale des pays par années, Singapour a intégré le classement PISA en 2015 :

```
1 # montrer toutes les colonnes qui concernent Singapour
2 Singapore = mysql_PISA=mysql_PISA[ 'Pays' ] == 'Singapore'
```

```
1 Singapore
```

	Année	Pays	Moyenne	Primaire invest	Secondaire invest	latitude	longitude
0	2015	Singapore	551.666667	NaN	NaN	1.35208	103.82

```
1 # montrer toutes les colonnes qui concernent le Japon
2 Japan = mysql_PISA=mysql_PISA[ 'Pays' ] == 'Japan'
```

```
1 Japan
```

	Année	Pays	Moyenne	Primaire invest	Secondaire invest	latitude	longitude
2	2015	Japan	528.777778	20.3	20.3	36.2048	138.253
44	2012	Japan	540.348889	22.3	22.3	36.2048	138.253
81	2009	Japan	529.555556	NaN	NaN	36.2048	138.253

```
1 # montrer toutes les colonnes qui concernent la France
2 France = mysql_PISA=mysql_PISA[ 'Pays' ] == 'France'
3 France
```

	Année	Pays	Moyenne	Primaire invest	Secondaire invest	latitude	longitude
23	2015	France	495.777778	74.5	74.5	46.2276	2.21375
58	2012	France	499.646667	69.9	69.9	46.2276	2.21375
95	2009	France	496.888889	71.7	71.7	46.2276	2.21375

Sélectionner uniquement les lignes contenant la même année pour pouvoir comparer les performances des pays :

Exemple avec l'année 2015

```
1 # montrer toutes les colonnes qui concernent l'année 2015
2 Année_2015 = mysql_PISA=mysql_PISA[ 'Année' ] == 2015
3 Année_2015
```

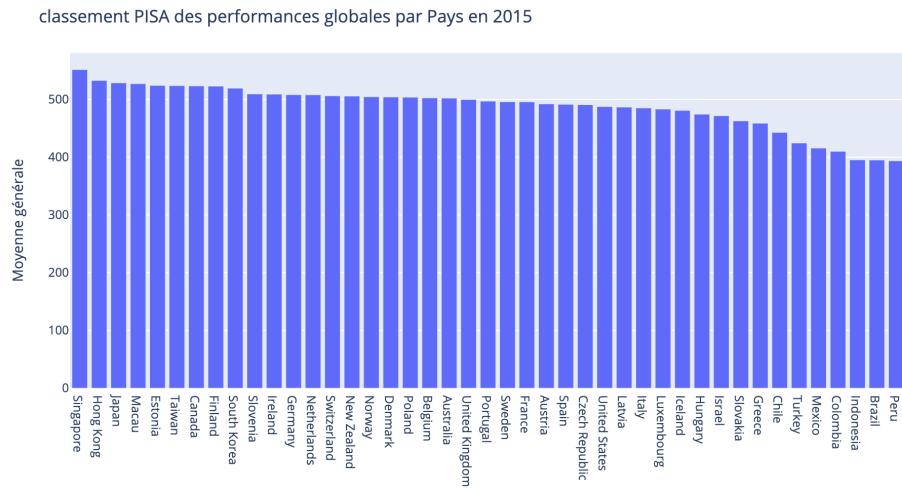
	Année	Pays	Moyenne	Primaire invest	Secondaire invest	latitude	longitude
0	2015	Singapore	551.666667	NaN	NaN	1.352080	103.82000
1	2015	Hong Kong	532.777778	NaN	NaN	22.396400	114.10900
2	2015	Japan	528.777778	20.3	20.3	36.204800	138.25300
3	2015	Macau	527.222222	NaN	NaN	22.198700	113.54400
4	2015	Estonia	524.333333	118.5	118.5	58.595300	25.01360
5	2015	Taiwan	523.888889	NaN	NaN	23.697800	120.96100
6	2015	Canada	523.444444	NaN	NaN	56.130400	-106.34700
7	2015	Finland	523.000000	123.7	123.7	61.924100	25.74820
8	2015	South Korea	519.444444	NaN	NaN	35.907800	127.76700
9	2015	Slovenia	509.555556	112.7	112.7	46.151200	14.99550
10	2015	Ireland	509.111111	7.4	7.4	53.412900	-8.24389
11	2015	Germany	508.111111	86.6	86.6	51.165700	10.45150
12	2015	Netherlands	508.000000	39.3	39.3	52.132600	5.29127
13	2015	Switzerland	506.222222	NaN	NaN	46.818200	8.22751
14	2015	New Zealand	505.888889	96.2	96.2	-40.900600	174.88600
15	2015	Norway	504.555556	182.5	182.5	60.472000	8.46895
16	2015	Denmark	504.333333	NaN	NaN	56.263900	9.50179

# —

## 03 VISUALISATION DES DONNÉES.

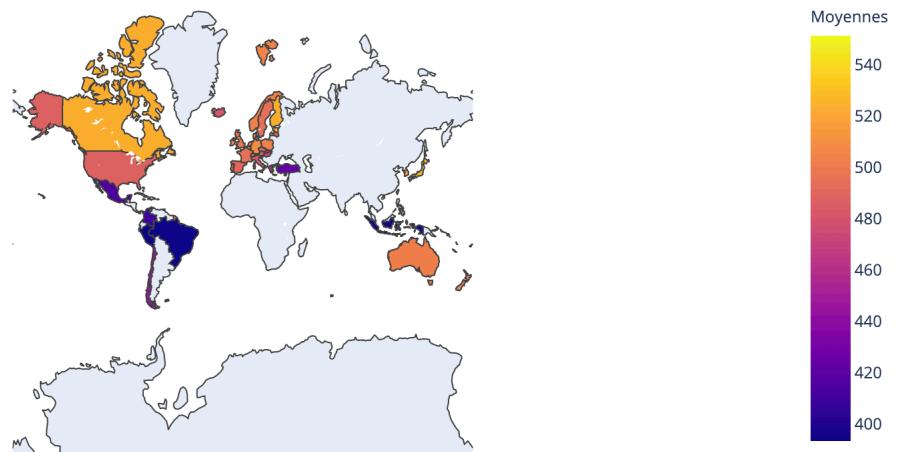
### CLASSEMENT PISA

Voici le classement PISA des performances globales par Pays en 2015 :

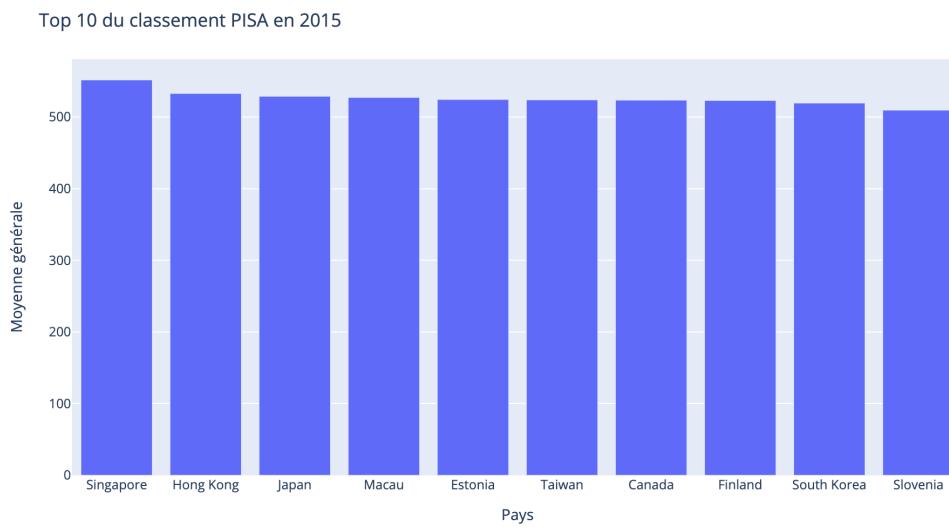


Voici la carte du classement PISA des performances globales par Pays en 2015 :

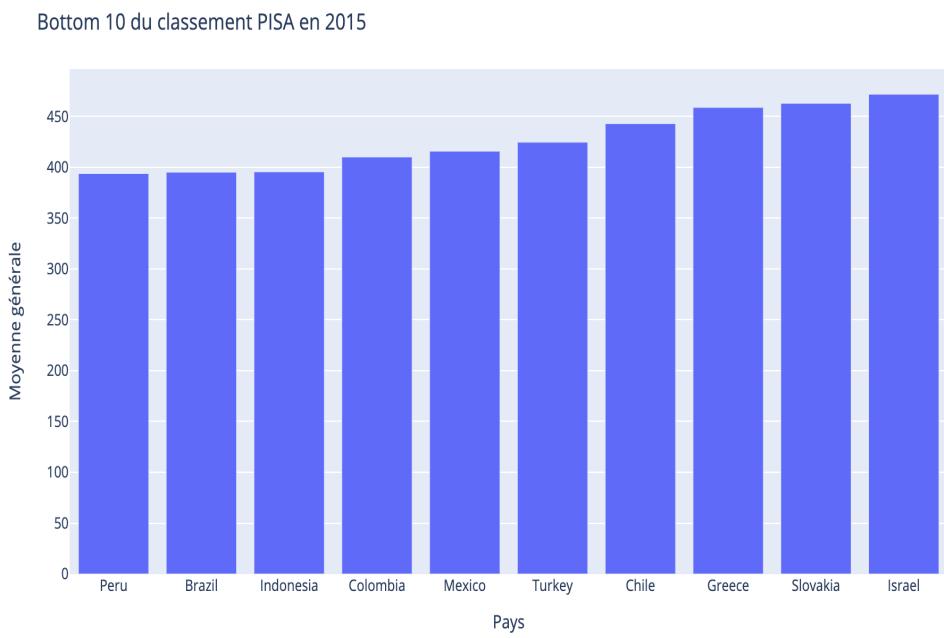
Classement PISA des pays en 2015



Voici le Top 10 des pays qui ont obtenu les meilleurs résultats en 2015 :



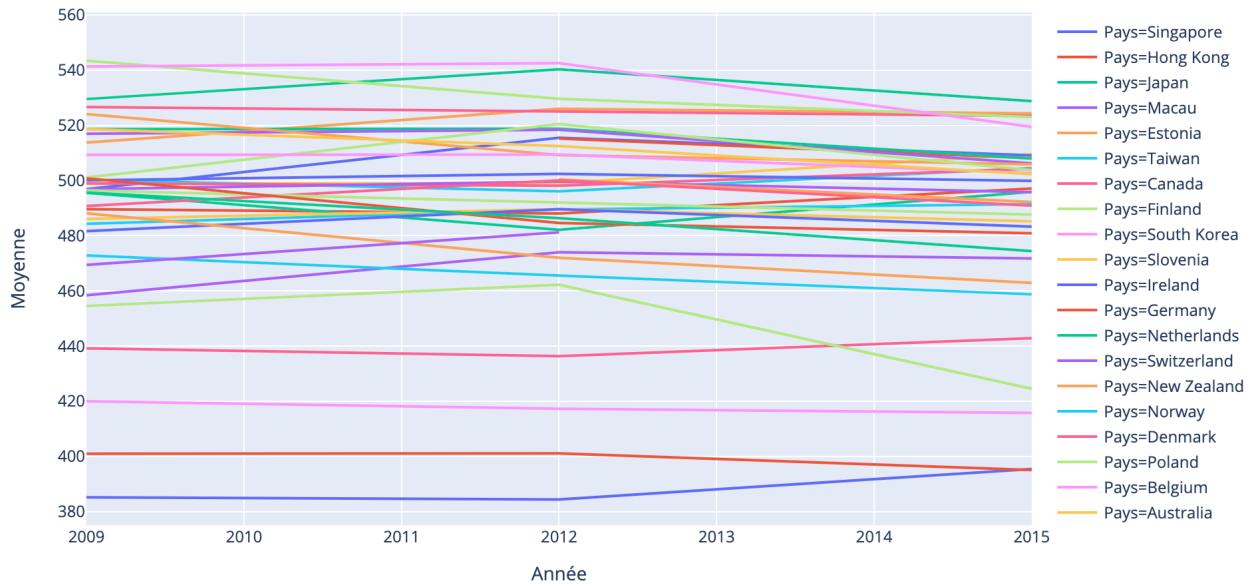
Voici le Bottom 10 des pays qui ont obtenu les moins bons résultats en 2015 :



Performance globale des Pays :

## Performances globales des pays depuis 2009:

```
1 fig = px.line(mysql_PISA, x="Année", y="Moyenne", color='Pays')
2 fig.show()
```

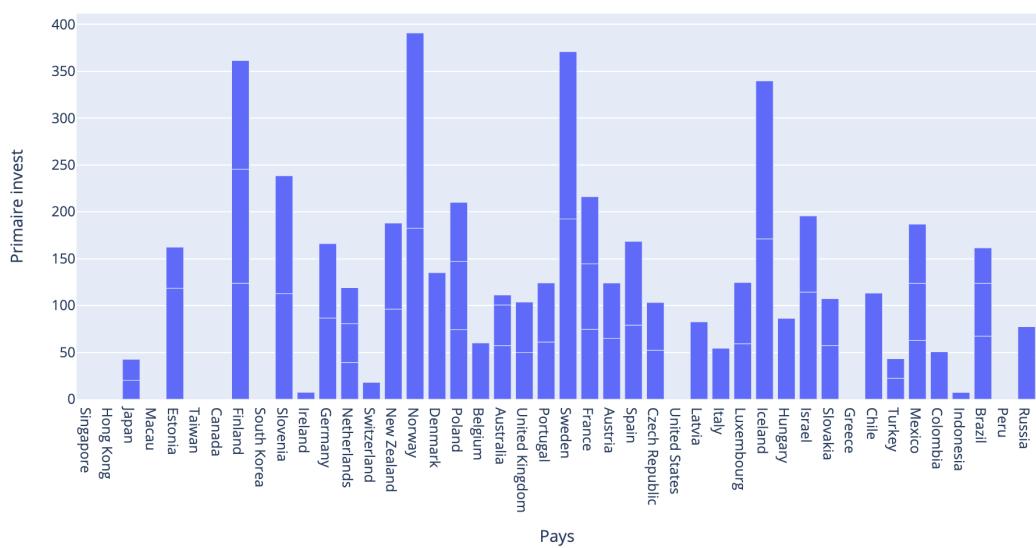


## ANALYSE INVESTISSEMENT

Investissements des pays entre 2009 et 2015, les données sont manquantes pour les pays qui ont un niveau d'investissement nul:

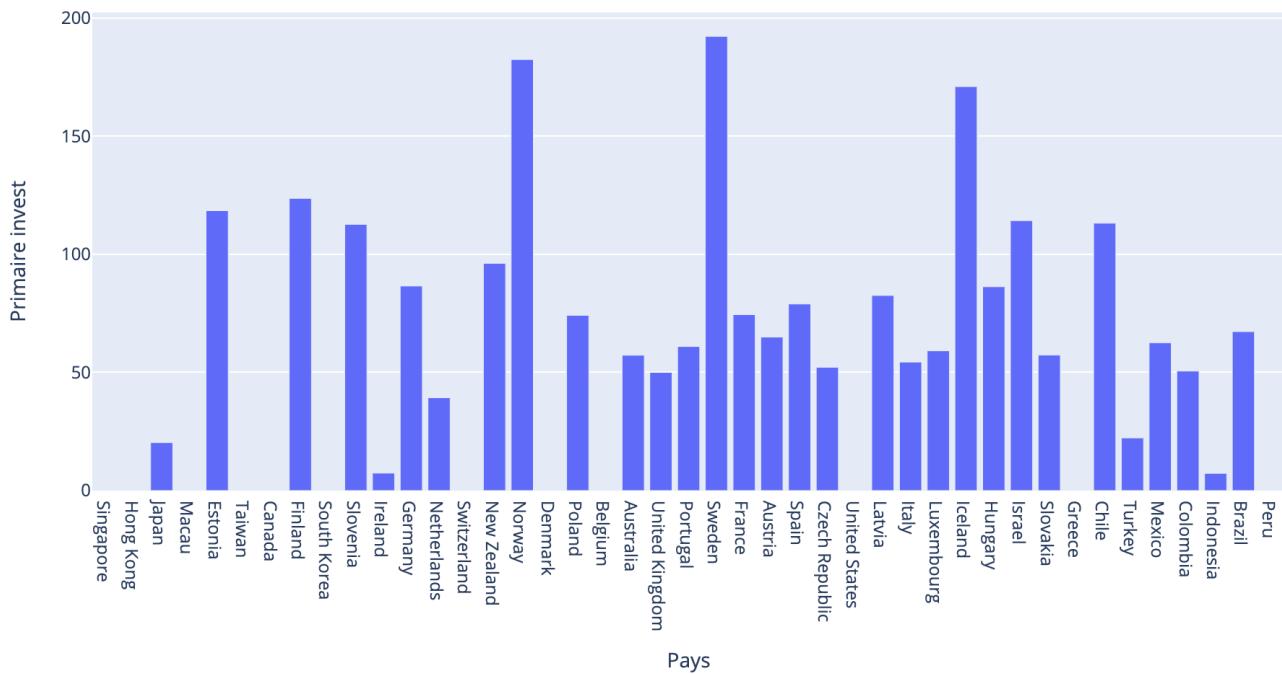
```
1 fig = px.bar(mysql_PISA, x='Pays', y='Primaire invest')
2
3 fig.update_layout(title_text="Investissement par Pays entre 2009-2015 dans l'ordre du classement PISA",
4 xaxis= dict(title= ' Pays', ticklen= 5,zeroline= False),
5 yaxis= dict(title= " Primaire invest", ticklen= 5,zeroline= False))
```

Investissement par Pays entre 2009-2015 dans l'ordre du classement PISA

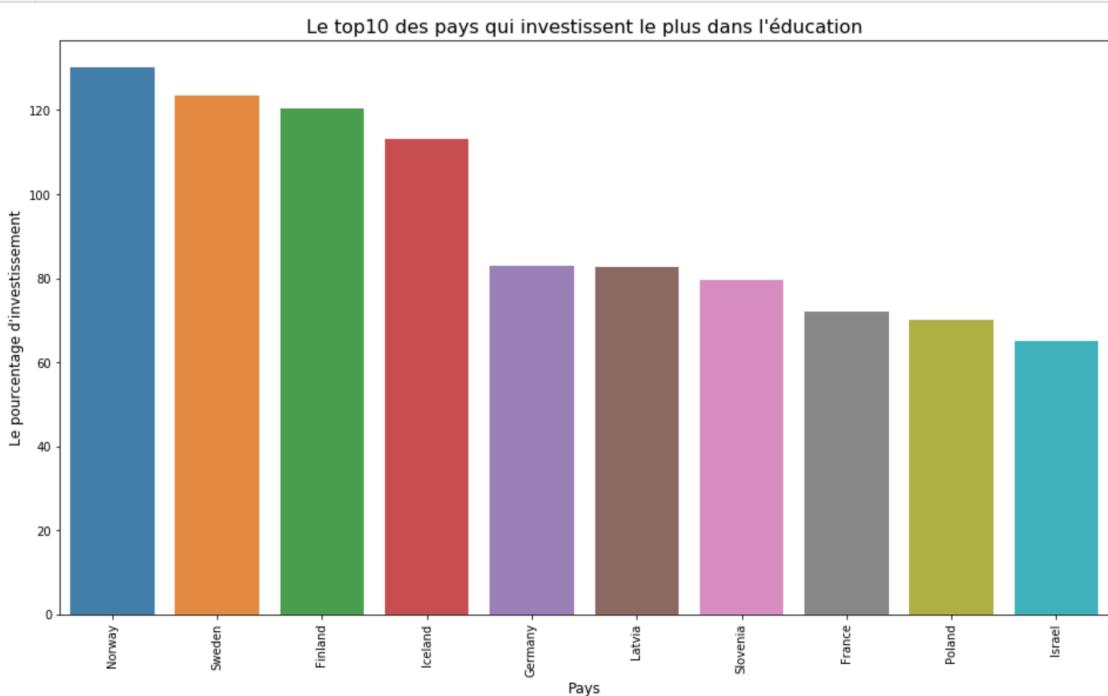


## Investissements par pays pour l'année 2015 dans l'ordre de performances PISA :

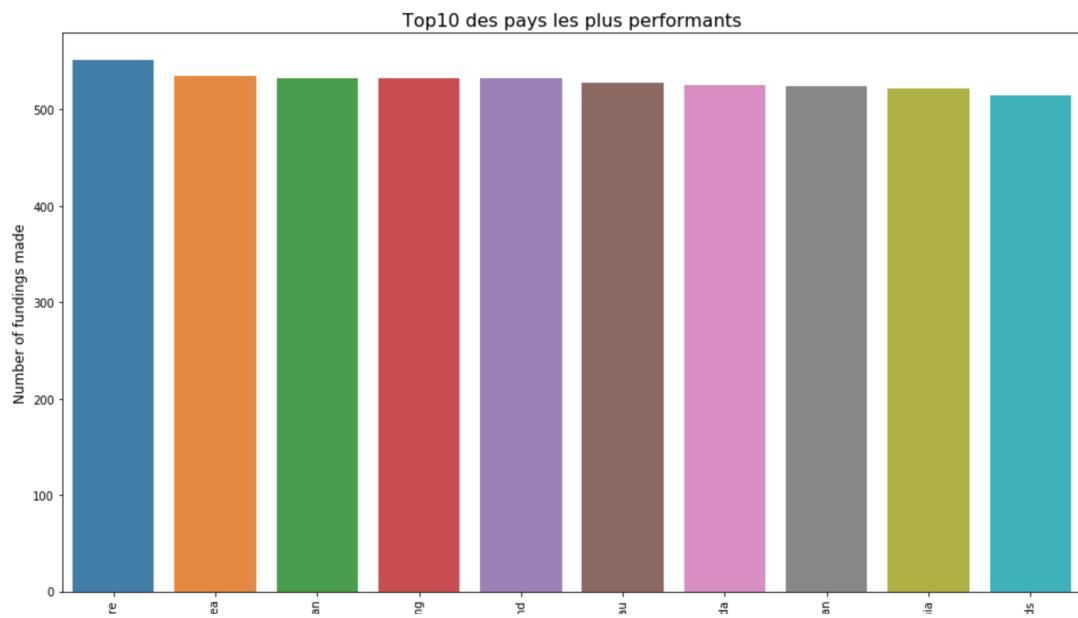
### Investissement par Pays en 2015 dans l'ordre du classement PISA



### Top10 : Les pays qui investissent le plus



## TOP 10 : les pays les plus performants au classement PISA

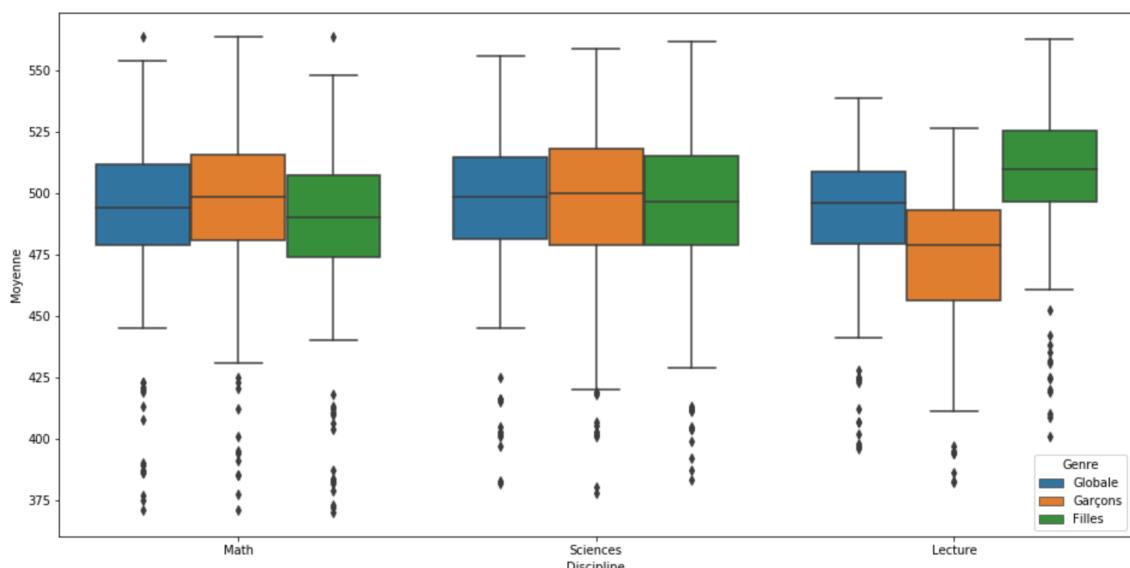


## CONCLUSION :

A l'issu des graphiques précédents, nous pouvons conclure que l'investissement des pays dans l'éducation ne conduit pas systématiquement à l'amélioration des résultats scolaires. L'investissement est essentiel mais n'est pas la cause essentielle des bonnes performances scolaires.

## ANALYSE par GENRE

Boxplot : Etude comparative des performances disciplinaires selon le genre :



Pour la plupart des sous-groupes, la médiane est dans le milieu de la boîte à moustache , ce qui suppose une distribution plus ou moins symétrique pour la plupart des performances des différentes disciplines. Néanmoins il existe des outliers(données aberrantes ). La discipline lecture montre une performance plus importante le groupe filles que pour le groupe garçon.

---

04

## CRÉATION D'UN SITE - DASHBOARD.



SITE EN  
MAINTENANCE..

# Remerciements.

Mes remerciements vont à la Fabrique Simplon Nanterre, qui m'a offert un cadre pour apprendre à mon rythme dans de bonnes conditions, sans préjugés et m'a permis de faire de belles rencontres.

Je remercie également son directeur pédagogique Kalidou Niang, très impliqué dans notre projet de reconversion professionnelle, ouvert et à l'écoute.

A mes encadrants qui ont su se montrer patients et instructifs Manel Boumaiza, Sayf Bejaoui, David Azria et Yacine Aslimi.

Enfin à tous mes camarades de classe qui ont donné à ses sept mois de formation une dimension joviale et bienveillante. Je remercie particulièrement mes camarades et amis Fatmaa Aarab, Youssef Kadri , Nadia Bibi et Ioan Hodor pour l'aide qu'il m'ont apporté dans la réalisation de cette étude.

