Analysez des données de systèmes éducatifs

PROJET 02/ Openclassrooms

Gulsum Kapanoglu

Problématique

Vous êtes Data Scientist dans une **start-up de la EdTech**, nommée ***academy***, qui propose des contenus de formation en ligne pour un public de niveau **lycée et université**.

Une première mission d'analyse exploratoire:

- Quels sont les pays avec un fort potentiel de clients pour nos services ?
- Pour chacun de ces pays, quelle sera l'évolution de ce potentiel de clients ?
- Dans quels pays l'entreprise doit-elle opérer en priorité ?

Mission

- Valider la qualité de ce jeu de données (comporte-t-il beaucoup de données manquantes, dupliquées ?)
- Décrire les informations contenues dans le jeu de données (nombre de colonnes ? nombre de lignes ?)
- Sélectionner les informations qui semblent pertinentes pour répondre à la problématique (quelles sont les colonnes contenant des informations qui peuvent être utiles pour répondre à la problématique de l'entreprise ?)
- Déterminer des ordres de grandeurs des indicateurs statistiques classiques pour les différentes zones géographiques et pays du monde (moyenne/médiane/écart-type par pays et par continent ou bloc géographique)

Mettre en place un environnement Python

Import des librairies

In [147]:
```
import pandas as pd
import numpy as np

import seaborn as sns
import os

import matplotlib as mpl
import matplotlib.pyplot as plt
pd.set_option('display.max_column',999)
pd.set_option('display.max_row',999)
np.set_printoptions(threshold=5)
```

In [148]:
```
path_to_file = '/Users/gulsumkapanoglu/Desktop/Edstats_csv/'
```

Import des données

In [149]:
```
all_data=pd.read_csv(path_to_file + 'EdStatsData.csv')
footnotes=pd.read_csv(path_to_file +  'EdStatsFootNote.csv')
series=pd.read_csv(path_to_file +  'EdStatsSeries.csv')
countryseries=pd.read_csv( path_to_file+  'EdStatsCountry-Series.csv')
country=pd.read_csv(path_to_file +  'EdStatsCountry.csv')
```

Décrire les informations

```
# Décrire les informations contenues dans le jeu de données
```

```
all_data.shape
```

```
(886930, 70)
```

```
all_data.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979
0	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.F	NaN	NaN	NaN	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	NaN	NaN	NaN
3	Arab World	ARB	Adjusted net enrolment rate, lower secondary, ...	UIS.NERA.2.M	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Arab World	ARB	Adjusted net enrolment rate, primary	SE.PRM.TENR	54.822121	54.894138	56.209438	57.267109	57.991138	59.36554	60.999962	61.92268	62.69342	64.383186	65

Décrire les informations

```
]: footnotes.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

```
In [153]: series.head()
```

Out[153]:

	Series Code	Topic	Indicator Name	Short definition	Long definition	Unit of measure	Periodicity	Base Period	Other notes	Aggregation method	Limitations and exceptions	Notes from original source	General comments	
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	NaN	NaN	NaN	http:
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	NaN	NaN	NaN	http:
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	NaN	NaN	NaN	http:
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	NaN	NaN	NaN	NaN	NaN	NaN	http:
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	NaN	NaN	NaN	http:

Décrire les informations

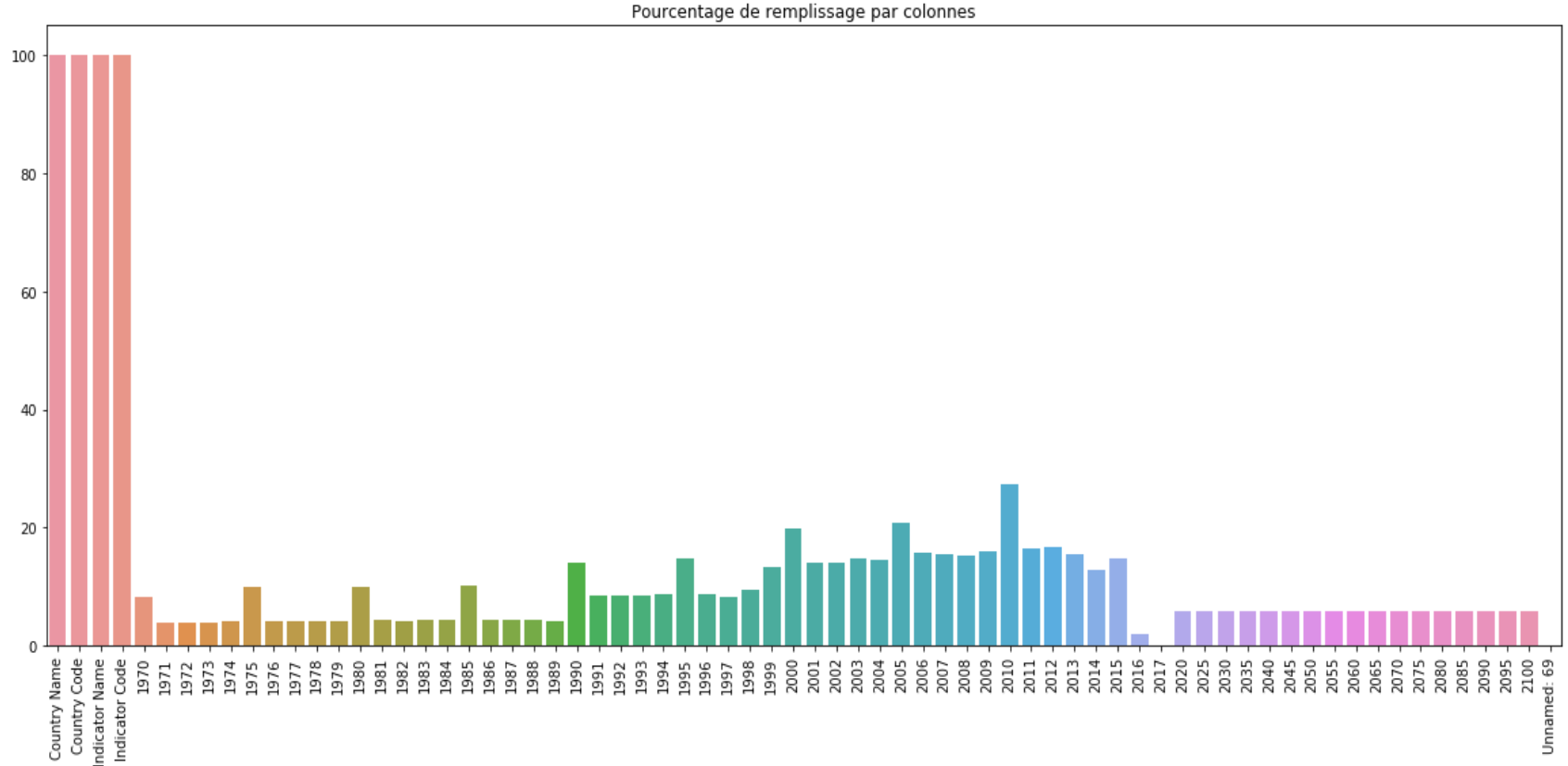
```
countryseries.head()
```

	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

```
country.head()
```

	Country Code	Short Name	Table Name	Long Name	2-alpha code	Currency Unit	Special Notes	Region	Income Group	WB-2 code	National accounts base year	National accounts reference year	SNA price valuation	Lending category	Other groups	Sy Nati Acco
0	ABW	Aruba	Aruba	Aruba	AW	Aruban florin	SNA data for 2000-2011 are updated from offici...	Latin America & Caribbean	High income: nonOECD	AW	2000	NaN	Value added at basic prices (VAB)	NaN	NaN	Co use Syste Nat Acc
1	AFG	Afghanistan	Afghanistan	Islamic State of Afghanistan	AF	Afghan afghani	Fiscal year end: March 20; reporting period fo...	South Asia	Low income	AF	2002/03	NaN	Value added at basic prices (VAB)	IDA	HIPC	Co use Syste Nat Acc
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	AO	2002	NaN	Value added at producer prices (VAP)	IBRD	NaN	Co use Syste Nat Acc
3	ALB	Albania	Albania	Republic of Albania	AL	Albanian lek	NaN	Europe & Central Asia	Upper middle income	AL	Original chained constant price data are resca...	1996.0	Value added at basic prices (VAB)	IBRD	NaN	Co use Syste Nat Acc
4	AND	Andorra	Andorra	Principality of Andorra	AD	Euro	NaN	Europe & Central Asia	High income: nonOECD	AD	1990	NaN	NaN	NaN	NaN	Co use Syste Nat Acc

Pour plus de détails



	Nb lignes	Nb colonnes	Taux remplissage moyen
EdStatsData.csv	886930	70	13.9%
EdStatsFootNote.csv	643638	5	80.0%
EdStatsSeries.csv	3665	21	28.3%
EdStatsCountry-Series.csv	613	4	75.0%
EdStatsCountry.csv	241	32	69.5%

Nettoyage des tables

Suppression des doublons

Là, on comprend qu'il n'y a pas de dublons car même nombre.

```
: all_data.drop_duplicates()
  print(all_data.shape)
```

```
(886930, 70)
```

```
filled_column=all_data.apply(lambda x: True if (x.notna().sum()/nb_row)*100 > 12 else False)
colonne=filled_column.index
values=filled_column.values
selected_colonnes=colonne[values]

all_data=all_data[selected_colonnes]
```

Les indicateurs

- Il y a 3665 indicateurs et 242 pays.
- Chaque indicateurs est disponible par années (1990, 1995,… 2010,2011..)
- Il y a des indicateurs sur le nombre d'étudiant en secondaire, en tertiaire, sur le PIB des pays etc…

Sélectionner les informations qui semblent pertinentes pour répondre à la problématique

Suppression des indicateurs inutiles

```
indicateurs_supprimés ='male|Male|primary|Primary|non-tertiary|teacher|Teacher|SABER|EGRA|PIAAC|PISA|LLECE|PASEC|TIMSS|
filter1_data = all_data.drop(all_data[all_data['Indicator Name'].str.contains(indicateurs_supprimés)].index)

print(len(filter1_data['Indicator Name'].unique().tolist()))
filter1_data['Indicator Name'].unique().tolist()
```

```
677

['Adjusted net enrolment rate, lower secondary, both sexes (%)',
 'Adjusted net enrolment rate, lower secondary, gender parity index (GPI)',
 'Adjusted net enrolment rate, upper secondary, both sexes (%)',
 'Adjusted net enrolment rate, upper secondary, gender parity index (GPI)',
 'Adult illiterate population, 15+ years, both sexes (number)',
 'Adult literacy rate, population 15+ years, both sexes (%)',
 'Adult literacy rate, population 15+ years, gender parity index (GPI)',
 'Africa Dataset: Percentage of lower secondary schools with access to electricity (%)',
 'Africa Dataset: Percentage of lower secondary schools with access to potable water (%)',
 'Africa Dataset: Percentage of lower secondary schools with mixed-sex toilets (%)',
 'Africa Dataset: Percentage of lower secondary schools with no information on electricity (%)',
 'Africa Dataset: Percentage of lower secondary schools with no information on potable water (%)',
 'Africa Dataset: Percentage of lower secondary schools with no information on toilets (%)',
 'Africa Dataset: Percentage of lower secondary schools with single-sex toilets (%)',
 'Africa Dataset: Percentage of lower secondary schools with toilets (%)',
 'Africa Dataset: Percentage of lower secondary schools without access to electricity (%)',
 'Africa Dataset: Percentage of lower secondary schools without access to potable water (%)',
 'Africa Dataset: Percentage of lower secondary schools without toilets (%)'
```

Les indicateurs qui nous intéressent

```
64

['Barro-Lee: Average years of secondary schooling, age 15-19, total',
 'Barro-Lee: Average years of secondary schooling, age 20-24, total',
 'Barro-Lee: Average years of tertiary schooling, age 15-19, total',
 'Barro-Lee: Average years of tertiary schooling, age 20-24, total',
 'Barro-Lee: Average years of total schooling, age 15-19, total',
 'Barro-Lee: Average years of total schooling, age 20-24, total',
 'Barro-Lee: Percentage of population age 15-19 with no education',
 'Barro-Lee: Percentage of population age 15-19 with secondary schooling. Completed Secondary',
 'Barro-Lee: Percentage of population age 15-19 with secondary schooling. Total (Incomplete and Completed Secondar
y)',
 'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Completed Tertiary',
 'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Total (Incomplete and Completed Tertiary)',
 'Barro-Lee: Percentage of population age 20-24 with no education',
 'Barro-Lee: Percentage of population age 20-24 with secondary schooling. Completed Secondary',
 'Barro-Lee: Percentage of population age 20-24 with secondary schooling. Total (Incomplete and Completed Secondar
y)',
 'Barro-Lee: Percentage of population age 20-24 with tertiary schooling. Completed Tertiary',
 'Barro-Lee: Percentage of population age 20-24 with tertiary schooling. Total (Incomplete and Completed Tertiary)',
 'Barro-Lee: Population in thousands, age 15-19, total',
 'Barro-Lee: Population in thousands, age 20-24, total',
 'DHS: Secondary completion rate',
 'DHS: Secondary completion rate. Quintile 1',
 'DHS: Secondary completion rate. Quintile 2',
 'DHS: Secondary completion rate. Quintile 3',
 'DHS: Secondary completion rate. Quintile 4',
 'DHS: Secondary completion rate. Quintile 5',
 'DHS: Secondary completion rate. Rural',
 'DHS: Secondary completion rate. Urban',
 'Enrolment in tertiary education per 100,000 inhabitants, both sexes',
 'GDP at market prices (constant 2005 US$)',
 'GDP at market prices (current US$)',
 'GDP per capita (constant 2005 US$)',
 'GDP per capita (current US$)',
 'GDP per capita, PPP (constant 2011 international $)',
```

Liste des indicateurs

```
nb_col=filter2_data.shape[1]
filter2_data.set_index(['Indicator Name'])
filter2_data['choosen'] = filter2_data.apply(lambda x:(x.notna().sum()/nb_col*100),axis=1)
```

```
filter2_data.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1990	1995	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	Arab World	ARB	Barro-Lee: Average years of secondary schoolin...	BAR.SEC.SCHL.1519	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Arab World	ARB	Barro-Lee: Average years of secondary schoolin...	BAR.SEC.SCHL.2024	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Arab World	ARB	Barro-Lee: Average years of tertiary schooling...	BAR.TER.SCHL.1519	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Arab World	ARB	Barro-Lee: Average years of tertiary schooling...	BAR.TER.SCHL.2024	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Arab World	ARB	Barro-Lee: Average years of total schooling, a...	BAR.SCHL.1519	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Jointure entre Filter_data et Country

```
column_list = ['Country Code','Region','Income Group']
country_filter = country[column_list]
filter2_data=filter2_data.merge(country_filter,on='Country Code',how='left')
```

```
print(len(filter2_data['Indicator Name'].unique()))
print(len(filter2_data['Country Name'].unique()))
```

```
64
242
```

```
filter2_data["choosen"].median()
```

```
26.08695652173913
```

Jointure entre Filter_data et Country

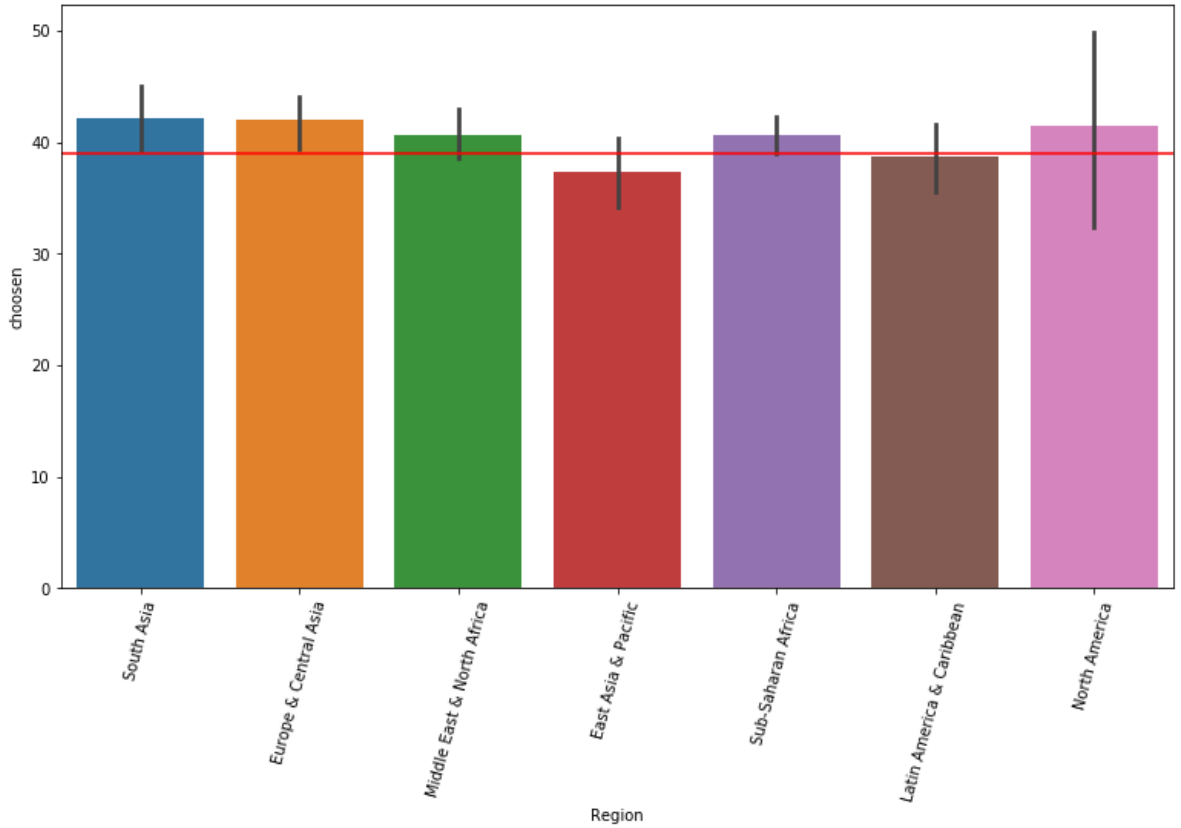
```
examine_par_pays=filter2_data.groupby(by=['Country Name','Region'])['choosen'].sum()/6400*100
examine_par_pays=pd.DataFrame(examine_par_pays)
examine_par_pays.reset_index(inplace=True)
examine_par_pays
```

	Country Name	Region	choosen
0	Afghanistan	South Asia	35.122283
1	Albania	Europe & Central Asia	42.391304
2	Algeria	Middle East & North Africa	41.847826
3	American Samoa	East Asia & Pacific	21.263587
4	Andorra	Europe & Central Asia	30.366848
5	Angola	Sub-Saharan Africa	32.404891
6	Antigua and Barbuda	Latin America & Caribbean	31.589674
7	Argentina	Latin America & Caribbean	50.135870
8	Armenia	Europe & Central Asia	45.448370
9	Aruba	Latin America & Caribbean	30.298913
10	Australia	East Asia & Pacific	47.350543

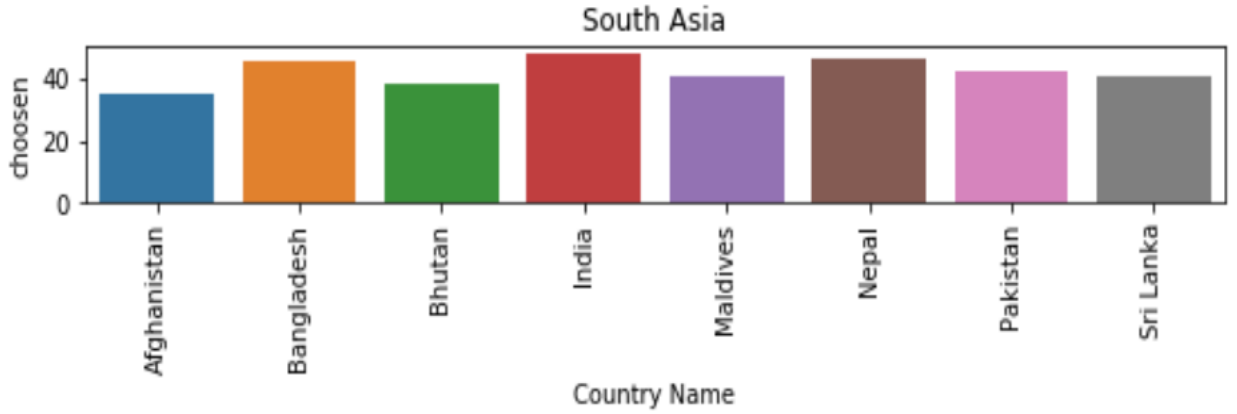
Jointure entre Filter_data et Country

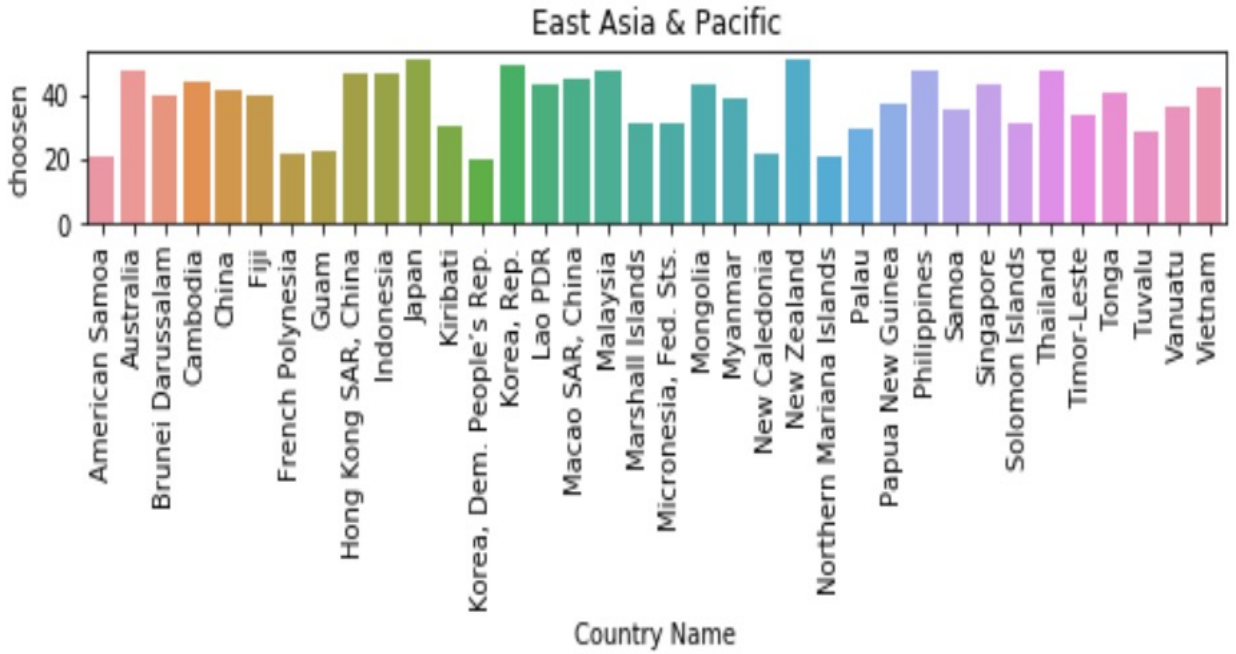
```
plt.subplots(figsize=(13,7))

sns.barplot(x = 'Region', y = 'choosen', data= examine_par_pays)
plt.xticks(rotation=75)
plt.axhline(y=39, color='red')
plt.show()
```

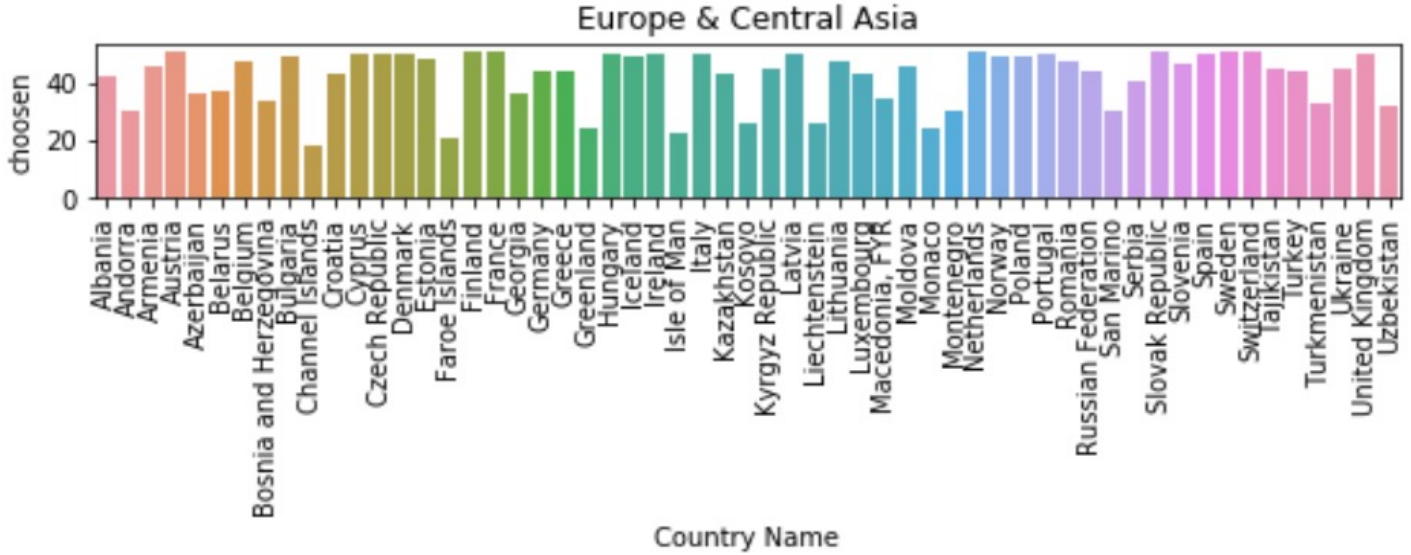


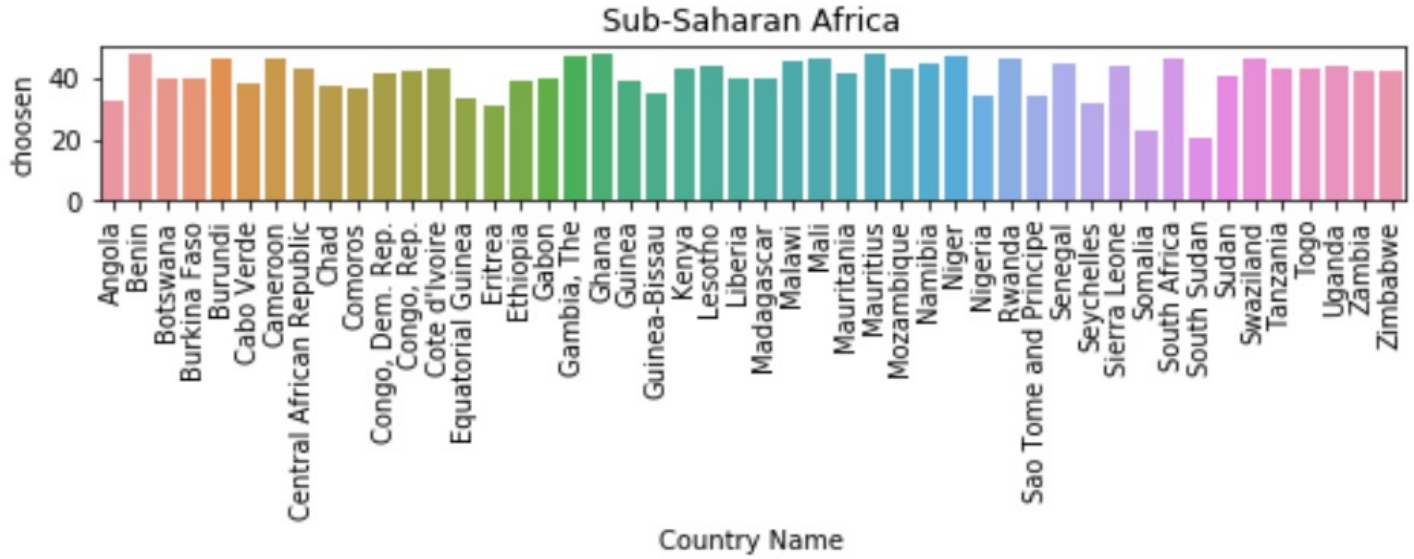
Les indicateurs par pays et régions

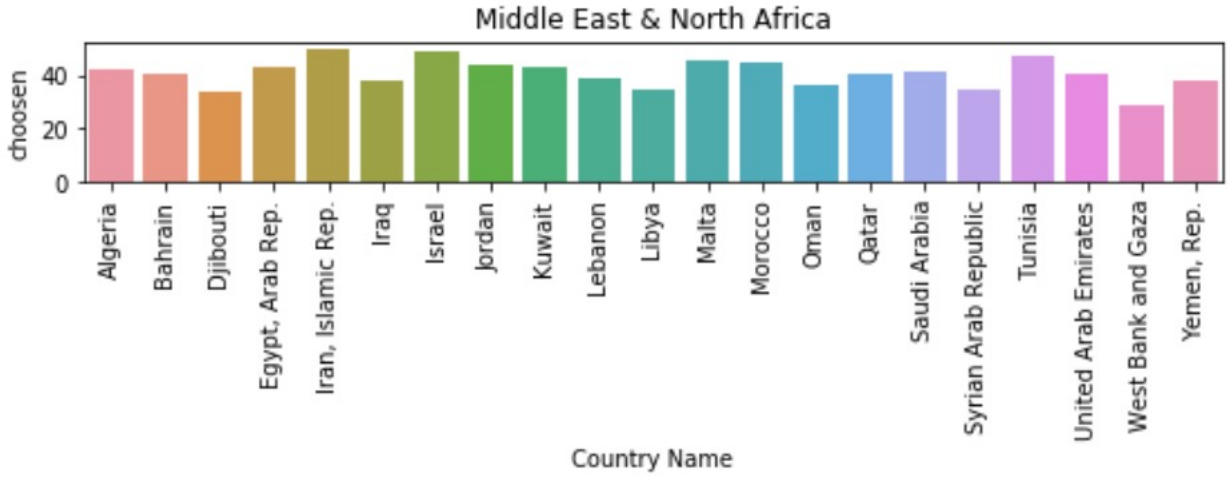


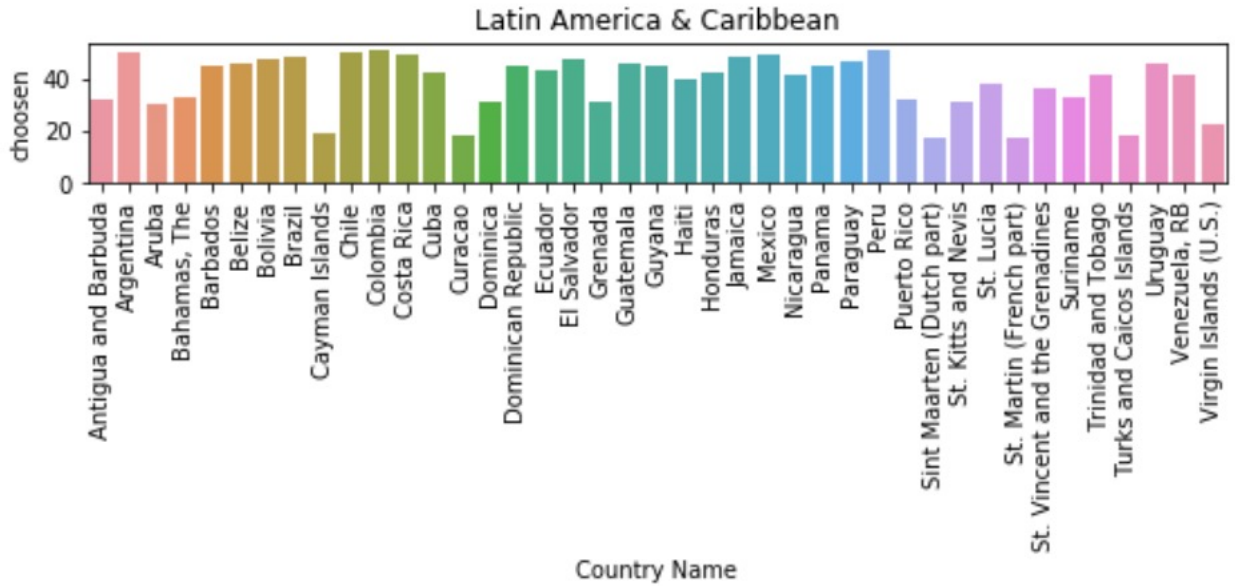


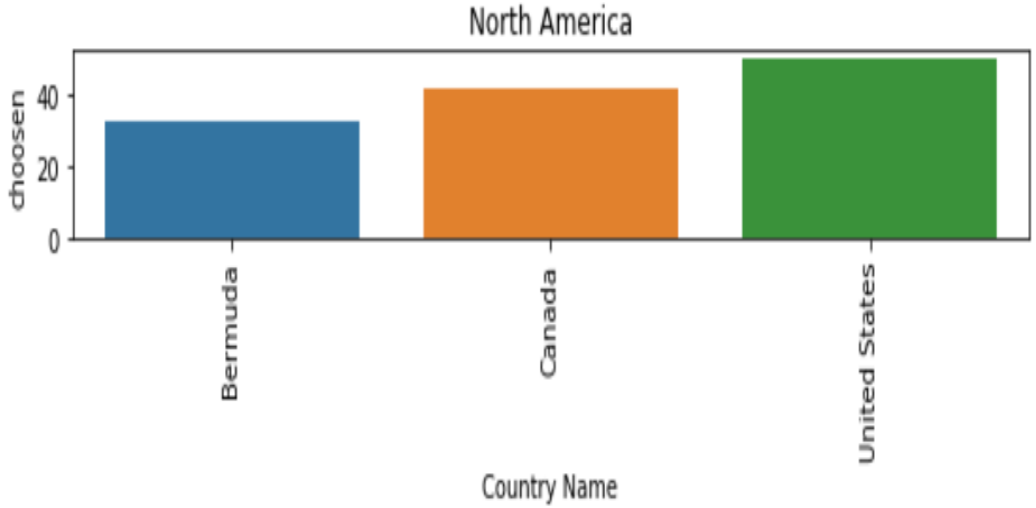
Les indicateurs par pays et régions






Middle East & North Africa
choosen
0
20
40
Algeria
Bahrain
Djibouti
Egypt, Arab Rep.
Iran, Islamic Rep.
Iraq
Israel
Jordan
Kuwait
Lebanon
Libya
Malta
Morocco
Oman
Qatar
Saudi Arabia
Syrian Arab Republic
Tunisia
United Arab Emirates
West Bank and Gaza
Yemen, Rep.
Country Name


Latin America & Caribbean
choosen
0
20
40
Antigua and Barbuda
Argentina
Aruba
Bahamas, The
Barbados
Belize
Bolivia
Brazil
Cayman Islands
Chile
Colombia
Costa Rica
Cuba
Curacao
Dominica
Dominican Republic
Ecuador
El Salvador
Grenada
Guatemala
Guyana
Haiti
Honduras
Jamaica
Mexico
Nicaragua
Panama
Paraguay
Peru
Puerto Rico
Sint Maarten (Dutch part)
St. Kitts and Nevis
St. Lucia
St. Martin (French part)
St. Vincent and the Grenadines
Suriname
Trinidad and Tobago
Turks and Caicos Islands
Uruguay
Venezuela, RB
Virgin Islands (U.S.)
Country Name


North America
choosen
0
20
40
Bermuda
Canada
United States
Country Name

Sélection les Pays et Régions

```
pays_selectionnes=examine_par_pays[examine_par_pays['choosen']>40]
pays_selectionnes=pays_selectionnes['Country Name']

pays_selectionnes;
```

```
pays_selectionnes.unique().tolist()
```

```
['Albania',
 'Algeria',
 'Argentina',
 'Armenia',
 'Australia',
```

```
region_selectionnes=examine_par_pays[examine_par_pays['choosen']>40]
region_selectionnes=region_selectionnes['Region']

region_selectionnes;
```

```
region_selectionnes.unique().tolist()
```

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

Filtre des indicateurs

```
filter3_data=filter2_data[filter2_data['Country Name'].isin(pays_selectionnes)]
len(filter3_data['Country Name'].unique())
```

```
133
```

```
examine_par_pays_ind=filter3_data.groupby(by='Indicator Name')['choosen'].sum()/13300*100
examine_par_pays_ind.to_frame()
```

Indicator Name	choosen
Barro-Lee: Average years of secondary schooling, age 15-19, total	39.130435
Barro-Lee: Average years of secondary schooling, age 20-24, total	39.130435
Barro-Lee: Average years of tertiary schooling, age 15-19, total	39.130435
Barro-Lee: Average years of tertiary schooling, age 20-24, total	39.130435
Barro-Lee: Average years of total schooling, age 15-19, total	39.130435
Barro-Lee: Average years of total schooling, age 20-24, total	39.130435
Barro-Lee: Percentage of population age 15-19 with no education	39.130435
Barro-Lee: Percentage of population age 15-19 with secondary schooling. Completed Secondary	39.130435
Barro-Lee: Percentage of population age 15-19 with secondary schooling. Total (Incomplete and Completed Secondary)	39.130435
Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Completed Tertiary	39.130435

```
filter3_data['choosen'].median()
```

```
39.130434782608695
```

Liste des indicateurs

```
ind_list=examine_par_pays_ind[examine_par_pays_ind>39]
ind_list=ind_list.index.tolist()
ind_list
```

```
['Barro-Lee: Average years of secondary schooling, age 15-19, total',
 'Barro-Lee: Average years of secondary schooling, age 20-24, total',
 'Barro-Lee: Average years of tertiary schooling, age 15-19, total',
 'Barro-Lee: Average years of tertiary schooling, age 20-24, total',
 'Barro-Lee: Average years of total schooling, age 15-19, total',
 'Barro-Lee: Average years of total schooling, age 20-24, total',
 'Barro-Lee: Percentage of population age 15-19 with no education',
 'Barro-Lee: Percentage of population age 15-19 with secondary schooling. Completed Secondary',
 'Barro-Lee: Percentage of population age 15-19 with secondary schooling. Total (Incomplete and Completed Secondar
y)',
 'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Completed Tertiary',
 'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Total (Incomplete and Completed Tertiary)',
 'Barro-Lee: Percentage of population age 20-24 with no education',
 'Barro-Lee: Percentage of population age 20-24 with secondary schooling. Completed Secondary',
 'Barro-Lee: Percentage of population age 20-24 with secondary schooling. Total (Incomplete and Completed Secondar
y)',
 'Barro-Lee: Percentage of population age 20-24 with tertiary schooling. Completed Tertiary',
 'Barro-Lee: Percentage of population age 20-24 with tertiary schooling. Total (Incomplete and Completed Tertiary)',
 'Barro-Lee: Population in thousands, age 15-19, total',
 'Barro-Lee: Population in thousands, age 20-24, total',
 'Enrolment in tertiary education per 100,000 inhabitants, both sexes',
 'GDP at market prices (constant 2005 US$)',
 'GDP at market prices (current US$)',
 'GDP per capita (constant 2005 US$)',
 'GDP per capita (current US$)',
 'GDP per capita, PPP (constant 2011 international $)',
 'GDP per capita, PPP (current international $)',
 'GDP, PPP (constant 2011 international $)',
 'GDP, PPP (current international $)',
 'Government expenditure in educational institutions as % of GDP (%)',
 'Government expenditure in secondary institutions education as % of GDP (%)',
 'Government expenditure in tertiary institutions as % of GDP (%)',
 'Government expenditure on education as % of GDP (%)',
 'Government expenditure on secondary education as % of GDP (%)',
 'Government expenditure on tertiary education as % of GDP (%)',
 'Government expenditure per lower secondary student as % of GDP per capita (%)',
 'Government expenditure per secondary student as % of GDP per capita (%)',
 'Government expenditure per tertiary student as % of GDP per capita (%)',
```

Déterminer des ordres de grandeurs des indicateurs statistiques classiques pour les différentes zones géographiques et pays du monde

Deuxième Sélection d'indicateurs

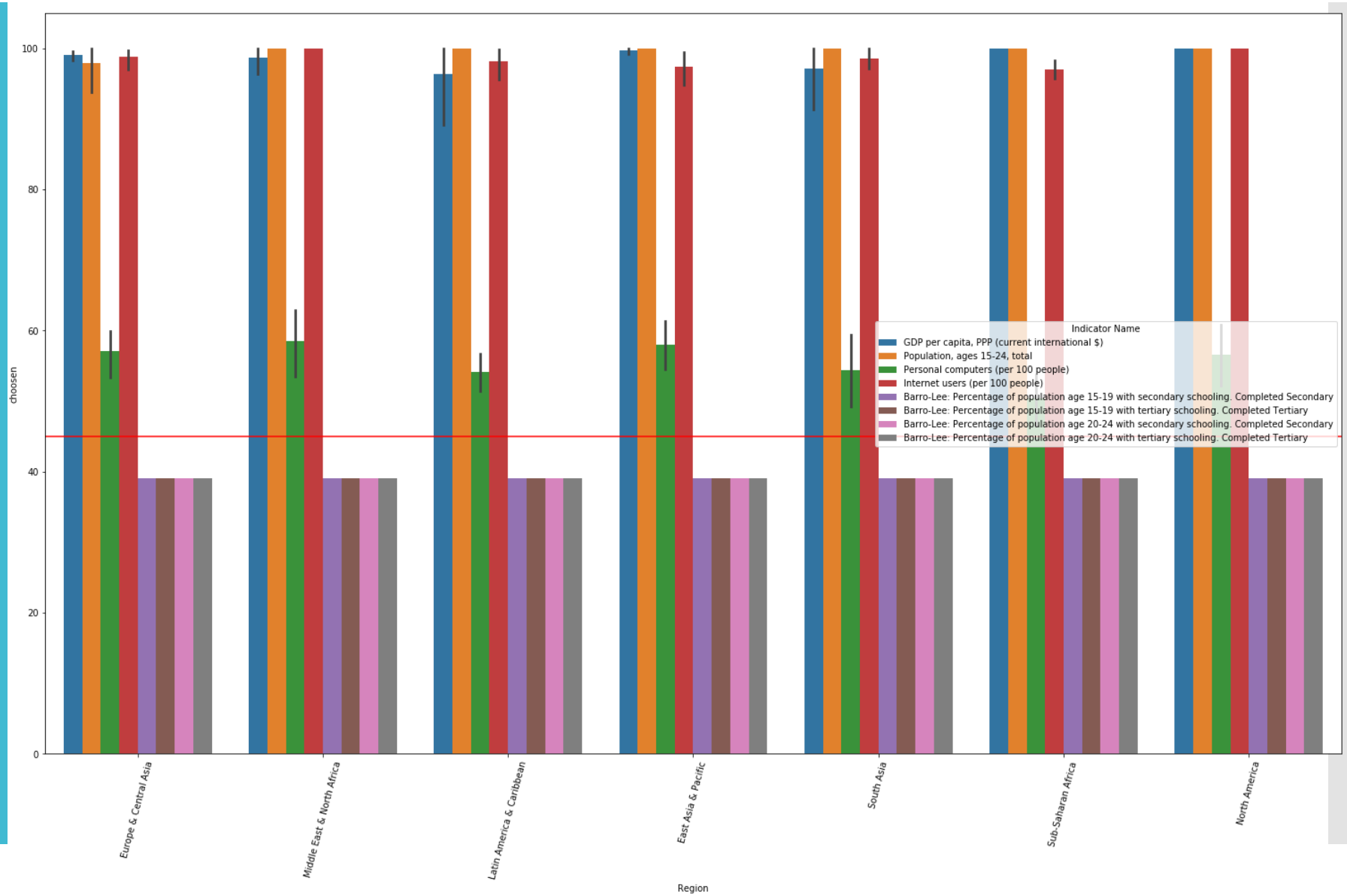
- *Internet users (per 100 people),Personal computers (per 100 people) : La condition la plus importante pour l'éducation en ligne est possesion d'un ordinateur personnel et l'accès à Internet.*
- *Pour Economique : GPD per capita: Cibler les pays à fort pouvoir d'achat*
- *Pour Demographie et education : On retient les données concernant la tranche d'age des 15-24 car notre cible est les lycéens et étudiants*

Au final, il nous reste 8 indicateurs.

```
8
['GDP per capita, PPP (current international $)', 'Population, ages 15-24, total', 'Personal computers (per 100 peopl
e)', 'Internet users (per 100 people)', 'Barro-Lee: Percentage of population age 15-19 with secondary schooling. Comp
leted Secondary', 'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Completed Tertiary', 'Barro
-Lee: Percentage of population age 20-24 with secondary schooling. Completed Secondary', 'Barro-Lee: Percentage of po
pulation age 20-24 with tertiary schooling. Completed Tertiary']
```

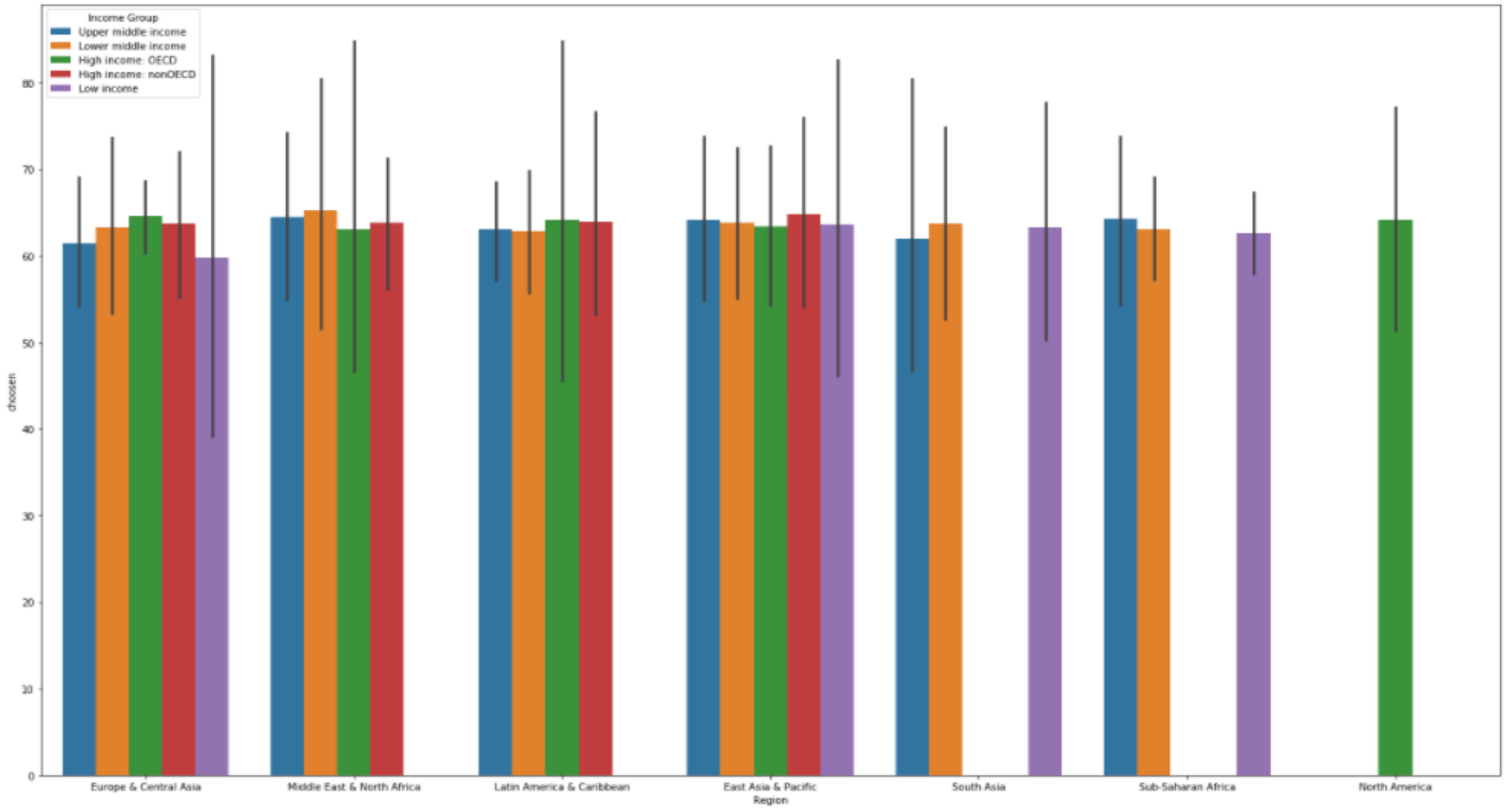
	Country Name	Country Code	Indicator Name	Indicator Code	1990	1995	1999	2000	2001	2002	20
0	Albania	ALB	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	2721.615212	2781.413989	3690.688729	4026.537422	4463.632986	4754.675856	5114.7847
1	Algeria	DZA	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	6616.408352	6777.297778	7725.883722	8093.530893	8416.752682	8911.680680	9621.1609
2	Argentina	ARG	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	6990.554045	10129.777793	11769.149975	11810.061364	11419.058866	10217.273100	11217.5719
3	Armenia	ARM	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	2418.860926	1584.600470	2126.915169	2318.238073	2613.784567	3020.453987	3531.9686
4	Australia	AUS	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	17373.768226	21005.202395	25299.359767	26406.130951	27431.075399	28717.289203	29723.6838

Les indicateurs sélectionnés selon Régions



Les indicateurs sélectionnés selon Income Group et Régions

```
f, ax = plt.subplots(figsize = (26,15))
sns.barplot(x = "Region", y = "choosen", hue = "Income Group", data = final_data);
color="salmon"
```



Sélection sur les pays riches

```
final_data['Income Group'].unique()
```

```
array(['Upper middle income', 'Lower middle income', 'High income: OECD',
       'High income: nonOECD', 'Low income'], dtype=object)
```

```
values=['High income: nonOECD', 'High income: OECD']
final_data = final_data[final_data['Income Group'].isin(values)]
```

```
view = final_data.set_index(['Region','Income Group','Country Name']).sort_index()
view
```

Region	Income Group	Country Name	index	Country Code	Indicator Name	Indicator Code	1990	1995	1999	2000	2001
East Asia & Pacific	High income: OECD	Australia	4	AUS	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	1.737377e+04	2.100520e+04	2.529936e+04	2.640613e+04	2.743108e+0
		Australia	137	AUS	Population, ages 15-24, total	SP.POP.1524.TO.UN	2.733117e+06	2.698761e+06	2.618375e+06	2.621168e+06	2.641779e+0
		Australia	270	AUS	Personal computers (per 100 people)	IT.CMP.PCMP.P2	1.495011e+01	2.747109e+01	4.204455e+01	4.673048e+01	5.131559e+0
					Internet						

```
len(final_data['Country Name'].unique())
```

```
48
```

Affectation de poids à chaque indicateur

```
final_data['Indicator Name'].unique()
```

```
array(['GDP per capita, PPP (current international $)',
       'Population, ages 15-24, total',
       'Personal computers (per 100 people)', ...,
       'Barro-Lee: Percentage of population age 15-19 with tertiary schooling. Completed Tertiary',
       'Barro-Lee: Percentage of population age 20-24 with secondary schooling. Completed Secondary',
       'Barro-Lee: Percentage of population age 20-24 with tertiary schooling. Completed Tertiary'],
      dtype=object)
```

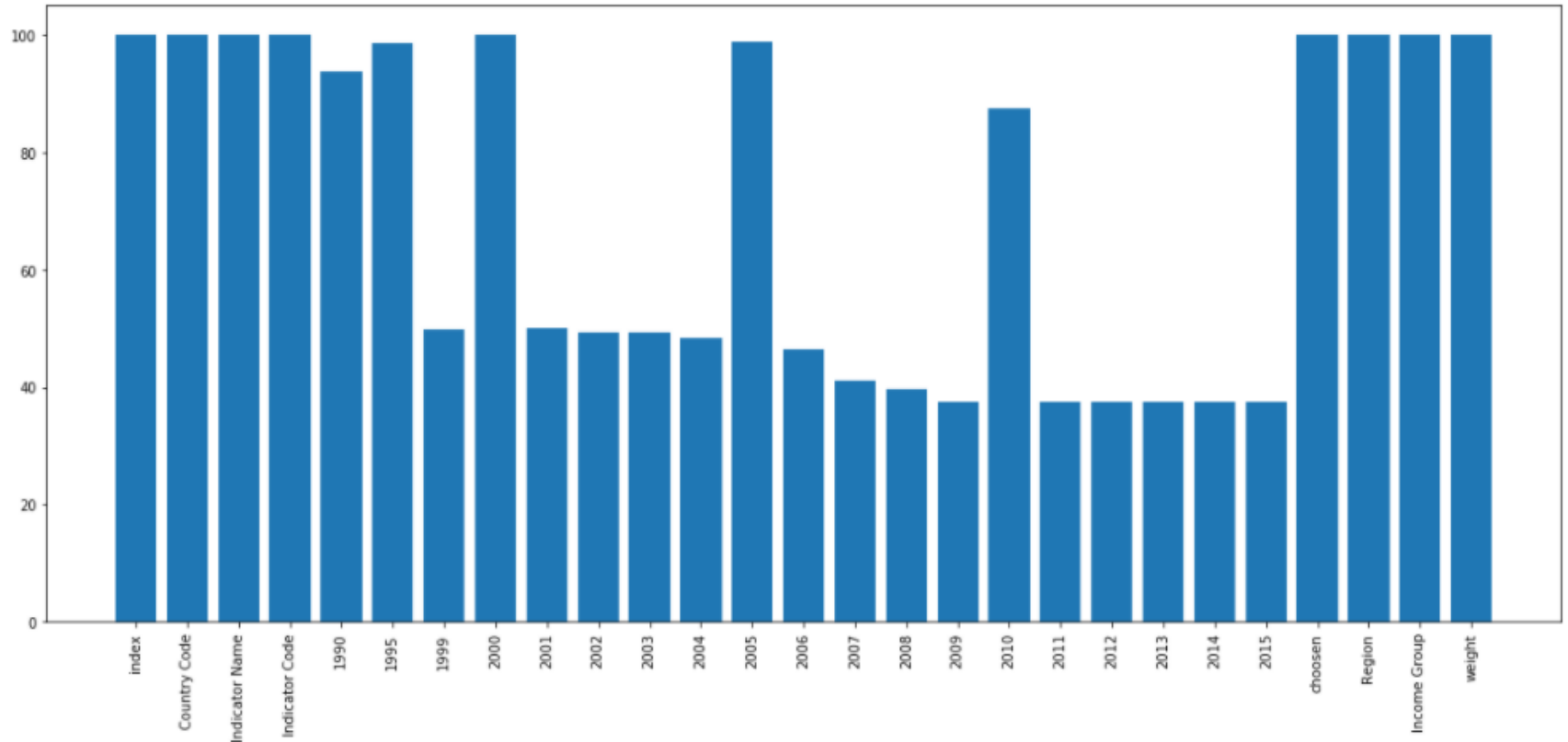
Nous allons maintenant créer un score entre 1 et 3 pour noter les indicators

```
iw = np.array([['GDP per capita, PPP (current international $)',2],['Personal computers (per 100 people)',2],['Internet
indicator_weight = pd.DataFrame(iw, index = [1,2,3,4,5,6,7,8], columns = ['Indicator Name', 'weight'])
indicator_weight['weight'] = pd.to_numeric(indicator_weight['weight'])
indicator_weight
```

	Indicator Name	weight
1	GDP per capita, PPP (current international $)	2
2	Personal computers (per 100 people)	2
3	Internet users (per 100 people)	2
4	Population, ages 15-24, total	2
5	Barro-Lee: Percentage of population age 15-19 ...	1
6	Barro-Lee: Percentage of population age 15-19 ...	1
7	Barro-Lee: Percentage of population age 20-24 ...	1
8	Barro-Lee: Percentage of population age 20-24 ...	1

Remplissage par colonnes dans le tableau filtré

```
nb_row=final_data.shape[0]
percentage=final_data.apply(lambda x:(x.notna().sum()/nb_row)*100)
columns=final_data.columns
plt.figure(figsize=(20,8))
plt.xticks(rotation=90)
plt.bar(columns,percentage,)
plt.show()
```



Sélectionner les informations qui semblent pertinentes pour répondre à la problématique

Calcul d'un score pour chaque pays par année

```
result_temp=final_data.apply(lambda x: x*final_data['weight'] if x.dtype=='float64' else x)
```

```
score1=result_temp.groupby('Country Name').sum().apply(lambda x: x/12 if x.dtype=='float64' else x)
col = ['Country Code','Region','Income Group']

score2=final_data[col].drop_duplicates()
score=score1.merge(score2,on='Country Name',how='left')
score
```

Country Name	index	1990	1995	1999	2000	2001	2002	2003	2004	2005	2006
Australia	3012	4.584283e+05	4.533082e+05	4.406262e+05	4.412904e+05	4.448857e+05	4.510703e+05	4.587693e+05	4.666712e+05	4.735758e+05	4.792566
Austria	3032	1.986700e+05	1.737674e+05	1.639594e+05	1.638533e+05	1.645403e+05	1.663097e+05	1.685305e+05	1.708755e+05	1.727621e+05	1.743149
Bahrain	3052	1.710988e+04	2.054694e+04	2.381112e+04	2.451165e+04	2.483821e+04	2.502099e+04	2.525290e+04	2.556389e+04	2.593219e+04	2.631878
Barbados	3092	1.021830e+04	9.446155e+03	9.118678e+03	9.163837e+03	9.117981e+03	9.167845e+03	9.279919e+03	9.393173e+03	9.559971e+03	9.741676

Filtrage par année 2015

```
global_seuil = score['2015'].quantile([0.90]).at[0.90]
print(global_seuil)

selected_global_country=score[score['2015']>= global_seuil]

selected_global_country.reset_index(drop=False, inplace=True)

selected_global_country=selected_global_country.set_index(['Region','Income Group','Country Name'])['2015']

selected_global_country=selected_global_country.to_frame()

selected_global_country.sort_index(axis=0)

selected_global_country['degree']=selected_global_country['2015'].rank(method='max',ascending=False)

selected_global_country
```

```
1276419.330180109
```

			2015	degree
Region	Income Group	Country Name		
Europe & Central Asia	High income: OECD	Germany	1.455109e+06	4.0
East Asia & Pacific	High income: OECD	Japan	2.033034e+06	3.0
Europe & Central Asia	High income: nonOECD	Russian Federation	2.447627e+06	2.0
	High income: OECD	United Kingdom	1.295623e+06	5.0
North America	High income: OECD	United States	7.534010e+06	1.0

Quantile

```
filter_seuil=score.groupby(by=['Region','Income Group','Country Name'])['2015'].quantile([0.90]).to_frame()
filter_seuil
```

				2015
Region	Income Group	Country Name		
East Asia & Pacific	High income: OECD	Australia	0.9	4.935301e+05
		Japan	0.9	2.033034e+06
		Korea, Rep.	0.9	1.081845e+06
		New Zealand	0.9	1.088560e+05
	High income: nonOECD	Hong Kong SAR, China	0.9	1.523163e+05
		Macao SAR, China	0.9	2.831354e+04
		Singapore	0.9	1.275819e+05
Europe & Central Asia	High income: OECD	Austria	0.9	1.691334e+05
		Belgium	0.9	2.146101e+05
		Czech Republic	0.9	1.829663e+05
		Denmark	0.9	1.265166e+05
		Estonia	0.9	2.790007e+04
		Finland	0.9	1.136278e+05
		France	0.9	1.268189e+06
		Germany	0.9	1.455109e+06
		Greece	0.9	1.900870e+05
		Iceland	0.9	1.539310e+04

Lister des Pays

```
selectionnes_pay=country_by_region_and_income[country_by_region_and_income['degree']>80]
selectionnes_pay=selectionnes_pay[['degree']]

selectionnes_pay;
```

```
main_country1 = selectionnes_pay.index.tolist()
main_country2 = selected_global_country.index.tolist()
main_country=[x[2] for x in main_country1]+[x[2] for x in main_country2]
main_country=list(set(main_country))
main_country
```

```
['Spain',
 'Latvia',
 'Slovak Republic',
 'Estonia',
 'Denmark',
 'Greece',
 'United Kingdom',
 'Sweden',
 'Italy',
 'Russian Federation',
 'United States',
 'Czech Republic',
 'Portugal',
 'Switzerland',
 'Norway',
 'Netherlands',
 'Austria',
 'Finland',
 'Ireland',
 'Luxembourg',
 'Germany',
 'Slovenia',
 'Poland',
 'Croatia',
 'Belgium',
 'Lithuania',
 'Japan']
```

Quelle sera l'évolution de ce potentiel de clients ?

- 'Wittgenstein Projection: Population age 15-19 in thousands by highest level of educational attainment. Post Secondary. Total',
- 'Wittgenstein Projection: Population age 15-19 in thousands by highest level of educational attainment. Upper Secondary. Total',
- 'Wittgenstein Projection: Population age 20-24 in thousands by highest level of educational attainment. Post Secondary. Total',
- 'Wittgenstein Projection: Population age 20-24 in thousands by highest level of educational attainment. Upper Secondary. Total']

Evolution

```
evolution=all_data[all_data['Country Name'].isin(main_country)]
evolution_ind=[
 'Wittgenstein Projection: Population age 15-19 in thousands by highest level of educational attainment. Post Secondary
 'Wittgenstein Projection: Population age 15-19 in thousands by highest level of educational attainment. Upper Secondar
 'Wittgenstein Projection: Population age 20-24 in thousands by highest level of educational attainment. Post Secondary
 'Wittgenstein Projection: Population age 20-24 in thousands by highest level of educational attainment. Upper Secondar
evolution=evolution[evolution['Indicator Name'].isin(evolution_ind)]
evolution=evolution.groupby(by=['Country Name']).sum()
nb_row=evolution.shape[0]
colonne=evolution.columns

values=[False if x==0 else True for x in np.count_nonzero(evolution, axis=0)]

selected_colonnes=colonne[values]
evolution=evolution[selected_colonnes]

evolution
```

Evolution

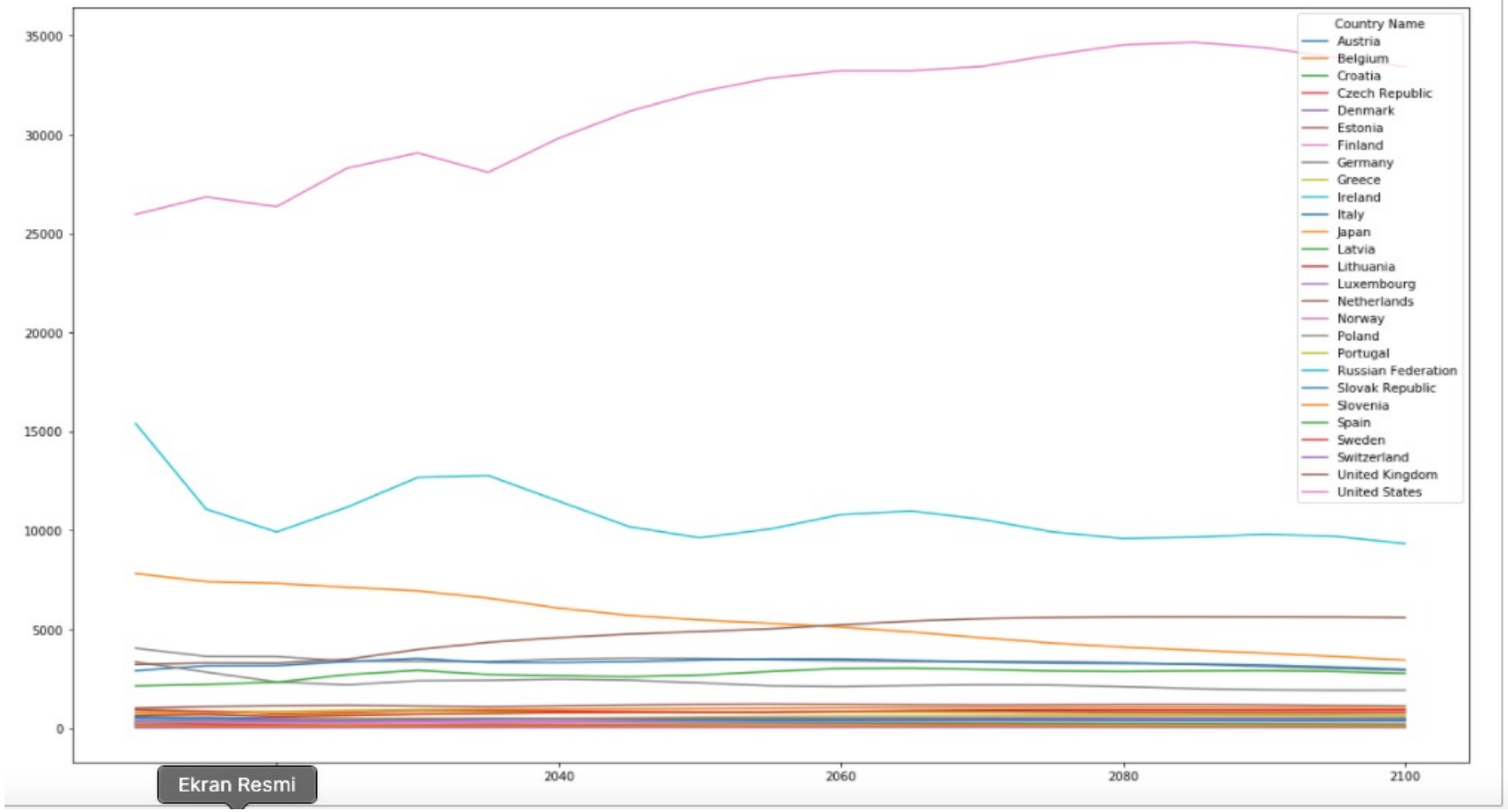
Country Name	2010	2015	2020	2025	2030	2035	2040	2045	2050	2055	2060	2065	2070	2075	2080
Austria	527.92	541.97	502.04	491.78	489.37	502.01	515.77	519.62	512.80	503.20	502.17	504.71	508.74	506.38	495
Belgium	806.00	842.68	851.38	897.38	936.37	948.23	968.57	986.31	1000.20	1015.26	1034.85	1045.49	1049.77	1050.78	1049
Croatia	389.82	344.26	322.95	299.97	303.19	295.85	289.61	279.63	269.14	260.91	255.92	251.37	245.50	238.02	230
Czech Republic	935.95	861.36	709.70	758.02	870.26	883.27	869.55	843.32	815.05	805.81	836.24	868.39	873.44	853.36	825
Denmark	260.77	301.91	305.62	302.77	308.88	290.44	316.91	349.65	375.24	383.78	380.98	377.85	385.30	402.50	417
Estonia	99.04	73.46	63.99	70.42	80.48	75.70	73.87	69.06	64.44	64.19	66.87	67.95	66.31	63.27	60
Finland	255.36	273.44	256.27	257.64	273.66	282.21	299.95	313.19	320.46	323.48	329.50	338.72	349.06	357.25	360
Germany	4041.73	3632.60	3623.01	3395.40	3387.79	3364.21	3491.47	3541.54	3524.74	3463.91	3409.25	3372.91	3370.17	3361.15	3313
Greece	773.64	764.27	779.54	833.22	880.56	830.91	804.50	790.04	797.88	813.71	819.21	800.08	769.44	745.93	734
Ireland	367.74	380.14	406.53	456.28	510.28	497.93	496.57	494.76	504.08	522.10	538.63	543.69	537.18	529.31	526
Italy	2905.84	3158.25	3165.75	3367.75	3527.67	3324.94	3316.63	3362.25	3444.03	3498.19	3498.91	3429.45	3341.56	3289.75	3272
Japan	7819.93	7408.43	7322.35	7124.27	6941.13	6576.83	6072.01	5703.07	5475.37	5301.31	5111.67	4865.42	4572.95	4302.74	4099
Latvia	163.05	118.16	94.10	103.40	113.43	106.80	103.74	94.93	85.05	81.42	83.24	83.94	81.34	76.31	70
Lithuania	237.32	200.74	157.70	141.75	159.85	178.34	173.65	160.37	141.70	128.16	128.40	134.63	136.05	129.49	119
Luxembourg	23.30	29.30	32.62	33.84	35.61	36.33	40.56	45.20	49.67	52.38	53.69	54.16	55.55	57.82	59
Netherlands	1015.76	1094.78	1128.87	1162.75	1115.36	1086.98	1120.68	1170.57	1211.47	1225.08	1211.55	1185.78	1174.70	1183.72	1196
Norway	252.26	302.03	303.55	302.49	321.99	328.53	356.62	385.39	405.35	413.81	419.87	426.98	439.79	454.69	464
Poland	3351.92	2839.29	2338.54	2191.79	2400.99	2418.84	2483.84	2429.65	2293.58	2141.99	2108.37	2157.07	2194.68	2175.43	2099
Portugal	400.03	432.12	468.06	506.26	507.32	488.54	509.35	535.91	570.08	599.46	616.45	619.30	625.11	634.12	642
Russian Federation	15395.93	11071.16	9919.47	11172.20	12673.45	12772.39	11475.21	10183.72	9626.55	10068.50	10793.88	10980.07	10556.14	9922.16	9588

Evolution

```
evolutionT = evolution.T
evolutionT.index = evolutionT.index.astype(int)
evolutionT = evolutionT[evolutionT.index>=2010]
```

```
evolutionT.plot(figsize=(20,12))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbb5b65b250>
```



Les Pays Prioritaires

Les Pays Prioritaires

```
conclusions_conclusions=country_by_region_and_income.copy()
conclusions_conclusions=conclusions_conclusions.rename(columns={"seuil": "seuil by region and income", "degree": "degre
conclusions_conclusions["degree global"]= selected_global_country["degree"]
conclusions_conclusions
```

			2015	seuil by region and income	degree by region and income	degree global
Region	Income Group	Country Name				
East Asia & Pacific	High income: OECD	Australia	4.935301e+05	1.088560e+05	6.0	NaN
		Japan	2.033034e+06	4.935301e+05	3.0	3.0
		Japan	2.033034e+06	1.081845e+06	3.0	3.0
		Japan	2.033034e+06	1.088560e+05	3.0	3.0
		Korea, Rep.	1.081845e+06	4.935301e+05	5.0	NaN
		Korea, Rep.	1.081845e+06	1.088560e+05	5.0	NaN
	High income: nonOECD	Hong Kong SAR, China	1.523163e+05	2.831354e+04	8.0	NaN
		Hong Kong SAR, China	1.523163e+05	1.275819e+05	8.0	NaN

Les Pays Prioritaires

```
conclusions_conclusions["degree by region and income"].median()
```

```
121.0
```

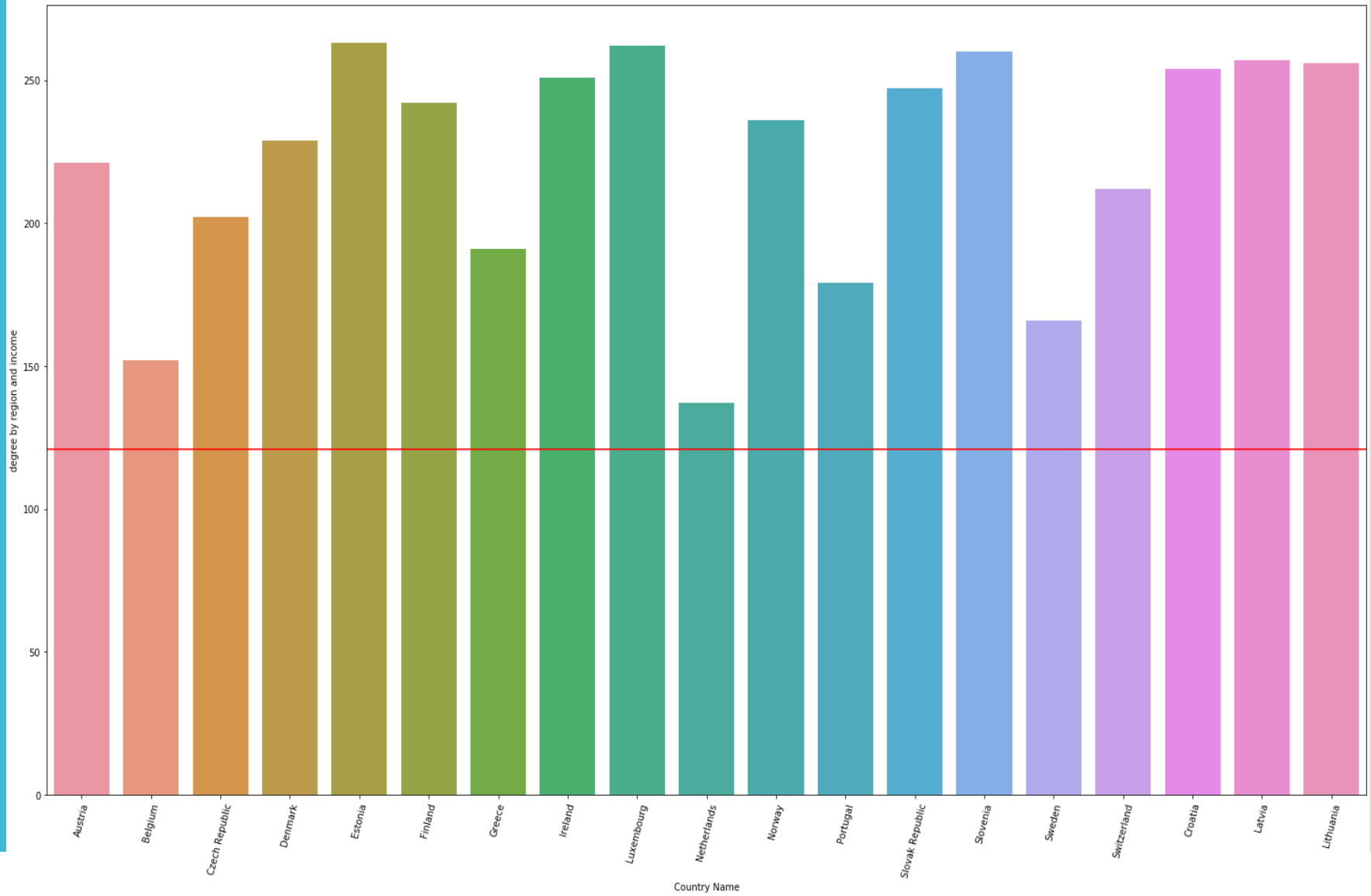
```
filter_paysprio1=conclusions_conclusions[conclusions_conclusions['degree by region and income']>121]
filter_paysprio1=filter_paysprio1[['degree by region and income']]

filter_paysprio1;
```

```
filter_paysprio1
```

	Region	Income Group	Country Name	degree by region and income
0	Europe & Central Asia	High income: OECD	Austria	221.0
1	Europe & Central Asia	High income: OECD	Belgium	152.0
2	Europe & Central Asia	High income: OECD	Czech Republic	202.0
3	Europe & Central Asia	High income: OECD	Denmark	229.0
4	Europe & Central Asia	High income: OECD	Estonia	263.0
5	Europe & Central Asia	High income: OECD	Finland	242.0
6	Europe & Central Asia	High income: OECD	Greece	191.0
7	Europe & Central Asia	High income: OECD	Ireland	251.0
8	Europe & Central Asia	High income: OECD	Luxembourg	262.0
9	Europe & Central Asia	High income: OECD	Netherlands	137.0
10	Europe & Central Asia	High income: OECD	Norway	236.0
11	Europe & Central Asia	High income: OECD	Portugal	179.0
12	Europe & Central Asia	High income: OECD	Slovak Republic	247.0
13	Europe & Central Asia	High income: OECD	Slovenia	260.0

Les Pays Prioritaires



Les Pays Prioritaires

```
prio_country1 = filter_paysprio1.index.tolist()

prio_country1=list(set(prio_country1))
prio_country1
```

```
[('Europe & Central Asia', 'High income: OECD', 'Czech Republic'),
 ('Europe & Central Asia', 'High income: OECD', 'Sweden'),
 ('Europe & Central Asia', 'High income: OECD', 'Portugal'),
 ('Europe & Central Asia', 'High income: OECD', 'Switzerland'),
 ('Europe & Central Asia', 'High income: nonOECD', 'Latvia'),
 ('Europe & Central Asia', 'High income: OECD', 'Slovenia'),
 ('Europe & Central Asia', 'High income: OECD', 'Belgium'),
 ('Europe & Central Asia', 'High income: OECD', 'Denmark'),
 ('Europe & Central Asia', 'High income: OECD', 'Ireland'),
 ('Europe & Central Asia', 'High income: OECD', 'Slovak Republic'),
 ('Europe & Central Asia', 'High income: OECD', 'Greece'),
 ('Europe & Central Asia', 'High income: OECD', 'Austria'),
 ('Europe & Central Asia', 'High income: OECD', 'Norway'),
 ('Europe & Central Asia', 'High income: nonOECD', 'Croatia'),
 ('Europe & Central Asia', 'High income: OECD', 'Luxembourg'),
 ('Europe & Central Asia', 'High income: OECD', 'Netherlands'),
 ('Europe & Central Asia', 'High income: OECD', 'Estonia'),
 ('Europe & Central Asia', 'High income: OECD', 'Finland'),
 ('Europe & Central Asia', 'High income: nonOECD', 'Lithuania')]
```

Conclusion

- **Quels sont les pays avec un fort potentiel de clients pour nos services ?**

Slovenia', 'Croatia', 'Netherlands', 'Norway', 'Spain', 'Japan', 'Italy', 'Luxembourg', 'Sweden', 'Switzerland', 'Poland', 'Belgium', 'Ireland', 'Slovak Republic', 'Denmark', 'Greece', 'Czech Republic', 'Russian Federation', 'Germany', 'Portugal', 'Finland', 'Lithuania', 'Estonia', 'United States', 'Austria', 'United Kingdom', 'Latvia'

- **Pour chacun de ces pays, quelle sera l'évolution de ce potentiel de clients ?**

United States, Russian Federation, Japan

- **Dans quels pays l'entreprise doit-elle opérer en priorité ?**

Czech Republic, Sweden, Portugal, Switzerland, Latvia, Slovenia, Belgium, Denmark, Ireland, Slovak Republic, Greece, Austria, Norway, Croatia, Luxemburg, Netherlands, Estonia, Finland, Lithuania