Analyse Exploratoire des Données (EDA) - Banque Mondiale

1. Imports

Nous configurons l'environnement et listons les sources de données brutes. Pour visualiser les données, nous définissons d'abord le chemin d'accès (**path**) où sont stockés les fichiers CSV. Nous récupérons ensuite la liste de ces fichiers dans une variable `all_files` en utilisant un pattern de recherche (**globbing**) avec l'extension `*.csv`.

2. Chargement des données

Boucle d'itération et rendu des données

Cette étape permet de valider l'intégrité des fichiers CSV et d'obtenir un premier aperçu visuel des structures.

- **Identification** : Nous affichons le nom du fichier pour confirmer la lecture.
- **Chargement** : Le contenu est chargé dans un **DataFrame**.
- **Rendu** : Nous utilisons `display(df.head())` pour générer un rendu visuel des 5 premières entrées.

Gestion des exceptions (Error Handling)

En cas d'erreur lors de la lecture ou de l'affichage, un bloc `try...except` permet de capturer l'exception. Le script affiche alors le nom du fichier problématique ainsi que le message d'erreur associé pour faciliter le débogage.

> **Note technique** :
>
> - **os** : Ce module fournit des fonctions pour interagir avec le système d'exploitation. Nous l'utilisons ici pour manipuler les chemins de fichiers (paths) de manière agnostique
> - **glob** : Cette bibliothèque est utilisée pour la recherche de fichiers via des patterns. Le globbing permet de lister tous les fichiers correspondant à une extension spécifique (ex: *.csv).
> - **pd.read_csv()** : C'est la fonction fondamentale de Pandas pour transformer un fichier CSV en un objet DataFrame (tableau structuré en lignes et colonnes).

> - **display()** : Une fonction spécifique à l'environnement IPython (Jupyter/DataSpell) qui permet un rendu HTML élégant des objets, plus lisible que le simple print().
> - **head(n)** : Cette méthode retourne les n premières lignes d'un DataFrame (par défaut 5).
> - **Gestion des exceptions : try... expect** : Ce bloc permet de tester un bloc de code (try). Si une erreur survient (fichier corrompu, chemin erroné), le programme ne plante pas brutalement mais exécute le bloc except, nous permettant d'afficher un message d'erreur personnalisé.

```
import numpy as np
import pandas as pd
import glob
import os
from IPython.display import Markdown
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

path = 'data'
all_files = glob.glob(os.path.join(path , "*.csv"))
dataframes = []

for file in all_files:
    try:
        file_name = os.path.basename(file)
        df = pd.read_csv(file)
        dataframes.append({"name": file_name, "data": df})
    except Exception as e:
        print(f"Erreur sur {file}: {e}")
```

```
display(Markdown(f"## Analyse du fichier : {dataframes[0]['name']}"))
```

Analyse du fichier : EdStatsCountry.csv

```
print(f"--- Fichier : {dataframes[0]['name']} ---")
display(dataframes[0]['data'].head())
```

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Income Group
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	High income: nonOECD
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	Low income
2	AGO	Angola	Angola	People's Republic of Angola	AO	Angolan kwanza	April 2013 database update: Based on IMF data,...	Sub-Saharan Africa	Upper middle income
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	Upper middle income
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	High income: nonOECD

5 rows × 32 columns

```
display(Markdown(f"## Analyse du fichier : {dataframes[1]['name']}"))
```

Analyse du fichier : EdStatsCountry-Series.csv

```
print(f"--- Fichier : {dataframes[1]['name']} ---")
display(dataframes[1]['data'].head())
```

	CountryCode	SeriesCode	DESCRIPTION	Unnamed: 3
0	ABW	SP.POP.TOTL	Data sources : United Nations World Population...	NaN
1	ABW	SP.POP.GROW	Data sources: United Nations World Population ...	NaN
2	AFG	SP.POP.GROW	Data sources: United Nations World Population ...	NaN
3	AFG	NY.GDP.PCAP.PP.CD	Estimates are based on regression.	NaN
4	AFG	SP.POP.TOTL	Data sources : United Nations World Population...	NaN

```
display(Markdown(f"## Analyse du fichier : {dataframes[2]['name']}"))
```

Analyse du fichier : EdStatsData.csv

```
print(f"--- Fichier : {dataframes[2]['name']} ---")
display(dataframes[2]['data'].head())
```

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1971	1972	1973	
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	NaN	
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	NaN	
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	NaN	
4	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	54.822121	54.894138	56.209438	57.267109	57.

5 rows × 70 columns

```
display(Markdown(f"## Analyse du fichier : {dataframes[3]['name']}"))
```

Analyse du fichier : EdStatsFootNote.csv

```
print(f"--- Fichier : {dataframes[3]['name']} ---")
display(dataframes[3]['data'].head())
```

	CountryCode	SeriesCode	Year	DESCRIPTION	Unnamed: 4
0	ABW	SE.PRE.ENRL.FE	YR2001	Country estimation.	NaN
1	ABW	SE.TER.TCHR.FE	YR2005	Country estimation.	NaN
2	ABW	SE.PRE.TCHR.FE	YR2000	Country estimation.	NaN
3	ABW	SE.SEC.ENRL.GC	YR2004	Country estimation.	NaN
4	ABW	SE.PRE.TCHR	YR2006	Country estimation.	NaN

```
display(Markdown(f"## Analyse du fichier : {dataframes[4]['name']}"))
```

Analyse du fichier : EdStatsSeries.csv

```
print(f"--- Fichier : {dataframes[4]['name']} ---")
display(dataframes[4]['data'].head())
```

	Series Code	Topic	Indicator Name	Short definition	Long definition	Unit of measure	Periodicity	E Pe
0	BAR.NOED.1519.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15-19 with...	Percentage of female population age 15-19 with...	NaN	NaN	
1	BAR.NOED.1519.ZS	Attainment	Barro-Lee: Percentage of population age 15-19 ...	Percentage of population age 15-19 with no edu...	Percentage of population age 15-19 with no edu...	NaN	NaN	
2	BAR.NOED.15UP.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15+ with n...	Percentage of female population age 15+ with n...	NaN	NaN	
3	BAR.NOED.15UP.ZS	Attainment	Barro-Lee: Percentage of population age 15+ wi...	Percentage of population age 15+ with no educa...	Percentage of population age 15+ with no educa...	NaN	NaN	
4	BAR.NOED.2024.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 20-24 with...	Percentage of female population age 20-24 with...	NaN	NaN	

5 rows × 21 columns

3. Nettoyage et Analyse Exploratoire des Données (EDA)

Cette section automatise le processus de nettoyage et d'analyse descriptive pour l'ensemble des fichiers chargés. L'objectif est de garantir l'intégrité des données et d'optimiser la structure des DataFrames avant l'analyse approfondie.

Méthodologie appliquée par fichier :

1. **Définition de l'unité d'observation (*Row definition*)** :
    - Identification de la granularité technique d'une ligne (ex: une observation unique pays/année/indicateur).
2. **Évaluation de la volumétrie (*Shape*)** :

- Calcul du nombre de lignes (*records*) et de colonnes (*features*) pour quantifier le dataset.

3. **Traitement des Redondances (*Deduplication*)** :
   - Détection et suppression des doublons pour éviter de biaiser les futurs calculs statistiques (moyennes, sommes).
4. **Analyse de la complétude (*Missing Values*)** :
   - Calcul de la proportion de valeurs manquantes (`NaN`) par colonne pour évaluer la fiabilité de chaque variable.
5. **Optimisation du Dataset (*Pruning*)** :
   - Suppression des colonnes jugées inutilisables (ex: colonnes techniques vides ou colonnes ayant plus de 90% de valeurs manquantes).
     - Justification technique :
       - **Significativité statistique** : Une variable renseignée à moins de 10% ne permet pas d'extraire des tendances représentatives et introduit un "bruit" analytique (*statistical noise*) qui fausse les mesures de tendance centrale comme les moyennes et les écart-types.
       - **Pertinence temporelle** : Dans le fichier `EdStatsData.csv`, ce seuil permet d'éliminer les projections à très long terme (ex: 2070-2100) qui sont quasi-intégralement vides, tout en conservant les données historiques réelles indispensables à l'analyse.
       - **Performance (*Memory Management*)** : La suppression de ces colonnes réduit l'empreinte mémoire du DataFrame, ce qui accélère les calculs ultérieurs (calculs vectorisés), une pratique essentielle sur des jeux de données dépassant les 800 000 lignes.
6. **Analyse Descriptive (*Numerical & Categorical Features*)** :
   - **Colonnes Numériques** : Application de `.describe()` pour obtenir les mesures de tendance centrale et de dispersion (Min, Max, Moyenne, Quartiles).
   - **Colonnes Catégorielles** : Calcul des occurrences via `.value_counts()` pour identifier les modalités dominantes et les déséquilibres potentiels.

> **Note technique :** Afin de garantir un rendu visuel stable dans l'IDE et un export PDF professionnel, les résultats catégoriels sont convertis en structures tabulaires (*DataFrames*) avant d'être affichés via la fonction `display()`.

3.1 Définition de l'unité d'observation

EdStatsCountry.csv

- **Définition** : Une ligne représente un pays unique ou une entité géographique (ex: une région comme l'Amérique Latine).

- **Clé primaire** : `CountryCode`
- **Contenu** : Toutes les caractéristiques fixes du pays (monnaie, région, système de recensement, etc...)

EdStatsSeries.csv

- **Définition** : Une ligne représente un indicateur statistique unique (un "Series").
- **Clé primaire** : `SeriesCode`
- **Contenu** : Les définitions, les sources et les méthodologies pour chaque type de donnée mesurée (ex: taux d'inscription scolaire).

EdStatsCountry-Series.csv

- **Définition** : Une ligne représente une relation spécifique entre un pays et un indicateur.
- **Clé composite** : `CountryCode` + `SeriesCode`
- **Contenu** : Il sert de table de liaison. Il précise souvent la source de données spécifique utilisée pour cet indicateur dans ce pays précis (colonne `DESCRIPTION`).

EdStatsFootNote.csv

- **Définition** : Une ligne représente une note de bas de page liée à une mesure spécifique.
- **Clé composite** : `CountryCode` + `SeriesCode` + `Year`
- **Contenu** : Une explication textuelle (`DESCRIPTION`) pour justifier une anomalie ou une estimation pour une année donnée.

EdStatsData.csv

- **Définition** : Une ligne représente l'évolution historique d'un indicateur pour un pays.
- **Clé composite** : `CountryCode` + `IndicatorCode`
- **Contenu** : Contrairement aux autres, ce fichier est "large" dans un format `pivoté` : il contient les valeurs numériques pour chaque année de 1970 à 2100 sur la même ligne.

```
display(Markdown(f"## Analyse du fichier : {dataframes[0]['name']}"))
```

Analyse du fichier : EdStatsCountry.csv

> **Note technique .shape :**
>
> - **Fonctionnement** : Contrairement à une méthode, .shape est un attribut (propriété) du DataFrame. Il ne prend pas de parenthèses (). Il renvoie un tuple contenant deux éléments : (nombre_de_lignes, nombre_de_colonnes).
> - **utilité** : C'est le premier indicateur de la complexité du fichier. Il permet également de valider l'impact de nos futurs filtrages en comparant les dimensions "avant" et

> "après".

```
rows, columns = dataframes[0]['data'].shape
print(f"Rows: {rows}, Columns: {columns}")
display(Markdown(f"### 3.2. Le fichier : {dataframes[0]['name']} comprend {rows} ligne
```

3.2. Le fichier : EdStatsCountry.csv comprend 241 lignes et 32 colonnes

> **Traitement des redondances**
>
> - **.duplicated()** : Cette méthode parcourt le DataFrame et renvoie un masque booléen (une série de `True` ou `False`). Elle marque True pour chaque ligne dont toutes les valeurs sont strictement identiques à une ligne précédente.
> - **.sum()** : En Python, les valeurs booléennes sont traitées numériquement (`True = 1` et `False = 0`). En appliquant `.sum()` sur le résultat de duplicated(), nous obtenons instantanément le nombre total de lignes dupliquées.

```
duplicate_count = dataframes[0]['data'].duplicated().sum()
display(Markdown(f"### 3.3. Le fichier : {dataframes[0]['name']} posséde {duplicate_c
```

3.3. Le fichier : EdStatsCountry.csv posséde 0 lignes dupliquées

> - **.drop_duplicates()** : Si le compte est supérieur à zéro, cette méthode permet de supprimer les copies pour ne conserver qu'une seule occurrence unique de chaque ligne.

```
if duplicate_count > 0:
    dataframes[0]['data'] = dataframes[0]['data'].drop_duplicates()
```

3.4. Calcul du pourcentage de valeurs manquantes par colonnes et affichage dans un nouveau dataframe trié par pourcentage décroissant.

> **Analyse de la complétude** :
>
> - **.isnull() ou .isna()** : Cette méthode génère un masque booléen où chaque cellule vide (NaN) est marquée comme True.
> - **len()** : En divisant la somme des valeurs nulles par la longueur totale du DataFrame (`len(df)`), nous obtenons une proportion que nous multiplions par 100 pour obtenir un pourcentage lisible.
> - **pd.DataFrame()** : Nous reconstruisons un nouveau tableau pour synthétiser ces résultats, ce qui facilite la lecture par rapport à une simple liste brute.
> - **.sort_values()** : Cette méthode permet de trier le résultat. En utilisant ascending=False, nous plaçons les colonnes les plus vides en haut de tableau

> pour identifier immédiatement les variables inutilisables.

```
percent_missing = dataframes[0]['data'].isnull().sum() * 100 / len(dataframes[0]['data
missing_value = pd.DataFrame({'column_name' : dataframes[0]['data'].columns, 'percent_
display(missing_value)
```

	column_name	percent_missing
Unnamed: 31	Unnamed: 31	100.000000
National accounts reference year	National accounts reference year	86.721992
Alternative conversion factor	Alternative conversion factor	80.497925
Other groups	Other groups	75.933610
Latest industrial data	Latest industrial data	55.601660
Vital registration complete	Vital registration complete	53.941909
External debt Reporting status	External debt Reporting status	48.547718
Latest household survey	Latest household survey	41.493776
Latest agricultural census	Latest agricultural census	41.078838
Lending category	Lending category	40.248963
PPP survey year	PPP survey year	39.834025
Special Notes	Special Notes	39.834025
Source of most recent Income and expenditure data	Source of most recent Income and expenditure data	33.609959
Government Accounting concept	Government Accounting concept	33.195021
Latest water withdrawal data	Latest water withdrawal data	25.726141
IMF data dissemination standard	IMF data dissemination standard	24.896266
Balance of Payments Manual in use	Balance of Payments Manual in use	24.896266
Latest trade data	Latest trade data	23.236515
SNA price valuation	SNA price valuation	18.257261
System of trade	System of trade	17.012448
National accounts base year	National accounts base year	14.937759
Latest population census	Latest population census	11.618257
Region	Region	11.203320
Income Group	Income Group	11.203320
Currency Unit	Currency Unit	10.788382
System of National Accounts	System of National Accounts	10.788382
2-alpha code	2-alpha code	1.244813
WB-2 code	WB-2 code	0.414938
Long Name	Long Name	0.000000
Short Name	Short Name	0.000000
Table Name	Table Name	0.000000

	column_name	percent_missing
Country Code	Country Code	0.000000

3.5. Néttoyage des collones atteignant plus de 90% de valeurs manquantes

> **Optimisation du Data Set**
>
> - **.dropna() avec un seuil** : axis=1 : Ce paramètre indique à Pandas d'agir sur les colonnes (l'axe vertical) plutôt que sur les lignes.
> - **thresh=limit** : C'est le paramètre de seuil (threshold). Il définit le nombre minimum de valeurs non-nulles requises pour que la colonne soit conservée.
> - **Logique de calcul** : Si on définis limit à 10% de la longueur totale (len(df) * 0.1), toute colonne ayant plus de 90% de données manquantes sera supprimée.
> - **Utilité** : Cette méthode est plus robuste que de supprimer les colonnes par leur nom, car elle s'adapte dynamiquement au contenu réel de chaque fichier sans hardcoding.

```
limit = len(dataframes[0]['data']) * 0.1
dataframes[0]['data'] = dataframes[0]['data'].dropna(axis=1, thresh=limit)
display(dataframes[0]['data'].head())
```

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Income Group
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	High income: nonOECD
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	Low income
2	AGO	Angola	Angola	People's Republic of Angola	AO	Angolan kwanza	April 2013 database update: Based on IMF data,...	Sub-Saharan Africa	Upper middle income
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	Upper middle income
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	High income: nonOECD

5 rows × 31 columns

```
display(Markdown(f"### 3.6.1. Obtention des mesures de tendances centrales et de dispe
```

3.6.1. Obtention des mesures de tendances centrales et de dispersions : EdStatsCountry.csv

> **Analyse descriptive (Numerical feature)** :
>
> - **select_dtypes(include=['number'])** : Cette méthode filtre le DataFrame pour ne conserver que les colonnes de type numérique (entiers ou flottants). C'est une

> sécurité indispensable avant de lancer des calculs mathématiques qui échoueraient sur du texte.
> - **.empty** : Un attribut qui renvoie True si le DataFrame résultant ne contient aucune donnée. Nous l'utilisons pour conditionner l'affichage et éviter des erreurs de rendu.
> - **.describe()** : La fonction maîtresse de l'analyse descriptive. Elle génère automatiquement un tableau de synthèse incluant :
> - count : Le nombre de valeurs renseignées (non-nulles).
> - mean : La moyenne arithmétique.
> - std : L'écart-type, mesurant la dispersion des données autour de la moyenne.
> - min / max : Les valeurs extrêmes du jeu de données.
> - 25%, 50% (médiane), 75% : Les quartiles, permettant de comprendre la répartition des données.

```
numeric_df = dataframes[0]['data'].select_dtypes(include=['number'])

if not numeric_df.empty:
    display(dataframes[0]['data'].describe(include=[np.number]))
else :
    display(Markdown(f" Le fichier : {dataframes[0]['name']} ne posséde pas de valeur
```

	National accounts reference year	Latest industrial data	Latest trade data
count	32.00000	107.000000	185.000000
mean	2001.53125	2008.102804	2010.994595
std	5.24856	2.616834	2.569675
min	1987.00000	2000.000000	1995.000000
25%	1996.75000	2007.500000	2011.000000
50%	2002.00000	2009.000000	2012.000000
75%	2005.00000	2010.000000	2012.000000
max	2012.00000	2010.000000	2012.000000

```
df_source = dataframes[0]['data']
cat_cols = df_source.select_dtypes(include=['object', 'string']).columns

report_chunks = []

for col in cat_cols:
    counts = df_source[col].value_counts().head(5).to_frame()

    counts = counts.reset_index()
    counts.columns = ['Valeur', 'Nombre']
    counts.insert(0, 'Variable', col)
    report_chunks.append(counts)

final_report = pd.concat(report_chunks, ignore_index=True)
```

```
display(Markdown(f"### 3.6.2. Rapport global des occurrences : {dataframes[0]['name']}
```

3.6.2. Rapport global des occurrences : EdStatsCountry.csv

> **Analyse des variables catégorielles**
>
> - **select_dtypes(include=['object', 'string'])** : Cette méthode cible exclusivement les colonnes textuelles (catégorielles).
> - **value_count().head(5)** : Pour chaque colonne, nous calculons la fréquence d'apparition de chaque texte et ne conservons que le "Top 5".
> - **reset_index et insert()** : Ces manipulations permettent de transformer un résultat de calcul (Series) en un tableau propre, en réintégrant le nom de la variable d'origine comme une colonne à part entière.
> - **pd.concat()** : Cette fonction est l'équivalent d'un "collage" de tableaux. Elle rassemble tous les petits rapports individuels (report_chunks) en un seul grand DataFrame final pour une lecture centralisée.
> - **pd.option_context** : C'est un "gestionnaire de contexte" (utilisé avec le mot-clé with). Il permet de modifier temporairement les réglages de Pandas uniquement pour le bloc de code qui suit.
> - **'display.max_rows'** : En réglant cette option sur None, on indique à Pandas qu'il n'y a plus de limite au nombre de lignes à afficher.
> - **Avantage** : Une fois sorti du bloc with, Pandas reprend ses réglages par défaut, ce qui évite d'encombrer le Notebook pour les autres affichages plus petits.

```
with pd.option_context('display.max_rows', None):
    display(final_report)
```

	Variable	Valeur	Nombre
0	Country Code	ABW	1
1	Country Code	AFG	1
2	Country Code	AGO	1
3	Country Code	ALB	1
4	Country Code	AND	1
5	Short Name	Aruba	1
6	Short Name	Afghanistan	1
7	Short Name	Angola	1
8	Short Name	Albania	1
9	Short Name	Andorra	1
10	Table Name	Aruba	1
11	Table Name	Afghanistan	1
12	Table Name	Angola	1
13	Table Name	Albania	1
14	Table Name	Andorra	1
15	Long Name	Aruba	1
16	Long Name	Islamic State of Afghanistan	1
17	Long Name	People's Republic of Angola	1
18	Long Name	Republic of Albania	1
19	Long Name	Principality of Andorra	1
20	2-alpha code	AW	1
21	2-alpha code	AF	1
22	2-alpha code	AO	1
23	2-alpha code	AL	1
24	2-alpha code	AD	1
25	Currency Unit	Euro	23
26	Currency Unit	U.S. dollar	14
27	Currency Unit	CFA franc	14
28	Currency Unit	East Caribbean dollar	6
29	Currency Unit	Australian dollar	3
30	Special Notes	April 2012 database update: Based on official ...	6

	Variable	Valeur	Nombre
31	Special Notes	Fiscal year end: March 31; reporting period fo...	4
32	Special Notes	Fiscal year end: June 30; reporting period for...	3
33	Special Notes	Fiscal year end: June 30; reporting period for...	2
34	Special Notes	April 2013 database update: Based on IMF data,...	2
35	Region	Europe & Central Asia	57
36	Region	Sub-Saharan Africa	48
37	Region	Latin America & Caribbean	41
38	Region	East Asia & Pacific	36
39	Region	Middle East & North Africa	21
40	Income Group	Upper middle income	55
41	Income Group	Lower middle income	50
42	Income Group	High income: nonOECD	44
43	Income Group	Low income	34
44	Income Group	High income: OECD	31
45	WB-2 code	AW	1
46	WB-2 code	AF	1
47	WB-2 code	AO	1
48	WB-2 code	AL	1
49	WB-2 code	AD	1
50	National accounts base year	2005	34
51	National accounts base year	Original chained constant price data are resca...	28
52	National accounts base year	2000	25
53	National accounts base year	2006	19
54	National accounts base year	1990	11
55	SNA price valuation	Value added at basic prices (VAB)	163
56	SNA price valuation	Value added at producer prices (VAP)	34
57	Lending category	IBRD	67
58	Lending category	IDA	59
59	Lending category	Blend	18

	Variable	Valeur	Nombre
60	Other groups	HIPC	40
61	Other groups	Euro area	18
62	System of National Accounts	Country uses the 1993 System of National Accou...	165
63	System of National Accounts	Country uses the 1968 System of National Accou...	42
64	System of National Accounts	Country uses the 2008 System of National Accou...	8
65	Alternative conversion factor	1990–95	8
66	Alternative conversion factor	1987–95	5
67	Alternative conversion factor	1993	3
68	Alternative conversion factor	1992–95	2
69	Alternative conversion factor	1991	2
70	PPP survey year	2005	98
71	PPP survey year	Rolling	37
72	PPP survey year	2011	10
73	Balance of Payments Manual in use	IMF Balance of Payments Manual, 6th edition.	181
74	External debt Reporting status	Actual	107
75	External debt Reporting status	Estimate	11
76	External debt Reporting status	Preliminary	6
77	System of trade	General trade system	106
78	System of trade	Special trade system	94
79	Government Accounting concept	Consolidated central government	95
80	Government Accounting concept	Budgetary central government	66
81	IMF data dissemination standard	General Data Dissemination System (GDDS)	110
82	IMF data dissemination standard	Special Data Dissemination Standard (SDDS)	71
83	Latest population census	2011	59
84	Latest population census	2010	49
85	Latest population census	2012	18
86	Latest population census	2009	14
87	Latest population census	2008	9

	Variable	Valeur	Nombre
88	Latest household survey	Multiple Indicator Cluster Survey (MICS), 2012	10
89	Latest household survey	World Health Survey (WHS), 2003	10
90	Latest household survey	Demographic and Health Survey (DHS), 2013	10
91	Latest household survey	Multiple Indicator Cluster Survey (MICS), 2011	9
92	Latest household survey	Multiple Indicator Cluster Survey (MICS), 2010	9
93	Source of most recent Income and expenditure data	Integrated household survey (IHS), 2012	15
94	Source of most recent Income and expenditure data	Integrated household survey (IHS), 2010	9
95	Source of most recent Income and expenditure data	Integrated household survey (IHS), 2011	9
96	Source of most recent Income and expenditure data	Integrated household survey (IHS), 2008	7
97	Source of most recent Income and expenditure data	Expenditure survey/budget survey (ES/BS), 2012	7
98	Vital registration complete	Yes	110
99	Vital registration complete	Yes. Vital registration for Guernsey and Jersey.	1
100	Latest agricultural census	2010	36
101	Latest agricultural census	2007	16
102	Latest agricultural census	2013	13
103	Latest agricultural census	2012	10
104	Latest agricultural census	2011	9
105	Latest water withdrawal data	2000	40
106	Latest water withdrawal data	2005	40
107	Latest water withdrawal data	2007	18
108	Latest water withdrawal data	2002	16
109	Latest water withdrawal data	2009	12

```
display(Markdown(f"## Analyse du fichier : {dataframes[1]['name']}"))
```

Analyse du fichier : EdStatsCountry-Series.csv

```
rows, columns = dataframes[1]['data'].shape
print(f"Rows: {rows}, Columns: {columns}")
display(Markdown(f"### 3.2. Le fichier : {dataframes[1]['name']} comprend {rows} ligne
```

3.2. Le fichier : EdStatsCountry-Series.csv comprend 613 lignes et 4 colonnes

```
duplicate_count = dataframes[1]['data'].duplicated().sum()
display(Markdown(f"### 3.3. Le fichier : {dataframes[1]['name']} posséde {duplicate_c
```

3.3. Le fichier : EdStatsCountry-Series.csv posséde 0 lignes dupliquées

```
if duplicate_count > 0:
    dataframes[1]['data'] = dataframes[1]['data'].drop_duplicates()
```

3.4. Calcul du pourcentage de valeurs manquantes par colonnes et affichage dans un nouveau dataframe trié par pourcentage décroissant.

```
percent_missing = dataframes[1]['data'].isnull().sum() * 100 / len(dataframes[1]['data
missing_value = pd.DataFrame({'column_name' : dataframes[1]['data'].columns, 'percent_
display(missing_value)
```

	column_name	percent_missing
Unnamed: 3	Unnamed: 3	100.0
CountryCode	CountryCode	0.0
SeriesCode	SeriesCode	0.0
DESCRIPTION	DESCRIPTION	0.0

3.5. Néttoyage des collones atteignant plus de 90% de valeurs manquantes

```
limit = len(dataframes[1]['data']) * 0.1
dataframes[1]['data'] = dataframes[1]['data'].dropna(axis=1, thresh=limit)
display(dataframes[1]['data'].head())
```

	CountryCode	SeriesCode	DESCRIPTION
0	ABW	SP.POP.TOTL	Data sources : United Nations World Population...
1	ABW	SP.POP.GROW	Data sources: United Nations World Population ...
2	AFG	SP.POP.GROW	Data sources: United Nations World Population ...
3	AFG	NY.GDP.PCAP.PP.CD	Estimates are based on regression.
4	AFG	SP.POP.TOTL	Data sources : United Nations World Population...

```
display(Markdown(f"### 3.6.1. Obtention des mesures de tendances centrales et de dispe
```

3.6.1. Obtention des mesures de tendances centrales et de dispersions : EdStatsCountry-Series.csv

```
numeric_df = dataframes[1]['data'].select_dtypes(include=['number'])

if not numeric_df.empty:
    display(dataframes[1]['data'].describe(include=[np.number]))
else :
    display(Markdown(f" Le fichier : {dataframes[1]['name']} ne posséde pas de valeur
```

Le fichier : EdStatsCountry-Series.csv ne posséde pas de valeur numérique

```
df_source = dataframes[1]['data']
cat_cols = df_source.select_dtypes(include=['object', 'string']).columns

report_chunks = []

for col in cat_cols:
    counts = df_source[col].value_counts().head(5).to_frame()

    counts = counts.reset_index()
    counts.columns = ['Valeur', 'Nombre']
    counts.insert(0, 'Variable', col)
    report_chunks.append(counts)

final_report = pd.concat(report_chunks, ignore_index=True)

display(Markdown(f"### 3.6.2. Rapport global des occurrences : {dataframes[1]['name']]
```

3.6.2. Rapport global des occurrences : EdStatsCountry-Series.csv

```
with pd.option_context('display.max_rows', None):
    display(final_report)
```

	Variable	Valeur	Nombre
0	CountryCode	GEO	18
1	CountryCode	MDA	18
2	CountryCode	CYP	12
3	CountryCode	MAR	12
4	CountryCode	MUS	12
5	SeriesCode	SP.POP.TOTL	211
6	SeriesCode	SP.POP.GROW	211
7	SeriesCode	NY.GDP.PCAP.PP.CD	19
8	SeriesCode	NY.GDP.PCAP.PP.KD	19
9	SeriesCode	NY.GNP.PCAP.PP.CD	19
10	DESCRIPTION	Data sources : United Nations World Population...	154
11	DESCRIPTION	Data sources: United Nations World Population ...	137
12	DESCRIPTION	Estimates are based on regression.	84
13	DESCRIPTION	Data sources : Eurostat	54
14	DESCRIPTION	Derived using ratio of age group from WPP and ...	24

```
display(Markdown(f"## Analyse du fichier : {dataframes[2]['name']}"))
```

Analyse du fichier : EdStatsData.csv

```
rows, columns = dataframes[2]['data'].shape
print(f"Rows: {rows}, Columns: {columns}")
display(Markdown(f"### 3.2. Le fichier : {dataframes[2]['name']} comprend {rows} ligne
```

3.2. Le fichier : EdStatsData.csv comprend 886930 lignes et 70 colonnes

```
duplicate_count = dataframes[2]['data'].duplicated().sum()
display(Markdown(f"### 3.3. Le fichier : {dataframes[2]['name']} posséde {duplicate_c
```

3.3. Le fichier : EdStatsData.csv posséde 0 lignes dupliquées

```
if duplicate_count > 0:
    dataframes[2]['data'] = dataframes[2]['data'].drop_duplicates()
```

3.4. Calcul du pourcentage de valuers manquantes par colonnes et affichage dans un nouveau dataframe trié par pourcentage décroissant

```
percent_missing = dataframes[2]['data'].isnull().sum() * 100 / len(dataframes[2]['data
missing_value = pd.DataFrame({'column_name' : dataframes[2]['data'].columns, 'percent_
display(missing_value)
```

	column_name	percent_missing
Unnamed: 69	Unnamed: 69	100.000000
2017	2017	99.983877
2016	2016	98.144160
1971	1971	95.993258
1973	1973	95.992356
...	...	...
2010	2010	72.665036
Country Code	Country Code	0.000000
Indicator Code	Indicator Code	0.000000
Indicator Name	Indicator Name	0.000000
Country Name	Country Name	0.000000

70 rows × 2 columns

3.5. Nettoyage des collones atteignant plus de 90% de valuers manquantes

```
limit = len(dataframes[3]['data']) * 0.1
dataframes[2]['data'] = dataframes[2]['data'].dropna(axis=1, thresh=limit)
display(dataframes[2]['data'].head())
```

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1975	1980	1985	
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	NaN	
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	NaN	
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	NaN	
4	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	54.822121	59.36554	65.617767	69.033211	71.9

5 rows × 34 columns

Observation des données brutes:

L'analyse du taux de valeurs manquantes (Missing Values) montre des seuils critiques pour les années post-2015:

- **2016** : 98,14 % de données manquantes.
- **2017** : 99,98 % de données manquantes.
- **2018 - 2027** : Taux de vacuité proche de 100 % pour les indicateurs thématiques sélectionnés (Education, IT, Learning Outcomes).

```
display(Markdown(f"### 3.6.1. Obtention des mesures de tendances centrales et de dispe
```

3.6.1. Obtention des mesures de tendances centrales et de dispersions : EdStatsData.csv

```
numeric_df = dataframes[2]['data'].select_dtypes(include=['number'])

if not numeric_df.empty:
    display(dataframes[2]['data'].describe(include=[np.number]))
else :
    display(Markdown(f" Le fichier : {dataframes[2]['name']} ne posséde pas de valeur
```

	1970	1975	1980	1985	1990	19
count	7.228800e+04	8.730600e+04	8.912200e+04	9.029600e+04	1.244050e+05	7.443700e+
mean	1.974772e+09	2.314288e+09	3.283898e+09	3.622763e+09	9.084424e+09	1.571674e-
std	1.211687e+11	1.375059e+11	1.780774e+11	2.002929e+11	3.665667e+11	4.881357e-
min	-1.435564e+00	-3.658569e+00	-1.404240e+00	-2.216315e+00	-1.803750e+00	-5.814339e+
25%	8.900000e-01	1.400000e+00	1.770000e+00	2.150000e+00	4.830000e+00	5.134554e-
50%	6.317724e+00	9.677420e+00	1.107000e+01	1.200000e+01	5.048379e+01	3.916000e+
75%	6.251250e+01	7.854163e+01	8.202760e+01	8.338313e+01	9.134300e+04	4.383130e+
max	1.903929e+13	2.300634e+13	2.784319e+13	3.166465e+13	4.714344e+13	4.781272e-

8 rows × 30 columns

3.6. Note sur l'analyse descriptive de EdStatsData.csv

Bien que la méthode `.describe()` s'exécute sans erreur technique sur ce fichier, les résultats statistiques globaux (moyenne, écart-type) sont **analytiquement inutilisables** en l'état pour les raisons suivantes :

- **Hétérogénéité des indicateurs (*Mixed Scales*)** : Chaque colonne "Année" mélange des données de natures totalement différentes. Faire la moyenne entre un PIB (en milliers de milliards), une population (en milliards) et un taux d'alphabétisation (en pourcentage) génère un chiffre dépourvu de sens métier.
- **Biais de dispersion (*Variance Bias*)** : L'écart-type (*standard deviation*) extrêmement élevé observé dans les résultats (ex: 1.2×10^{11} pour 1970) confirme que les données ne suivent pas une distribution commune. Ce "bruit" statistique masque la réalité de chaque indicateur individuel.
- **Interprétation** : Pour obtenir des statistiques descriptives cohérentes, il est impératif d'effectuer un **filtrage préalable** sur la colonne `Indicator Code` afin d'isoler une seule métrique avant d'appliquer des calculs d'agrégation.

> **Conclusion de l'audit** : Ce tableau global n'est conservé ici qu'à titre de validation technique de la lecture des données numériques. Ce jeu de données servira exclusivement de base à une analyse par filtrage sélectif (méthodes `.loc` ou `.query`). Cette approche est la seule permettant de garantir la pertinence des calculs en isolant chaque métrique de son contexte d'origine.

```
df_source = dataframes[1]['data']
cat_cols = df_source.select_dtypes(include=['object', 'string']).columns

report_chunks = []

for col in cat_cols:
    counts = df_source[col].value_counts().head(5).to_frame()

    counts = counts.reset_index()
    counts.columns = ['Valeur', 'Nombre']
    counts.insert(0, 'Variable', col)
    report_chunks.append(counts)

final_report = pd.concat(report_chunks, ignore_index=True)

display(Markdown(f"### 3.6.2. Rapport global des occurrences : {dataframes[1]['name']}
```

3.6.2. Rapport global des occurrences : EdStatsCountry-Series.csv

```
with pd.option_context('display.max_rows', None):
    display(final_report)
```

	Variable	Valeur	Nombre
0	CountryCode	GEO	18
1	CountryCode	MDA	18
2	CountryCode	CYP	12
3	CountryCode	MAR	12
4	CountryCode	MUS	12
5	SeriesCode	SP.POP.TOTL	211
6	SeriesCode	SP.POP.GROW	211
7	SeriesCode	NY.GDP.PCAP.PP.CD	19
8	SeriesCode	NY.GDP.PCAP.PP.KD	19
9	SeriesCode	NY.GNP.PCAP.PP.CD	19
10	DESCRIPTION	Data sources : United Nations World Population...	154
11	DESCRIPTION	Data sources: United Nations World Population ...	137
12	DESCRIPTION	Estimates are based on regression.	84
13	DESCRIPTION	Data sources : Eurostat	54
14	DESCRIPTION	Derived using ratio of age group from WPP and ...	24

```
display(Markdown(f"## Analyse du fichier : {dataframes[2]['name']}"))
```

Analyse du fichier : EdStatsData.csv

```
rows, columns = dataframes[3]['data'].shape
print(f"Rows: {rows}, Columns: {columns}")
display(Markdown(f"### 3.2. Le fichier : {dataframes[3]['name']} comprend {rows} ligne
```

3.2. Le fichier : EdStatsFootNote.csv comprend 643638 lignes et 5 colonnes

```
duplicate_count = dataframes[3]['data'].duplicated().sum()
display(Markdown(f"### 3.3. Le fichier : {dataframes[3]['name']} posséde {duplicate_c
```

3.3. Le fichier : EdStatsFootNote.csv posséde 0 lignes dupliquées

```
if duplicate_count > 0:
    dataframes[3]['data'] = dataframes[3]['data'].drop_duplicates()
```

3.4. Calcul du pourcentage de valuers manquantes par colonnes et affichage dans un nouveau dataframe trié par pourcentage décroissant

```
percent_missing = dataframes[3]['data'].isnull().sum() * 100 / len(dataframes[3]['data
missing_value = pd.DataFrame({'column_name' : dataframes[3]['data'].columns, 'percent_
display(missing_value)
```

	column_name	percent_missing
Unnamed: 4	Unnamed: 4	100.0
CountryCode	CountryCode	0.0
SeriesCode	SeriesCode	0.0
Year	Year	0.0
DESCRIPTION	DESCRIPTION	0.0

3.5. Nettoyage des collones atteignant plus de 90% de valuers manquantes

```
limit = len(dataframes[3]['data']) * 0.1
dataframes[3]['data'] = dataframes[3]['data'].dropna(axis=1, thresh=limit)
display(dataframes[3]['data'].head())
```

	CountryCode	SeriesCode	Year	DESCRIPTION
0	ABW	SE.PRE.ENRL.FE	YR2001	Country estimation.
1	ABW	SE.TER.TCHR.FE	YR2005	Country estimation.
2	ABW	SE.PRE.TCHR.FE	YR2000	Country estimation.
3	ABW	SE.SEC.ENRL.GC	YR2004	Country estimation.
4	ABW	SE.PRE.TCHR	YR2006	Country estimation.

```
display(Markdown(f"### 3.6.1. Obtention des mesures de tendances centrales et de dispe
```

3.6.1. Obtention des mesures de tendances centrales et de dispersions : EdStatsFootNote.csv

```
numeric_df = dataframes[3]['data'].select_dtypes(include=['number'])

if not numeric_df.empty:
    display(dataframes[3]['data'].describe(include=[np.number]))
else :
    display(Markdown(f" Le fichier : {dataframes[3]['name']} ne posséde pas de valeur
```

Le fichier : EdStatsFootNote.csv ne posséde pas de valeur numérique

```
df_source = dataframes[3]['data']
cat_cols = df_source.select_dtypes(include=['object', 'string']).columns

report_chunks = []

for col in cat_cols:
    counts = df_source[col].value_counts().head(5).to_frame()

    counts = counts.reset_index()
    counts.columns = ['Valeur', 'Nombre']
    counts.insert(0, 'Variable', col)
    report_chunks.append(counts)

final_report = pd.concat(report_chunks, ignore_index=True)

display(Markdown(f"### 3.6.2. Rapport global des occurrences : {dataframes[3]['name']}
```

3.6.2. Rapport global des occurrences : EdStatsFootNote.csv

```
with pd.option_context('display.max_rows', None):
    display(final_report)
```

	Variable	Valeur	Nombre
0	CountryCode	LIC	7320
1	CountryCode	CYP	7183
2	CountryCode	LDC	6481
3	CountryCode	SSA	6389
4	CountryCode	SSF	6336
5	SeriesCode	SH.DYN.MORT	9226
6	SeriesCode	SE.PRM.AGES	8771
7	SeriesCode	SE.PRM.DURS	8771
8	SeriesCode	SE.SEC.DURS	8619
9	SeriesCode	SE.SEC.AGES	8581
10	Year	YR2004	27128
11	Year	YR2005	25992
12	Year	YR2002	25687
13	Year	YR2003	25683
14	Year	YR2000	25093
15	DESCRIPTION	Country Data	191188
16	DESCRIPTION	UNESCO Institute for Statistics (UIS) estimate	171527
17	DESCRIPTION	Estimated	117155
18	DESCRIPTION	UIS Estimation	31395
19	DESCRIPTION	Country estimation.	26308

```
display(Markdown(f"## Analyse du fichier : {dataframes[4]['name']}"))
```

Analyse du fichier : EdStatsSeries.csv

```
rows, columns = dataframes[4]['data'].shape
print(f"Rows: {rows}, Columns: {columns}")
display(Markdown(f"### 3.2. Le fichier : {dataframes[4]['name']} comprend {rows} ligne
```

3.2. Le fichier : EdStatsSeries.csv comprend 3665 lignes et 21 colonnes

```
duplicate_count = dataframes[4]['data'].duplicated().sum()
display(Markdown(f"### 3.3. Le fichier : {dataframes[4]['name']} posséde {duplicate_c
```

3.3. Le fichier : EdStatsSeries.csv posséde 0 lignes dupliquées

```
if duplicate_count > 0:
    dataframes[4]['data'] = dataframes[4]['data'].drop_duplicates()
```

3.4. Calcul du pourcentage de valuers manquantes par colonnes et affichage dans un nouveau dataframe trié par pourcentage décroissant

```
percent_missing = dataframes[4]['data'].isnull().sum() * 100 / len(dataframes[4]['data
missing_value = pd.DataFrame({'column_name' : dataframes[4]['data'].columns, 'percent_
display(missing_value)
```

	column_name	percent_missing
Unnamed: 20	Unnamed: 20	100.000000
Notes from original source	Notes from original source	100.000000
License Type	License Type	100.000000
Related indicators	Related indicators	100.000000
Other web links	Other web links	100.000000
Unit of measure	Unit of measure	100.000000
Development relevance	Development relevance	99.918145
General comments	General comments	99.618008
Limitations and exceptions	Limitations and exceptions	99.618008
Statistical concept and methodology	Statistical concept and methodology	99.372442
Aggregation method	Aggregation method	98.717599
Periodicity	Periodicity	97.298772
Related source links	Related source links	94.133697
Base Period	Base Period	91.432469
Other notes	Other notes	84.938608
Short definition	Short definition	41.173261
Topic	Topic	0.000000
Source	Source	0.000000
Long definition	Long definition	0.000000
Indicator Name	Indicator Name	0.000000
Series Code	Series Code	0.000000

3.5. Nettoyage des collones atteignant plus de 90% de valuers manquantes

```
limit = len(dataframes[4]['data']) * 0.1
dataframes[4]['data'] = dataframes[4]['data'].dropna(axis=1, thresh=limit)
display(dataframes[4]['data'].head())
```

	Series Code	Topic	Indicator Name	Short definition	Long definition	Other notes	Source
0	BAR.NOED.1519.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15-19 with...	Percentage of female population age 15-19 with...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
1	BAR.NOED.1519.ZS	Attainment	Barro-Lee: Percentage of population age 15-19 ...	Percentage of population age 15-19 with no edu...	Percentage of population age 15-19 with no edu...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
2	BAR.NOED.15UP.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15+ with n...	Percentage of female population age 15+ with n...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
3	BAR.NOED.15UP.ZS	Attainment	Barro-Lee: Percentage of population age 15+ wi...	Percentage of population age 15+ with no educa...	Percentage of population age 15+ with no educa...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
4	BAR.NOED.2024.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 20-24 with...	Percentage of female population age 20-24 with...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...

```
display(Markdown(f"### 3.6.1. Obtention des mesures de tendances centrales et de dispe
```

3.6.1. Obtention des mesures de tendances centrales et de dispersions : EdStatsSeries.csv

```
numeric_df = dataframes[4]['data'].select_dtypes(include=['number'])

if not numeric_df.empty:
    display(dataframes[4]['data'].describe(include=[np.number]))
else :
    display(Markdown(f" Le fichier : {dataframes[4]['name']} ne posséde pas de valeur
```

Le fichier : EdStatsSeries.csv ne posséde pas de valeur numérique

```
df_source = dataframes[4]['data']
cat_cols = df_source.select_dtypes(include=['object', 'string']).columns
```

```
report_chunks = []

for col in cat_cols:
    counts = df_source[col].value_counts().head(5).to_frame()

    counts = counts.reset_index()
    counts.columns = ['Valeur', 'Nombre']
    counts.insert(0, 'Variable', col)
    report_chunks.append(counts)

final_report = pd.concat(report_chunks, ignore_index=True)

display(Markdown(f"### 3.6.2. Rapport global des occurrences : {dataframes[4]['name']}
```

3.6.2. Rapport global des occurrences : EdStatsSeries.csv

```
with pd.option_context('display.max_rows', None):
    display(final_report)
```

	Variable	Valeur	Nombre
0	Series Code	BAR.NOED.1519.FE.ZS	1
1	Series Code	BAR.NOED.1519.ZS	1
2	Series Code	BAR.NOED.15UP.FE.ZS	1
3	Series Code	BAR.NOED.15UP.ZS	1
4	Series Code	BAR.NOED.2024.FE.ZS	1
5	Topic	Learning Outcomes	1046
6	Topic	Attainment	733
7	Topic	Education Equality	426
8	Topic	Secondary	256
9	Topic	Primary	248
10	Indicator Name	Barro-Lee: Percentage of female population age...	1
11	Indicator Name	Barro-Lee: Percentage of population age 15-19 ...	1
12	Indicator Name	Barro-Lee: Percentage of female population age...	1
13	Indicator Name	Barro-Lee: Percentage of population age 15+ wi...	1
14	Indicator Name	Barro-Lee: Percentage of female population age...	1
15	Short definition	Data Interpretation: 1=Latent; 2=Emerging; 3=E...	215
16	Short definition	Percentage of students who were unable to read...	51
17	Short definition	Average total number of invented/nonsense word...	51
18	Short definition	Share of students who scored zero percent on t...	51
19	Short definition	Share of students who scored 80 percent or hig...	51
20	Long definition	Data Interpretation: 1=Latent; 2=Emerging; 3=E...	215
21	Long definition	Percentage of students who were unable to read...	51
22	Long definition	Average total number of invented/nonsense word...	51
23	Long definition	Share of students who scored zero percent on t...	51
24	Long definition	Share of students who scored 80 percent or hig...	51
25	Other notes	EGRA	403
26	Other notes	Health: Population: Structure	52
27	Other notes	Single Level Attainment/ Not Cumulative	21
28	Other notes	Proficiency	20
29	Other notes	Cumulative Attainment	16
30	Source	UNESCO Institute for Statistics	1269
31	Source	Early Grade Reading Assessment (EGRA): https:/...	403

	Variable	Valeur	Nombre
32	Source	Robert J. Barro and Jong-Wha Lee: http://www.b...	360
33	Source	Wittgenstein Centre for Demography and Global ...	308
34	Source	Systems Approach for Better Education Results ...	215

4. Nettoyage des entités non-étatiques (Faux pays)

Cette étape vise à garantir la cohérence référentielle du dataset en isolant les véritables pays des agrégats statistiques (ex: World, High Income).

Méthodologie appliquée :

1. **Identification dans `EdStatsCountry.csv`** :
   - Audit de la colonne `Region` : les entités n'ayant pas d'affectation régionale sont identifiées comme des agrégats économiques ou mondiaux.
2. **Purification du référentiel** :
   - Suppression des lignes identifiées dans `EdStatsCountry.csv` pour créer une "Source de Vérité" fiable.
   - Extraction de la "Liste Blanche" : Création de la liste des Country Code valides à partir du fichier fraîchement nettoyé.
3. **Propagation du nettoyage aux autres DataFrames** :
   - **Méthode par liste :** Utilisation de la liste extraite précédemment pour filtrer les autres tables.
   - **Méthode par jointure :** Utilisation d'un `inner join` (jointure interne) pour ne conserver que les enregistrements dont la clé (`Country Code`) existe dans le référentiel nettoyé.

4.2.1. Suppression des lignes identifiées dans `EdStatsCountry.csv` pour créer une "Source de Vérité" fiable.

```
dataframes[0]['data'] = dataframes[0]['data'].dropna(subset=['Region'])
display(dataframes[0]['data'])
```

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Inc G
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	inc nonC
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	in
2	AGO	Angola	Angola	People's Republic of Angola	AO	Angolan kwanza	April 2013 database update: Based on IMF data,...	Sub-Saharan Africa	U m in
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	U m in
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	inc nonC
...	...	...	...	...	...	...	...	...	
236	XKX	Kosovo	Kosovo	Republic of Kosovo	NaN	Euro	Kosovo became a World Bank member on June 29, ...	Europe & Central Asia	L m in
237	YEM	Yemen	Yemen, Rep.	Republic of Yemen	YE	Yemeni rial	Based on official government statistics and In...	Middle East & North Africa	L m in
238	ZAF	South Africa	South Africa	Republic of South Africa	ZA	South African rand	Fiscal year end: March 31; reporting period fo...	Sub-Saharan Africa	U m in
239	ZMB	Zambia	Zambia	Republic of Zambia	ZM	New Zambian	National accounts	Sub-Saharan	L m

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Inc G
						kwacha	data have rebased to reflect...	Africa	in
240	ZWE	Zimbabwe	Zimbabwe	Republic of Zimbabwe	ZW	U.S. dollar	Fiscal year end: June 30; reporting period for...	Sub-Saharan Africa	in

214 rows × 31 columns

4.2.2. Extraction de la "Liste Blanche" : Création de la liste des Country Code valides à partir du fichier fraîchement nettoyé.

> **Extraction de la 'white list'** :
>
> - **.unique()** : Cette méthode de Pandas identifie toutes les valeurs distinctes dans une colonne. Elle élimine les doublons potentiels et renvoie un objet de type Numpy Array.
> - **.tolist()** : Comme son nom l'indique, cette méthode convertit l'objet Pandas/Numpy en une liste standard Python. C'est une étape cruciale car les listes Python sont plus simples à manipuler pour les opérations de filtrage itératives.
> - **[:5] (Slicing)** : Puisque la liste contient des centaines d'éléments, nous utilisons le découpage (slicing) pour n'afficher que les 5 premiers. Cela permet de valider visuellement le format des données (ex: ['ABW', 'AFG', ...]) sans encombrer le Notebook.

```
list_countryCode = dataframes[0]['data']['Country Code'].unique().tolist()
display(list_countryCode[:5])
```

```
['ABW', 'AFG', 'AGO', 'ALB', 'AND']
```

4.3.1. Utilisation de la liste extraite précédemment pour filtrer les autres tables.

```
display(Markdown(f" Nombre de lignes avant filtrage pour le fichier {dataframes[1]['na
```

Nombre de lignes avant filtrage pour le fichier EdStatsCountry-Series.csv : 613

> **Filtrage par appartenance**

- **.isin()** : .isin(liste) : Cette méthode est un filtre puissant qui compare chaque valeur d'une colonne avec une liste de référence. Elle renvoie True si la valeur est présente dans la liste, et False sinon.
- **Indexation booléenne** : En plaçant ce filtre entre crochets df[mask], Pandas ne conserve que les lignes où la condition est True. Ici, cela permet d'éliminer instantanément toutes les lignes correspondant à des agrégats ou à des entités non-étatiques.
- **Réaffectation** : En réaffectant le résultat à `dataframes[1]['data']`, nous mettons à jour le DataFrame en mémoire avec sa version nettoyée.
- .head() : Comme vu précédemment, nous terminons par un affichage des premières lignes pour confirmer visuellement que le format des données reste cohérent après le filtrage.

```
dataframes[1]['data'] = dataframes[1]['data'][dataframes[1]['data']['CountryCode'].is
display(dataframes[1]['data'].head())
```

	CountryCode	SeriesCode	DESCRIPTION
0	ABW	SP.POP.TOTL	Data sources : United Nations World Population...
1	ABW	SP.POP.GROW	Data sources: United Nations World Population ...
2	AFG	SP.POP.GROW	Data sources: United Nations World Population ...
3	AFG	NY.GDP.PCAP.PP.CD	Estimates are based on regression.
4	AFG	SP.POP.TOTL	Data sources : United Nations World Population...

```
display(Markdown(f" Nombre de lignes aprés filtrage pour le fichier {dataframes[1]['na
```

Nombre de lignes aprés filtrage pour le fichier EdStatsCountry-Series.csv : 611

```
display(Markdown(f" Nombre de lignes avant filtrage pour le fichier {dataframes[2]['na
```

Nombre de lignes avant filtrage pour le fichier EdStatsData.csv : 886930

```
dataframes[2]['data'] = dataframes[2]['data'][dataframes[2]['data']['Country Code'].is
display(dataframes[2]['data'].head())
```

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1975	1980	1985	1990	1991	.
91625	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	NaN	NaN	.
91626	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	NaN	NaN	NaN	.
91627	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	NaN	NaN	NaN	.
91628	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	NaN	NaN	NaN	.
91629	Afghanistan	AFG	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	NaN	NaN	NaN	NaN	NaN	NaN	.

5 rows × 34 columns

```
display(Markdown(f" Nombre de lignes aprés filtrage pour le fichier {dataframes[2]['na
```

Nombre de lignes aprés filtrage pour le fichier EdStatsData.csv : 784310

4.3.2. Utilisation d'un `inner join` (jointure interne) pour ne conserver que les enregistrements dont la clé (`Country Code`) existe dans le référentiel nettoyé.

```
display(Markdown(f" Nombre de lignes avant filtrage pour le fichier {dataframes[3]['na
```

Nombre de lignes avant filtrage pour le fichier EdStatsFootNote.csv : 643638

> **Synchronisation par jointure relationnelle** :

> - **.merge()** : Fonctionnement : Cette méthode réalise une jointure (join) entre deux DataFrames, à la manière d'une base de données SQL. Elle combine les colonnes de deux tableaux en se basant sur une valeur commune.
> - **how='inner** : Ce paramètre définit une jointure "interne". Pandas ne conserve que les lignes dont la clé de jointure est présente simultanément dans les deux fichiers. Les "faux pays" absents du référentiel nettoyé sont donc automatiquement éliminés.
> - **left_on et right_on** : Ces arguments sont utilisés lorsque les colonnes de jointure n'ont pas le même nom. Ici, nous lions la colonne CountryCode (sans espace) du fichier FootNote à la colonne Country Code (avec espace) du référentiel.
> - **Avantage** : Contrairement au filtrage simple, le merge permet de ramener des informations complémentaires d'une table vers une autre tout en assurant l'intégrité référentielle.

```
dataframes[3]['data'] = dataframes[3]['data'].merge(dataframes[0]['data']['Country Co
display(dataframes[3]['data'].head())
```

	CountryCode	SeriesCode	Year	DESCRIPTION	Country Code
0	ABW	SE.PRE.ENRL.FE	YR2001	Country estimation.	ABW
1	ABW	SE.TER.TCHR.FE	YR2005	Country estimation.	ABW
2	ABW	SE.PRE.TCHR.FE	YR2000	Country estimation.	ABW
3	ABW	SE.SEC.ENRL.GC	YR2004	Country estimation.	ABW
4	ABW	SE.PRE.TCHR	YR2006	Country estimation.	ABW

```
display(Markdown(f" Nombre de lignes aprés filtrage pour le fichier {dataframes[3]['na
```

Nombre de lignes aprés filtrage pour le fichier EdStatsFootNote.csv : 515752

> **Note sur `EdStatsFootNote`** : Le fichier `EdStatsFootNote.csv` contient deux colonnes `Country Code` il sera prévus de supprimer la colonne redondante.

> **Note sur le périmètre de filtrage** : Le fichier `EdStatsSeries.csv` n'est pas concerné par ce nettoyage. En tant que référentiel métier des indicateurs, il ne contient aucune dimension géographique (`Country Code`) et ses définitions restent valides pour l'ensemble du domaine d'étude.

5. Réduction du périmètre analytique

L'analyse initiale a révélé un volume d'indicateurs trop important pour une exploitation manuelle pertinente. Cette section détaille la stratégie de filtrage thématique et temporel pour répondre aux

besoins d'expansion internationale de la start-up Academy.

Méthodologie appliquée :

1. **Filtrage thématique :**
   - **Analyse des catégories :** Nous utilisons la colonne `Topic` du fichier `EdStatsSeries` pour identifier les thématiques structurantes.
   - **Critères de sélection :**
     - Inclusion des catégories liées au secondaire (Lycée) et au tertiaire (Université).
     - Inclusion des données sur l'accès aux technologies (IT) et les résultats d'apprentissage.
     - Exclusion des thématiques économiques pures ou de santé sans lien direct avec l'éducation.
   - **Impact :** Réduction du nombre de `Series Code` pour gagner en lisibilité et en performance de calcul.
   - **Catégories retenues** :
     - **Secondary & Tertiary** : Identification directe du volume de clients cibles (Lycée/Université).
     - **Learning Outcomes** : Mesure de la performance éducative réelle (besoin potentiel en soutien scolaire).
     - **Literacy** : Évaluation du socle de compétences de base de la population.
     - **Attainment** : Pour mesurer les diplômes déjà obtenus.
     - **Infrastructure: Communications** : Vérification de la faisabilité technique pour l'enseignement en ligne (connectivité).
2. **Filtrer l'ensemble des jeux de données** :
   - **Séparation des flux de traitement** :
     - Table de liaison (Country-Series) : Isolation dans des variables spécifiques pour diagnostic (résultats nuls observés).
     - Tables transactionnelles : Écrasement séléctif pour optimisation de la mémoire vive
3. **Filtrage temporel** : Filtrez l'ensemble des jeux de données pour ne garder que les indicateurs sélectionnés. **Objectif :** Se concentrer sur les données actuelles et exploitables pour une décision d'implantation immédiate.
   - **Interprétation des années futures :** Les colonnes allant de 2025 à 2100 dans EdStatsData correspondent à des projections démographiques théoriques.
   - **Hypothèse métier :** Pour une stratégie d'expansion à court et moyen terme, ces projections lointaines sont jugées hors périmètre et présentent un taux de remplissage quasi nul.
   - **Action :** Suppression des colonnes d'années futures via une liste générée dynamiquement (`np.arange`).
   - **Plage temporelle retenues** :
     - **2010 - 2019** : Fournit un historique stable pour comprendre l'évolution éducative sur le long terme.

    - **2020 - 2025** : Capture l'impact de la numérisation accélérée de l'enseignement (période COVID et post-COVID).
    - **2026 - 2027** : Offre une vision "présent et futur proche" pour la prise de décision immédiate.
4. **Filtrer le fichier Data** : Filtrer les années en conséquences de l'analyse de la plage temporelle pertinantes.

```
target_topics = [
    'Secondary',
    'Tertiary',
    'Infrastructure: Communications',
    'Learning Outcomes',
    'Attainment',
    'Literacy'
]

df_series_filtered = dataframes[4]['data'][dataframes[4]['data']['Topic'].isin(target_
valid_series_codes = df_series_filtered['Series Code'].unique().tolist()
```

5.1. Filtrage thématique

Nous extrayons les codes d'indicateurs (`Series Code`) issus du référentiel thématique pour les utiliser comme clé de filtrage sur le dataset principal. Création d'un dataframe basé sur EdStatsSeries contenant uniquement les `Topic` séléctionnés précédemment via `isin()` à partie d'une white list.

```
display(df_series_filtered)
```

	Series Code	Topic	Indicator Name	Short definition	Long definition	Other notes	Source
0	BAR.NOED.1519.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15-19 with...	Percentage of female population age 15-19 with...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
1	BAR.NOED.1519.ZS	Attainment	Barro-Lee: Percentage of population age 15-19 ...	Percentage of population age 15-19 with no edu...	Percentage of population age 15-19 with no edu...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
2	BAR.NOED.15UP.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 15+ with n...	Percentage of female population age 15+ with n...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
3	BAR.NOED.15UP.ZS	Attainment	Barro-Lee: Percentage of population age 15+ wi...	Percentage of population age 15+ with no educa...	Percentage of population age 15+ with no educa...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
4	BAR.NOED.2024.FE.ZS	Attainment	Barro-Lee: Percentage of female population age...	Percentage of female population age 20-24 with...	Percentage of female population age 20-24 with...	NaN	Robert J. Barro and Jong-Wha Lee: http://www.b...
...	...	...	...	...	...	...	...
3552	UIS.thDur.4.A.GPV	Tertiary	Theoretical duration of post-secondary non-ter...	NaN	Number of grades (years) in post-secondary edu...	NaN	UNESCO Institute for Statistics
3553	UIS.TranRA.23.GPV.GPI	Secondary	Effective transition rate from primary to lowe...	NaN	The ratio of the female transition rate to the...	NaN	UNESCO Institute for Statistics
3578	UIS.UAPP.23	Secondary	Under-age enrolment ratio in secondary educati...	NaN	Percentage of the secondary school age populat...	NaN	UNESCO Institute for Statistics
3579	UIS.UAPP.23.F	Secondary	Under-age enrolment ratio in	NaN	Percentage of the female secondary	NaN	UNESCO Institute for Statistics

	Series Code	Topic	Indicator Name	Short definition	Long definition	Other notes	Source
			secondary educati...		school age ...		
3580	UIS.UAPP.23.M	Secondary	Under-age enrolment ratio in secondary educati...	NaN	Percentage of the male secondary school age po...	NaN	UNESCO Institute for Statistics

2227 rows × 7 columns

```
display(Markdown(f"Nombre de lignes dans `EdStatsSeries.csv` : {len(dataframes[4]['da
```

Nombre de lignes dans `EdStatsSeries.csv` : 3665, nombres de lignes restant aprés filtrage par `Topic` : 2227

```
display(Markdown(f"### Nombre d'indicateur restant {len(valid_series_codes)}"))
```

Nombre d'indicateur restant 2227

Création d'une liste de `Series Code` associé aux `Topic` séléctionnés.

```
display(valid_series_codes[:5])
```

```
['BAR.NOED.1519.FE.ZS',
 'BAR.NOED.1519.ZS',
 'BAR.NOED.15UP.FE.ZS',
 'BAR.NOED.15UP.ZS',
 'BAR.NOED.2024.FE.ZS']
```

5.2. Filtrer l'ensemble des jeux de données

```
df_data = dataframes[1]['data']
df_data_filtered_EdStatsCountry_Series = df_data[df_data['SeriesCode'].isin(valid_ser
display(df_data_filtered_EdStatsCountry_Series)
```

	CountryCode	SeriesCode	DESCRIPTION

```
df_data = dataframes[2]['data']
df_data_filtered = df_data[df_data['Indicator Code'].isin(valid_series_codes)]
display(Markdown(f"Nombre de ligne avant filtrage : {len(dataframes[2]['data'])}"))
```

Nombre de ligne avant filtrage : 784310

```
dataframes[2]['data'] = df_data_filtered
display(dataframes[2]['data'])
```

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1975	1980	1985	1990
91625	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	NaN
91626	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	NaN	NaN
91627	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	NaN	NaN
91628	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	NaN	NaN
91633	Afghanistan	AFG	Adjusted net enrolment rate, upper secondary, ...	UIS.NERA.3	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
886922	Zimbabwe	ZWE	Youth illiterate population, 15-24 years, % fe...	UIS.LPP.AG15T24	NaN	NaN	NaN	NaN	NaN
886926	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, b...	SE.ADT.1524.LT.ZS	NaN	NaN	NaN	NaN	NaN
886927	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, f...	SE.ADT.1524.LT.FE.ZS	NaN	NaN	NaN	NaN	NaN

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1975	1980	1985	1990
886928	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, g...	SE.ADT.1524.LT.FM.ZS	NaN	NaN	NaN	NaN	NaN
886929	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, m...	SE.ADT.1524.LT.MA.ZS	NaN	NaN	NaN	NaN	NaN

470158 rows × 34 columns

```
display(Markdown(f"Nombre de lignes aprés filtrage : {len(dataframes[2]['data'])}"))
```

Nombre de lignes aprés filtrage : 470158

```
df_data = dataframes[3]['data']
df_data_filtered = df_data[df_data['SeriesCode'].isin(valid_series_codes)]
display(Markdown(f"Nombre de ligne avant filtrage : {len(dataframes[3]['data'])}"))
```

Nombre de ligne avant filtrage : 515752

```
dataframes[3]['data'] = df_data_filtered
display(dataframes[3]['data'])
```

	CountryCode	SeriesCode	Year	DESCRIPTION	Country Code
3	ABW	SE.SEC.ENRL.GC	YR2004	Country estimation.	ABW
6	ABW	SE.SEC.ENRL.VO.FE	YR2005	Country estimation.	ABW
7	ABW	SE.SEC.ENRL.GC	YR2003	Country estimation.	ABW
12	ABW	SE.SEC.ENRL.VO.FE.ZS	YR2002	Country estimation.	ABW
14	ABW	SE.SEC.ENRL.VO.FE.ZS	YR2007	Country estimation.	ABW
...	...	...	...	...	...
515701	ZWE	SE.TER.ENRR.MA	YR1992	Country Data	ZWE
515702	ZWE	SE.TER.GRAD.FE.ZS	YR1981	Country Data	ZWE
515733	ZWE	SE.SEC.ENRL.GC	YR1999	Country estimation.	ZWE
515734	ZWE	SE.SEC.ENRL.GC.FE	YR1998	UIS estimation.	ZWE
515735	ZWE	SE.SEC.ENRL.GC.FE.ZS	YR2000	Country estimation.	ZWE

218137 rows × 5 columns

```
display(Markdown(f"Nombre de lignes aprés filtrage : {len(dataframes[3]['data'])}"))
```

Nombre de lignes aprés filtrage : 218137

5.3. Filtrage temporel

Création d'une liste contenant les années non conservé via `np.range`. Cette liste ce compose de la concaténation des années passé plus les années futures.

```
past_years = np.arange(1970, 2010).astype(str).tolist()
future_years = np.arange(2028, 2101).astype(str).tolist()
columns_to_delete = past_years + future_years

display(columns_to_delete[:5])
```

```
['1970', '1971', '1972', '1973', '1974']
```

5.4. Filtrer le fichier Data

```
display(Markdown(f"### Dimensions du dataset avant filtrage temporel : {dataframes[2]
```

Dimensions du dataset avant filtrage temporel : (470158, 34)

```
dataframes[2]['data'].drop(columns=columns_to_delete, errors='ignore', inplace=True)
display(dataframes[2]['data'].head())
```

	Country Name	Country Code	Indicator Name	Indicator Code	2010	2011	2012	2013	2014
91625	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	47.436790	50.62723
91626	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	34.073261	37.64154
91627	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	0.567060	0.59837
91628	Afghanistan	AFG	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	60.087059	62.90695
91633	Afghanistan	AFG	Adjusted net enrolment rate, upper secondary, ...	UIS.NERA.3	NaN	NaN	NaN	31.332621	32.41703

```
display(Markdown(f"### Dimensions du dataset aprés filtrage temporel : {dataframes[2]
```

Dimensions du dataset aprés filtrage temporel : (470158, 10)

Analyse de la disponibilité temporelle

Bien que notre périmètre d'étude cible initialement la plage 2010-2027, l'audit de complétude révèle une attrition massive des données sur la période récente.

Limiter l'analyse à 2010-2015

D'un point de vue statistique et métier, conserver les années au-delà de 2015 introduirait des risques majeurs:

1. **Absence de représentativité** : Un taux de complétude inférieur à 2 % signifie que moins d'une cinquantaine de points de données subsistent pour l'ensemble du globe. Une décision d'investissement ne peut reposer sur un échantillon aussi restreint.
2. **Biais de sélection** : Seuls quelques pays très spécifiques (souvent les plus développés ou ayant des programmes de collecte particuliers) renseignent ces années récentes. Analyser 2025 reviendrait à ignorer 99 % des marchés potentiels d'Academy.
3. **Nature des données futures** : Les colonnes 2020-2027 dans ce dataset sont essentiellement des projections théoriques. En l'absence de valeurs réelles saisies, elles n'offrent aucune plus-value pour un scoring de pays comparatif.

Conclusion de l'audit : Nous privilégions la qualité à la quantité. La fenêtre 2010-2015 constitue le segment le plus récent offrant une masse critique de données (complétude ~28 %) permettant d'établir un diagnostic international fiable et comparable.

6 : Analyse de la complétude (Data Fill Rate)

Cette section constitue le cœur de notre audit statistique. L'objectif est de quantifier la densité des données pour identifier les indicateurs présentant une masse critique suffisante pour un scoring international fiable.

Méthodologie appliquée :

1. **Développement d'un indicateur de densité :**
   - Création d'une fonction `calculate_data_fill_rate` permettant de mesurer la proportion de valeurs non-nulles (`.notnull()`) sur n'importe quel axe du DataFrame.
   - **Axe temporel (Années)** : Identification des années les mieux renseignées pour l'ensemble des thématiques.
   - **Axe thématique (Indicateurs)** : Évaluation de la régularité de mise à jour de chaque métrique sur la période 2010-2015.
2. **Mesure de la couverture géographique :**
   - Utilisation de la fonction `groupby('Indicator Name').count()` pour dénombrer, pour chaque indicateur, le nombre de pays ayant fourni au moins une valeur réelle.
   - **Scoring cumulé** : Création de la variable `Somme_Pays_Total` qui agrège la présence de données sur l'ensemble de la fenêtre temporelle pour classer les indicateurs par "richesse".
3. **Stratégie de sélection finale :**
   - **Arbitrage Quantité/Qualité** : Nous trions les résultats par ordre décroissant pour isoler le "Top 15" des indicateurs.
   - **Filtre Métier** : Parmi les plus riches, nous ne conservons que ceux répondant aux trois piliers d'Academy :
     - **Potentiel client** : Volume d'étudiants (Secondary/Tertiary).
     - **Faisabilité technique** : Accès au numérique (Internet Users).
     - **Besoin en soutien** : Performance éducative (Learning Outcomes).

6.1. Développement d'un indicateur de densité :

Pour évaluer la fiabilité de notre dataset, nous avons développé une fonction utilitaire permettant de mesurer la densité des données. Cette approche évite la répétition de code (principe DRY - Don't Repeat Yourself) et garantit une mesure homogène sur tous les axes du DataFrame.

Justification de la logique appliquée :

- `.notnull()` : Cette méthode crée un masque booléen identifiant chaque cellule contenant une information réelle. Les valeurs NaN (manquantes) sont ignorées.
- `.mean(axis=axis)` : En calculant la moyenne de ce masque booléen (True valant 1 et False valant 0), nous obtenons directement le taux de remplissage exprimé entre 0 et 1.
- Flexibilité de l'axe (`axis`) :
  - `axis=0` (par colonne) : Permet d'identifier les années les plus riches en données pour l'ensemble des pays
  - `axis=1` (par ligne) : Permet de calculer un score de fiabilité pour chaque indicateur spécifique, en mesurant sa régularité sur la plage temporelle choisie.

> **Note métier** : Cette fonction nous permet de ne retenir que les segments ayant une masse critique suffisante pour que les conclusions stratégiques d'Academy soient statistiquement significatives.

```
def calculate_data_fill_rate(df_analys, axis=0):
    return df_analys.notnull().mean(axis=axis)
```

1. **Fonctionnement technique : Le Slicing avec `.loc`**
   - `.loc` : C'est une méthode d'accès par étiquettes (labels).
   - `:` : Le premier argument (avant la virgule) signifie qu'on sélectionne toutes les lignes.
   - `'2010':'2015'` : Le deuxième argument définit la plage de colonnes. Grâce à l'opérateur `:`, Pandas récupère toutes les colonnes situées entre 2010 et 2015 inclus.

```
df_years_only = dataframes[2]['data'].loc[:, '2010':'2015']
display(df_years_only)
```

	2010	2011	2012	2013	2014	2015
91625	NaN	NaN	NaN	47.436790	50.627232	NaN
91626	NaN	NaN	NaN	34.073261	37.641541	NaN
91627	NaN	NaN	NaN	0.567060	0.598370	NaN
91628	NaN	NaN	NaN	60.087059	62.906952	NaN
91633	NaN	NaN	NaN	31.332621	32.417030	NaN
...	...	...	...	...	...	...
886922	NaN	43.61436	NaN	NaN	35.887100	NaN
886926	NaN	90.93070	NaN	NaN	90.428120	NaN
886927	NaN	92.12456	NaN	NaN	93.188350	NaN
886928	NaN	1.02828	NaN	NaN	1.063890	NaN
886929	NaN	89.59058	NaN	NaN	87.591860	NaN

470158 rows × 6 columns

> **Fonctionnement technique** :
>
> - `calculate_data_fill_rate(..., axis=0)` : En passant l'argument axis=0, on indiques à la fonction de calculer la moyenne verticalement (colonne par colonne).
> - `.to_frame(name='...')` : Par défaut, le résultat est une Series Pandas (une simple liste indexée). Cette méthode la convertit en un DataFrame, ce qui facilite la lecture et l'exportation future du rapport.

```
data_fill_rate_per_year = calculate_data_fill_rate(df_years_only, axis=0)
display(data_fill_rate_per_year.to_frame(name='Ratio de complétude'))
```

	Ratio de complétude
2010	0.302315
2011	0.096706
2012	0.101542
2013	0.085497
2014	0.074739
2015	0.151392

> **Fonctionnement technique** :

> - `calculate_data_fill_rate(..., axis=1)` : En passant l'argument axis=0, on indiques à la fonction de calculer la moyenne verticalement (ligne par ligne).

```
data_fill_rate_per_indicator = calculate_data_fill_rate(df_years_only, axis=1)
display(data_fill_rate_per_indicator.to_frame(name='Ratio de complétude'))
```

	Ratio de complétude
91625	0.333333
91626	0.333333
91627	0.333333
91628	0.333333
91633	0.333333
...	...
886922	0.333333
886926	0.333333
886927	0.333333
886928	0.333333
886929	0.333333

470158 rows × 1 columns

> **Fonctionnement technique** :
>
> Le groupby est une opération en trois étapes souvent appelée **Split-Apply-Combine**.
>
> - **Split** : Pandas rassemble toutes les lignes qui ont le même Indicator Name (par exemple, toutes les lignes "Internet users" de tous les pays).
> - **Apply** : La fonction `.count()` est appliquée à chaque groupe.
>   - Note importante : `.count()` ne compte que les valeurs non-nulles. Si un pays n'a pas de donnée pour 2010, il n'est pas compté.
> - **Combine** : Pandas rassemble les résultats dans un nouveau tableau où chaque ligne est un indicateur unique, et chaque colonne contient le nombre de pays l'ayant renseigné.
>
> **Mesure de la couverture géographique** : On ne regarde plus si une cellule est vide, on regarde si l'indicateur est global. Un indicateur présent dans 200 pays a beaucoup plus de valeur qu'un indicateur présent dans seulement 5 pays.
>
> **Tri stratégique** : Le `sort_values` permet d'isoler immédiatement les "indicateurs stars" (comme Internet users ou Secondary enrolment) qui seront les piliers de la stratégie d'Academy.

```
df_density = dataframes[2]['data'][['Indicator Name', '2010', '2011', '2012', '2013',
indicators_per_countries = df_density.groupby('Indicator Name').count()
indicators_per_countries['Somme_Pays_Total'] = indicators_per_countries.sum(axis=1)
sort_indicators = indicators_per_countries.sort_values(by='Somme_Pays_Total', ascendi
display(sort_indicators.head(30))
```

Indicator Name	2010	2011	2012	2013	2014	2015	Somme_Pays_Total
Official entrance age to lower secondary education (years)	202	201	201	202	202	202	1210
Theoretical duration of secondary education (years)	202	201	201	202	202	202	1210
Theoretical duration of upper secondary education (years)	202	201	201	202	202	202	1210
Internet users (per 100 people)	201	203	201	201	201	201	1208
Enrolment in secondary general, both sexes (number)	156	162	161	155	154	139	927
Percentage of students in secondary general education who are female (%)	156	162	161	154	154	139	926
Enrolment in secondary general, female (number)	156	162	161	154	154	139	926
Gross enrolment ratio, lower secondary, both sexes (%)	151	159	156	148	147	129	890
Gross enrolment ratio, lower secondary, female (%)	151	157	155	147	146	129	885
Gross enrolment ratio, lower secondary, male (%)	151	157	155	147	146	129	885
Enrolment in secondary education, both sexes (number)	150	155	148	140	139	122	854
Percentage of students in secondary education who are female (%)	150	154	147	137	137	121	846
Enrolment in secondary education, female (number)	150	154	147	137	137	121	846
Gross enrolment ratio, upper secondary, both sexes (%)	144	149	144	136	133	114	820
Gross enrolment ratio, secondary, both sexes (%)	145	149	142	134	133	117	820
Gross enrolment ratio, secondary, male (%)	145	148	141	131	131	116	812
Gross enrolment ratio, secondary, gender parity index (GPI)	145	148	141	131	131	116	812
Gross enrolment ratio, secondary, female (%)	145	148	141	131	131	116	812
Gross enrolment ratio, upper secondary, male (%)	144	147	143	134	131	113	812

Indicator Name	2010	2011	2012	2013	2014	2015	Somme_Pays_Total
Enrolment in lower secondary general, female (number)	161	166	167	157	135	8	794
Enrolment in lower secondary general, both sexes (number)	161	166	167	157	135	8	794
Percentage of students in lower secondary general education who are female (%)	161	166	167	157	135	8	794
Enrolment in tertiary education, all programmes, both sexes (number)	140	142	141	131	124	109	787
Gross enrolment ratio, primary and secondary, gender parity index (GPI)	140	145	138	125	127	111	786
Percentage of students in tertiary education who are female (%)	138	138	138	127	121	106	768
Percentage of students in upper secondary general education who are female (%)	157	161	159	153	130	8	768
Enrolment in upper secondary general, female (number)	157	161	159	153	130	8	768
Enrolment in tertiary education, all programmes, female (number)	138	138	138	127	121	106	768
Enrolment in upper secondary general, both sexes (number)	157	161	159	153	130	8	768
Percentage of enrolment in secondary education in private institutions (%)	129	134	130	130	130	114	767

7. Construction du Dataset Décisionnel (Pivotement & Agrégation)

L'objectif de cette phase est de restructurer nos données pour obtenir une **"Source Unique de Vérité"** par pays. Nous passons d'un format "long" (plusieurs lignes d'indicateurs et plusieurs colonnes d'années par pays) à un format "large", où chaque pays est décrit par une ligne unique regroupant l'ensemble de nos variables stratégiques.

Méthodologie appliquée :

1. Filtrage Final du Périmètre :

- **Indicateurs** : Utilisation de la liste `final_indicators` regroupant les 15 indicateurs clés retenus pour leur pertinence métier (Lycée, Université, IT).
- **Années** : Focalisation sur la fenêtre **2010-2015**, identifiée précédemment comme le segment le plus dense et le plus fiable en données exploitables.

2. Stratégie d'Agrégation Temporelle :

- Puisque nous disposons de données réparties sur 6 ans, il est nécessaire de résumer cette période en une statistique unique par couple (Pays, Indicateur).
- **Choix de la Moyenne (`mean`)** : Nous optons pour la moyenne arithmétique afin de lisser les éventuelles variations annuelles. Cela permet d'obtenir une tendance stable du niveau d'éducation et d'équipement sur la période choisie.

3. Pivotement des Données (`pivot_table`) :

- **Restructuration** : Le DataFrame est pivoté pour que chaque `Indicator Name` devienne une colonne distincte.
- **Unicité** : Chaque ligne représentera désormais un **pays unique**, facilitant ainsi le futur calcul de scoring et la comparaison directe entre les marchés potentiels.

4. Gestion de la Fiabilité :

- Grâce au nettoyage préalable par seuil (`thresh`), nous garantissons que les moyennes calculées reposent sur une base de données suffisante.
- Cette rigueur limite l'impact des valeurs aberrantes ou isolées, assurant une base solide pour les recommandations stratégiques finales.

7.1. Sélection finale des indicateurs clés

Cette liste constitue la variable selection_indicateurs_finale demandée par l'exercice. Elle équilibre la richesse des données (complétude) et la pertinence stratégique.

1. **Metriques coeur (Indispensables)** : Ces indicateurs mesurent directement la taille du marché et la faisabilité technique de l'offre en ligne d'Academy.
   - **Internet users (per 100 people)** : Indispensable pour valider la faisabilité du e-learning.
   - **Enrolment in secondary education, both sexes (number)** : Mesure la taille brute du marché "Lycée".
   - **Enrolment in tertiary education, all programmes, both sexes (number)** : Mesure la taille brute du marché "Université".
   - **Gross enrolment ratio, secondary, both sexes (%)** : Indique le taux de scolarisation (pénétration du système éducatif).
   - **Gross enrolment ratio, tertiary, both sexes (%)** : Indique l'accès à l'enseignement supérieur.
   - **Population, ages 15-24, total** : Identifie le bassin démographique cible (cœur d'audience).

- **Personal computers (per 100 people)** : Complète la donnée Internet pour valider l'équipement des foyers.

2. **Metriques contextuels (Utiles)** : Ces indicateurs permettent d'affiner le "scoring" en mesurant le niveau de développement éducatif et le besoin potentiel.
   - **Literacy rate, adult total (% of people ages 15 and above)** : Indicateur du socle de compétences de base du pays.
   - **Government expenditure on education, total (% of GDP)** : Mesure l'investissement public et la stabilité du secteur.
   - **Pupil-teacher ratio in secondary education (headcount basis)** : Un ratio élevé peut indiquer un besoin accru en soutien scolaire privé.
   - **Lower secondary completion rate, both sexes (%)** : Mesure le flux d'étudiants arrivant vers le Lycée.
   - **Secondary education, duration (years)** : Utile pour adapter la durée des programmes d'accompagnement.
   - **Tertiary education, academic staff (% female)** : Indicateur de la structure du corps enseignant supérieur.
   - **Barro-Lee: Average years of total secondary education, ages 15+, total** : Donne une vision du stock de compétences "Lycée" déjà présent.
   - **Barro-Lee: Percentage of population age 15+ with secondary schooling. Completed Secondary** : Précise le niveau de diplôme moyen de la cible.

```
final_indicators = [
    'Internet users (per 100 people)',
    'Enrolment in secondary education, both sexes (number)',
    'Enrolment in tertiary education, all programmes, both sexes (number)',
    'Gross enrolment ratio, secondary, both sexes (%)',
    'Gross enrolment ratio, tertiary, both sexes (%)',
    'Population, ages 15-24, total',
    'Personal computers (per 100 people)',
    'Literacy rate, adult total (% of people ages 15 and above)',
    'Government expenditure on education, total (% of GDP)',
    'Pupil-teacher ratio in secondary education (headcount basis)',
    'Lower secondary completion rate, both sexes (%)',
    'Secondary education, duration (years)',
    'Tertiary education, academic staff (% female)',
    'Barro-Lee: Average years of total secondary education, ages 15+, total',
    'Barro-Lee: Percentage of population age 15+ with secondary schooling. Completed S
]
display(final_indicators)
```

```
['Internet users (per 100 people)',
 'Enrolment in secondary education, both sexes (number)',
 'Enrolment in tertiary education, all programmes, both sexes (number)',
 'Gross enrolment ratio, secondary, both sexes (%)',
 'Gross enrolment ratio, tertiary, both sexes (%)',
 'Population, ages 15-24, total',
 'Personal computers (per 100 people)',
 'Literacy rate, adult total (% of people ages 15 and above)',
 'Government expenditure on education, total (% of GDP)',
 'Pupil-teacher ratio in secondary education (headcount basis)',
 'Lower secondary completion rate, both sexes (%)',
 'Secondary education, duration (years)',
 'Tertiary education, academic staff (% female)',
 'Barro-Lee: Average years of total secondary education, ages 15+, total',
 'Barro-Lee: Percentage of population age 15+ with secondary schooling. Completed Sec
ondary']
```

```python
# Filtrage du dataframe final pour ne garder que ces 15 indicateurs
df_final_academy = dataframes[2]['data'][dataframes[2]['data']['Indicator Name'].isin
with pd.option_context('display.float_format', '{:.2f}'.format):
    display(df_final_academy.head())
```

	Country Name	Country Code	Indicator Name	Indicator Code	2010	2011	
92002	Afghanistan	AFG	Barro-Lee: Percentage of population age 15+ wi...	BAR.SEC.CMPT.15UP.ZS	8.65	NaN	
92816	Afghanistan	AFG	Enrolment in secondary education, both sexes (...	SE.SEC.ENRL	2044157.00	2208963.00	24158
92829	Afghanistan	AFG	Enrolment in tertiary education, all programme...	SE.TER.ENRL	NaN	97504.00	
92960	Afghanistan	AFG	Gross enrolment ratio, secondary, both sexes (%)	SE.SEC.ENRR	53.25	54.62	
92964	Afghanistan	AFG	Gross enrolment ratio, tertiary, both sexes (%)	SE.TER.ENRR	NaN	3.76	