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.

```
import numpy as np
import pandas as pd
import glob
import os
from IPython.display import Markdown
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

```
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

```
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

```
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.

```
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

```
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

```
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

```
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
...	...	...	...
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

110 rows × 3 columns

```
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

```
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[2]['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	1980	1985	1990	1995	
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	65.617767	69.033211	71.995819	71.81176	76.25

5 rows × 25 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 : 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
```

	1980	1985	1990	1995	1999	20
count	8.912200e+04	9.029600e+04	1.244050e+05	1.313610e+05	1.188390e+05	1.766760e+
mean	3.283898e+09	3.622763e+09	9.084424e+09	1.052543e+10	1.331558e+10	9.423384e+
std	1.780774e+11	2.002929e+11	3.665667e+11	4.285218e+11	5.153472e+11	4.442374e
min	-1.404240e+00	-2.216315e+00	-1.803750e+00	-2.697722e+00	-6.526000e+04	-6.759300e+
25%	1.770000e+00	2.150000e+00	4.830000e+00	5.200000e+00	1.749051e+01	5.699035e+
50%	1.107000e+01	1.200000e+01	5.048379e+01	5.018663e+01	1.251000e+03	5.078717e-
75%	8.202760e+01	8.338313e+01	9.134300e+04	7.954000e+04	1.867360e+05	3.343950e+
max	2.784319e+13	3.166465e+13	4.714344e+13	5.275448e+13	6.040632e+13	6.327293e-

8 rows × 21 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

```
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

```
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

```
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