Projet 2 : Analysez des données de systèmes éducatifs

Lecerf Defer Amandine

Mise en place de l'environnement de travail

In [ ]:

```
import pandas as pd
import numpy as  np
import collections
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
import inspect
import missingno as msno
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [ ]:

```
#Optimisation automatique des colonnes pour avoir un maximum d'information
pd.options.display.width = 0
```

Découverte des données

EdStatsCountry

In [ ]:

```
Stats_country = pd.read_csv("Data/EdStatsCountry.csv")
Stats_country.head()
```

Out[ ]:

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Income Group	WB-2 code	...	IMF data dissemination standard
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	High income: nonOECD	AW	...	NaN
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	Low income	AF	...	General Data Dissemination System (GDDS)
							April 2013					General Data

2	AGO	Angola	Angola	People's Republic of Angola	AO	Angolan kwanza	database update: Special Nov 15 data,...	Sub-Saharan Africa	Upper middle Income	AO	...	General Data Dissemination System (GDDS) IMF data dissemination standard
	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit		Region	Group	WB-2 code	...	
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	Upper middle income	AL	...	General Data Dissemination System (GDDS)
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	High income: nonOECD	AD	...	NaN

5 rows × 32 columns

Ce premier fichier nous donne des informations globales sur l'économie de chaque pays.

In [ ]:

```
Stats_country.shape
```

Out[ ]:

```
(241, 32)
```

Ce fichier possède 241 lignes et 32 colonnes (correspondants à des informations générales sur l'économie).

In [ ]:

```
Stats_country["Country Code"].nunique()
```

Out[ ]:

```
241
```

Chaque ligne correspond à un pays unique, il y a donc 241 pays analysés dans ce fichier.

In [ ]:

```
Stats_country.isna().sum().sort_values(ascending=False)
```

Out[ ]:

```
Unnamed: 31                                          241
National accounts reference year                     209
Alternative conversion factor                        194
Other groups                                         183
Latest industrial data                               134
Vital registration complete                          130
External debt Reporting status                       117
Latest household survey                              100
Latest agricultural census                            99
Lending category                                      97
PPP survey year                                       96
Special Notes                                         96
Source of most recent Income and expenditure data     81
Government Accounting concept                         80
Latest water withdrawal data                          62
Balance of Payments Manual in use                     60
IMF data dissemination standard                       60
Latest trade data                                     56
```

```
SNA price valuation                44
System of trade                    41
National accounts base year        36
Latest population census           28
Income Group                       27
Region                             27
Currency Unit                      26
System of National Accounts        26
2-alpha code                        3
WB-2 code                           1
Long Name                           0
Table Name                          0
Short Name                          0
Country Code                        0
dtype: int64
```

Dans la plupart des colonnes il y a des données manquantes. La colonne Unnamed: 31 ne contient que des NaNs, elle n'est d'aucune utilité pour réaliser l'analyse, on peut donc la supprimer.

In [ ]:

```
Stats_country = Stats_country.drop(columns='Unnamed: 31')
```

In [ ]:

```
Stats_country.duplicated(keep=False).sum()
```

Out[ ]:

```
0
```

Le fichier ne contient pas de doublon.

EdStatsCountry-Series

In [ ]:

```
Country_series = pd.read_csv("Data/EdStatsCountry-Series.csv")
Country_series.head()
```

Out[ ]:

	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

Ce fichier contient les sources des données contenues dans le fichier précédent.

In [ ]:

```
Country_series.shape
```

Out[ ]:

```
(613, 4)
```

(613, 4)

Ce fichier possède 613 lignes et 4 colonnes.

In [ ]:

```
Country_series["CountryCode"].nunique()
```

Out[ ]:

```
211
```

Il y a presque 3 fois plus de lignes que de country code (1 country code représente un pays unique), ce qui signifie qu'il y a plusieurs lignes par pays.

In [ ]:

```
Country_series.isna().sum().sort_values(ascending=False)
```

Out[ ]:

```
Unnamed: 3     613
DESCRIPTION      0
SeriesCode       0
CountryCode      0
dtype: int64
```

Il n'y a pas de donnée manquante sauf pour la colonne Unnamed: 3 qui ne contient que des NaNs, elle n'est d'aucune utilité pour réaliser l'analyse, on peut donc la supprimer.

In [ ]:

```
Country_series = Country_series.drop(columns='Unnamed: 3')
```

In [ ]:

```
Country_series.duplicated(keep=False).sum()
```

Out[ ]:

```
0
```

Le fichier ne contient pas de doublon.

EdStatsData

In [ ]:

```
Stats_Data = pd.read_csv("Data/EdStatsData.csv")
Stats_Data.head()
```

Out[ ]:

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1971	1972	1973	1974	1975	...	2060
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1971	1972	1973	1974	1975	...	2060
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	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	57.991138	59.36554	...	NaN

5 rows × 70 columns

Ce fichier donne l'évolution sur de nombreux indicateurs pour tous les pays de l'année 1970 à 2020.

In [ ]:

```
Stats_Data.shape
```

Out[ ]:

```
(886930, 70)
```

Ce fichier possède 886930 lignes et 70 colonnes.

In [ ]:

```
Stats_Data["Country Code"].nunique()
```

Out[ ]:

```
242
```

Il y a beaucoup plus de lignes que de nombre de pays (environ 3665 lignes par pays). Au vu de la nature du fichier, on peut supposer qu'il y a 3665 indicateurs renseignés pour chaque pays.

In [ ]:

```
Stats_Data.isna().sum().sort_values(ascending=False)
```

Out[ ]:

```
Unnamed: 69       886930
2017              886787
2016              870470
1971              851393
1973              851385
                   ...
2010              644488
Indicator Code         0
Indicator Name         0
```

```
Country Code            0
Country Name            0
Length: 70, dtype: int64
```

Les colonnes représentent 70 ans d'étude. Il y a beaucoup de données manquantes pour toutes les colonnes années, il faudra donc choisir les années sur lesquelles réaliser l'étude. La colonne Unnamed: 69 ne contient que des NaNs, elle n'est d'aucune utilité pour réaliser l'analyse, on peut donc la supprimer.

In [ ]:

```
Stats_Data = Stats_Data.drop(columns='Unnamed: 69')
```

In [ ]:

```
Stats_Data.duplicated(keep=False).sum()
```

Out[ ]:

```
0
```

Le fichier ne contient pas de doublon.

EdStatsFootNote

In [ ]:

```
FootNote = pd.read_csv("Data/EdStatsFootNote.csv")
FootNote.head()
```

Out[ ]:

	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

Ce fichier donne des informations sur l'année d'origine des données correspondant à chaque indicateur pour chaque pays.

In [ ]:

```
FootNote.shape
```

Out[ ]:

```
(643638, 5)
```

Ce fichier possède 643638 lignes et 5 colonnes.

In [ ]:

```
FootNote["CountryCode"].nunique()
```

Out[ ]:

```
239
```

Dans ce dataset, 239 pays sont représentés.

In [ ]:

```
FootNote.isna().sum().sort_values(ascending=False)
```

Out[ ]:

```
Unnamed: 4      643638
DESCRIPTION          0
Year                 0
SeriesCode           0
CountryCode          0
dtype: int64
```

Il n'y a pas de donnée manquante. Seule la colonne Unnamed: 4 ne contient que des NaNs, elle n'est d'aucune utilité pour réaliser l'analyse, on peut donc la supprimer.

In [ ]:

```
FootNote = FootNote.drop(columns='Unnamed: 4')
```

In [ ]:

```
FootNote.duplicated(keep=False).sum()
```

Out[ ]:

```
0
```

Le fichier ne contient pas de doublon.

EdStatsSeries

In [ ]:

```
StatsSeries = pd.read_csv("Data/EdStatsSeries.csv")
StatsSeries.head()
```

Out[ ]:

	Series Code	Topic	Indicator Name	Short definition	Long definition	Unit of measure	Periodicity	Base Period	Other notes	Aggregation method
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	NaN	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	NaN	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	NaN	NaN	NaN
3			Barro-Lee: Percentage of	Percentage of population	Percentage of population					

	Series Code	Topic	Indicator Name	Short definition	Long definition	Unit of measure	Periodicity	Base Period	Other notes	Aggregation method
3	BAR.NOED.15UP.ZS	Attainment	population age 15+	age 15+	age 15+	NaN	NaN	NaN	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	NaN	NaN	NaN

5 rows × 21 columns

Ce fichier donne des informations sur les indicateurs socio-économiques étudiés pour chaque pays.

In [ ]:

```
StatsSeries.shape
```

Out[ ]:

```
(3665, 21)
```

Ce fichier possède 3665 lignes et 21 colonnes.

In [ ]:

```
StatsSeries["Indicator Name"].nunique()
```

Out[ ]:

```
3665
```

Dans ce dataset, il y a des informations pour les 3665 indicateurs étudiés dans chaque pays

In [ ]:

```
StatsSeries.isna().sum().sort_values(ascending=False)
```

Out[ ]:

```
Unnamed: 20                          3665
Related indicators                   3665
Other web links                      3665
Unit of measure                      3665
License Type                         3665
Notes from original source           3665
Development relevance                3662
General comments                     3651
Limitations and exceptions           3651
Statistical concept and methodology  3642
Aggregation method                   3618
Periodicity                          3566
Related source links                 3450
Base Period                          3351
Other notes                          3113
Short definition                     1509
Source                                  0
Long definition                         0
Indicator Name                          0
Topic                                   0
Series Code                             0
dtype: int64
```

Il y a de nombreuses données manquantes. Plusieurs colonnes (Unnamed: 20, Related indicators, Other web links, Unit of measure, License Type, Notes from original source) ne contiennent que des NaNs. La colonne

Unname: 20 n'est d'aucune utilité pour réaliser l'analyse, on peut donc la supprimer.

In [ ]:

```
StatsSeries = StatsSeries.drop(columns='Unnamed: 20')
```

In [ ]:

```
StatsSeries.duplicated(keep=False).sum()
```

Out[ ]:

```
0
```

Le fichier ne contient pas de doublon.

Bilan sur les Datasets

In [ ]:

```
def retrieve_name(var):
    '''afficher le nom d\'une variable '''
    callers_local_vars = inspect.currentframe().f_back.f_locals.items()
    return [var_name for var_name, var_val in callers_local_vars if var_val is var]
list_dataset = [Stats_country, Country_series, Stats_Data, FootNote, StatsSeries]
for dataset in list_dataset:
    print('Le Dataset {} possède {} lignes et {} colonnes.'.format(retrieve_name(dataset
)[0], dataset.shape[0], dataset.shape[1]))
```

```
Le Dataset Stats_country possède 241 lignes et 31 colonnes.
Le Dataset Country_series possède 613 lignes et 3 colonnes.
Le Dataset Stats_Data possède 886930 lignes et 69 colonnes.
Le Dataset FootNote possède 643638 lignes et 4 colonnes.
Le Dataset StatsSeries possède 3665 lignes et 20 colonnes.
```

- **Le Dataset Stats_country contient des informations globales sur l'économie de chaque pays du monde. Il y a certaines valeurs manquantes pour de nombreux pays. Il n'y a aucun doublon.**
- **Le Dataset Country_series contient la source des données des informations contenues dans EdStatsCountry.csv. Il n'y a pas de donnée manquante, ni de doublon.**
- **Le Dataset Stats_Data donne l'évolution de plusieurs indicateurs pour tous les pays du monde. Il y a beaucoup de données manquantes pour plusieurs indicateurs et pour de nombreuses années surtout celles antérieures à 2020. Il n'y a pas de doublon.**
- **Le Dataset FootNote donne des informations sur l'année d'origine des données et leur description. Il n'y a pas de donnée manquante ni de doublon.**
- **Le Dataset StatsSeries donne des données descriptives sur les indicateurs socio économiques étudiés. Il y a cinq colonnes qui ne contiennent que des données manquantes, on pourrait les supprimer sans perdre d'information. Dans les autres colonnes, il y a plus de 50% de données manquantes et il n'y a pas de doublon.**

Bilan sur les données disponibles

- **Le dataset comporte bien des données éducatives**
- **Tous les pays du monde semblent y être intégrés**
- **Il semble y avoir un nombre important de données par pays pour procéder à des analyses comparatives**

Pré Analyse des données

Choix de la période d'étude

In [ ]:

```
#Formatage des noms de Year pour supprimer le YR et n'avoir que l'année, exemple : YR2010 devient 2010
FootNote['Year'] = FootNote['Year'].apply(lambda x : x[2:])
```

In [ ]:

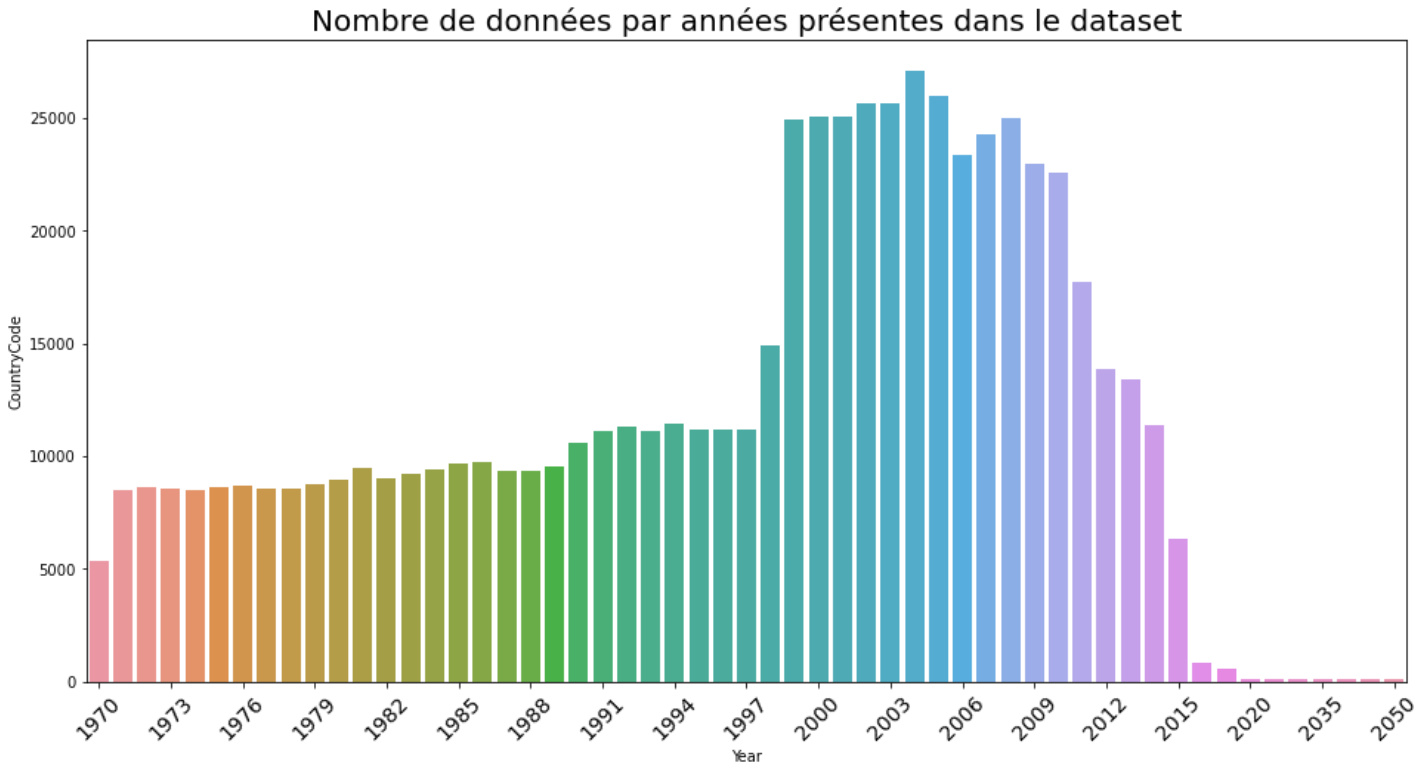
```
FootNote['Year'].sort_values().unique()
```

Out[ ]:

```
array(['1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
       '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985',
       '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993',
       '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001',
       '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009',
       '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017',
       '2020', '2025', '2030', '2035', '2040', '2045', '2050'],
      dtype=object)
```

In [ ]:

```
plt.figure(figsize=(16, 8))
data = FootNote.groupby('Year').count().reset_index()
#data["Year"]=int(data["Year"])
sns.barplot(y = data["CountryCode"], x=data["Year"])
plt.xticks(rotation=45, size=14)
plt.title("Nombre de données par années présentes dans le dataset", size=20)
plt.gca().xaxis.set_major_locator(plt.MultipleLocator(3))
plt.show()
```



On retrouve la majorité des données entre 1998 et 2013. Ce sont donc les données correspondantes à cette période qui sont utilisées pour l'analyse.

Choix des pays pour l'étude

Liste des régions

In [ ]:

```
region = Stats_country["Region"].unique()
region
```

Out[ ]:

```
array(['Latin America & Caribbean', 'South Asia', 'Sub-Saharan Africa',
       'Europe & Central Asia', nan, 'Middle East & North Africa',
       'East Asia & Pacific', 'North America'], dtype=object)
```

On remarque que les pays sont répartis dans différentes régions du monde que l'on retrouve dans la colonne Region du fichier Stats_country. Cependant, on voit qu'il y a 7 régions bien définies et des NaNs. Il faut donc rechercher dans la colonne Long Name à quoi correspond les NaNs. Hypothèse : Des régions ont été mal répertoriées.

Recherche des lignes contenant NaNs comme valeur pour région et céation d'un DataFrame intermédiaire

In [ ]:

```
Liste_indexNAN = Stats_country.loc[pd.isna(Stats_country["Region"]), :].index
Liste_indexNAN
```

Out[ ]:

```
Int64Index([  5,  57,  58,  59,  60,  63,  68,  78,  89,  92, 116, 122, 123,
            124, 127, 128, 140, 143, 148, 157, 166, 168, 187, 198, 200, 225,
            234],
           dtype='int64')
```

In [ ]:

```
Find_Region = pd.DataFrame(columns=["Country Code","Short Name","Table Name","Long Name"
,"2-alpha code","Currency Unit","Special Notes","Region","Income Group","WB-2 code","Nati
onal accounts base year","National accounts reference year","SNA price valuation","Lendin
g category","Other groups","System of National Accounts","Alternative conversion factor",
"PPP survey year","Balance of Payments Manual in use","External debt Reporting status","S
ystem of trade","Government Accounting concept","IMF data dissemination standard","Latest
population census","Latest household survey","Source of most recent Income and expenditur
e data","Vital registration complete","Latest agricultural census","Latest industrial dat
a","Latest trade data","Latest water withdrawal data",])
```

In [ ]:

```
a = Find_Region.append(Stats_country.iloc[5])
b = a.append(Stats_country.iloc[57])
c = b.append(Stats_country.iloc[58])
a = c.append(Stats_country.iloc[59])
b = a.append(Stats_country.iloc[60])
c = b.append(Stats_country.iloc[63])
a = c.append(Stats_country.iloc[68])
b = a.append(Stats_country.iloc[78])
c = b.append(Stats_country.iloc[89])
a = c.append(Stats_country.iloc[92])
b = a.append(Stats_country.iloc[116])
c = b.append(Stats_country.iloc[122])
a = c.append(Stats_country.iloc[123])
b = a.append(Stats_country.iloc[124])
c = b.append(Stats_country.iloc[127])
a = c.append(Stats_country.iloc[128])
b = a.append(Stats_country.iloc[140])
c = b.append(Stats_country.iloc[143])
a = c.append(Stats_country.iloc[148])
b = a.append(Stats_country.iloc[157])
c = b.append(Stats_country.iloc[166])
a = c.append(Stats_country.iloc[168])
```

```
b = a.append(Stats_country.iloc[187])
c = b.append(Stats_country.iloc[198])
a = c.append(Stats_country.iloc[200])
b = a.append(Stats_country.iloc[225])
Find_Region = b.append(Stats_country.iloc[234])
Find_Region.head()
```

Out[ ]:

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Income Group	WB-2 code	...	Government Accounting concept	IMF dissemi sta
5	ARB	Arab World	Arab World	Arab World	1A	NaN	Arab World aggregate. Arab World is composed o...	NaN	NaN	1A	...	NaN	
57	EAP	East Asia & Pacific (developing only)	East Asia & Pacific	East Asia & Pacific (developing only)	4E	NaN	East Asia and Pacific regional aggregate (does...	NaN	NaN	4E	...	NaN	
58	EAS	East Asia & Pacific (all income levels)	East Asia & Pacific (all income levels)	East Asia & Pacific (all income levels)	Z4	NaN	East Asia and Pacific regional aggregate (incl...	NaN	NaN	Z4	...	NaN	
59	ECA	Europe & Central Asia (developing only)	Europe & Central Asia	Europe & Central Asia (developing only)	7E	NaN	Europe and Central Asia regional aggregate (do...	NaN	NaN	7E	...	NaN	
60	ECS	Europe & Central Asia (all income levels)	Europe & Central Asia (all income levels)	Europe & Central Asia (all income levels)	Z7	NaN	Europe and Central Asia regional aggregate (in...	NaN	NaN	Z7	...	NaN	

5 rows × 31 columns

In [ ]:

```
find_region = Find_Region["Short Name"].unique()
find_region
```

Out[ ]:

```
array(['Arab World', 'East Asia & Pacific (developing only)',
       'East Asia & Pacific (all income levels)',
       'Europe & Central Asia (developing only)',
       'Europe & Central Asia (all income levels)', 'Euro area',
       'European Union', 'Gibraltar', 'High income',
       'Heavily indebted poor countries (HIPC)',
       'Latin America & Caribbean (developing only)',
       'Latin America & Caribbean (all income levels)',
       'Least developed countries: UN classification', 'Low income',
       'Lower middle income', 'Low & middle income',
       'Middle East & North Africa (all income levels)', 'Middle income',
       'Middle East & North Africa (developing only)', 'North America',
```

```
       'Nauru', 'OECD members', 'South Asia',
       'Sub-Saharan Africa (developing only)',
       'Sub-Saharan Africa (all income levels)', 'Upper middle income',
       'World'], dtype=object)
```

On voit qu'il y a, dans region et find region, un certain nombre de noms qui correspondent en fait à des zones. Une liste areas est créée et contient les différentes régions du monde représentées.

In [ ]:

```
areas = ['Arab World',
         'East Asia & Pacific',
         'Europe & Central Asia',
         'Latin America & Caribbean',
         'Middle East & North Africa',
         'North America',
         'South Asia',
         'Sub-Saharan Africa'
        ]
```

In [ ]:

```
len(areas)
```

Out[ ]:

```
8
```

On peut enlever certaines régions car elles ne permettent pas de donner des informations précises : Euro area','European Union', 'OECD members','Heavily indebted poor countries (HIPC)', 'High income', 'Least developed countries: UN classification', 'Low & middle income','Low income','Lower middle income', 'Middle income','Upper middle income', 'World'.

Répartition des pays selon leur région, leur devise et leur Revenu

Répartition des pays selon leur région

Comment sont répartis les pays étudiés dans les différentes zones du monde. Quelles sont les zones les plus représentées ?

In [ ]:

```
y = Stats_country.groupby('Region')[['Short Name']].count()['Short Name'].sort_values(as
cending=False)
y
```

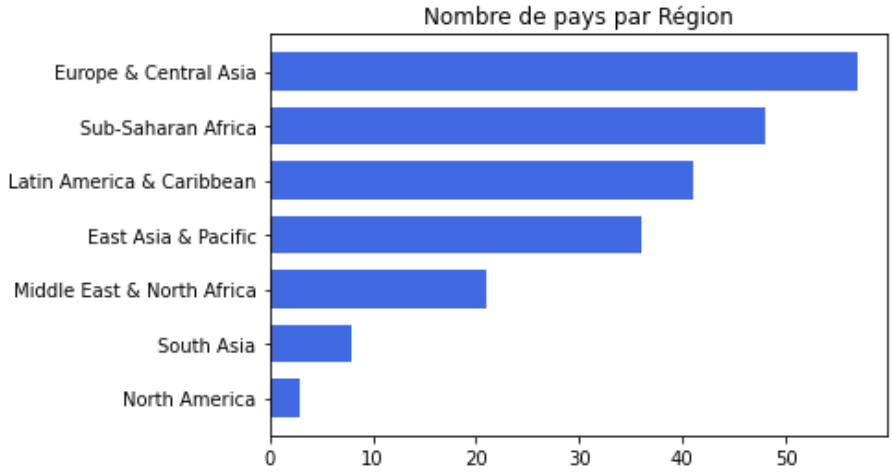
Out[ ]:

```
Region
Europe & Central Asia         57
Sub-Saharan Africa            48
Latin America & Caribbean     41
East Asia & Pacific           36
Middle East & North Africa    21
South Asia                     8
North America                  3
Name: Short Name, dtype: int64
```

In [ ]:

```
plt.figure
plt.barh(y = Stats_country.groupby('Region')[['Short Name']].count().reset_index().sort_
values(by='Short Name')['Region'], width = Stats_country.groupby("Region")[["Short Name"
]].count()["Short Name"].sort_values(), height=0.7, align='center', color = "royalblue",
linewidth = 0.8)
```

```
ax = plt.gca()
plt.title('Nombre de pays par Région')
plt.show()
```



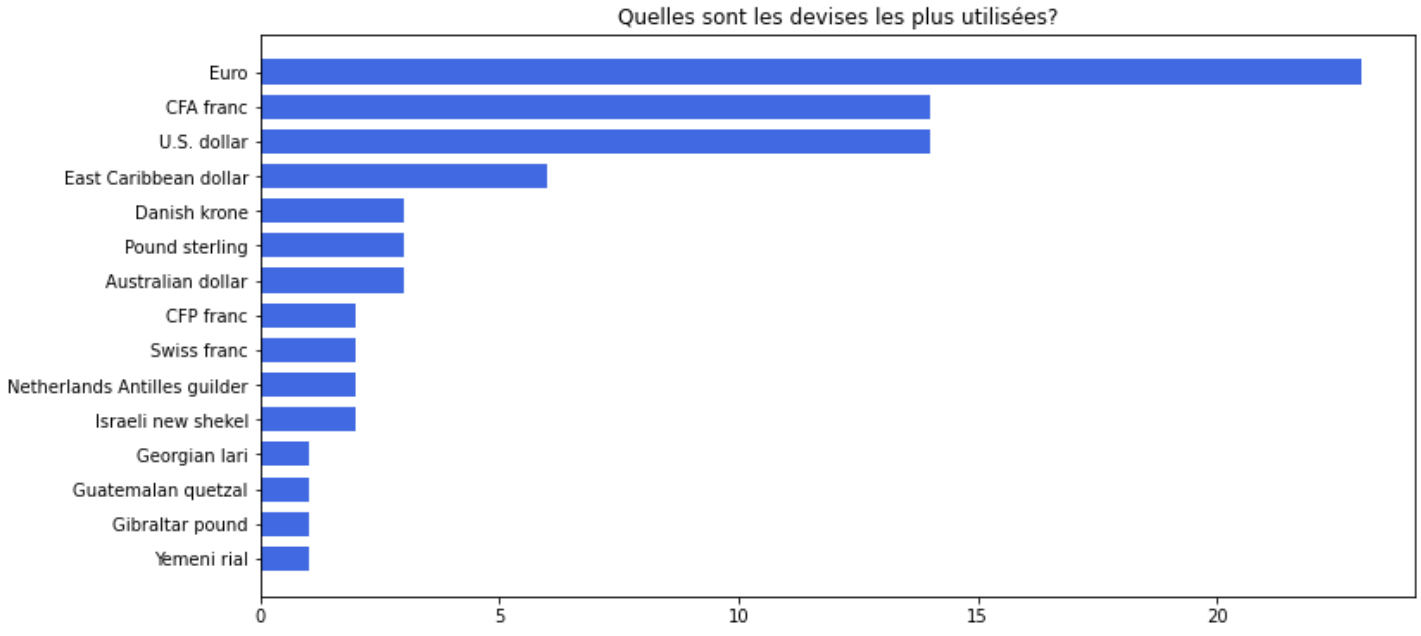
Dans cette étude, Europe & Central Asia, Sub-Saharan Africa, Latin America & Caribbea, East Asia & Pacific sont les régions les plus représentées.

Répartition des pays selon leur devise

Quelle sont les devises les plus représentées ?

In [ ]:

```
plt.figure(figsize=(12,6))
plt.barh(y = Stats_country.groupby('Currency Unit')[['Short Name']].count().reset_index(
).sort_values(by='Short Name')['Currency Unit'].tail(15), width = Stats_country.groupby(
'Currency Unit')[['Short Name']].count()['Short Name'].sort_values().tail(15), height=0.
7, align='center', color = "royalblue")
plt.title('Quelles sont les devises les plus utilisées?')
plt.show()
```



L'Euro est fortement majoritaire au sein des pays étudiés.

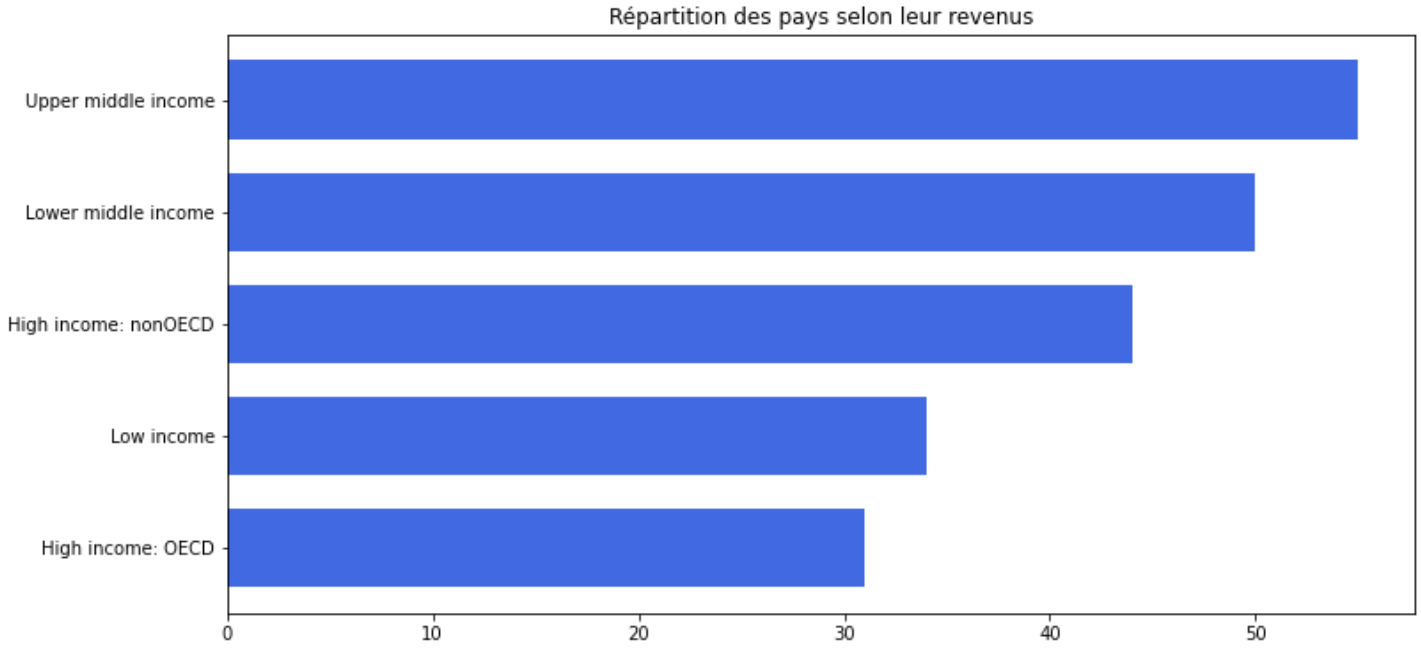
Répartition des pays selon leur Revenu

Quels sont les niveaux économiques les plus représentés dans cette étude ?

In [ ]:

```
plt.figure(figsize=(12,6))
plt.barh(y = Stats_country.groupby('Income Group')[['Short Name']].count().reset_index()
.sort_values(by='Short Name')['Income Group'], width = Stats_country.groupby("Income Gro
up")[["Short Name"]].count()["Short Name"].sort_values(), height=0.7, align='center', co
lor = "royalblue",linewidth = 1)
ax = plt.gca()
plt.title('Répartition des pays selon leur revenus')
plt.show()

# , edgecolor = "green"
```



Les pays les plus représentés dans l'étude ont des revenus corrects (moyen).

Filtration des pays

Liste des pays

Quels sont les pays qui se trouvent dans les zones géographiques précédement étudiées ?

In [ ]:

```
Stats_country[~Stats_country['Short Name'].isin(areas)]['Short Name'].unique().shape
```

Out[ ]:

```
(238,)
```

Les pays répartis dans les zones géographiques sont au nombre de 228.

In [ ]:

```
List_countries = Stats_country[~Stats_country['Short Name'].isin(areas)]['Short Name'].u
nique().tolist()
#list(List_countries)
```

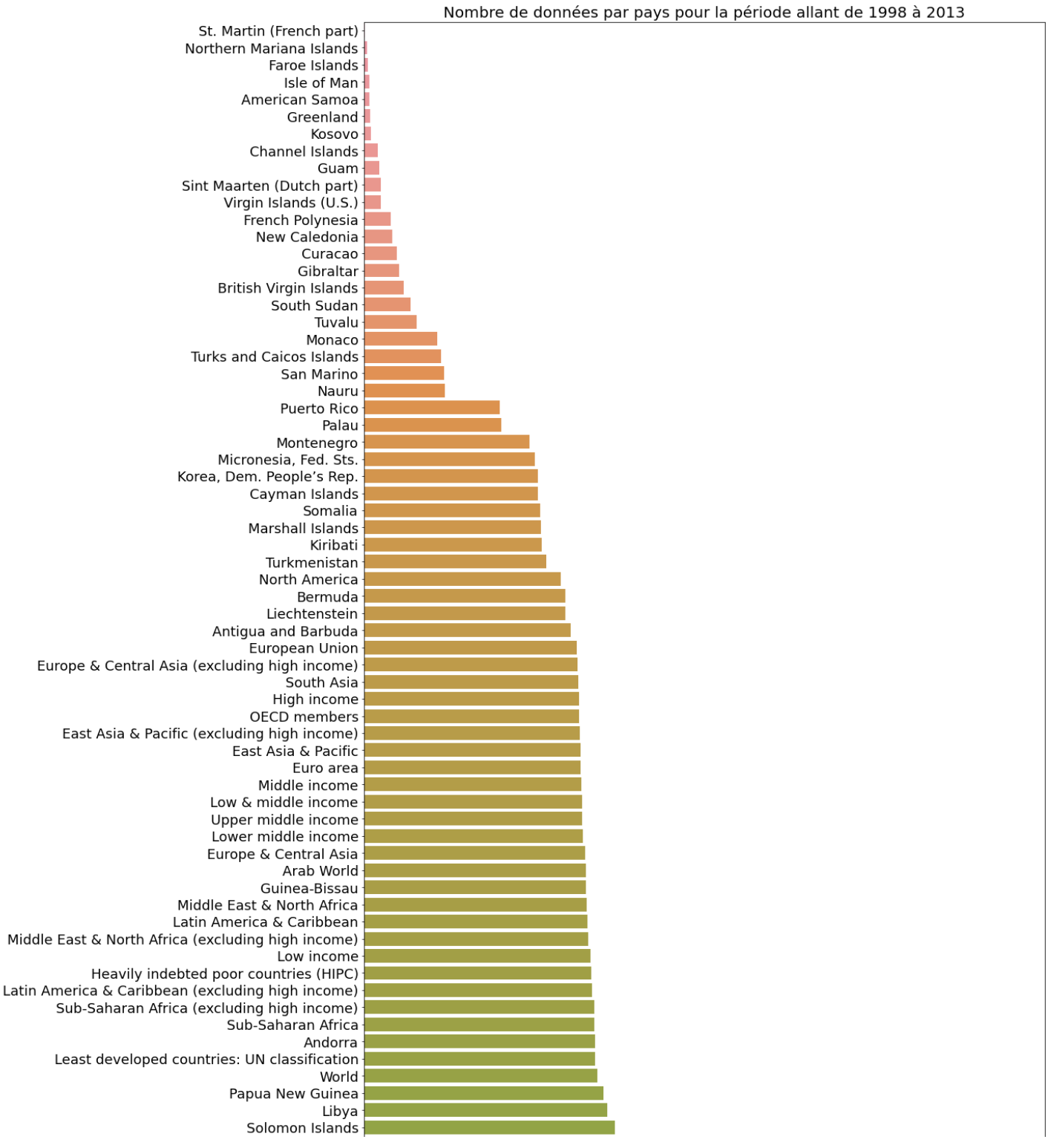
Il y a en tout 228 pays étudiés et on voit que certains "pays" sont des subdivisions d'autres pays (tel que : St Martin, Isle of Man, French polynesia, etc.)

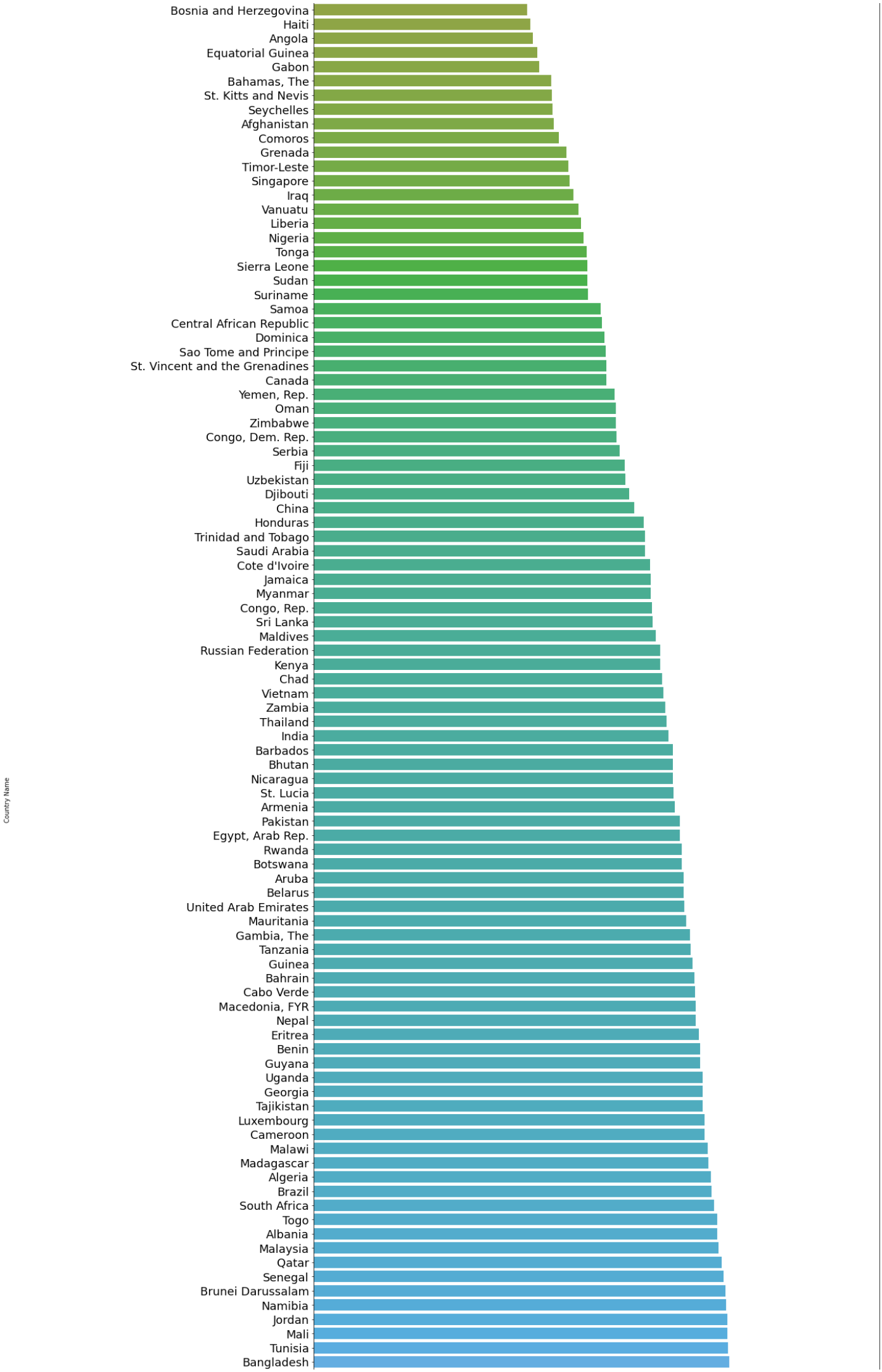
Filtration du nombre de pays

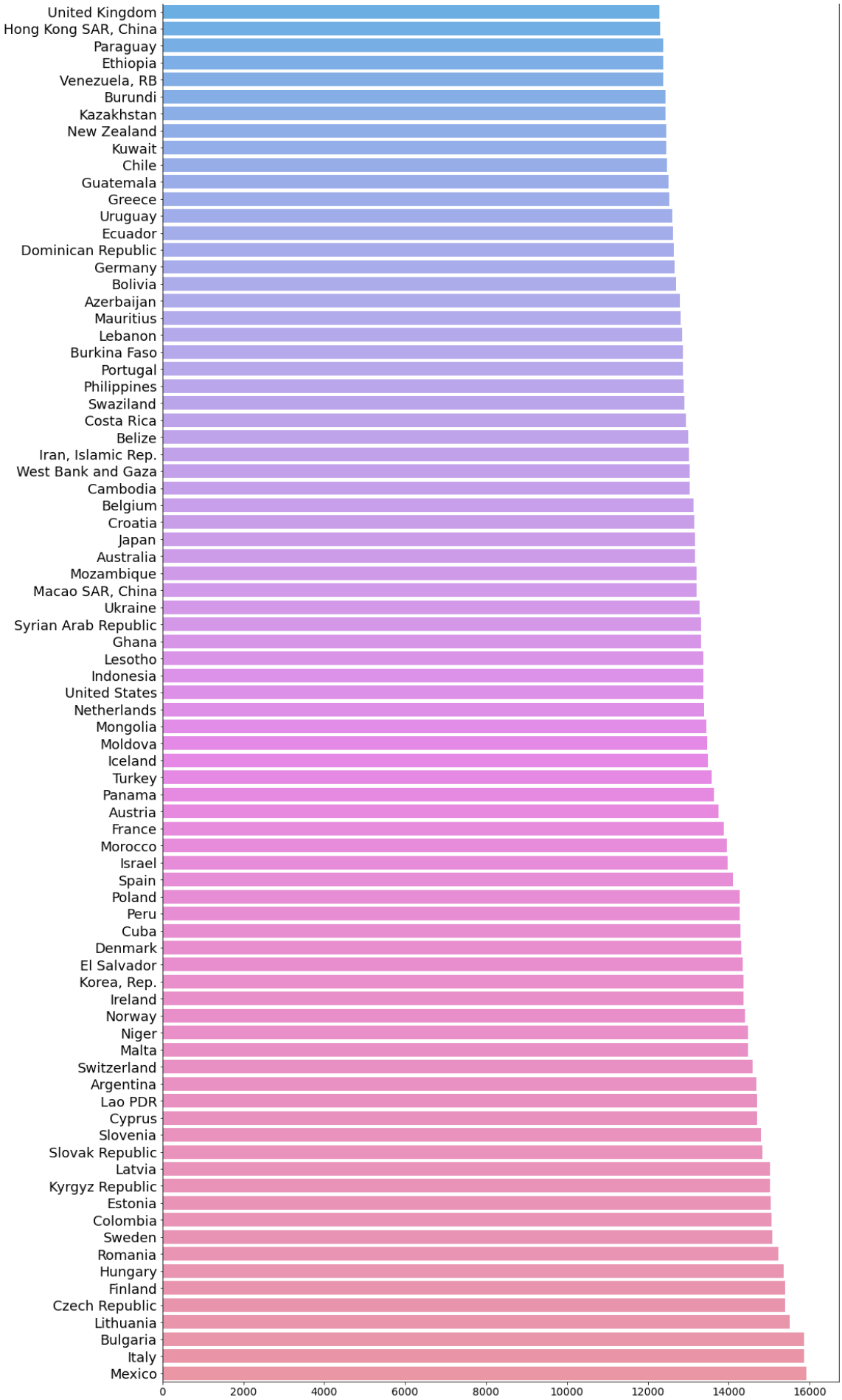
Quels sont les pays que l'on peut supprimer de l'étude pour affiner nos recherches ?

In [ ]:

```
plt.figure(figsize=(16, 100))
plt.title("Nombre de données par pays pour la période allant de 1998 à 2013", size=20)
data = Stats_Data
sns.barplot(x = data.groupby('Country Name')[['1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005',
       '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013']].count().sum(axis=1).sort_values(ascending=True).values,
                  y = data.groupby('Country Name')[['1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005',
       '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013']].count().sum(axis=1).sort_values(ascending=True).index)
plt.yticks(fontsize=18)
plt.xticks(fontsize=14)
plt.show()
```




Country Name
Bosnia and Herzegovina
Haiti
Angola
Equatorial Guinea
Gabon
Bahamas, The
St. Kitts and Nevis
Seychelles
Afghanistan
Comoros
Grenada
Timor-Leste
Singapore
Iraq
Vanuatu
Liberia
Nigeria
Tonga
Sierra Leone
Sudan
Suriname
Samoa
Central African Republic
Dominica
Sao Tome and Principe
St. Vincent and the Grenadines
Canada
Yemen, Rep.
Oman
Zimbabwe
Congo, Dem. Rep.
Serbia
Fiji
Uzbekistan
Djibouti
China
Honduras
Trinidad and Tobago
Saudi Arabia
Cote d'Ivoire
Jamaica
Myanmar
Congo, Rep.
Sri Lanka
Maldives
Russian Federation
Kenya
Chad
Vietnam
Zambia
Thailand
India
Barbados
Bhutan
Nicaragua
St. Lucia
Armenia
Pakistan
Egypt, Arab Rep.
Rwanda
Botswana
Aruba
Belarus
United Arab Emirates
Mauritania
Gambia, The
Tanzania
Guinea
Bahrain
Cabo Verde
Macedonia, FYR
Nepal
Eritrea
Benin
Guyana
Uganda
Georgia
Tajikistan
Luxembourg
Cameroon
Malawi
Madagascar
Algeria
Brazil
South Africa
Togo
Albania
Malaysia
Qatar
Senegal
Brunei Darussalam
Namibia
Jordan
Mali
Tunisia
Bangladesh



In [ ]:

```
to_remove = (data.groupby('Country Name')[['1998', '1999', '2000', '2001', '2002', '2003
', '2004', '2005',
       '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013']].count().sum(axis
=1).sort_values(ascending=True).values)<6000
to_remove = to_remove.tolist()
to_remove.count(True)
```

Out[ ]:

Out[ ]:

```
64
```

Les 64 premiers pays (lorsque le classement des pays selon le nombre de données est croissant) sont ceux qui ont moins de 6000 données sur la période d'étude de 1998 à 2013.

In [ ]:

```
pays_total = data.groupby('Country Name')[['1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005',
       '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013']].count().sum(axis=1).sort_values(ascending=True).index.tolist()
remove = pays_total[:64]
#remove
```

Les pays pour lesquels on a le moins d'informations (moins de 6000 données pour la période d'étude choisie) sont :

- **Les petits pays**
- **Les nouveaux pays (kosovo)**
- **Les régions de certains pays : st martin, french polynésia, ...**

Je décide de supprimer certains pays où il serait difficile de développer notre activité au vu de la situation du pays (démographie, économique, éducatif, ...).

Je décide aussi de supprimer les régions de certains pays (st martin, french polynésia, ...).

Je décide aussi de supprimer des catégories qui ne sont pas des pays et qui ne nous donne pas d'information précise (World, Least developed countries: UN classification, Low income, Lower middle income, Low & middle income).

In [ ]:

```
countries = ['Albania', 'United Arab Emirates', 'Argentina', 'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Burundi', 'Belgium', 'Bulgaria', 'Bahrain', 'The Bahamas', 'Bosnia and Herzegovina', 'Belarus', 'Belize', 'Bolivia', 'Brazil', 'Barbados', 'Bhutan', 'Canada', 'Switzerland', 'Chile', 'China', "Côte d'Ivoire", 'Colombia', 'Costa Rica', 'Cuba', 'Cyprus', 'Czech Republic', 'Germany', 'Denmark', 'Dominican Republic', 'Algeria', 'Ecuador', 'Egypt', 'Eritrea', 'Spain', 'Estonia', 'Finland', 'France', 'United Kingdom', 'Georgia', 'Guinea', 'Greece', 'Grenada', 'Guatemala', 'Guyana', 'Hong Kong SAR, China', 'Honduras', 'Croatia', 'Hungary', 'Indonesia', 'India', 'Ireland', 'Iran', 'Iraq', 'Iceland', 'Israel', 'Italy', 'Jamaica', 'Jordan', 'Japan', 'Kenya', 'Cambodia', 'Korea', 'Kuwait', 'Lao PDR', 'Lebanon', 'Sri Lanka', 'Lesotho', 'Lithuania', 'Luxembourg', 'Latvia', 'Morocco', 'Moldova', 'Madagascar', 'Mexico', 'Macedonia', 'Malta', 'Myanmar', 'Mongolia', 'Mauritius', 'Malaysia', 'Niger', 'Nigeria', 'Nicaragua', 'Netherlands', 'Norway', 'New Zealand', 'Oman', 'Pakistan', 'Panama', 'Peru', 'Philippines', 'Poland', "Dem. People's Rep. Korea", 'Portugal', 'Paraguay', 'Qatar', 'Romania', 'Russia', 'Saudi Arabia', 'Sudan', 'Senegal', 'Singapore', 'Serbia', 'Suriname', 'Slovak Republic', 'Slovenia', 'Sweden', 'Chad', 'Togo', 'Thailand', 'Tunisia', 'Turkey', 'Tanzania', 'Uganda', 'Ukraine','Uruguay', 'United States', 'Venezuela', 'Vietnam', 'Yemen', 'South Africa']
#list(countries)
```

In [ ]:

```
len(countries)
```

Out[ ]:

```
124
```

Après cette première filtration, il reste 124 pays (contenus dans countries) sur les 228 proposés.

Création d'un nouveau Dataset de façon à avoir une moyenne des données pour la période étudiée et les régions pour chaque pays.

In [ ]:

```
Data_Study = Stats_Data.merge(right = Stats_country[['Country Code', 'Region']],
                  on='Country Code', how='left')
```

In [ ]:

```
Data_Study["Study_years"] = Stats_Data[[str(year) for year in [1999,2010]]].mean(1)
Data_Study = Data_Study[["Country Name", "Country Code", "Indicator Name", "Indicator Cod
e", "Region", "Study_years"]]
Data_Study
```

Out[ ]:

	Country Name	Country Code	Indicator Name	Indicator Code	Region	Study_years
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN
2	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.GPI	NaN	NaN
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN
4	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	NaN	80.733158
...	...	...	...	...	...	...
886925	Zimbabwe	ZWE	Youth illiterate population, 15-24 years, male...	UIS.LP.AG15T24.M	Sub-Saharan Africa	NaN
886926	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, b...	SE.ADT.1524.LT.ZS	Sub-Saharan Africa	NaN
886927	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, f...	SE.ADT.1524.LT.FE.ZS	Sub-Saharan Africa	NaN
886928	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, g...	SE.ADT.1524.LT.FM.ZS	Sub-Saharan Africa	NaN
886929	Zimbabwe	ZWE	Youth literacy rate, population 15-24 years, m...	SE.ADT.1524.LT.MA.ZS	Sub-Saharan Africa	NaN

886930 rows × 6 columns

Pour faciliter notre étude et pour une analyse plus fiable, les données de 1998 à 2013 sont regroupées dans une variable Study_years sous forme de moyenne. De plus, chaque pays a sa région grâce à la concaténation de deux fichiers.

Choix des indicateurs pour l'étude

Liste des indicateurs

In [ ]:

```
List_indicator = Stats_Data["Indicator Name"].unique()
#list(List_indicator)
```

In [ ]:

```
Stats_Data["Indicator Name"].nunique()
```

Out[ ]:

```
3665
```

Pour notre étude nous avons à notre disposition 3665 indicateurs.

In [ ]:

```
x=Stats_Data.groupby('Country Name')[['Indicator Code']].count().reset_index()
dico = x.set_index('Country Name').T.to_dict('list')
```

In [ ]:

```
for k,v in dico.items():
    if v != [3665]:
        print("Le {0} ne traite pas la totalité des indicateurs, il en traite {1}".format
(k,v))
print("Si vous ne voyez pas de sortie, celà signifie que tous les pays étudiés traitent l
es 3665 indicateurs")
```

```
Si vous ne voyez pas de sortie, celà signifie que tous les pays étudiés traitent les 3665
indicateurs
```

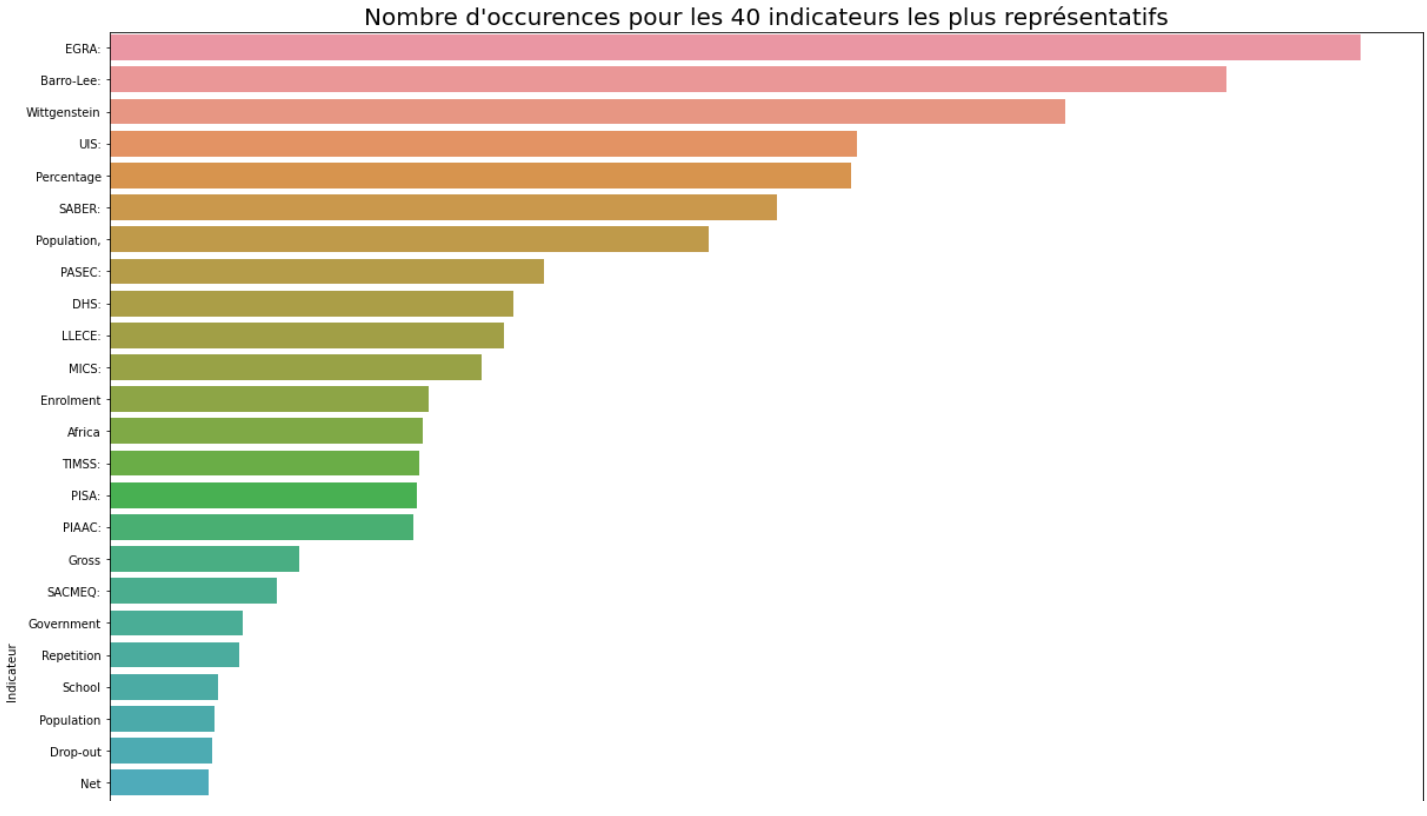
Tout les pays sélectionnés précédement traitent les 3665 indicateurs (mais il peut y avoir des données manquantes).

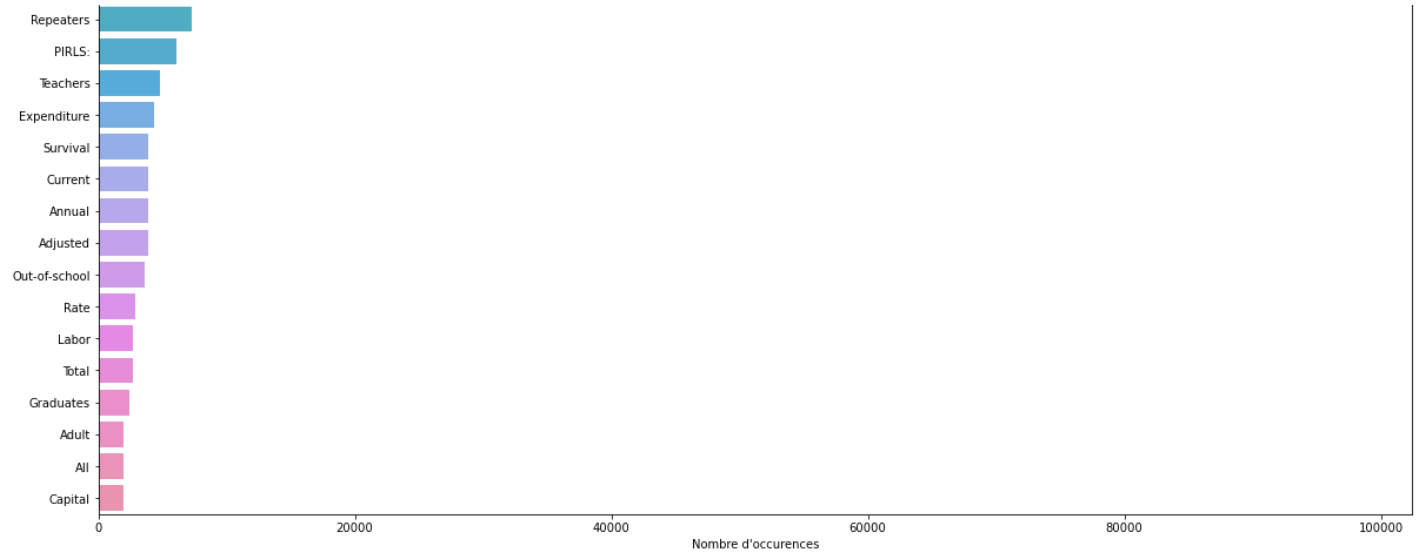
Quels sont les indicateurs les plus utilisés ?

In [ ]:

```
liste_indicateurs = [i.split(' ', 1)[0] for i in Stats_Data['Indicator Name'].tolist()]
indicateurs_populaires = collections.Counter(liste_indicateurs).most_common(40)
DataFrame_indicateurs_populaires = pd.DataFrame(indicateurs_populaires, columns = ['Indi
cateur', 'Nombre Occurences']).sort_values(by='Nombre Occurences', ascending=False)

plt.figure(figsize=(20, 20))
sns.barplot(y = DataFrame_indicateurs_populaires["Indicateur"], x=DataFrame_indicateurs_
populaires["Nombre Occurences"])
plt.title("Nombre d\'occurences pour les 40 indicateurs les plus représentatifs", size=20
)
plt.xlabel('Nombre d\'occurences')
plt.show()
```





Les indicateurs que l'on retrouve le plus avec un grand nombre d'occurences sont en relation avec l'éducation.

Parmis les 40 occurences les plus représentées dans ce jeu de données, 10 indicateurs sont en rapport avec l'éducation :

- **EGRA : Early Grade Reading Assessment (évaluations internationales de l'apprentissage, lecture);**
- **Barro-lee : Dataset relatif à l'éducation**
- **Wittgenstein (Wittgenstein Centre Human Capital Data Explore)**
- **UIS : UNESCO Institut de Statistiques**
- **PISA : Tests comparatifs de compétences pour les élèves**
- **school**
- **Teachers;**
- **out of school**
- **graduate.**
- **Net : L'accès à internet**

Quels sont les indicateurs avec le plus de données ?

In [ ]:

```
Data_Study[["Indicator Name", "Study_years"]].groupby("Indicator Name").count().sort_values(by="Study_years", ascending=False)
```

Out[ ]:

Indicator Name	Study_years
Population, total	240
Population growth (annual %)	240
GDP per capita (current US$)	232
GDP at market prices (current US$)	232
Internet users (per 100 people)	229
...	...
EGRA: Oral Reading Fluency - Share of students with a zero score (%). Songhoi. 2nd Grade	0
EGRA: Oral Reading Fluency - Share of students with a zero score (%). Spanish. 2nd Grade	0
EGRA: Correct Non-Words Read Per Minute (Mean). Nzema. 2nd Grade	0
EGRA: Correct Non-Words Read Per Minute (Mean). Luvale. 2nd Grade	0
PASEC: Mean performance on the mathematics scale for 6th grade students. Male	0

Indicator Name	Study_years

3665 rows × 1 columns

Indicateurs dont nous avons besoin pour répondre à notre étude :

- **indicateur pour quantifier l'utilisation d'internet par pays**
- **indicateur pour quantifier le nombre d'etudiants / lycéens**

Indicateurs choisis

En observant les indicateurs avec le plus de données, on peut déterminer 4 indicateurs pour la suite de l'étude.

indicateur précis

education

- **SE.SEC.ENRL** = Enrolment in secondary education, both sexes (number)
- **SE.TER.ENRL** = Enrolment in tertiary education, all programmes, both sexes (number)

démographie

- **SP.POP.1564.TO** = population, age 15-64, total : De nos jours, dans le monde du travail, il est essentiel de se former tout au long de sa carrière. C'est pourquoi notre service de cours en ligne ne touche pas seulement les personnes qui se trouvent dans le secondaire ou le tertiaire.

Utilisation internet

- **IT.NET.USER.P2** = internet user (per 100 people)

Nombre de données pour chaque indicateur choisi

In [ ]:

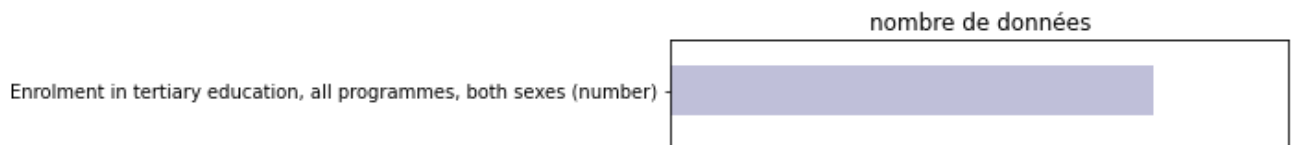
```
indicateurs = ["SE.SEC.ENRL", "SE.TER.ENRL", "SP.POP.1564.TO", "IT.NET.USER.P2"]
df = Data_Study[Data_Study["Indicator Code"].isin(indicateurs)][["Indicator Name", "Indicator Code", "Study_years"]].groupby(['Indicator Name', 'Indicator Code']).count().reset_index().sort_values(by='Study_years',ascending=False)
df
```

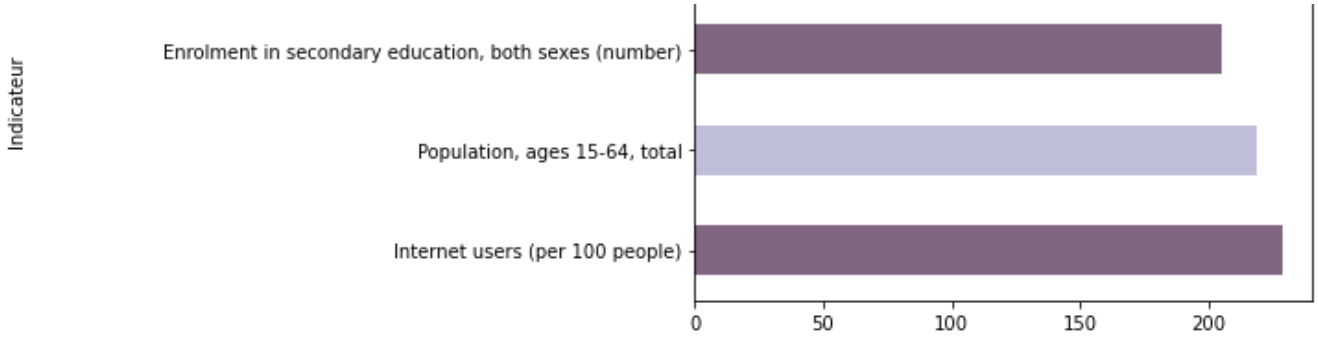
Out[ ]:

	Indicator Name	Indicator Code	Study_years
2	Internet users (per 100 people)	IT.NET.USER.P2	229
3	Population, ages 15-64, total	SP.POP.1564.TO	219
0	Enrolment in secondary education, both sexes (...	SE.SEC.ENRL	205
1	Enrolment in tertiary education, all programme...	SE.TER.ENRL	188

In [ ]:

```
my_colors = [(0.5,0.4,0.5), (0.75, 0.75, 0.85)]
df.plot.barh(x='Indicator Name', y='Study_years', title="nombre de données", xlabel='Indicateur', color=my_colors, legend=False)
plt.show()
```





Les 4 indicateurs choisis ont environ 200 données chacun.

Création de deux Dataset finaux filtrés en fonction de nos indicateurs choisis.

Récupération des lignes traitant uniquement nos indicateurs et en fonction des régions

In [ ]:

```
data_areas = Data_Study[Data_Study['Country Name'].isin(areas)][Data_Study['Indicator Code'].isin(indicateurs)]
#data_areas.head()
data_areas["Indicator Name"] = data_areas["Indicator Name"].apply(lambda x: x.replace("Enrolment in secondary education, both sexes (number)", "Secondary education"))
data_areas["Indicator Name"] = data_areas["Indicator Name"].apply(lambda x: x.replace("Enrolment in tertiary education, all programmes, both sexes (number)","Tertiary education"))
data_areas["Indicator Name"] = data_areas["Indicator Name"].apply(lambda x: x.replace("Internet users (per 100 people)", "Internet users"))
data_areas["Indicator Name"] = data_areas["Indicator Name"].apply(lambda x: x.replace("Population, ages 15-64, total", "Population (15-64)"))
data_areas.head()
```

Out[ ]:

	Country Name	Country Code	Indicator Name	Indicator Code	Region	Study_years
1191	Arab World	ARB	Secondary education	SE.SEC.ENRL	NaN	2.626794e+07
1204	Arab World	ARB	Tertiary education	SE.TER.ENRL	NaN	6.795613e+06
1375	Arab World	ARB	Internet users	IT.NET.USER.P2	NaN	1.367292e+01
2486	Arab World	ARB	Population (15-64)	SP.POP.1564.TO	NaN	1.868050e+08
4856	East Asia & Pacific	EAS	Secondary education	SE.SEC.ENRL	NaN	1.475797e+08

Récupération des lignes traitant uniquement nos indicateurs et en fonction des pays choisis

In [ ]:

```
data_countries = Data_Study[Data_Study['Country Name'].isin(countries)][Data_Study['Indicator Code'].isin(indicateurs)]
data_countries["Indicator Name"] = data_countries["Indicator Name"].apply(lambda x: x.replace("Enrolment in secondary education, both sexes (number)", "Secondary education"))
data_countries["Indicator Name"] = data_countries["Indicator Name"].apply(lambda x: x.replace("Enrolment in tertiary education, all programmes, both sexes (number)","Tertiary education"))
data_countries["Indicator Name"] = data_countries["Indicator Name"].apply(lambda x: x.replace("Internet users (per 100 people)", "Internet users"))
data_countries["Indicator Name"] = data_countries["Indicator Name"].apply(lambda x: x.replace("Population, ages 15-64, total", "Population (15-64)"))
data_countries.head()
```

Out[ ]:

	Country Name	Country Code	Indicator Name	Indicator Code	Region	Study_years
96481	Albania	ALB	Secondary education	SE.SEC.ENRL	Europe & Central Asia	3.596865e+05
96494	Albania	ALB	Tertiary education	SE.TER.ENRL	Europe & Central Asia	8.041400e+04
96665	Albania	ALB	Internet users	IT.NET.USER.P2	Europe & Central Asia	2.254072e+01
97776	Albania	ALB	Population (15-64)	SP.POP.1564.TO	Europe & Central Asia	1.941948e+06
100146	Algeria	DZA	Secondary education	SE.SEC.ENRL	Middle East & North Africa	3.800576e+06

Analyse

Analyse de nos indicateurs.

Analyse statistique sans distinstion de pays ou de région.

In [ ]:

```
Data_Study[Data_Study["Indicator Code"].isin(indicateurs)][["Indicator Name", "Indicator Code", "Study_years"]].groupby(['Indicator Name', 'Indicator Code']).describe()
```

Out[ ]:

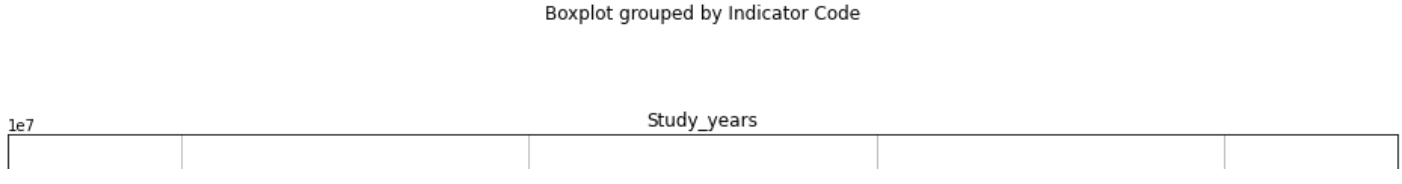
		Study_years							
		count	mean	std	min	25%	50%	75%	max
Indicator Name	Indicator Code								
Enrolment in secondary education, both sexes (number)	SE.SEC.ENRL	205.0	1.602213e+07	5.782920e+07	1114.0	1.056080e+05	5.915490e+05	3.800576e+06	4.916
Enrolment in tertiary education, all programmes, both sexes (number)	SE.TER.ENRL	188.0	4.770593e+06	1.583654e+07	5.0	2.362825e+04	1.955342e+05	1.310947e+06	1.381
Internet users (per 100 people)	IT.NET.USER.P2	229.0	2.057119e+01	1.853226e+01	0.0	4.220329e+00	1.614744e+01	3.307811e+01	8.000
Population, ages 15-64, total	SP.POP.1564.TO	219.0	1.275306e+08	4.760221e+08	54314.0	1.678121e+06	5.298222e+06	2.558678e+07	4.160

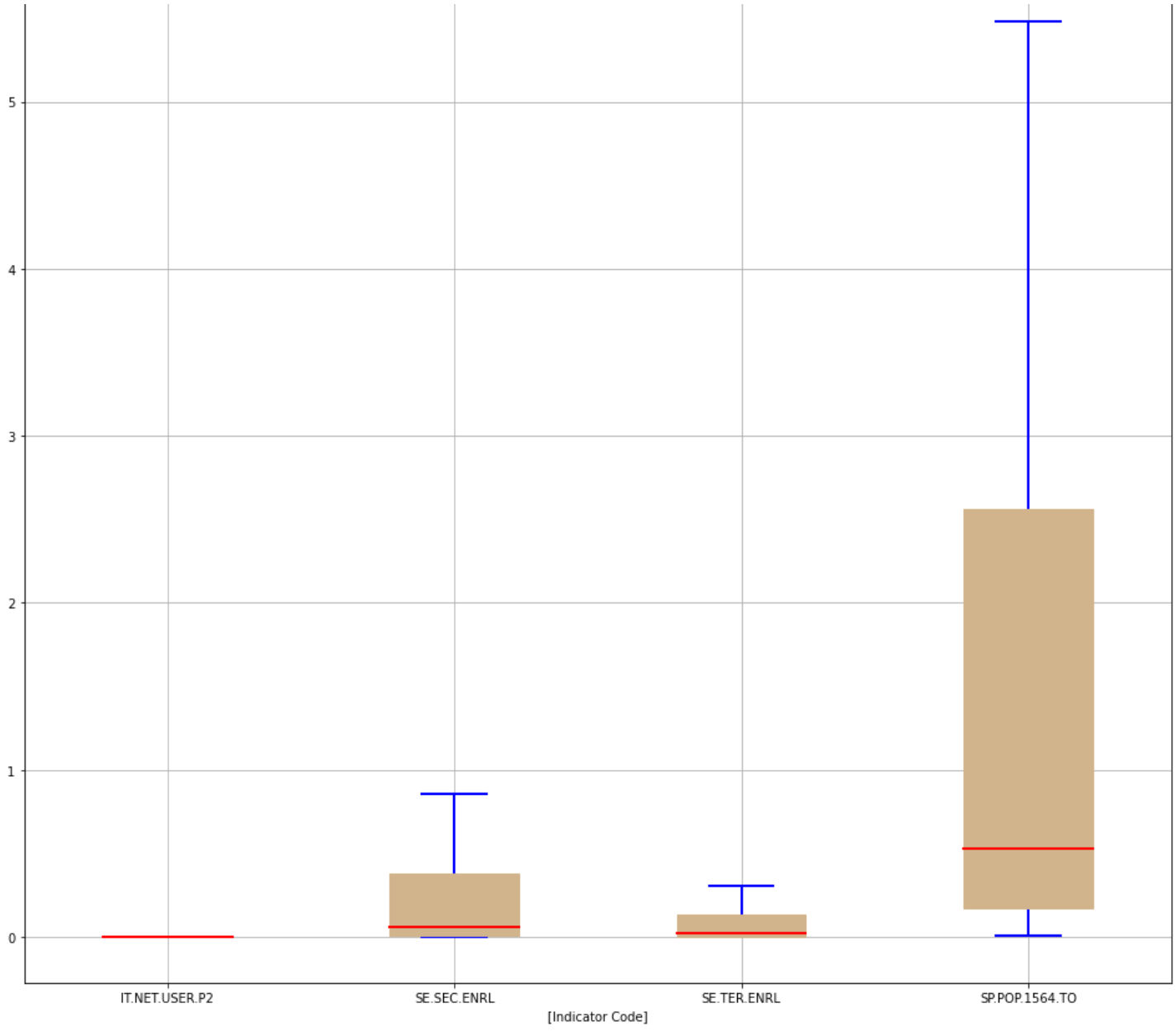
In [ ]:

```
graphe = Data_Study[Data_Study['Indicator Code'].isin(indicateurs)][['Indicator Code', 'Study_years']]
graphe.boxplot(by='Indicator Code', showfliers=False, figsize=(16,15), showcaps=True, patch_artist=True, color='tan',medianprops={'linestyle': '-', 'linewidth': 2, 'color': 'red',}, whiskerprops={'linestyle': '-', 'linewidth': 2, 'color' : 'blue'}, capprops={'linestyle': '-', 'linewidth': 2, 'color':'blue'})
```

Out[ ]:

```
<AxesSubplot:title={'center':'Study_years'}, xlabel='[Indicator Code]'>
```





On remarque que l'indicateur qui représente la population entre 15 et 64 ans, a une distribution beaucoup plus dispersée que les autres indicateurs. Hypothèse : Les personnes inscrites dans le secondaire et le tertiaire ne représentent qu'une infime partie des personnes qui ont entre 15 et 64 ans. L'indicateur sur l'utilisation d'internet étant représenté en pourcentage (données sur une base de 100 personnes), il n'est pas possible de le comparer aux trois autres indicateurs.

In [ ]:

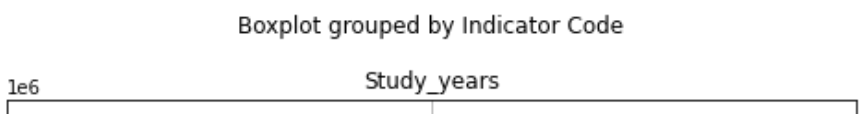
```
SEC = ["SE.SEC.ENRL"]
TER = ["SE.TER.ENRL"]
NET = ["IT.NET.USER.P2"]
POP = ["SP.POP.1564.TO"]
```

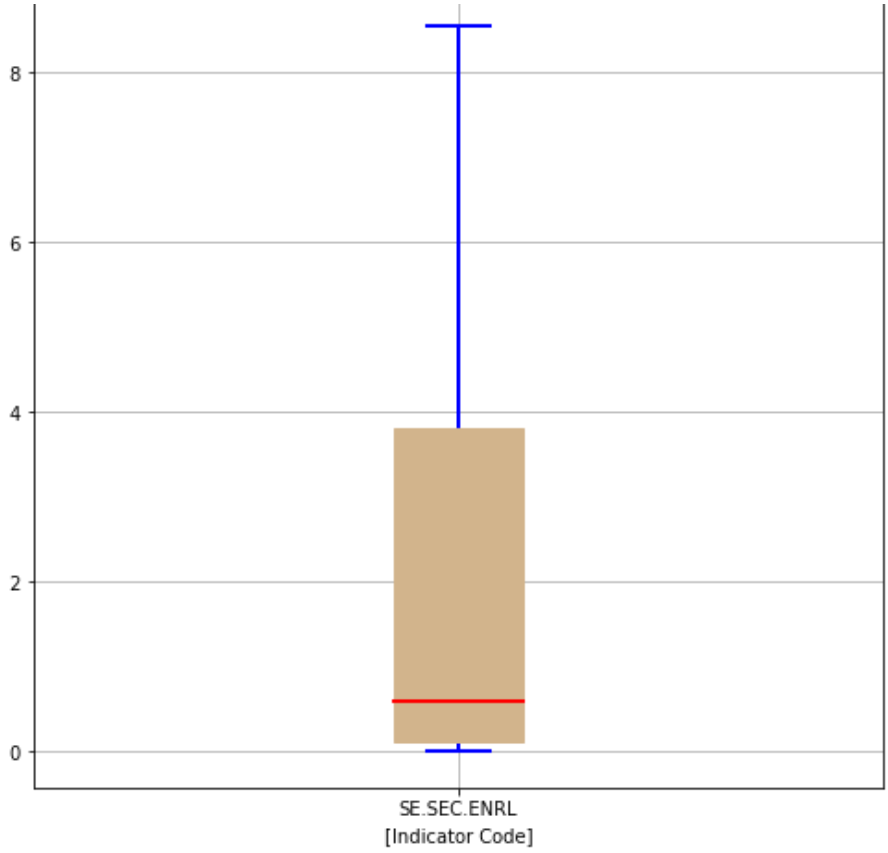
In [ ]:

```
graphe = Data_Study[Data_Study['Indicator Code'].isin(SEC)][['Indicator Code', 'Study_years']]
graphe.boxplot(by='Indicator Code', showfliers=False, figsize=(8, 8), showcaps=True, patch_artist=True, color='tan',medianprops={'linestyle': '-', 'linewidth': 2, 'color': 'red',}, whiskerprops={'linestyle': '-', 'linewidth': 2, 'color' : 'blue'}, capprops={'linestyle': '-', 'linewidth': 2, 'color':'blue'})
```

Out[ ]:

```
<AxesSubplot:title={'center':'Study_years'}, xlabel='[Indicator Code]'>
```





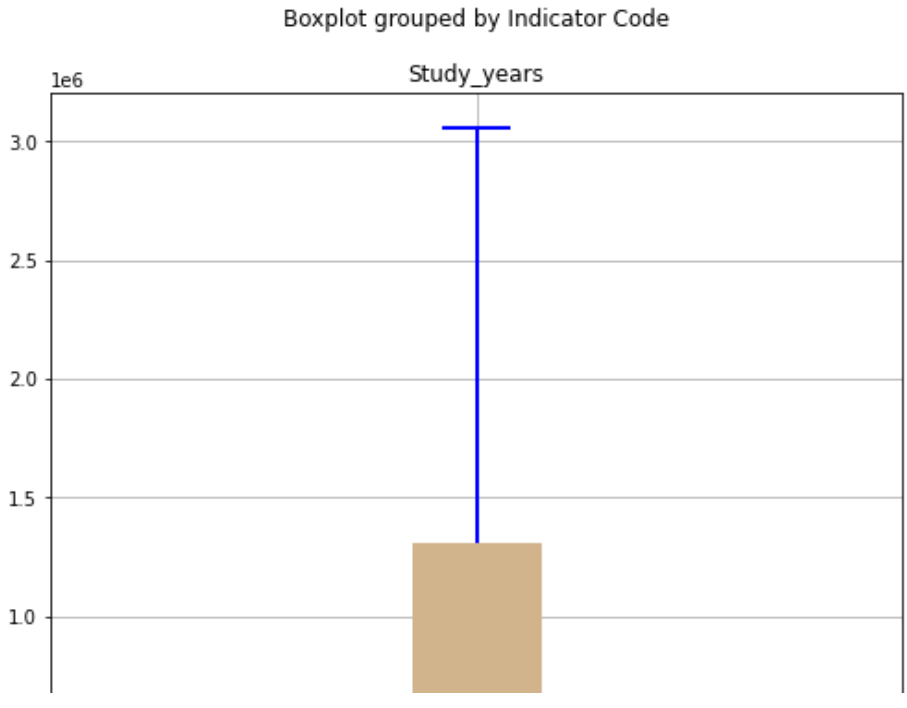
- **Environ 25% des pays étudiés ont moins de 23 628 personnes qui étudient dans le secondaire (il s'agit de pays défavorisés) et environ 75% des pays étudiés ont plus de 23 628 personnes qui étudient dans le secondaire (25% ont plus de 3 800 576 personnes étudiant dans le secondaire, il s'agit de pays développés).**
- **Environ 50% des pays étudiés ont moins de 591 549 personnes qui étudient dans le secondaire.**

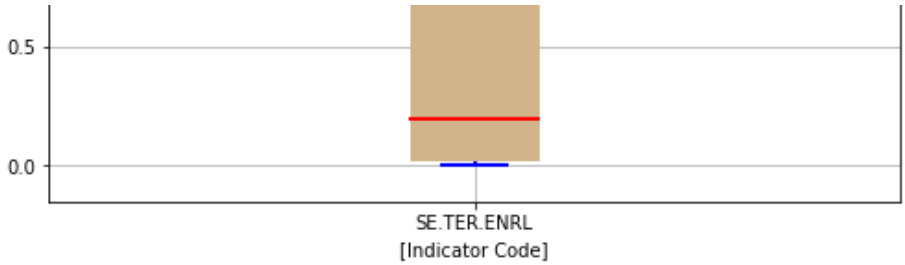
In [ ]:

```
graphe = Data_Study[Data_Study['Indicator Code'].isin(TER)][['Indicator Code', 'Study_years']]
graphe.boxplot(by='Indicator Code', showfliers=False, figsize=(8, 8), showcaps=True, patch_artist=True, color='tan',medianprops={'linestyle': '-', 'linewidth': 2, 'color': 'red',}, whiskerprops={'linestyle': '-', 'linewidth': 2, 'color' : 'blue'}, capprops={'linestyle': '-', 'linewidth': 2, 'color':'blue'})
```

Out[ ]:

```
<AxesSubplot:title={'center':'Study_years'}, xlabel='[Indicator Code]'>
```





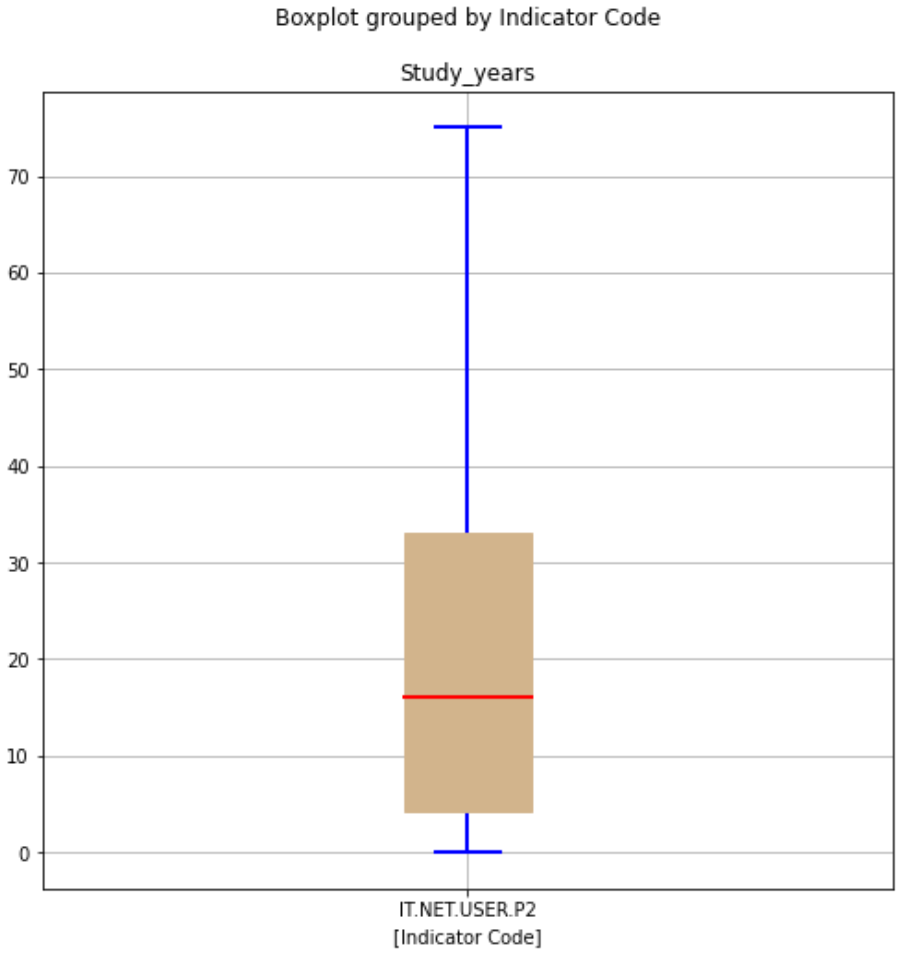
- **Environ 25% des pays étudiés ont moins de 105 608 personnes qui étudient dans le tertiaire (il s'agit de pays pauvres) et environ 75% des pays étudiés ont plus de 105 608 personnes qui étudient dans le secondaire (25% ont plus de 1 310 947 personnes étudiant dans le secondaire, il s'agit de pays riches).**
- **Environ 50% des pays étudiés ont moins de 195 534 personnes qui étudient dans le tertiaire.**

In [ ]:

```
graphe = Data_Study[Data_Study['Indicator Code'].isin(NET)][['Indicator Code', 'Study_years']]
graphe.boxplot(by='Indicator Code', showfliers=False, figsize=(8, 8), showcaps=True, patch_artist=True, color='tan',medianprops={'linestyle': '-', 'linewidth': 2, 'color': 'red',}, whiskerprops={'linestyle': '-', 'linewidth': 2, 'color' : 'blue'}, capprops={'linestyle': '-', 'linewidth': 2, 'color':'blue'})
```

Out[ ]:

```
<AxesSubplot:title={'center':'Study_years'}, xlabel='[Indicator Code]'>
```



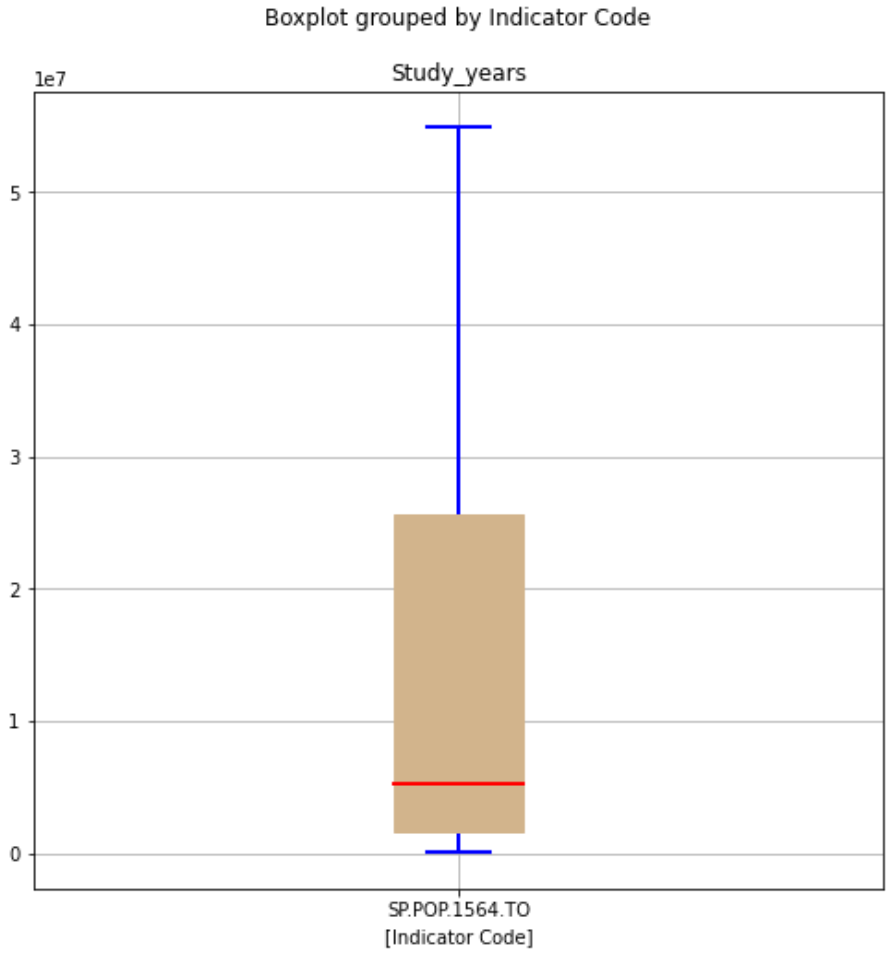
- **Environ 25% des pays étudiés ont moins de 4% de leur population qui utilisent internet (il s'agit de pays pauvres) et environ 75% des pays étudiés ont beaucoup plus de personnes qui ont accès à internet (25% des pays ont plus de 33% de leur population qui utilise internet).**
- **Environ 50% des pays étudiés ont moins de 16% de leur population qui a accès à internet.**

In [ ]:

```
graphe = Data_Study[Data_Study['Indicator Code'].isin(POP)][['Indicator Code', 'Study_years']]
graphe.boxplot(by='Indicator Code', showfliers=False, figsize=(8, 8), showcaps=True, patch_artist=True, color='tan',medianprops={'linestyle': '-', 'linewidth': 2, 'color': 'red',}, whiskerprops={'linestyle': '-', 'linewidth': 2, 'color' : 'blue'}, capprops={'linestyle': '-', 'linewidth': 2, 'color':'blue'})
```

Out[ ]:

```
<AxesSubplot:title={'center':'Study_years'}, xlabel='[Indicator Code]'>
```



- **Environ 25% des pays étudiés ont moins de 1 678 121 personnes qui ont entre 15 et 64 ans (il s'agit de petits pays). Environ 75% des pays étudiés ont plus de 1 678 121 personnes qui ont entre 15 et 64 ans (25% ont plus de 25 586 780 personnes ont entre 15 et 64 ans).**
- **Environ 50% des pays étudiés ont moins de 5 298 222 personnes qui ont entre 15 et 64 ans.**

Analyse statistique de nos indicateurs par zone géographique.

In [ ]:

```
def description_indicateurs(df, indicateurs, groupe):
    df_per_indicator = pd.DataFrame()
    for indicateur in indicateurs:
        df_temp = df[df['Indicator Code'] == indicateur]
        df_temp[indicateur] = df['Study_years']
        df_temp.drop(['Study_years', 'Indicator Name', 'Indicator Code'], inplace=True, axis=1)

        if df_per_indicator.empty is False:
            df_per_indicator = df_per_indicator.merge(right = df_temp, how = 'outer', on = ['Country Code', 'Country Name', 'Region']).sort_values(by='Country Name')
        else:
            df_per_indicator = df_temp
```

```
    description = df_per_indicator.groupby([groupe])[indicateurs].describe(percentiles =
[0.5])
    for indicateur in indicateurs:
        description[(indicateur, 'sum')] = description[(indicateur, 'count')] * descript
ion[(indicateur, 'mean')]
    colonnes = []
    for indicateur in indicateurs:
        for stat in ['mean', 'std', '50%', 'sum'] :
            colonnes += [(indicateur, stat)]
    description = description[colonnes]
    description.reset_index(inplace=True)
    return description, df_per_indicator
```

In [ ]:

```
description_countries, df_countries = description_indicateurs(data_countries, indicateurs
, 'Region')
description_zones, df_zones     = description_indicateurs(data_areas, indicateurs, 'Coun
try Name')
```

In [ ]:

```
description_zones
```

Out[ ]:

	Country Name	SE.SEC.ENRL				SE.TER.ENRL				SP.POP.1564.TO	
		mean	std	50%	sum	mean	std	50%	sum	mean	
0	Arab World	26267941.0	NaN	26267941.0	26267941.0	6.795613e+06	NaN	6.795613e+06	6.795613e+06	1.868050e+08	N
1	East Asia & Pacific	147579732.0	NaN	147579732.0	147579732.0	3.945356e+07	NaN	3.945356e+07	3.945356e+07	1.458420e+09	N
2	Europe & Central Asia	80480044.0	NaN	80480044.0	80480044.0	3.318544e+07	NaN	3.318544e+07	3.318544e+07	5.871547e+08	N
3	Latin America & Caribbean	57958326.0	NaN	57958326.0	57958326.0	1.632724e+07	NaN	1.632724e+07	1.632724e+07	3.557445e+08	N
4	Middle East & North Africa	33923334.0	NaN	33923334.0	33923334.0	9.274999e+06	NaN	9.274999e+06	9.274999e+06	2.172816e+08	N
5	North America	25884831.0	NaN	25884831.0	25884831.0	1.839120e+07	NaN	1.839120e+07	1.839120e+07	2.173912e+08	N
6	South Asia	111357252.0	NaN	111357252.0	111357252.0	1.710642e+07	NaN	1.710642e+07	1.710642e+07	9.199538e+08	N
7	Sub-Saharan Africa	35142246.0	NaN	35142246.0	35142246.0	4.452334e+06	NaN	4.452334e+06	4.452334e+06	4.048978e+08	N

Quelles sont les régions du monde à privilégier pour notre expension à l'international ?

In [ ]:

```
df_pv=pd.pivot_table(data_areas, index=['Indicator Name', 'Country Name'], values='Study
_years', aggfunc='sum')
df_pv.head()
```

Out[ ]:

	Study_years

Indicator Name	Country Name	Study_years
Internet users	Arab World	13.672917
	East Asia & Pacific	18.818623
	Europe & Central Asia	32.681569
	Latin America & Caribbean	18.381056
	Middle East & North Africa	12.885774

In [ ]:

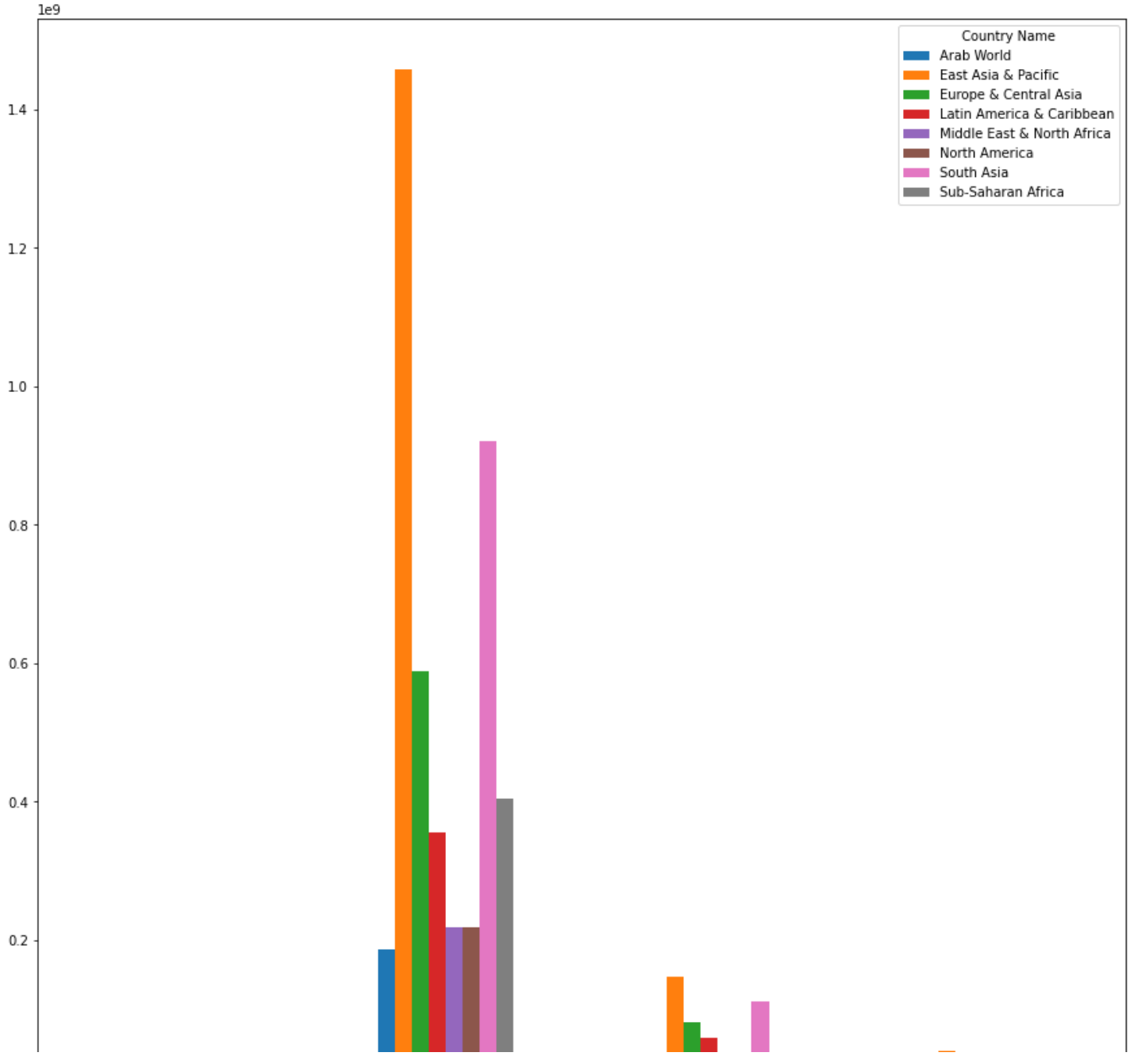
```
df_pv_areas = pd.pivot_table(data_areas, index = 'Indicator Name', columns = 'Country Name', values = 'Study_years', aggfunc = 'sum')
```

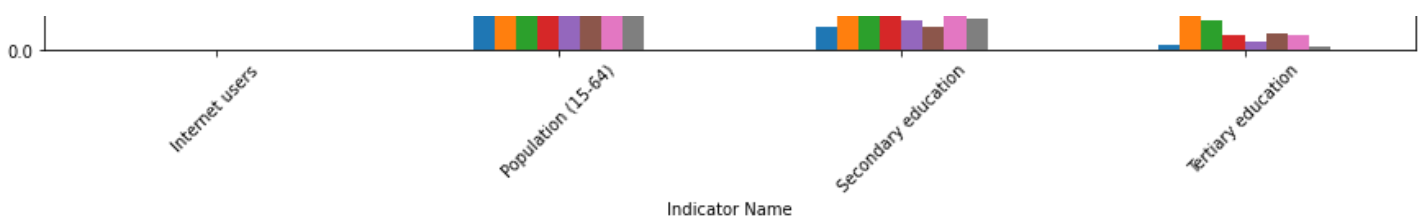
In [ ]:

```
df_pv_areas.plot( kind='bar', figsize=(15,15))
plt.xticks(rotation=45)
```

Out[ ]:

```
(array([0, 1, 2, 3]),
 [Text(0, 0, 'Internet users'),
  Text(1, 0, 'Population (15-64)'),
  Text(2, 0, 'Secondary education'),
  Text(3, 0, 'Tertiary education')])
```

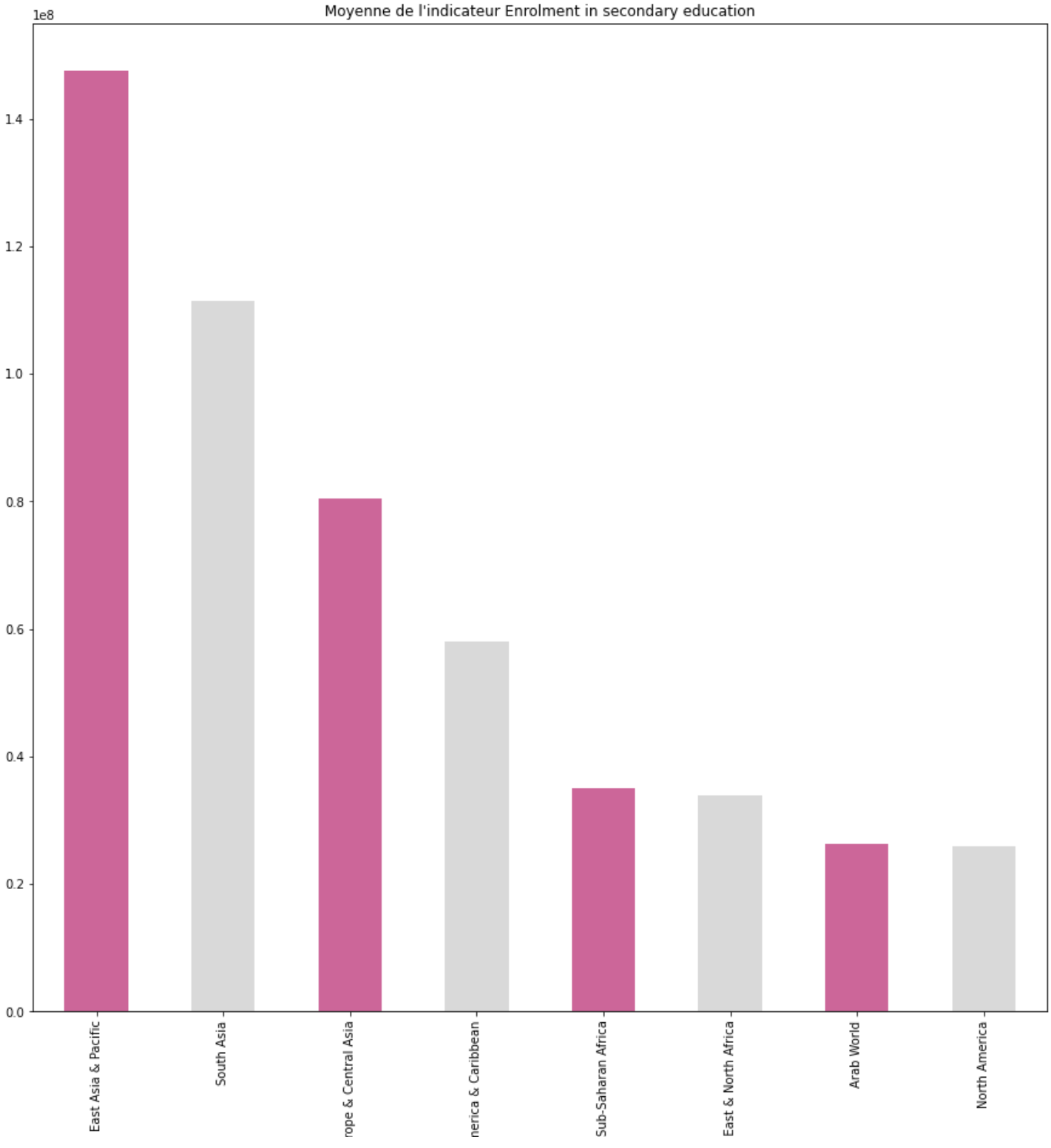




On peut voir que certaines régions du monde comme East Asia & Pacific, South Asia et Europe et central Asia sont les régions qui ressortent de cette étude. Cependant il est diffcile d'analyser correctement le graphique, c'est pourquoi chaque indicateur va être analysé séparement.

In [ ]:

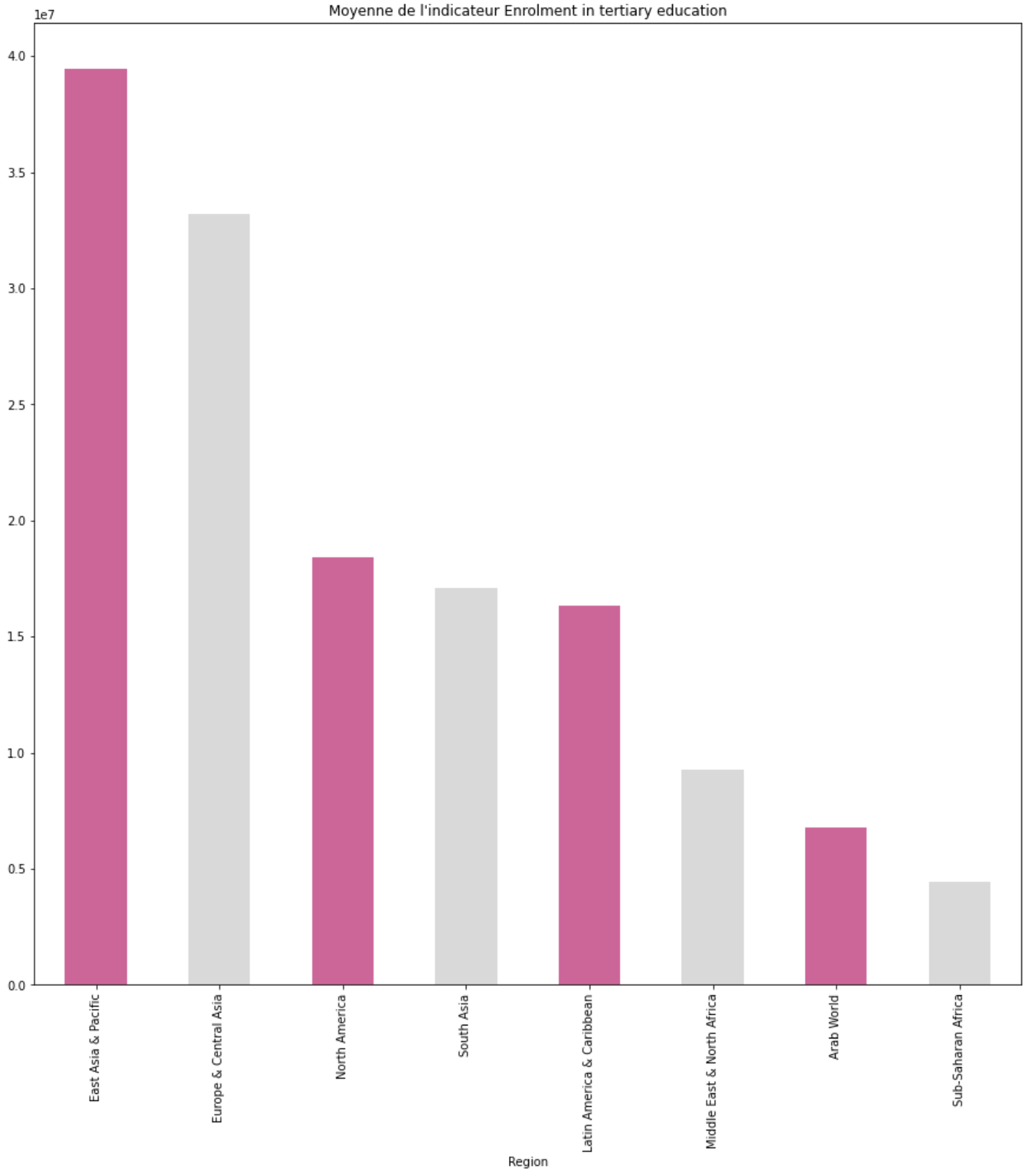
```
indicator_1a = df_pv_areas.loc["Secondary education"].sort_values(ascending=False).head(
15)
my_colors = [(0.8,0.4,0.6), (0.85, 0.85, 0.85)]
ax = indicator_1a.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Enrol
ment in secondary education", xlabel='Region', color=my_colors)
plt.show()
```



En observant le nombre moyen de personnes inscrites dans le secondaire dans chaque région du monde, on remarque que la majorité des lycéens se trouve dans les régions suivantes : East Asia & Pacific, South Asia, Europe & central Asia et Latin America & Caribbean.

In [ ]:

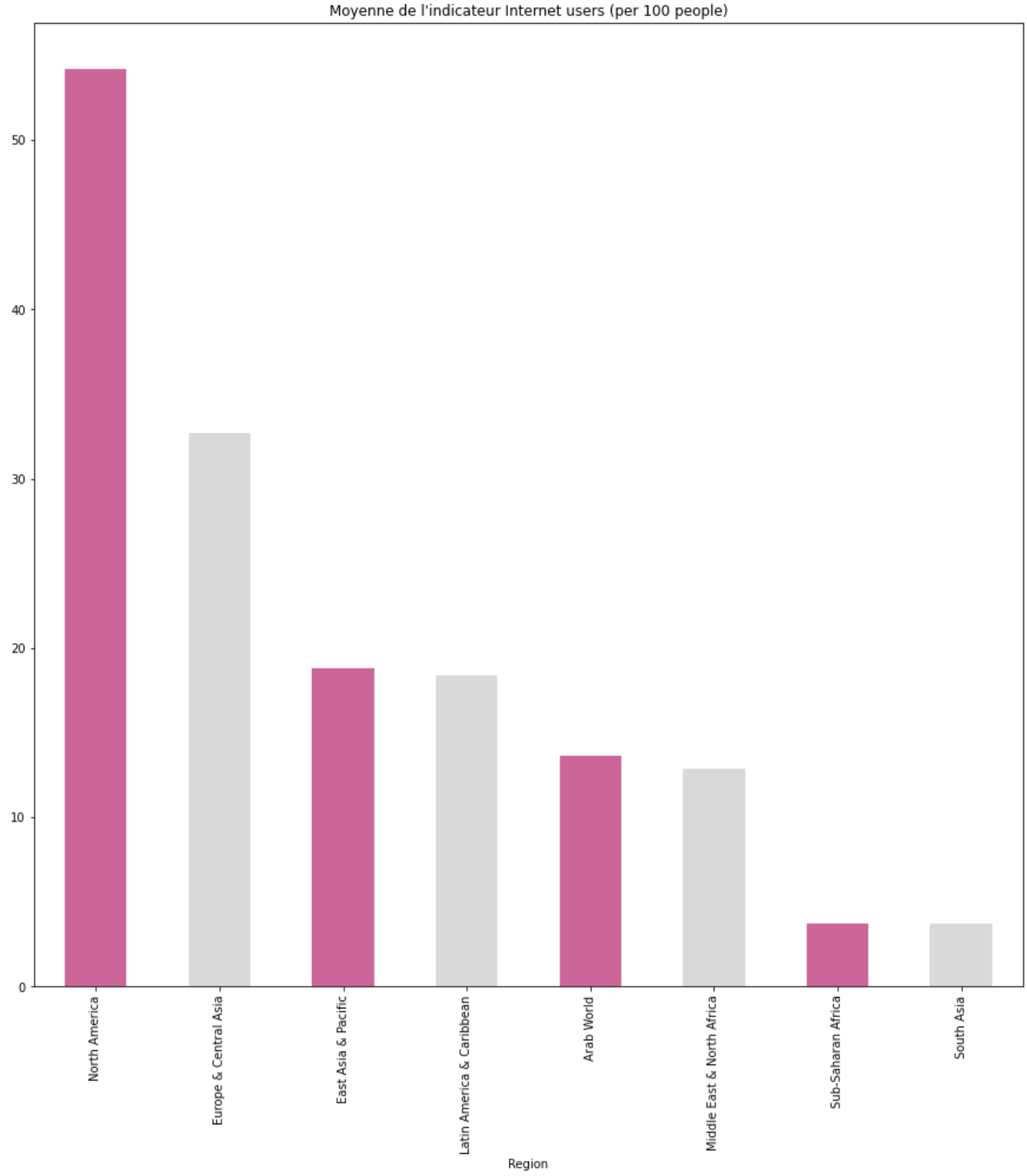
```
indicator_2a = df_pv_areas.loc["Tertiary education"].sort_values(ascending=False).head(15)
my_colors = [(0.8,0.4,0.6), (0.85, 0.85, 0.85)]
ax = indicator_2a.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Enrolment in tertiary education", xlabel='Region', color=my_colors)
plt.show()
```



En observant le nombre moyen de personnes inscrites dans le tertiaire dans chaque région du monde, on remarque que la majorité des étudiants se trouve dans les régions suivantes : East Asia & Pacific, Europe & central Asia, North America et South Asia.

In [ ]:

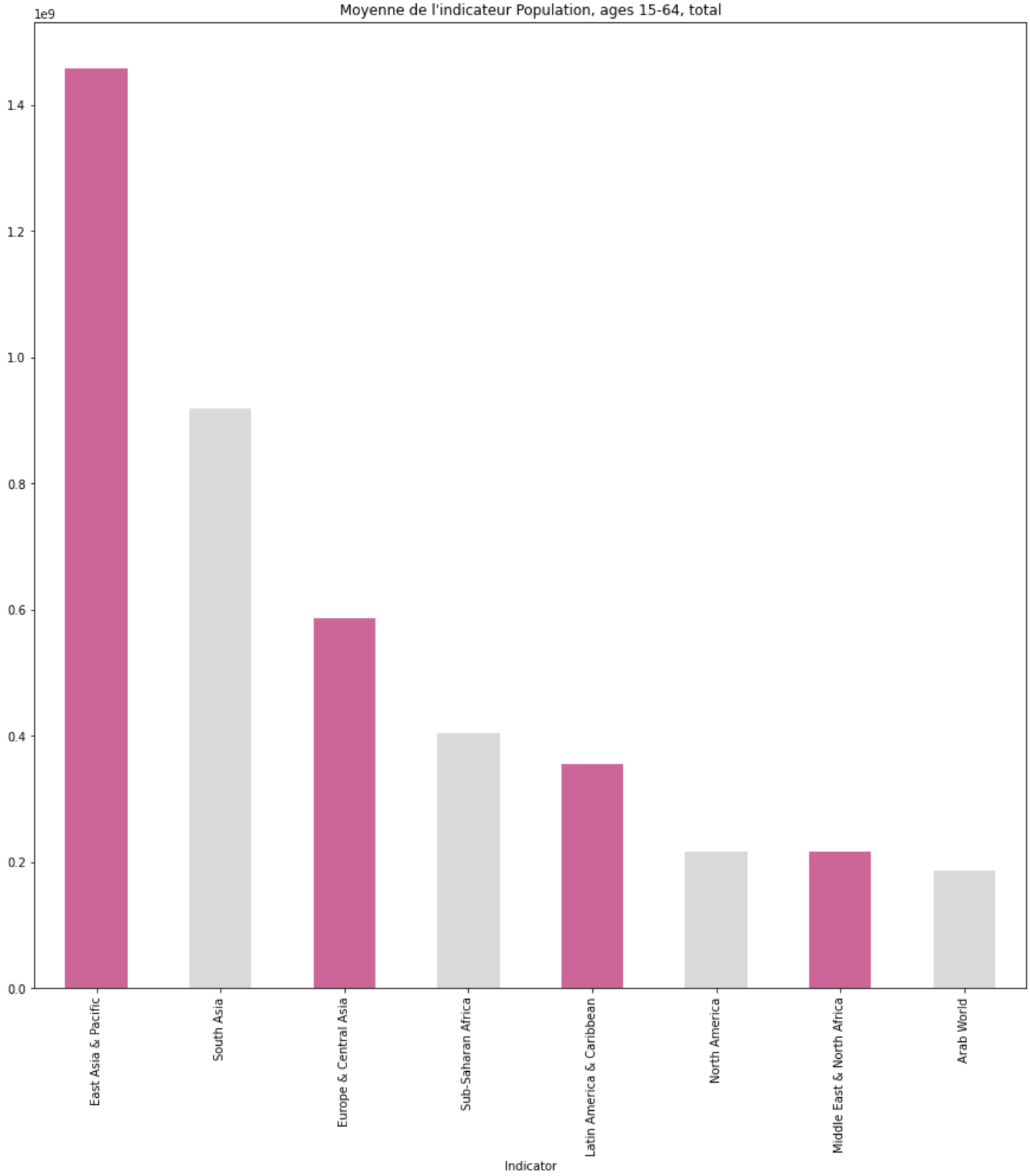
```
indicator_3a = df_pv_areas.loc["Internet users"].sort_values(ascending=False).head(15)
my_colors = [(0.8,0.4,0.6), (0.85, 0.85, 0.85)]
ax = indicator_3a.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Internet users (per 100 people)", xlabel='Region', color=my_colors)
fig = ax.get_figure()
plt.show()
```



En observant le taux moyen d'accès à internet par région, il est plus aisé d'utiliser internet dans les régions suivantes : North America, Europe & Central Asia, East Asia & Pacific et Latin America & Caribbean.

In [ ]:

```
indicator_4a = df_pv_areas.loc["Population (15-64)"].sort_values(ascending=False).head(15)
my_colors = [(0.8,0.4,0.6), (0.85, 0.85, 0.85)]
ax = indicator_4a.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Population, ages 15-64, total", xlabel='Region', color=my_colors)
fig = ax.get_figure()
ax.set_xlabel("Indicator")
plt.show()
```



En observant le nombre moyen de personnes agées entre 15 et 64 ans, on remarque que la majorité des personnes susceptibles de s'intéresser à nos services se situe dans les régions suivantes : East Asia & Pacific, South Asia, Europe & central Asia.

Quels sont les pays à privilégier pour notre expension à l'internationnal ?

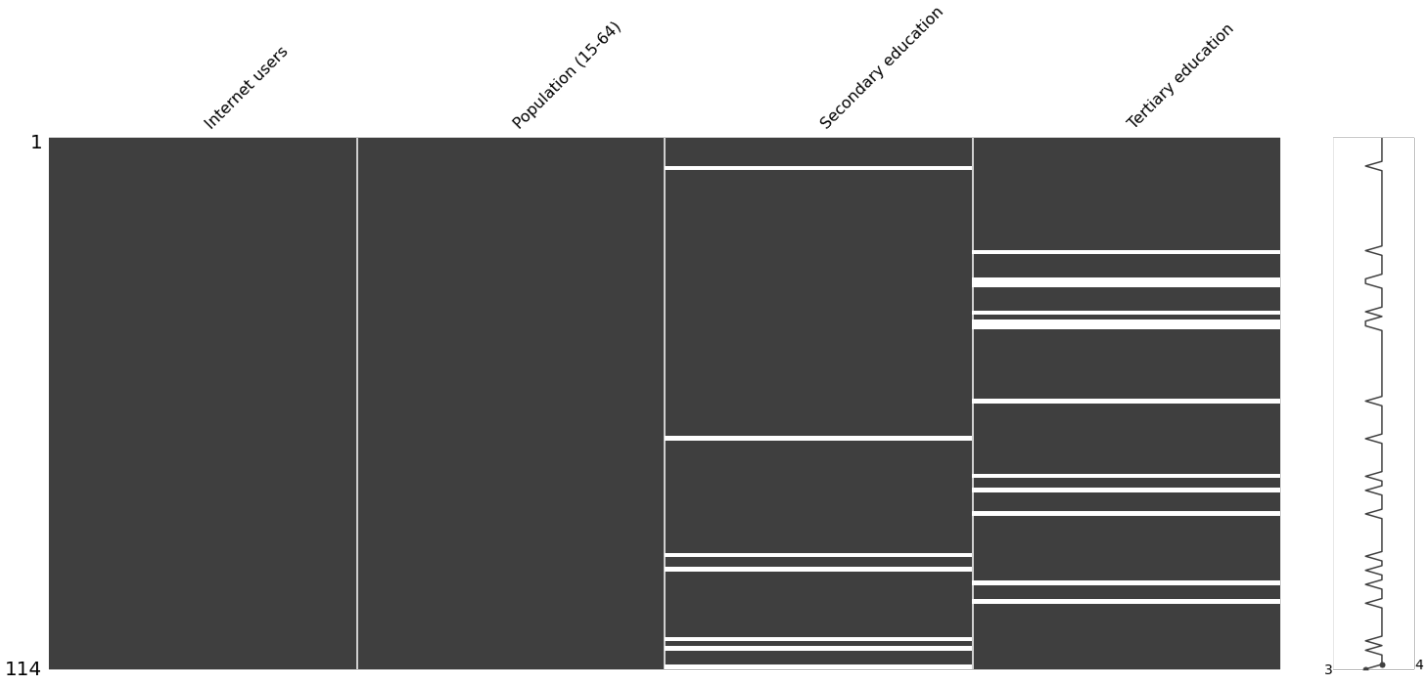
Taux de remplissage des données selon les indicateurs et les pays choisis

In [ ]:

```
df_pv_c = pd.pivot_table(data_countries, index=['Country Name', 'Indicator Name'], value
s='Study_years', dropna=False)
df_pv_countries = pd.pivot_table(df_pv_c, index = 'Indicator Name', columns = 'Country N
ame', values = 'Study_years', aggfunc = 'min', dropna=False)
df_pv_countries_T=df_pv_countries.T
msno.matrix(df_pv_countries_T)
```

Out[ ]:

```
<AxesSubplot:>
```



Les pays et les indicateurs choisis sont des choix judicieux car il y a peu de données manquantes, généralement il ne manque des données que sur un des quatres indicateurs. L'indicateur, essentiel dans notre étude sur le développement de cours e-learning, qui quantifie l'accès à internet est traité par la totalité des pays.

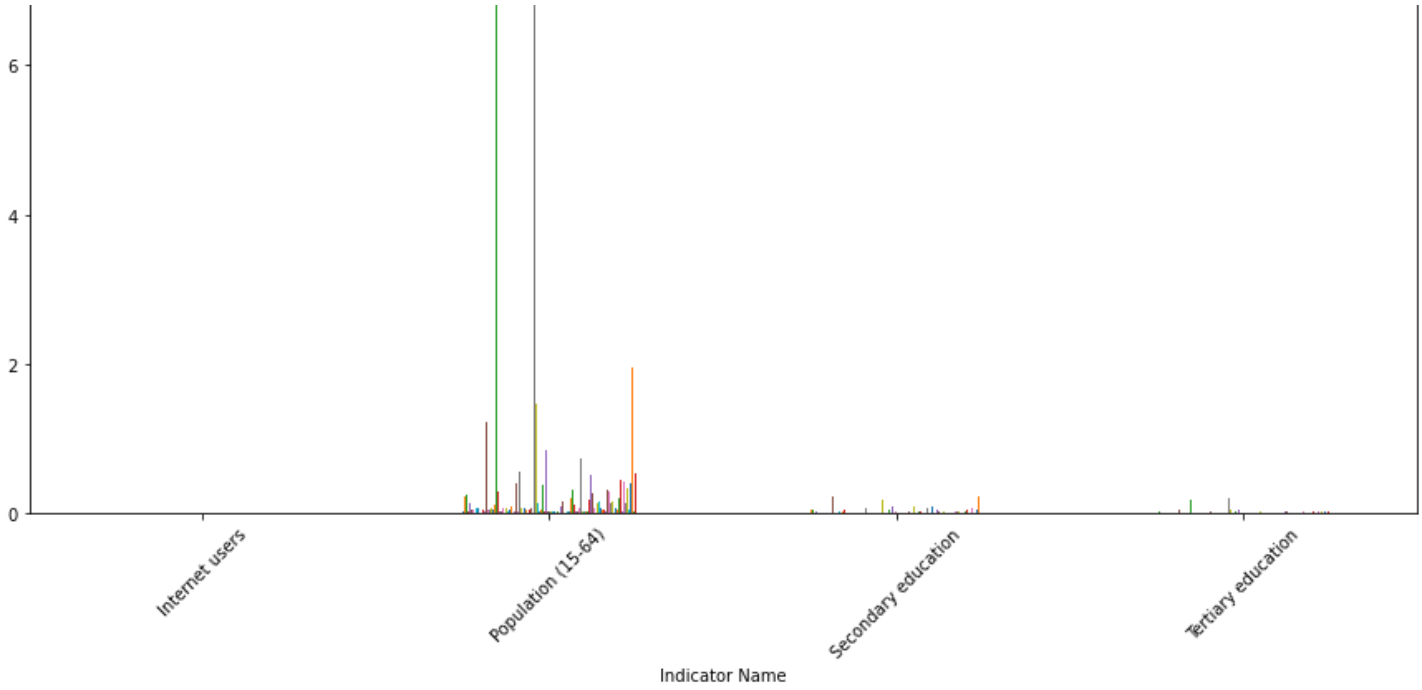
Quels sont les pays satisfaisant nos critères ?

In [ ]:

```
df_pv_countries.plot( kind='bar', figsize=(15,8), legend=None)
plt.xticks(rotation=45)
```

Out[ ]:

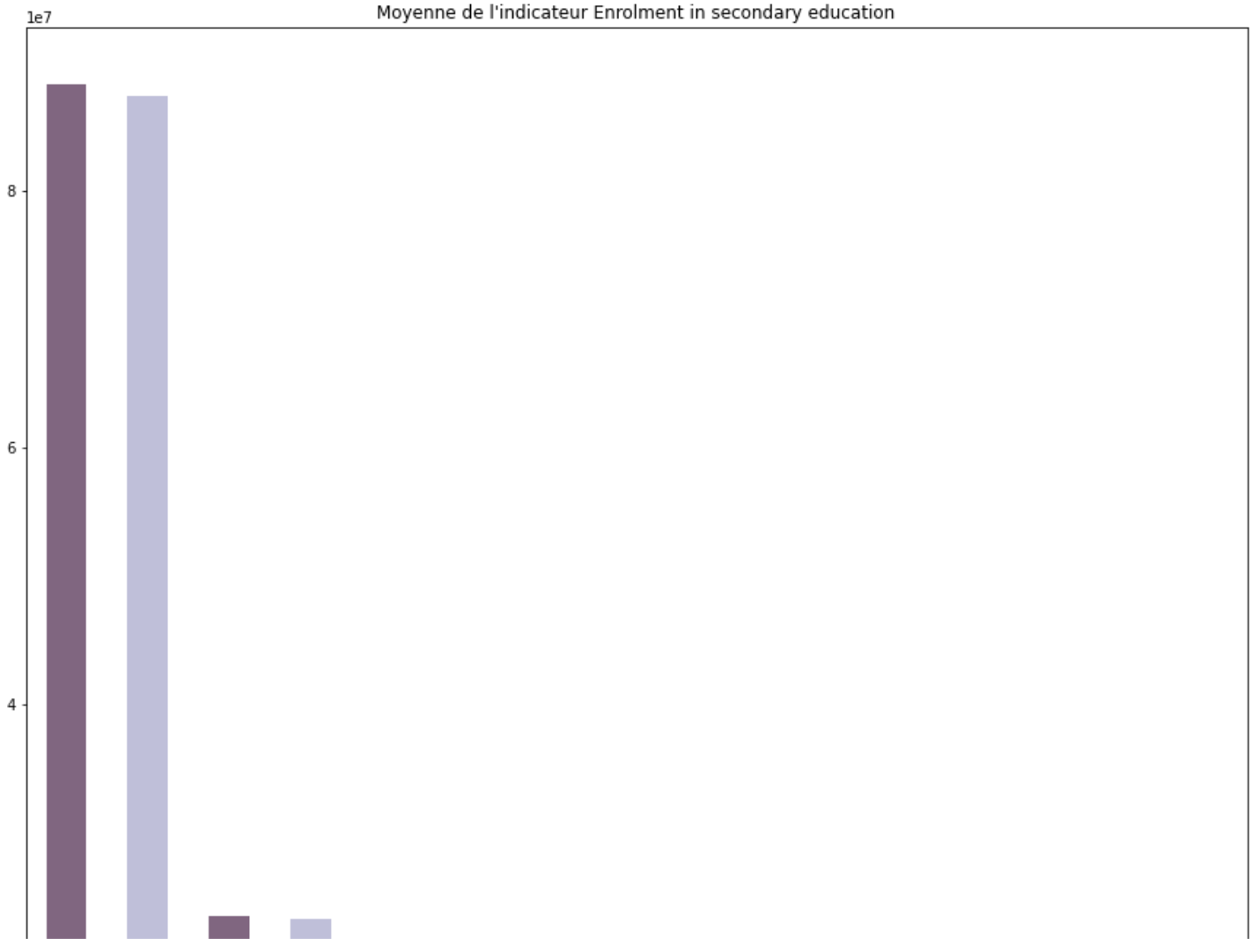
```
(array([0, 1, 2, 3]),
 [Text(0, 0, 'Internet users'),
  Text(1, 0, 'Population (15-64)'),
  Text(2, 0, 'Secondary education'),
  Text(3, 0, 'Tertiary education')])
```

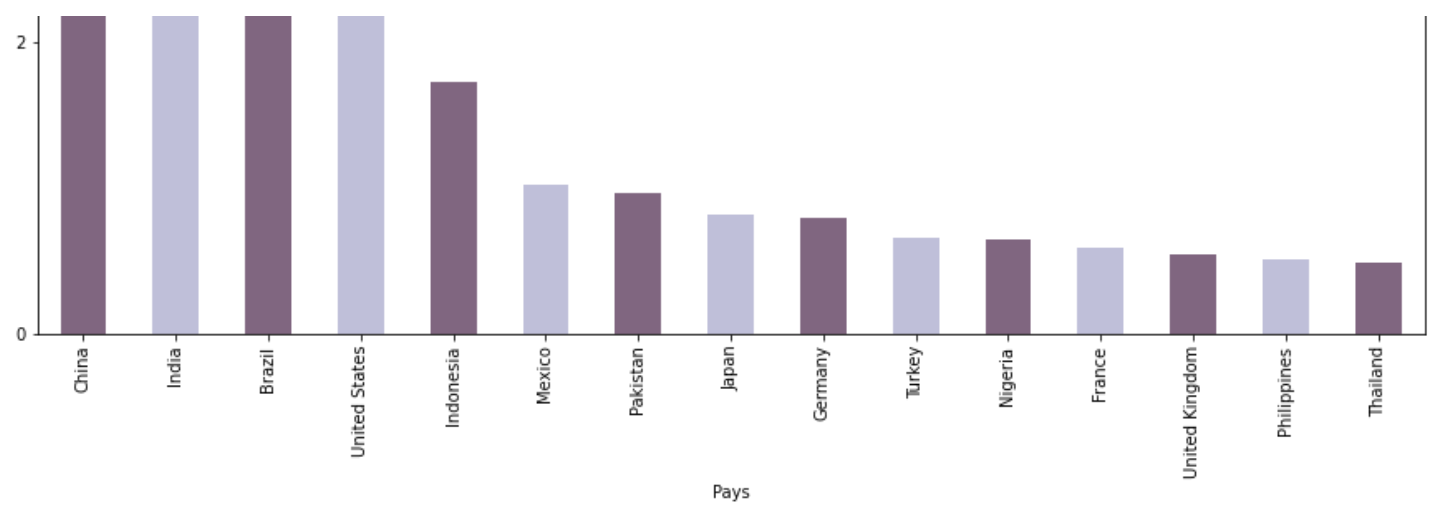


On peut voir que certains pays ressortent de cette étude. Cependant il est diffcile d'analyser correctement le graphique, c'est pourquoi chaque indicateur va être analysé séparement.

In [ ]:

```
indicator_1c = df_pv_countries.loc["Secondary education"].sort_values(ascending=False).h
ead(15)
my_colors = [(0.5,0.4,0.5), (0.75, 0.75, 0.85)]
ax = indicator_1c.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Enrol
ment in secondary education", xlabel='Pays', color=my_colors)
fig = ax.get_figure()
plt.show()
```

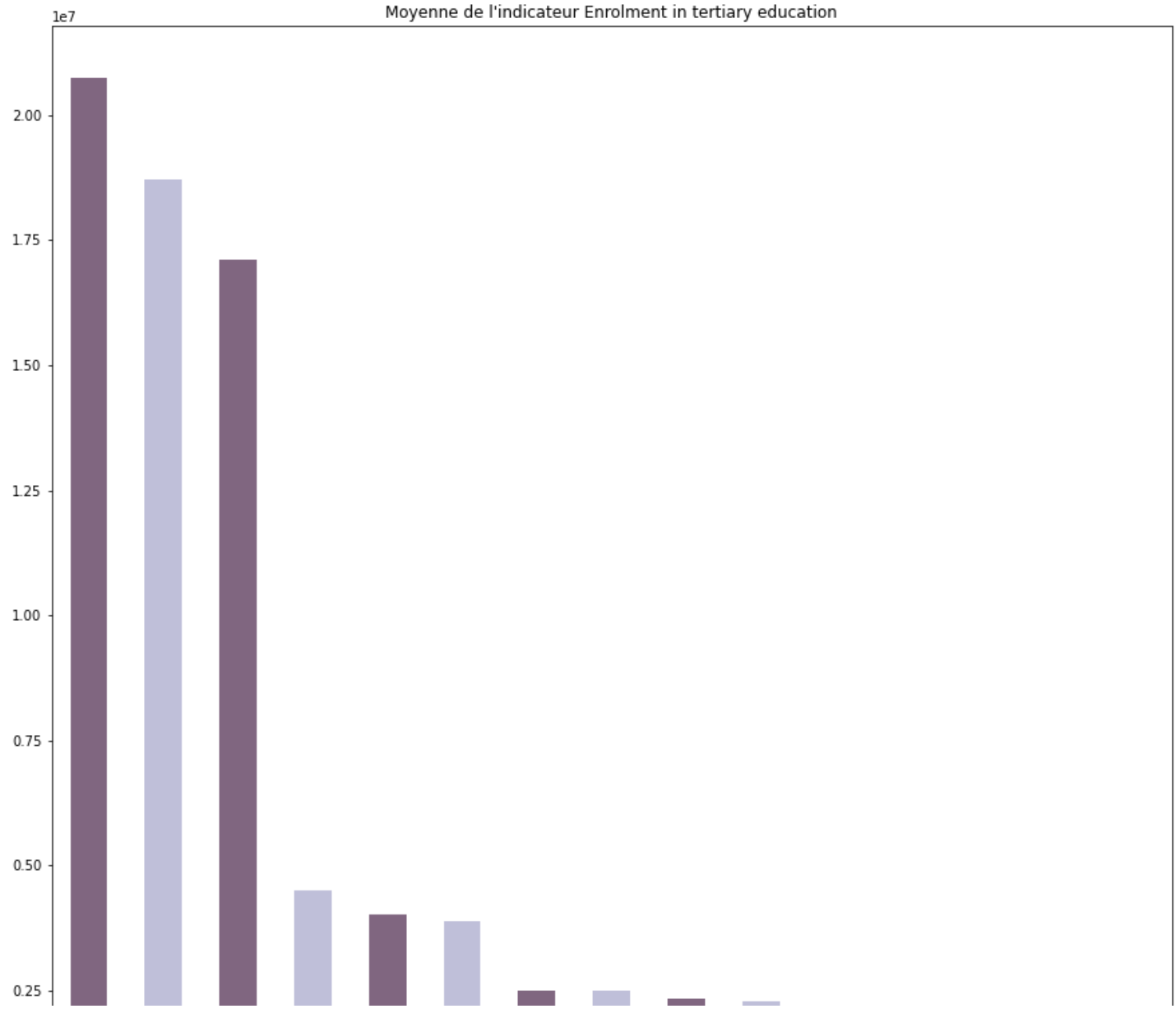


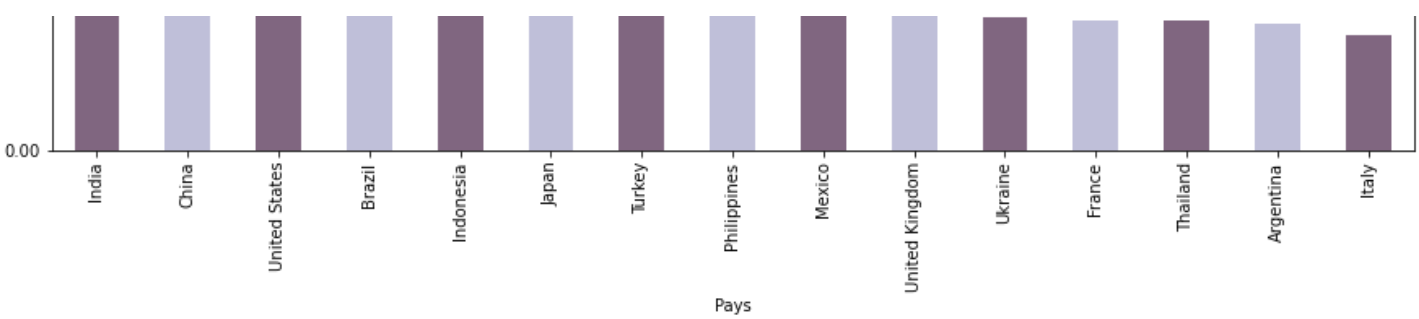


En observant le nombre moyen de personnes inscrites dans le secondaire dans chaque pays du monde, on remarque que la majorité des lycéens se trouve dans les pays suivants : Chine, Inde , Brésil, Etat-Unis, Indonésie.

In [ ]:

```
indicator_2c = df_pv_countries.loc["Tertiary education"].sort_values(ascending=False).hea
d(15)
my_colors = [(0.5,0.4,0.5), (0.75, 0.75, 0.85)]
ax = indicator_2c.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Enrol
ment in tertiary education", xlabel='Pays', color=my_colors)
fig = ax.get_figure()
plt.show()
```

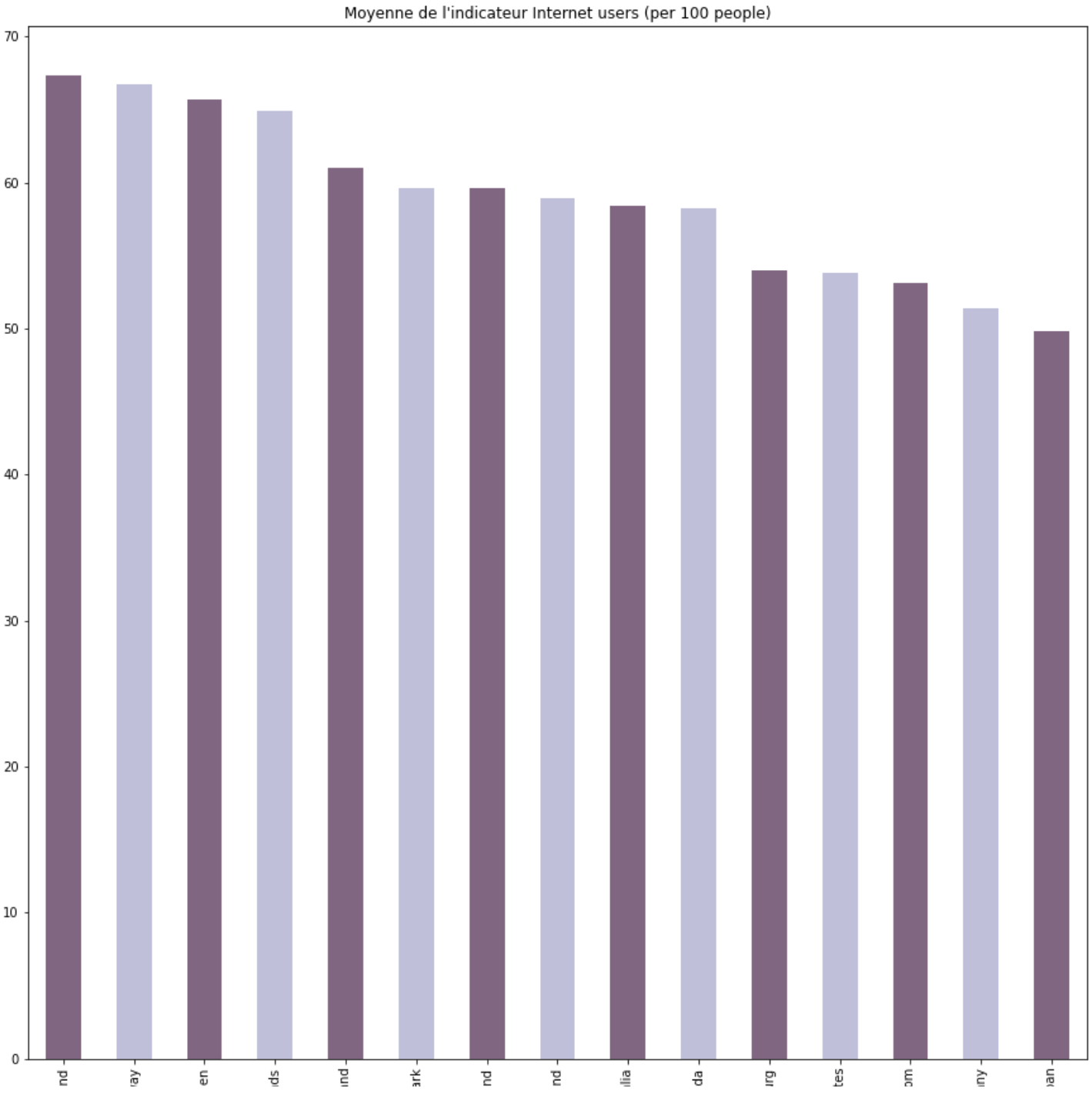




En observant le nombre moyen de personnes inscrites dans le tertiaire dans chaque pays du monde, on remarque que la majorité des étudiants se trouve dans les pays suivants : Inde, Chine, Etat-Unis, Brésil, Indonésie.

In [ ]:

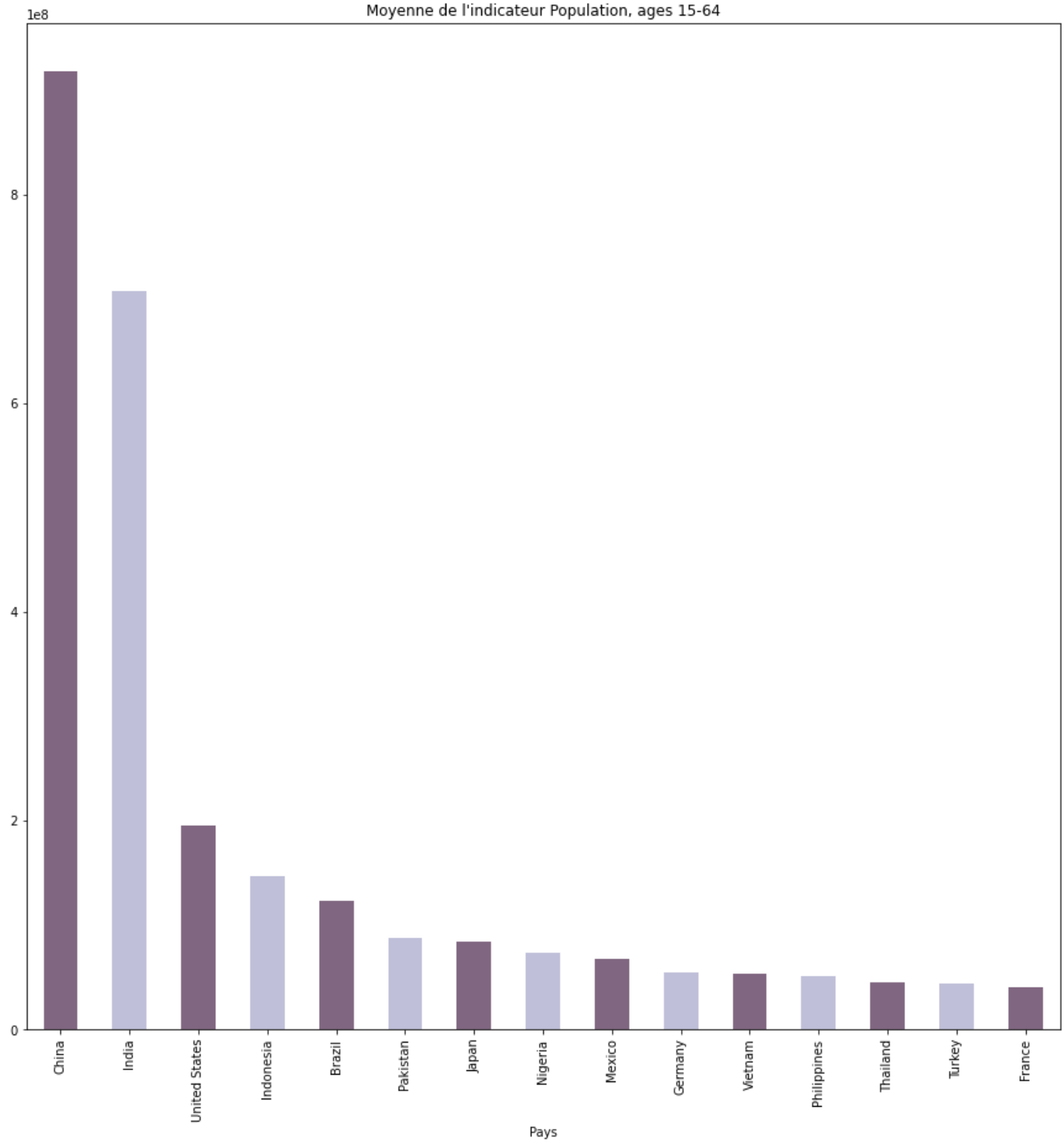
```
indicator_3c = df_pv_countries.loc["Internet users"].sort_values(ascending=False).head(15)
my_colors = [(0.5,0.4,0.5), (0.75, 0.75, 0.85)]
ax = indicator_3c.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Internet users (per 100 people)", xlabel='Pays', color=my_colors)
fig = ax.get_figure()
plt.show()
```



En observant le taux moyen d'accès à internet par pays, il est possible de dire que tous les pays riches, développés ont un accès facile à internet.

In [ ]:

```
indicator_4c = df_pv_countries.loc["Population (15-64)"].sort_values(ascending=False).hea
d(15)
my_colors = [(0.5,0.4,0.5), (0.75, 0.75, 0.85)]
ax = indicator_4c.plot(kind="bar", figsize=(15,15), title="Moyenne de l'indicateur Popul
ation, ages 15-64", xlabel='Pays', color=my_colors)
fig = ax.get_figure()
plt.show()
```



En observant le nombre moyen de personnes agées entre 15 et 64 ans, on remarque que la majorité des

personnes susceptibles de s'intéresser à nos services se trouve dans les pays suivants : Chine, Inde, Etat-Unis, Indonésie, Brésil, .

Première conclusion : Où est-il intéressant de s'implanter ?

Après cette première analyse, il serait intéressant de s'implanter en Asie de l'Est et Pacific, en Asie du Sud, en Europe et Asie centrale, en Amérique du Nord et en Latin America & Caribbean. Et plus précisément **en Chine, en Inde, aux Etats-Unis, au Brésil et en Indonésie.**

Analyses complémentaires pour conforter notre première conclusion

Evolution de nos indicateurs en fonction des 5 pays qui ressortent de la première étude.

In [ ]:

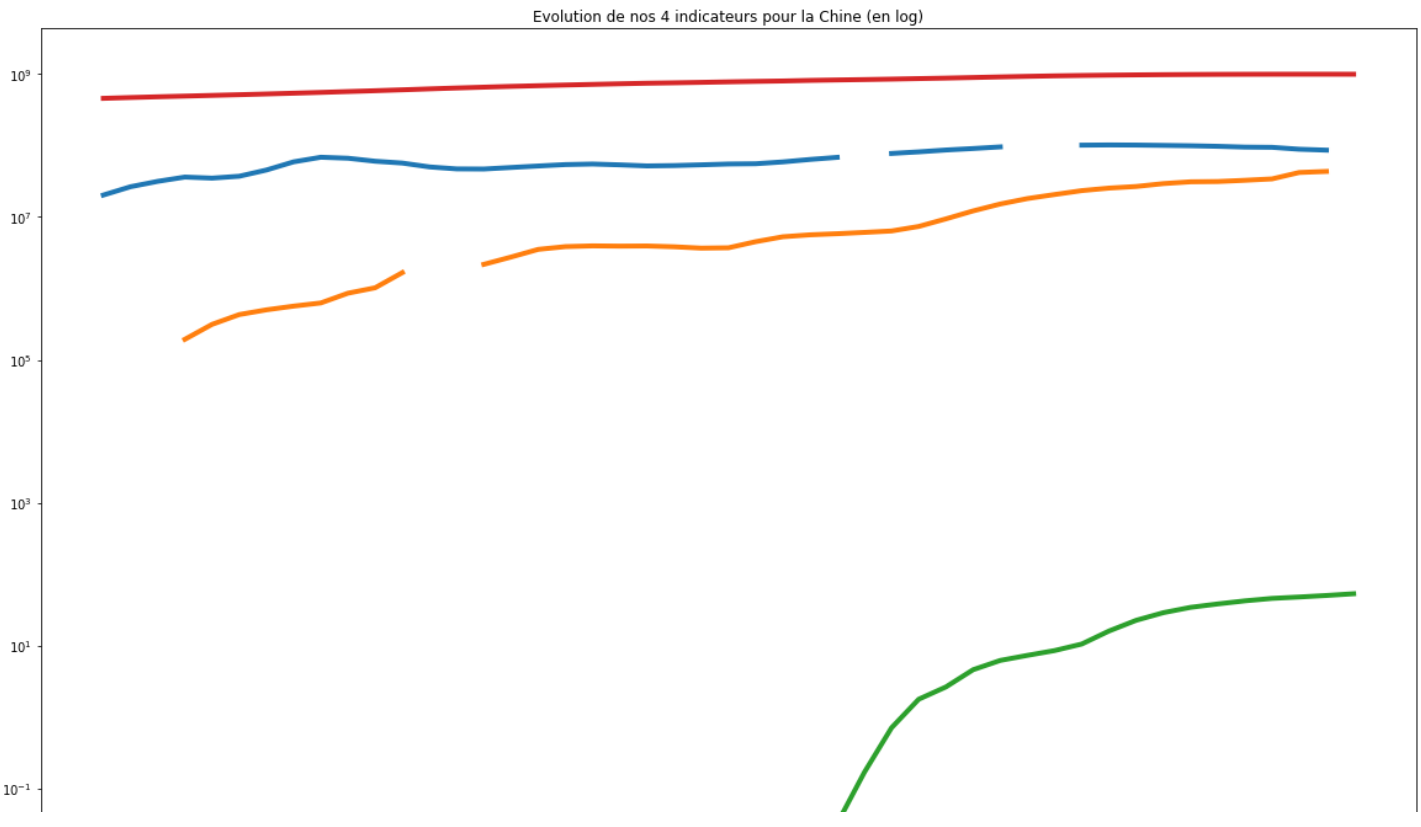
```
win_countries = ['China', 'India', 'United States', 'Indonesia', 'Brazil']
ch = ['China']
ind = ['India']
USA = ['United States']
indo = ['Indonesia']
braz = ['Brazil']
data_winners = Stats_Data[Stats_Data['Country Name'].isin(win_countries)][Stats_Data['Ind
icator Code'].isin(indicateurs)]
```

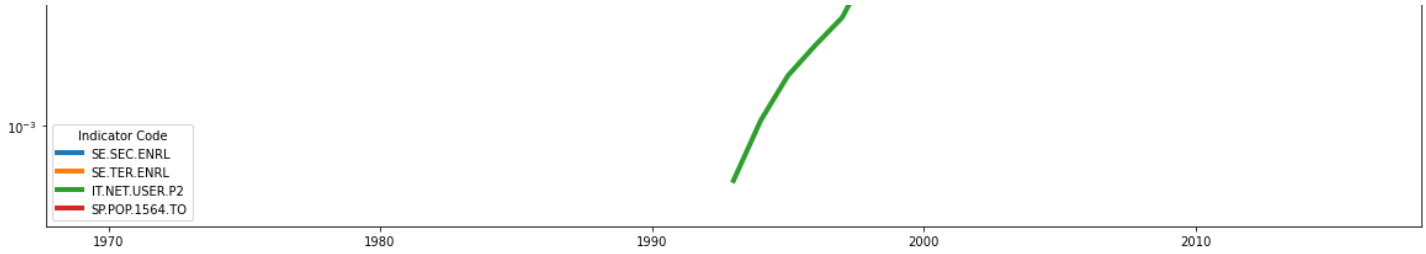
Formation d'un nouveau Dataset contenant uniquement les lignes concernant nos pays et nos indicateurs.

La Chine.

In [ ]:

```
CHINA = data_winners[data_winners['Country Name'].isin(ch)]
CHINA = CHINA.drop(columns=['Country Name', 'Country Code', 'Indicator Name'])
CHINA = CHINA.set_index("Indicator Code")
CHINA = CHINA.T
CHINA.plot(linewidth=4,kind='line',figsize=(20,15), title='Evolution de nos 4 indicateur
s pour la Chine (en log)').set_yscale('log')
```



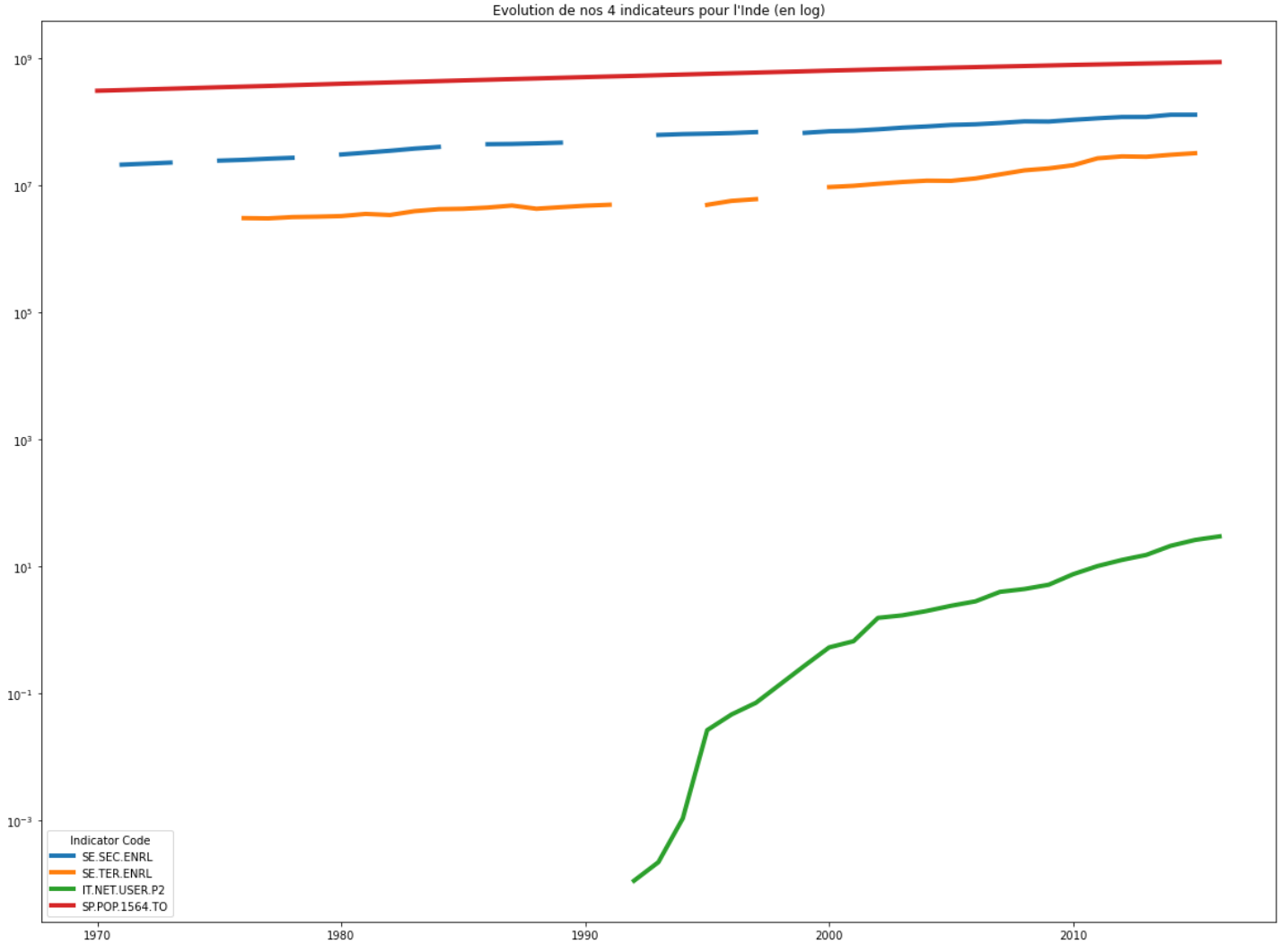


La conversion des données en logarithme permet une comparaison des différents indicateurs. Depuis le début de la prises des données, la population est jeune et il y a au fil des années une légère augmentation de ce nombre mais il reste constant. Le nombre de personnes dans le secondaire est en légère augmentation contrairement au nombre de personnes inscrites dans le tertiaire qui lui est en forte évolution constante jusqu'à rejoindre la courbe du nombre de personne inscrites dans le secondaire ce qui signifie que de nos jour, la très grande majorité des chinois inscrits dans le secondaire poursuivent leur études dans le tertiaire. Enfin, suite à l'apparition d'internet, la population chinoise s'est rapidement équipée et de nos jours il y a un plateau pour cet indicateur car quasiment toute la population possède internet.

L'Inde.

In [ ]:

```
INDIA = data_winners[data_winners['Country Name'].isin(ind)]
INDIA = INDIA.drop(columns=['Country Name', 'Country Code', 'Indicator Name'])
INDIA = INDIA.set_index("Indicator Code")
INDIA = INDIA.T
INDIA.plot(linewidth=4, kind='line',figsize=(20,15), title='Evolution de nos 4 indicateurs pour l\'Inde (en log)').set_yscale('log')
```



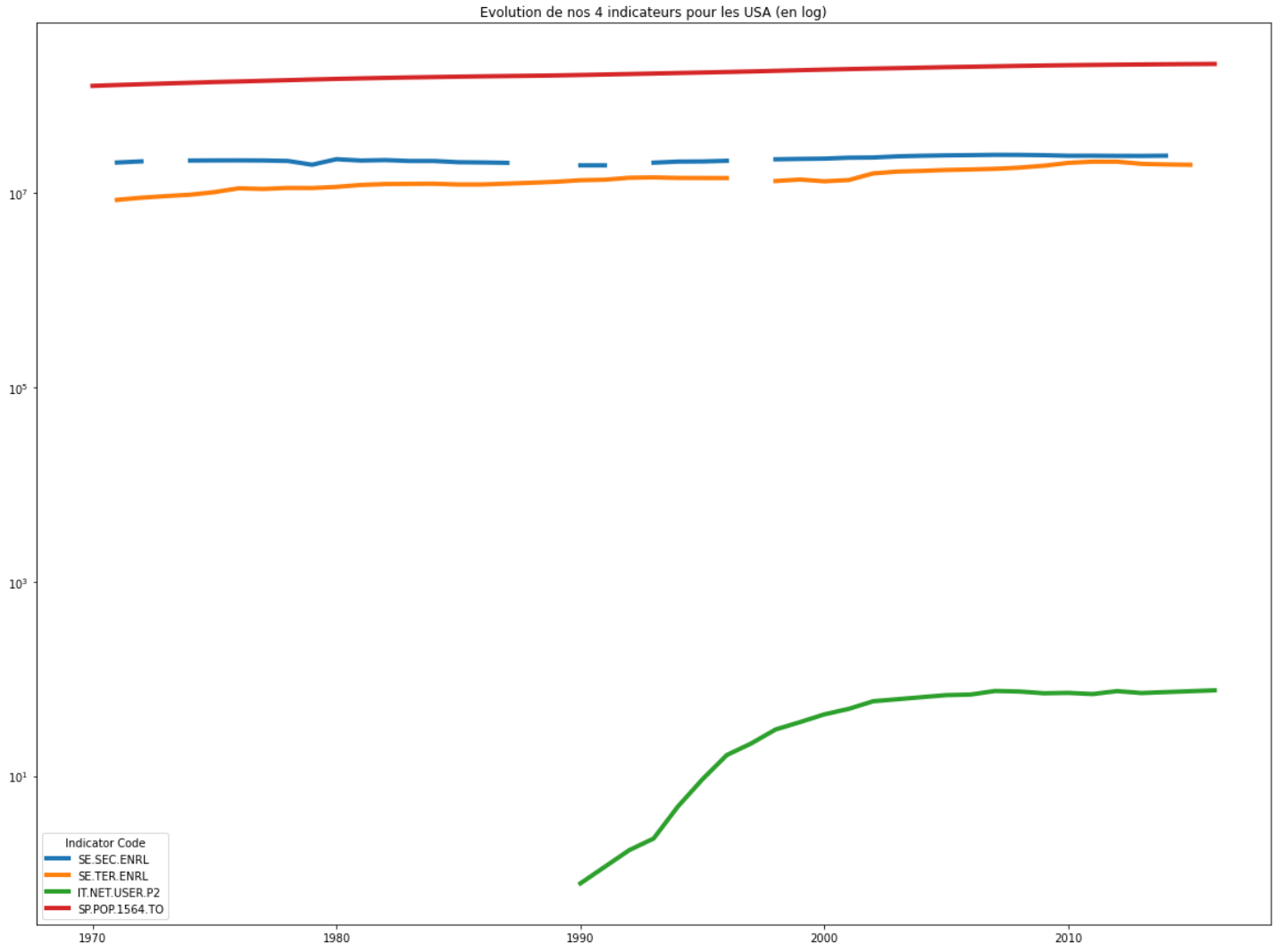
La population est depuis le début des prises des données jeune mais il y a au fil des années une augmentation de ce nombre. Le nombre de personne dans le secondaire est en augmentation ainsi que le nombre de personnes inscrites dans le tertiaire qui est en forte évolution. Cependant, une partie des personnes inscrites

dans le secondaire ne continuent pas leurs études dans le tertiaire. Enfin, suite à l'apparition d'internet, la population a commencé à s'équiper mais l'acquisition d'internet s'effectue au fil des années et continue toujours aujourd'hui car cet indicateur est encore en augmentation.

Les Etats Unis.

In [ ]:

```
USA = data_winners[data_winners['Country Name'].isin(USA)]
USA = USA.drop(columns=['Country Name', 'Country Code', 'Indicator Name'])
USA = USA.set_index("Indicator Code")
USA = USA.T
USA.plot(linewidth=4,kind='line',figsize=(20,15), title='Evolution de nos 4 indicateurs pour les USA (en log)').set_yscale('log')
```



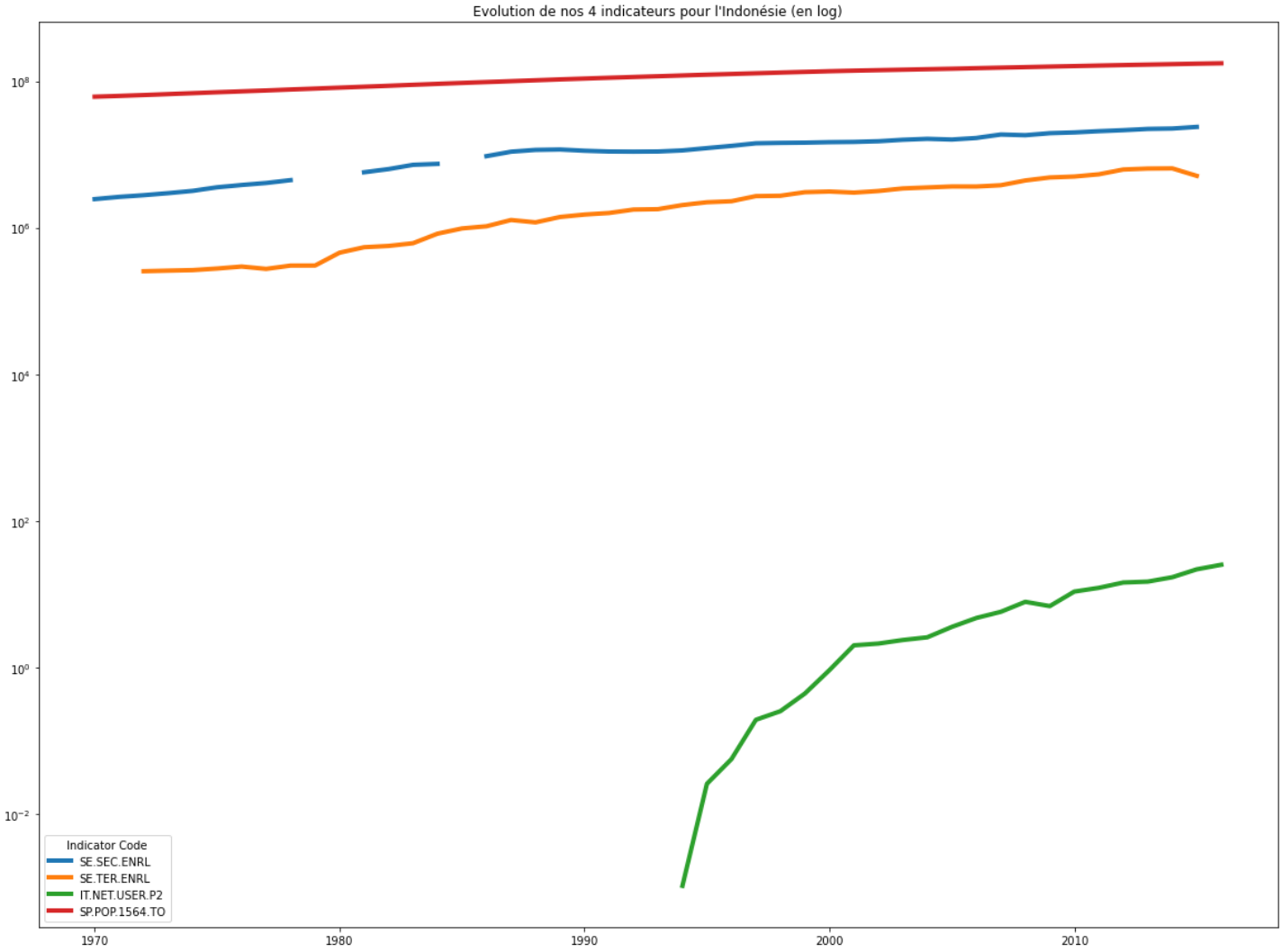
La population est depuis le début des prises des données jeune mais il y a au fil des années une petite augmentation de ce nombre. Le nombre de personne dans le secondaire est constant ce qui montre que l'éducation prend une part importante dans ce pays. Cependant, le nombre de personnes inscrites dans le tertiaire est en légère augmentation jusqu'à rejoindre la courbe du nombre de personnes inscrites dans le secondaire ce qui signifie que, la très grande majorité des inscrits dans le secondaire poursuivent leur études dans le tertiaire. Enfin, suite à l'apparition d'internet, la population s'est rapidement équipée et de nos jours il y a un plateau pour cet indicateur car quasiment toute la population possède internet.

L'Indonésie.

In [ ]:

```
INDO = data_winners[data_winners['Country Name'].isin(indo)]
```

```
INDO = INDO.drop(columns=['Country Name', 'Country Code', 'Indicator Name'])
INDO = INDO.set_index("Indicator Code")
INDO = INDO.T
INDO.plot(linewidth=4,kind='line',figsize=(20,15), title='Evolution de nos 4 indicateurs pour l\'Indonésie (en log)').set_yscale('log')
```

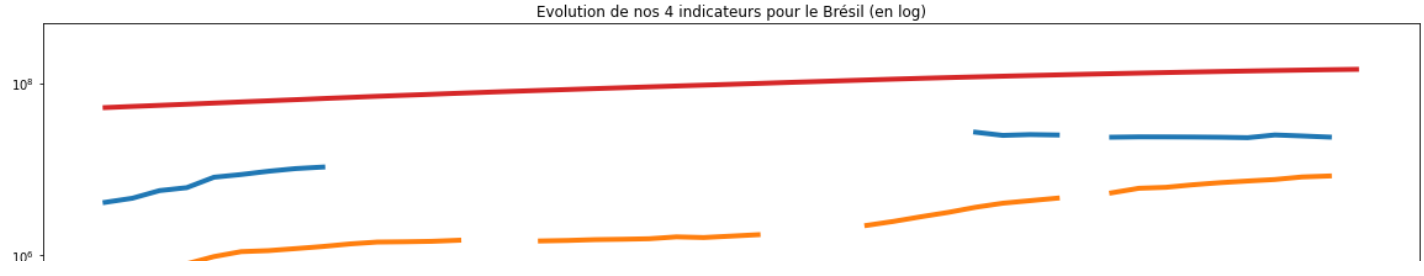


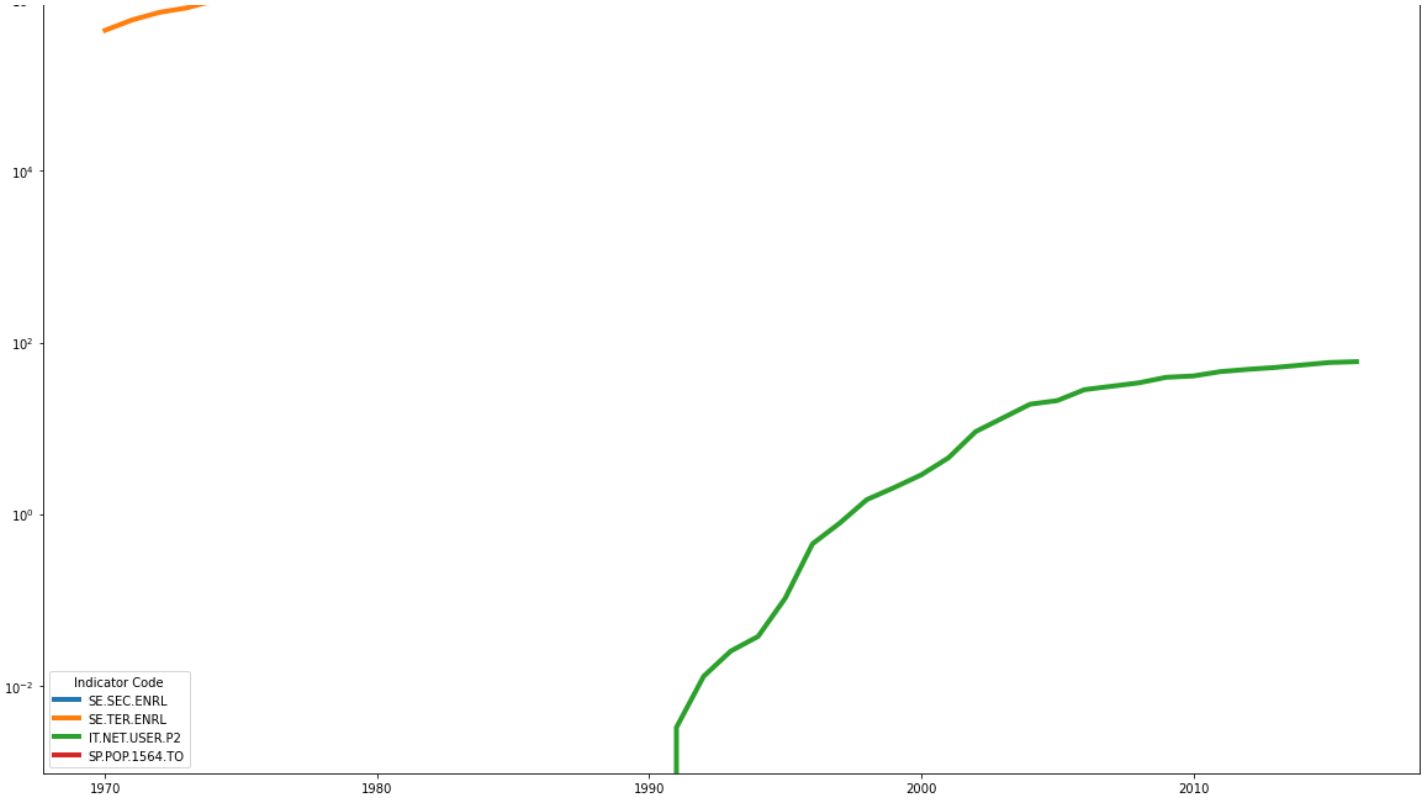
La population est depuis le début des prises des données jeune mais il y a au fil des années une petite augmentation de ce nombre. Le nombre de personne dans le secondaire est en augmentation ainsi que le nombre de personnes inscrites dans le tertiaire. Cependant, une partie les personnes inscrite dans le secondaire ne continuent pas leurs études dans le tertiaire. Enfin, suite à l'apparition d'internet, la population a commencée à s'équiper mais l'acquisition d'internet s'effectue au fil des années et continue toujours aujourd'hui car cet indicateur est encore en augmentation dûe à une forte population.

Le Brésil.

In [ ]:

```
BRAZ = data_winners[data_winners['Country Name'].isin(braz)]
BRAZ = BRAZ.drop(columns=['Country Name', 'Country Code', 'Indicator Name'])
BRAZ = BRAZ.set_index("Indicator Code")
BRAZ = BRAZ.T
BRAZ.plot(linewidth=4,kind='line',figsize=(20,15), title='Evolution de nos 4 indicateurs pour le Brésil (en log)').set_yscale('log')
```





La population est depuis le début des prises des données jeune mais il y a au fil des années une petite augmentation de ce nombre. Le nombre de personne dans le secondaire est difficilement exploitable dû à un gros manque de données mais on peut voir que le nombre de personnes inscrites dans le tertiaire est en augmentation. On peut donc supposer que même si il y a un manque de données, le nombre de personnes inscrites en secondaire est en constante augmentation car il est obligatoire de finir ses études dans le secondaires pour effectuer des études dans le tertiaire. Concernant l'utilisation d'internet, le nombre d'utilisateurs a fortement augmmenté pendant plusieurs années car le contexte économique est en progression au fil des années ce qui permet à la population de s'équiper.

Pays avec un fort potentiel de clients pour nos services.

Nombre de lycéens et d'étudiants par pays.

In [ ]:

```
df_countries.head()
```

Out[ ]:

	Country Name	Country Code	Region	SE.SEC.ENRL	SE.TER.ENRL	SP.POP.1564.TO	IT.NET.USER.P2
0	Albania	ALB	Europe & Central Asia	359686.5	80414.0	1941947.5	22.540719
1	Algeria	DZA	Middle East & North Africa	3800576.5	800314.5	21449356.0	6.349762
2	Argentina	ARG	Latin America & Caribbean	3939500.5	2060933.5	24484033.0	24.142241
3	Armenia	ARM	Europe & Central Asia	328578.5	117693.0	1984802.5	12.985369
4	Australia	AUS	East Asia & Pacific	2386912.5	1061062.0	13755396.5	58.391892

In [ ]:

```
df_countries['Student'] = df_countries['SE.SEC.ENRL'] + df_countries['SE.TER.ENRL']
print('liste des pays avec le plus de lycéens et étudiants')
df_countries.sort_values(by='Student', ascending=False)[['Country Name', 'Student']].hea
d(20)
```

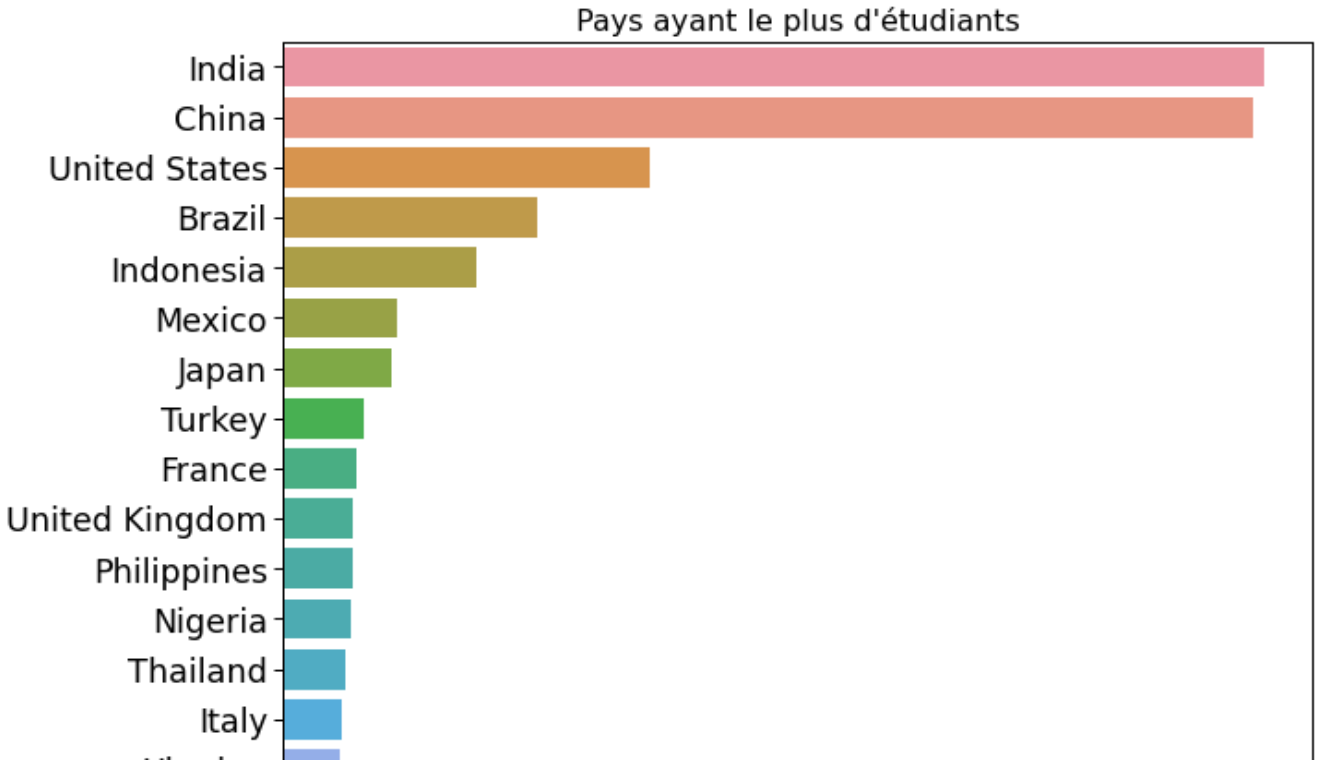
```
liste des pays avec le plus de lycéens et étudiants
```

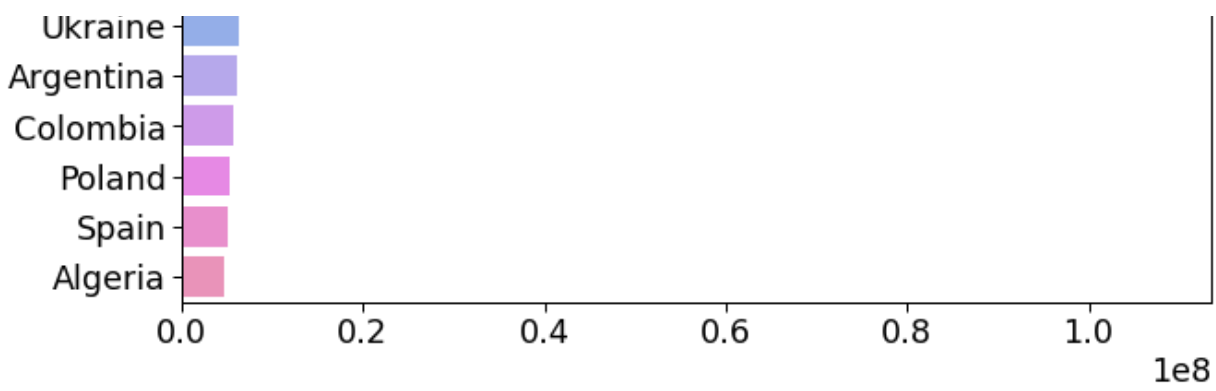
Out[ ]:

	Country Name	Student
47	India	108129118.0
22	China	107033356.5
111	United States	40417344.0
15	Brazil	28043550.0
48	Indonesia	21269591.5
68	Mexico	12544258.0
54	Japan	12016049.5
106	Turkey	9023825.0
35	France	8042944.0
110	United Kingdom	7650387.0
84	Philippines	7608960.5
77	Nigeria	7497523.5
103	Thailand	6927429.5
52	Italy	6427039.0
108	Ukraine	6359540.5
2	Argentina	6000434.0
23	Colombia	5610760.5
85	Poland	5186721.0
96	Spain	5075114.0
1	Algeria	4600891.0

In [ ]:

```
plt.figure(figsize = (10,10))
sns.set_context("paper", font_scale=2)
sns.barplot(x = df_countries.sort_values(by='Student', ascending=False)['Student'].head(
20), y=df_countries.sort_values(by='Student', ascending=False)['Country Name'].head(20))
plt.title('Pays ayant le plus d\'étudiants', size=16)
plt.xlabel(None)
plt.ylabel(None)
plt.show()
```





Nos services s'adressent en particulier aux personnes en cours de scolarité (Lycéens et étudiants, mais également aux employés, qui de nos jours, doivent sans cesse se former). Les pays qui ont le plus d'étudiants sont : L'Inde, la Chine, les Etats-Unis, le Brésil et l'Indonésie.

Estimation du nombre d'utilisateurs potentiels utilisant internet par pays.

In [ ]:

```
df_countries['futurs_client'] = df_countries['Student'] * df_countries['IT.NET.USER.P2']
df_countries.sort_values(by='futurs_client', ascending=False)[['Country Name', 'Student'
, 'IT.NET.USER.P2', 'futurs_client']].head(20)
```

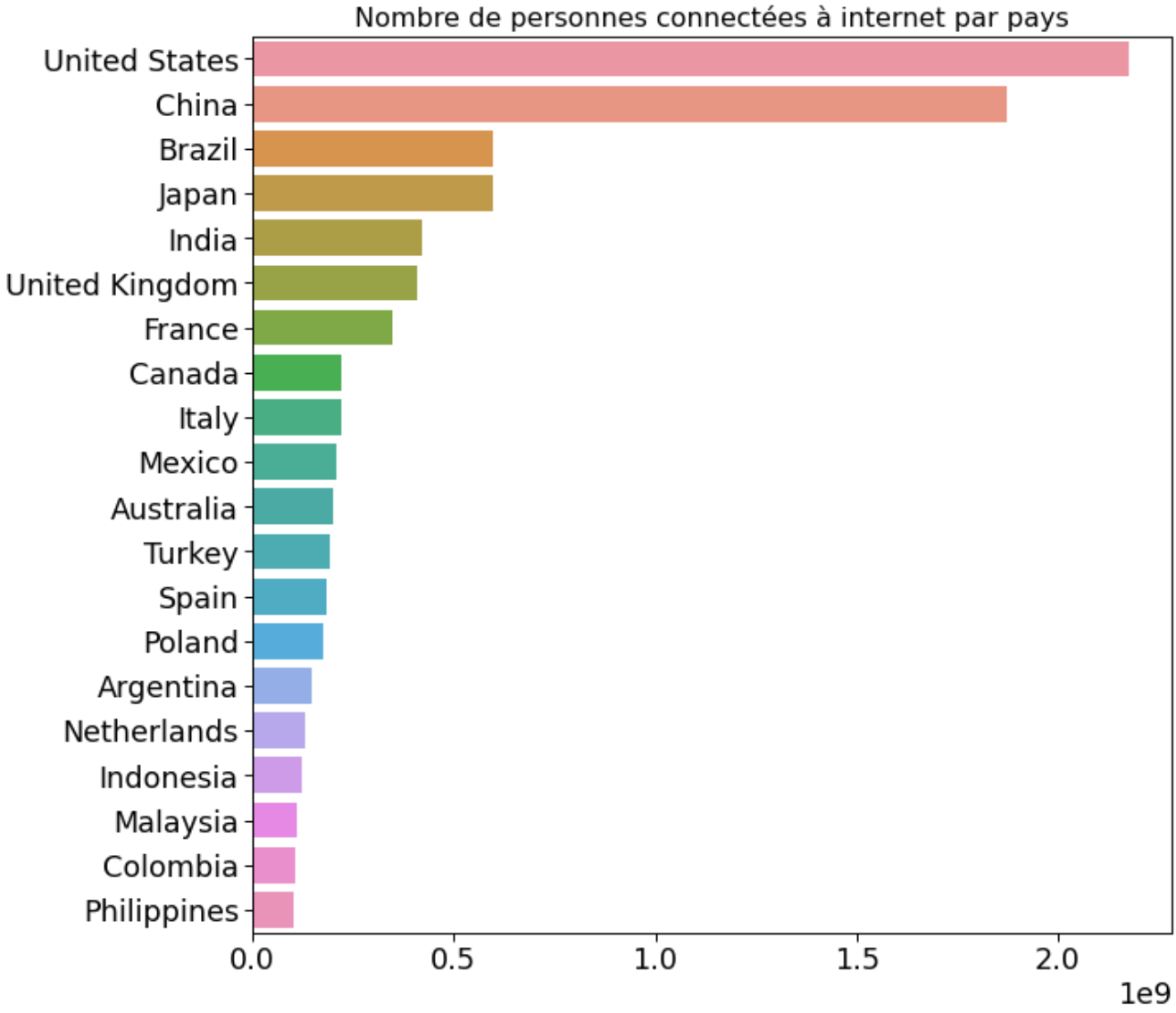
Out[ ]:

	Country Name	Student	IT.NET.USER.P2	futurs_client
111	United States	40417344.0	53.769362	2.173215e+09
22	China	107033356.5	17.504094	1.873522e+09
15	Brazil	28043550.0	21.344366	5.985718e+08
54	Japan	12016049.5	49.800645	5.984070e+08
47	India	108129118.0	3.886612	4.202559e+08
110	United Kingdom	7650387.0	53.146819	4.065937e+08
35	France	8042944.0	43.202659	3.474766e+08
19	Canada	3782219.0	58.243220	2.202886e+08
52	Italy	6427039.0	34.029205	2.187070e+08
68	Mexico	12544258.0	16.453718	2.063997e+08
4	Australia	3447974.5	58.391892	2.013338e+08
106	Turkey	9023825.0	21.056059	1.900062e+08
96	Spain	5075114.0	36.442520	1.849499e+08
85	Poland	5186721.0	33.888222	1.757688e+08
2	Argentina	6000434.0	24.142241	1.448639e+08
73	Netherlands	1980289.5	64.948004	1.286158e+08
48	Indonesia	21269591.5	5.682208	1.208582e+08
65	Malaysia	3163679.0	34.302751	1.085229e+08
23	Colombia	5610760.5	19.098865	1.071592e+08
84	Philippines	7608960.5	13.215804	1.005585e+08

In [ ]:

```
clients_potentiels = df_countries.sort_values(by='futurs_client', na_position='first', a
scending=True)[['Country Name', 'IT.NET.USER.P2', 'Student', 'futurs_client']].tail(20).
sort_values(by='futurs_client',ascending=False)
```

```
plt.figure(figsize = (10, 10))
plt.title('Nombre de personnes connectées à internet par pays', fontsize=16)
sns.barplot(x = clients_potentiels['futurs_client'], y=clients_potentiels['Country Name']
)
plt.xlabel(None)
plt.ylabel(None)
plt.show()
```



In [ ]:

```
selected_countries=clients_potentiels[clients_potentiels['futurs_client'] > 500000]['Coun
try Name'].tolist()
print(selected_countries)
```

```
['United States', 'China', 'Brazil', 'Japan', 'India', 'United Kingdom', 'France', 'Canad
a', 'Italy', 'Mexico', 'Australia', 'Turkey', 'Spain', 'Poland', 'Argentina', 'Netherland
s', 'Indonesia', 'Malaysia', 'Colombia', 'Philippines']
```

Potentiel : Evolution du pourcentage d'utilisateur d'internet par pays.

In [ ]:

```
selected_data=Stats_Data[Stats_Data['Country Name'].isin(win_countries)][Stats_Data['Indi
cator Code'].isin(indicateurs)]
```

In [ ]:

```
data_final = Stats_Data[Stats_Data['Country Name'].isin(selected_countries) & Stats_Data[
'Indicator Code'].isin(['SE.SEC.ENRL', 'SE.TER.ENRL', 'IT.NET.USER.P2'])]
```

In [ ]:

```
data_final_melt = data_final.melt(id_vars = ['Country Name', 'Country Code', 'Indicator
Name', 'Indicator Code'], value_vars = ['1970', '1971', '1972', '1973', '1974', '1975',
'1976', '1977', '1978',
       '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986', '1987',
       '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996',
       '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004', '2005',
       '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014',
       '2015', '2016', '2017', '2020', '2025', '2030', '2035', '2040', '2045',
       '2050', '2055', '2060', '2065', '2070', '2075', '2080', '2085', '2090',
       '2095', '2100'], var_name = 'Year', value_name = 'Value')
#data_final_melt
```

In [ ]:

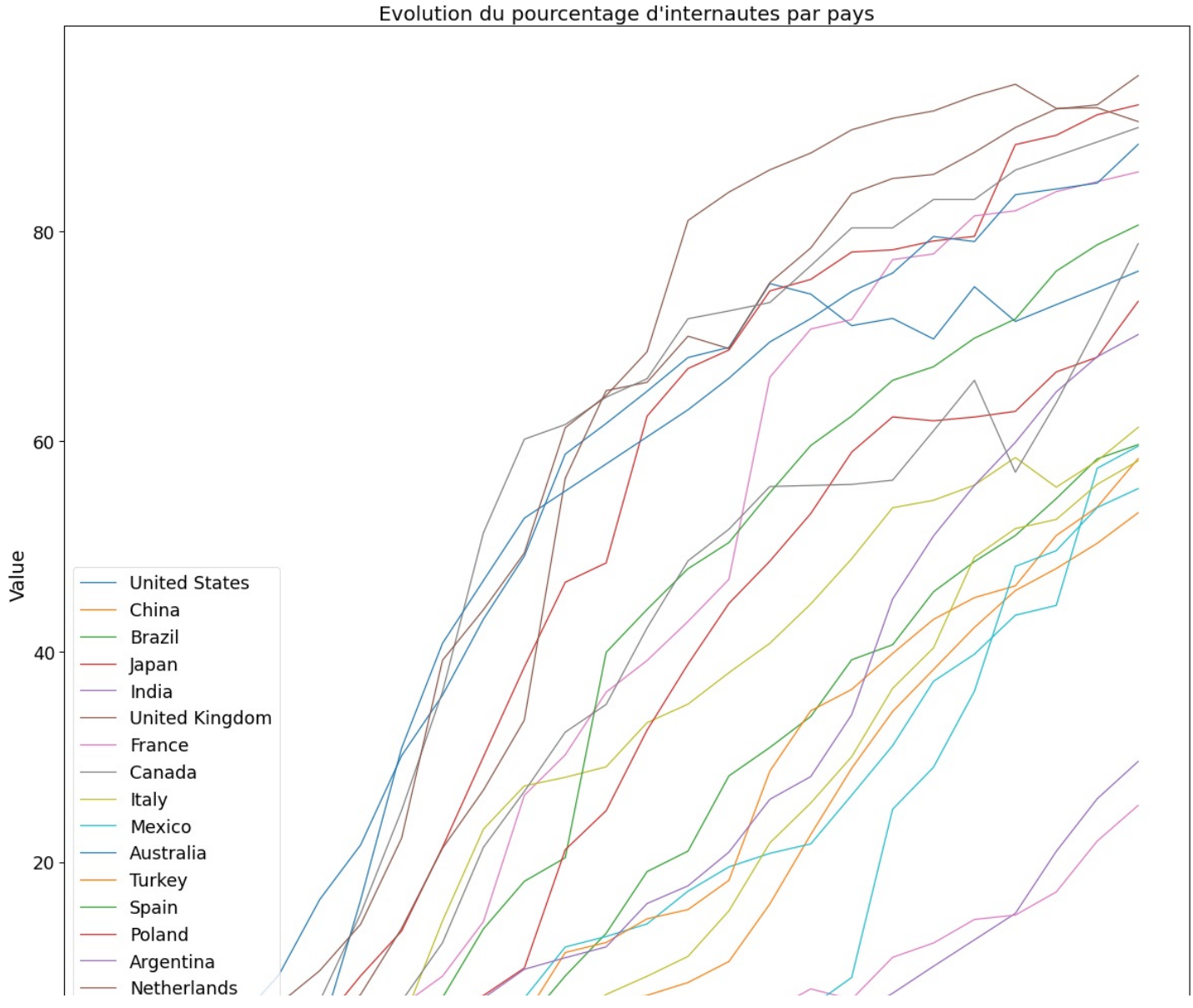
```
data_final_melt['Year'] = data_final_melt['Year'].astype('int32')
```

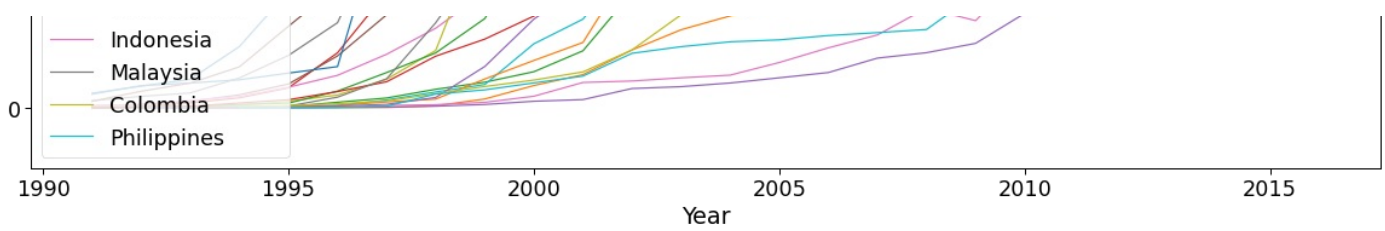
In [ ]:

```
data_final_melt = data_final_melt[data_final_melt['Year'] > 1990]
#data_final_melt
```

In [ ]:

```
plt.figure(figsize = (20,20))
for country in selected_countries:
    sns.lineplot(data_final_melt[data_final_melt['Indicator Code'] == 'IT.NET.USER.P2'][
data_final_melt['Country Name'] == country ]['Year'],
            data_final_melt[data_final_melt['Indicator Code'] == 'IT.NET.USER.P2'][data_
final_melt['Country Name'] == country ]['Value'])
plt.legend(selected_countries, loc = 'lower left', )
plt.title('Evolution du pourcentage d\'internautes par pays', size=20)
plt.show()
```





Les données ne nous donne pas directement d'information car le dataset ne nous fourni pas de prédiction. Cependant, on peut supposer :

- **que les pays qui ont une tendance haussière importante du nombre d'étudiant et de personnes entre 15 et 64 ans, vont continuer à voir cette part croitre dans les prochaines années**
- **que les pays où internet est moins implanté vont continuer à voir l'utilisation d'internet augmenter, et donc le nombre de clients**

Deuxième conclusion : Où est-il intéressant de s'implanter ?

Suite à cette seconde analyse, les pays qui sont intéressants pour le développement à l'internatonal sont : les Etats-Unis, la Chine, le Brésil, le Japon, l'Inde et le Royaume-Uni.

Conclusion

Dans quels pays développer nos services en priorité ?

Les pays qui ressortent de la première analyse sont : la Chine, l'Inde, les Etats-Unis, le Brésil et l'Indonésie.

Les pays qui ressortent de la deuxième analyse sont : les Etats-Unis, la Chine, le Brésil, le Japon, l'Inde et le Royaume-Unis.

En recoupant les résultats obtenus, il est possible de conclure qu'il serait intéressant de s'implanter en Asie de l'Est et Pacific, en Asie du Sud, en Europe et Asie centrale, en Amérique du Nord et en Latin America & Caribbean. Et plus particulièrement en Chine, en Inde, aux Etats-Unis et au Brésil .

Sur le Dataset

L'étude de ce dataset est pertinente pour répondre à la problématique car il possède de nombreuses données pour comparer les pays :

- **il y a les sources de chaque organisme pour les données récoltées**
- **Tous les pays du monde sont abordés dans cette étude**
- **Données de nature diverses : démographique, réussite aux examens, nombre d'inscriptions dans les études, taux d'utilisation d'internet.**

Cependant, le dataset contient de nombreuses données manquantes et avec les données que nous donne ce dataset, il n'est pas possible de faire des projections pour les années à venir pour nos indicateurs choisis.

Il aurait été utile pour notre problématique de savoir le taux d'utilisation d'e-learning dans chaque pays pour affiner l'analyse.