academy

Soutenance projet 02 –Data Science

Analyse des données de systèmes éducatifs

Etudiante : Bouchra MEKHALDI
Mentor : Souhail TOUMDI
Evaluateur :Ahmed BOULMANE
Date :31/01/22

Problématique

- ACADEMY est une start-up de la EdTech
- Il fournit du contenu de formation
- Niveau lycée et université
- Volonté d'expansion à l'international

- quels sont les **pays** avec un fort potentiel de clients pour les 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**?

Processus de l’étude

Présenter les données

- Quelles informations?

Identifier les indicateurs exploitables

- Quels indicateurs choisir?

Comparer les pays

- Analyse des résultats obtenus

- Quels sont les pays à cibler

- Recommandation des pays

l’évolution de potentiel de clients

- Comment identifier le potentiel des pays choisis?

Conclusion

Présenter les données

Description des jeux de données

Le portail “EdStatsAll IndicatorQuery” de la Banque mondiale répertorie 4000 indicateurs internationaux décrivant l’accès à l’éducation, l’obtention de diplômes et des informations relatives aux professeurs, aux dépenses liées à l’éducation.

Contenues dans 5 Datasets.

→Pour en savoir plus :

http://datatopics.worldbank.org/education/

→Site de la Banque Mondiale de données :

http://datatopics.worldbank.org/education/

EdStatsData

Des séries temporelles des indicateurs pour tous les pays de 1970 et 2100

Taille:886930 lignes et 70 colonnes

Variables qualitatives:4 variables

Variables quantitatives: 66 variables

Valeurs manquante: 86,1% (variables quantitatives)

Doublon :0

EdStatsCountry

Informations sur les pays : région, monnaie, Système de commerce..

Taille:241 lignes et, 32 colonnes

Variables qualitatives: 28

Variables quantitatives: 3

Valeurs manquante: 30,5%

Doublon :0

EdStatsSeries

Les informations sur les indicateurs: définitions, année d'apparition, méthode..

Taille:3665 lignes(codes) et 21 colonnes

Variables qualitatives:15

Variables quantitatives:6

Valeurs manquante: 71,7%

Doublon :0

EdStatsCountry-Series

Informations sur les source des données contenues dans EdStatsData pour les indicateur population

Taille:613 lignes et 4 colonnes

Variables qualitatives:3

Variables quantitatives:1

Valeurs manquante: 25%

Doublon :0

EdStatsFootNote

Des informations complémentaire sur les indicateurs de chaque pays : leurs source ,méthode de calcul (juste 1558 indicateurs)

Taille:643638 lignes et 4 colonnes

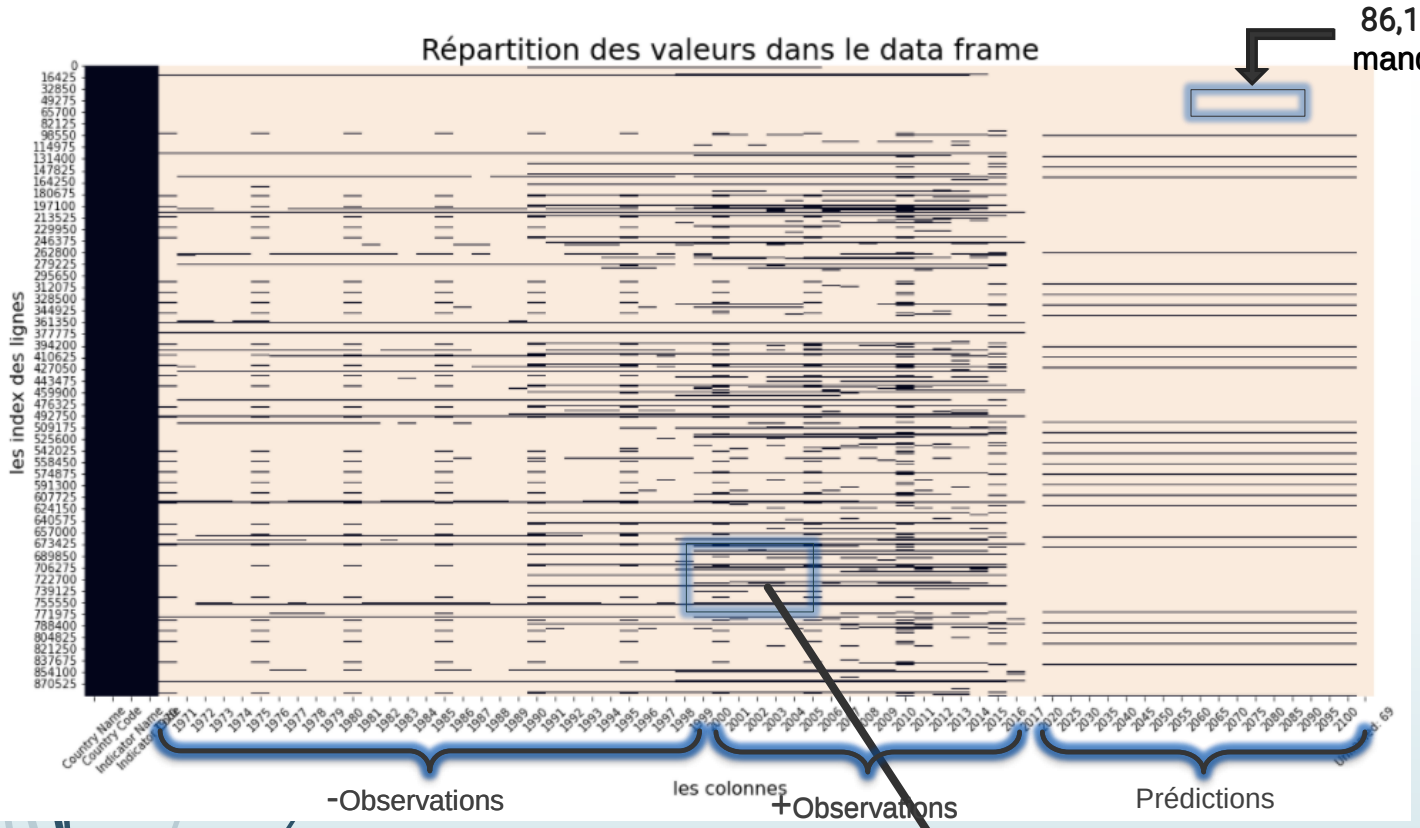
Variables qualitatives:5

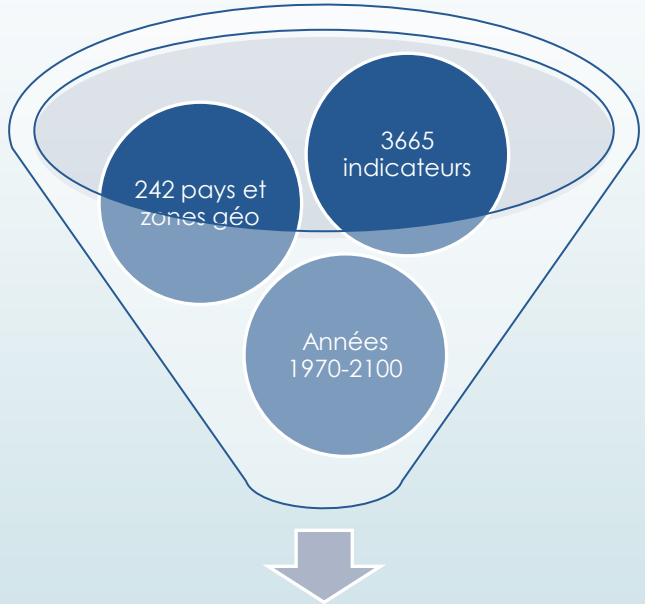
Variables quantitatives:1

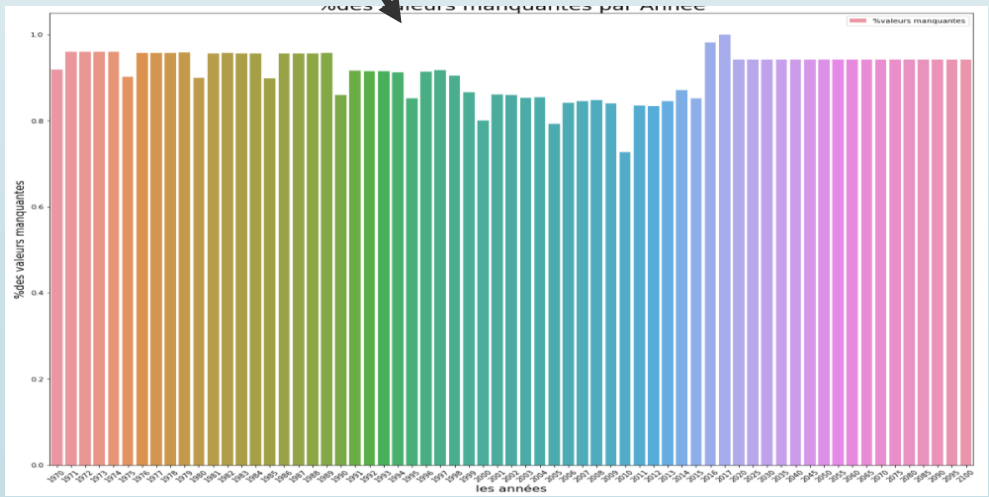
Valeurs manquante: 20%

Doublon :0

EdStats Data



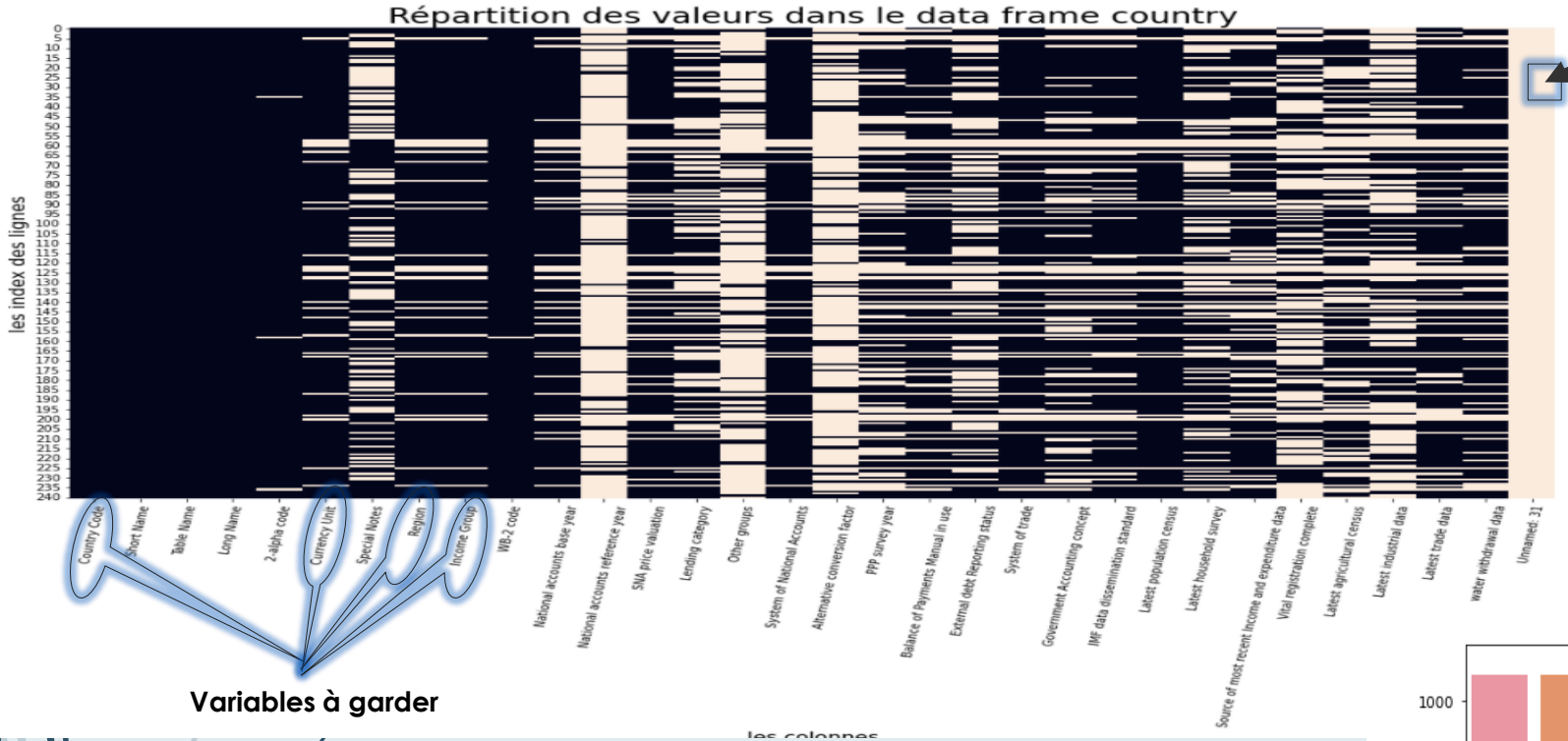




%des valeurs manquantes par Année

les données utiles

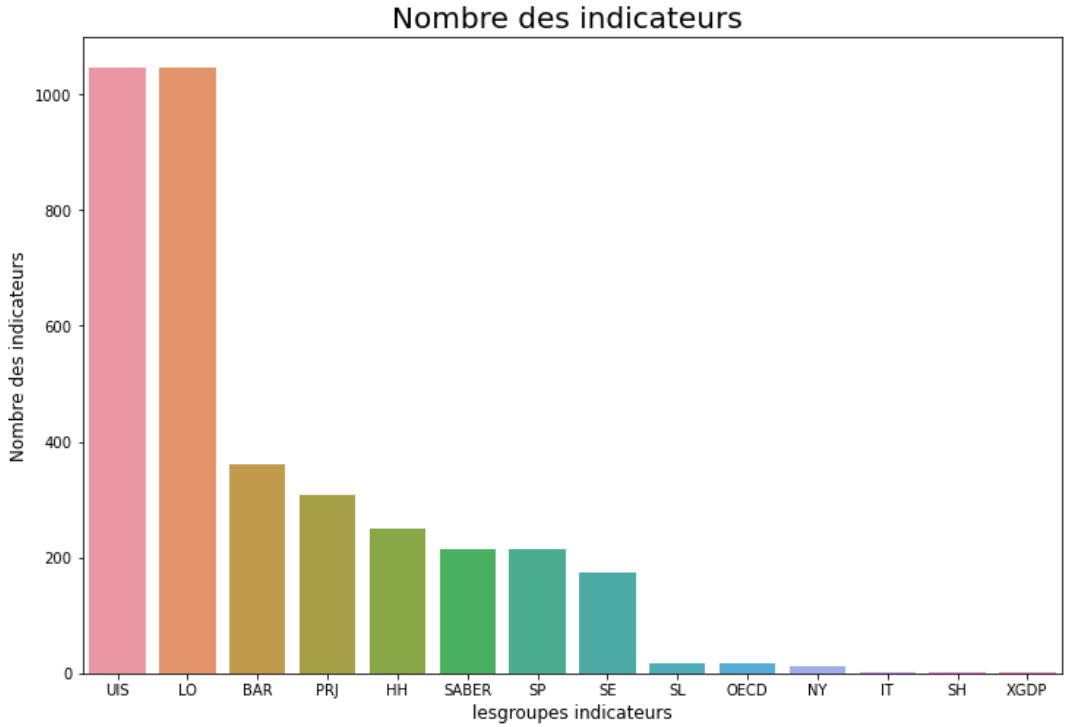
- Plage Temporelle?
- Indicateurs pertinents?
- Variables utiles?



30,5 % de valeurs manquantes

Variables à garder

EdStats Country



EdStats Series

Les groups d'indicateurs
SP : Social Population
SE : Social Education
NY : National Accounts, produits intérieurs et nationaux
IT : Infrastructure : utilisateurs internet et ordinateurs

nous intéressons aux groupes suivants
IT, NY, SE, SP

Approche méthodologique

1

Filtrer et nettoyer de la Variable qualitative **Country_Name** à partir de la Table EdStatsCountry.

2

Sélectionner la variable qualitative **Indicator_Code** à partir de la Table EdStatsSeries.

3

Nettoyage des variables quantitatives (**Années**)

4

Appliquer les **filtres** sur la DataEdStats puis récupérer et sauvegarder le jeu de donnée nettoyé et prêt pour l'analyse

Filtrage et nettoyage de la variable **Country_Name**

Nettoyage de DF

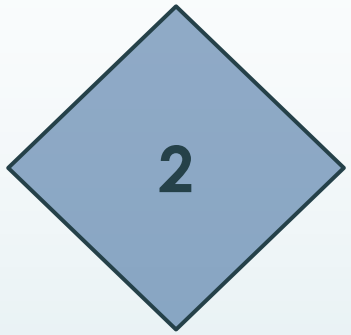
```
df_data=data.copy()
df_data =data.drop(columns=df_data.loc[:,'1970':'1999'])
df_data=df_data.drop(columns=df_data.loc[:,'2017':'Unnamed: 69'])
df_data.drop(['Indicator Name',], axis=1, inplace=True)
df_data.head()
```

Fusionner les DFs

```
df_data=df_data.merge(right = country[['Country Code', 'Region', 'Income Group','Currency Unit']],
                      on = 'Country Code')

df_data.head()
```

Suppression des régions

```
df_data.dropna(subset=['Currency Unit'],inplace=True)
df_data.head()
```



Identifier les indicateurs exploitables

Les groupes → Les cibles → Mots clés

```
# Cible : moyen de communication internet
liste_it=[row for row in df_indic['groupe_indic'] if ('IT') in row ]
df_indic[df_indic['groupe_indic'].isin(liste_it)][['Series Code','Indicator Name','Long definition']]
```

	Series Code	Indicator Name	Long definition
610	IT.CMP.PCMP.P2	Personal computers (per 100 people)	Personal computers are self-contained computer...
611	IT.NET.USER.P2	Internet users (per 100 people)	Internet users are individuals who have used t...

numérique

IT.NET.USER.P2
Accès à internet

Économique

NY.GNP.PCAP.PP.CD
Revenus par habitant

démographique

SP.POP.1524.TO.UN
Population des 15-24 ans

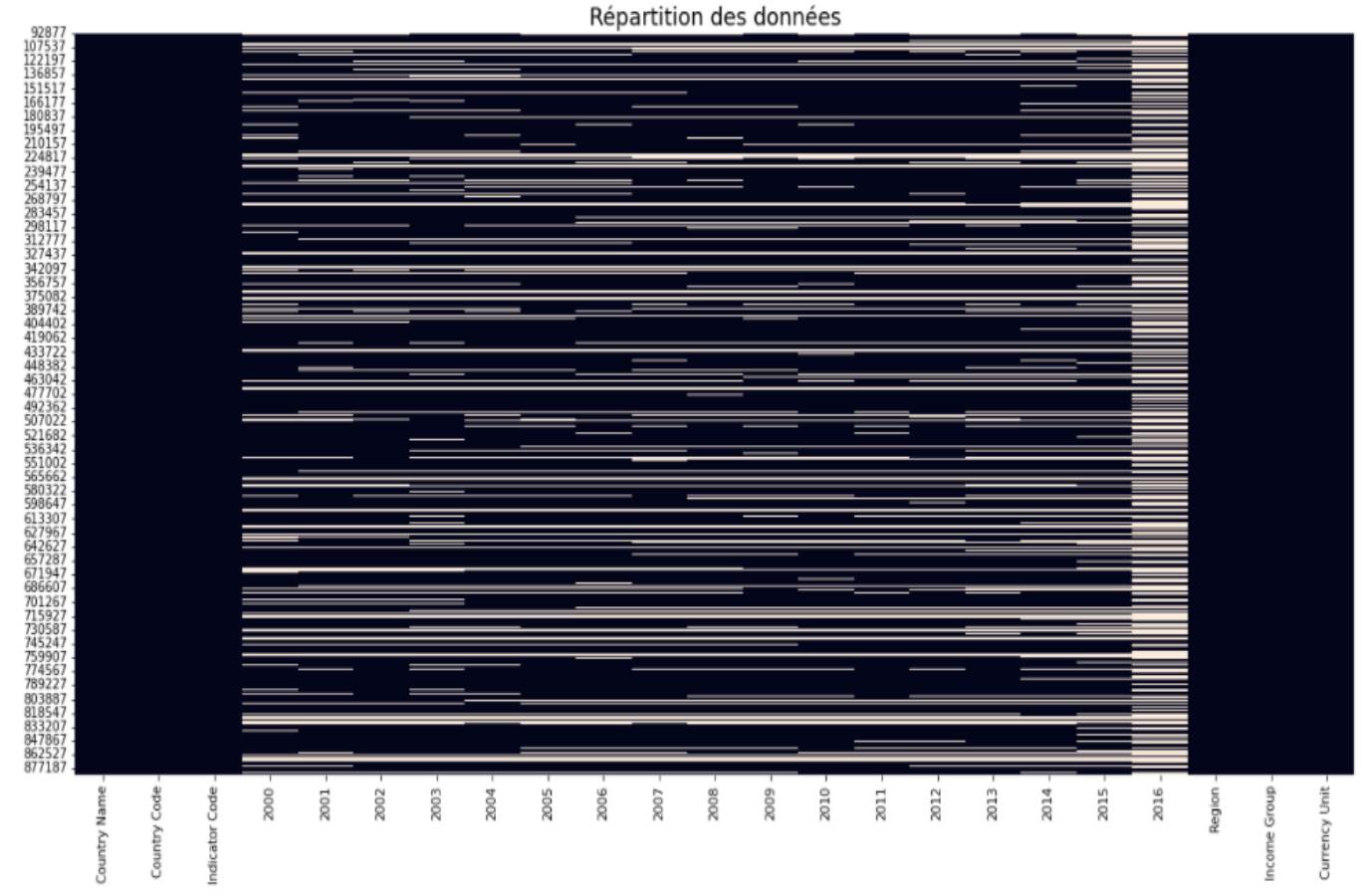
SP.POP.TOTL
Population total

Éducatif

SE.SEC.ENRR
Taux de scolarisation secondaire

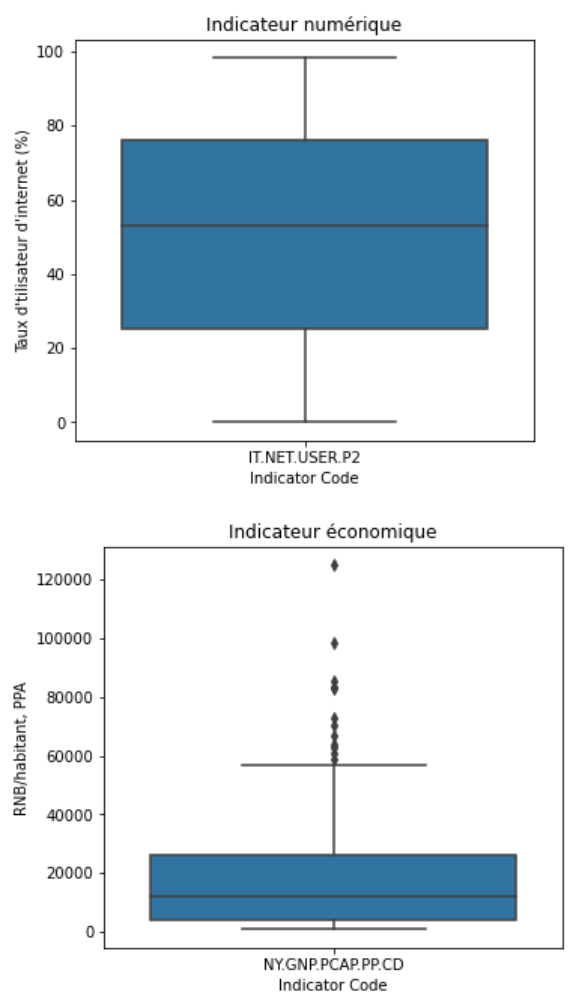
SE.TER.ENRRS
Taux de scolarisation tertiaire

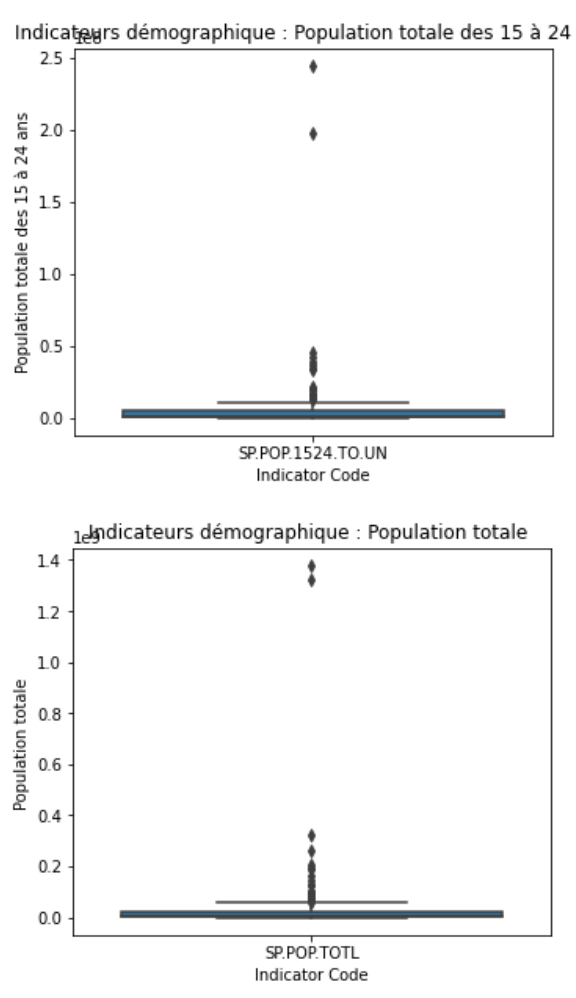
Répartition des données après filtrage des indicateurs

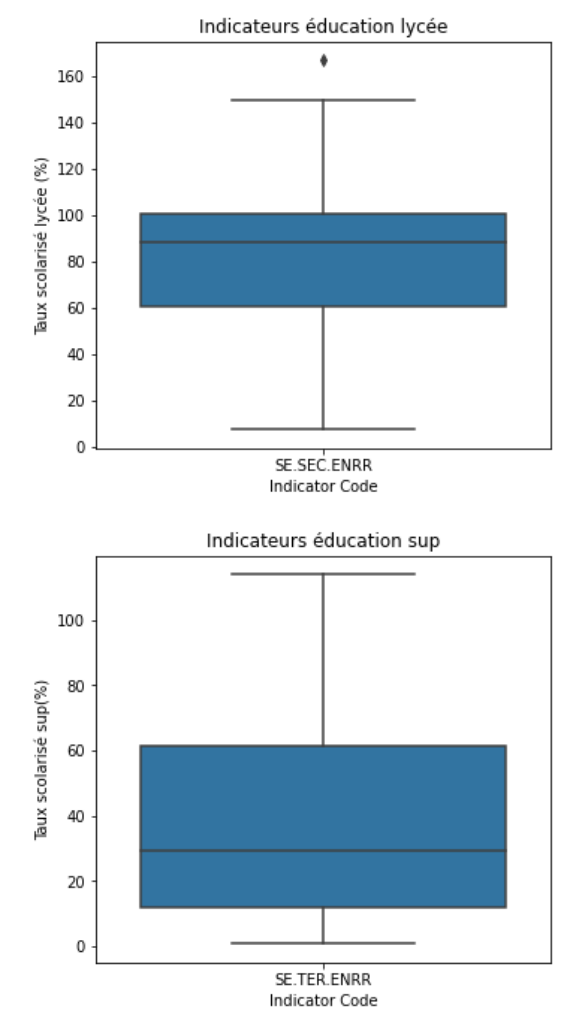


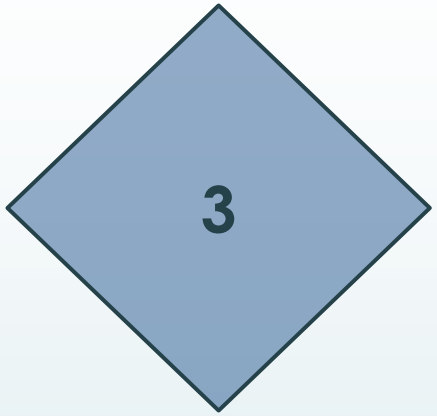

```
: # Fonction qui permet de renseigner la derniere années ou la valeurs est non null  :
def annee_valeur(row):
    if row.first_valid_index() is None:
        return None
    else:
        return (row.first_valid_index(),row[row.first_valid_index()])
```

Data Visualisation pour la DF pour chaque indicateur









Comparer les pays

Calcul du score

Création de DF

```
# création d'un tableau pivo pour les pays

pays_final = df_final.pivot_table(index= 'Country Name', columns="Indicator Code")['Dernière Valeur Non Null']
pays_final=pays_final.rename_axis('Country Name').reset_index()#la colonne country_name devient colonne et non index
pays_final.head()
```

Indicator Code	Country Name	IT.NET.USER.P2	NY.GNP.PCAP.PP.CD	SE.SEC.ENRR	SE.TER.ENRR	SP.POP.1524.TO.UN	SP.POP.TOTL
0	Afghanistan	10.595726	1900.0	55.644409	8.662800	7252785.0	34656032.0
1	Albania	66.363445	11670.0	95.765488	58.109951	556269.0	2876101.0

```
def Score(x,p,pas): # fonction pour le calcule d percentile
    q=[]
    for i in range (1,pas+1):
        q += [i/pas] # choix de pourcentage
    quantiles = df_score.quantile(q)
    for j in range (1,pas+1):
        if x<=quantiles[p][j/pas]:
            return j
```

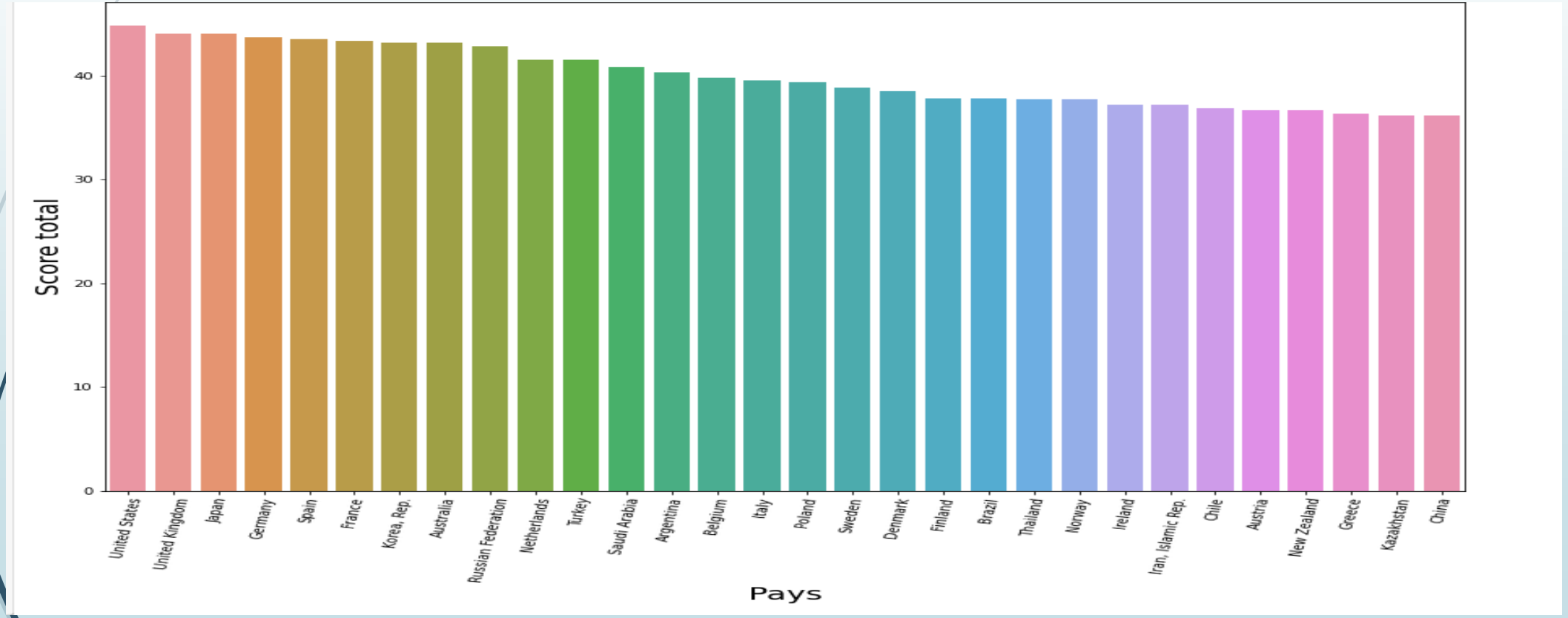
Fonction pour calculer le score

Fonction pour remplissage du table

```
# Remplissage du table
df_score=pays_final.copy()
for j in range (1,7): # choix des colonnes
    nom =df_score.columns[j]
    df_score['scor',df_score.columns[j]]=df_score[nom].apply(Score, args=(nom,50),) # le score associé pour chaque indicateur
df_score.head()
```

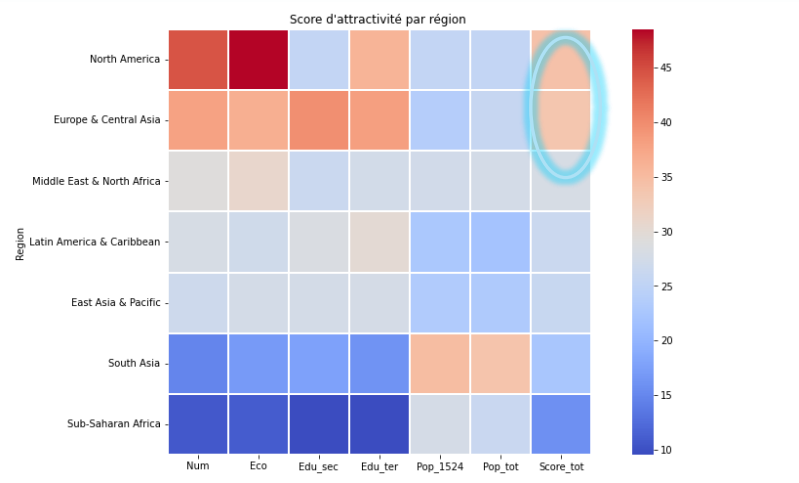
```python
# fonction pour calculr le score final
coef=[1,1,1,1,1,1] # list pour des coefficients aattribué pour chaque indinateur: coef[0]est le poids attribué pour le 1er indi

def calcul_score (df,coef):
    n=len(coef)
    s=0 # pour calculer la somme
    for i in range (1,n+1):
        s += df.iloc[:,i]*coef[i-1] # la somme des produits
    score= s/sum(coef) # calcul des moyens
    return round(score,2)
```

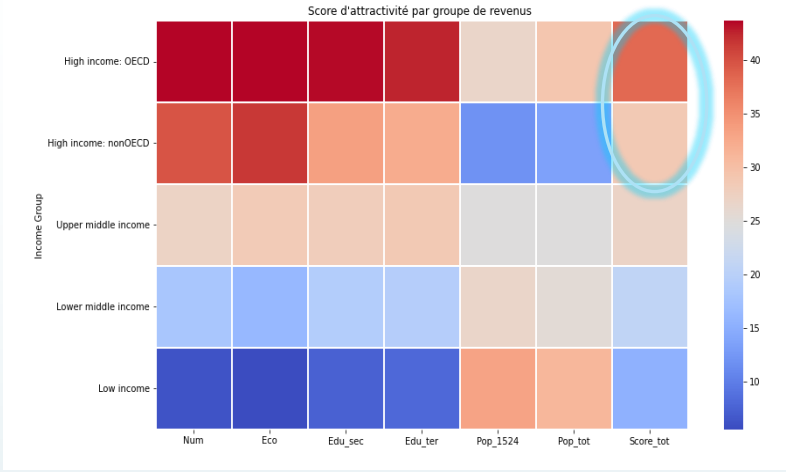
Calcul de score finale

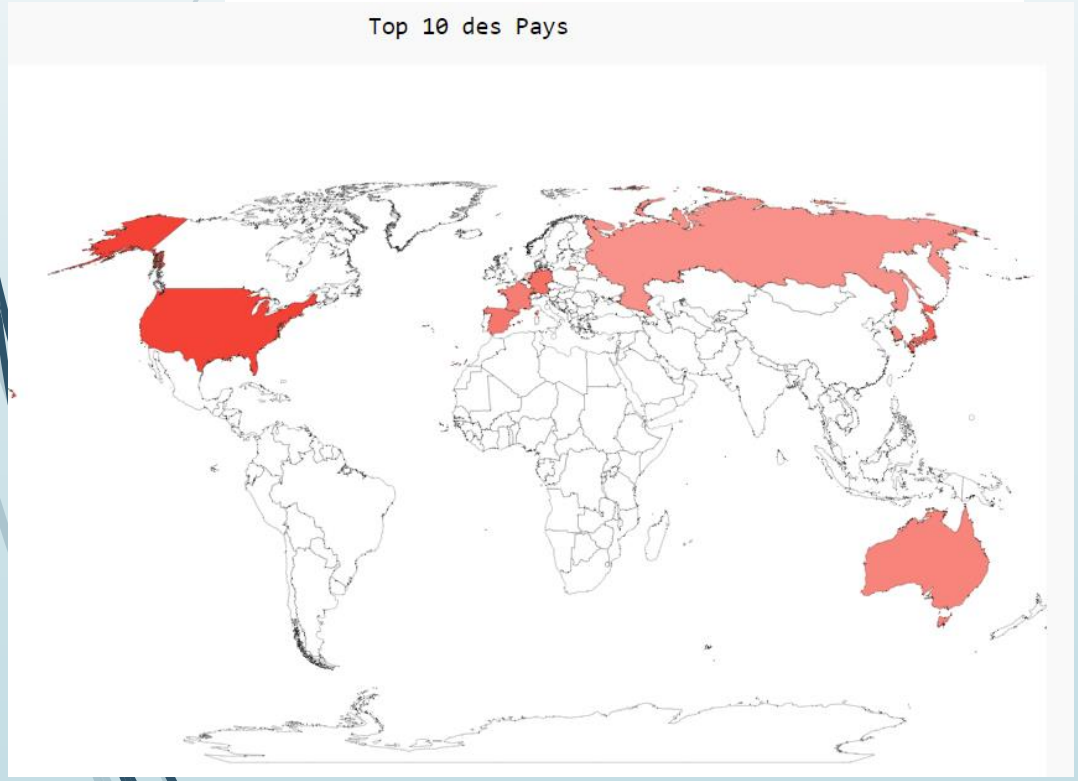


Classement des 30 premiers pays

Score d'attractivité par région et par groupe de revenus







Les 10 pays les plus attractifs

Place	Pays	Region	Income Group	Score_tot
1	United States	North America	High income: OECD	44.83
2	United Kingdom	Europe & Central Asia	High income: OECD	44.00
3	Japan	East Asia & Pacific	High income: OECD	44.00
4	Germany	Europe & Central Asia	High income: OECD	43.67
5	Spain	Europe & Central Asia	High income: OECD	43.50
6	France	Europe & Central Asia	High income: OECD	43.33
7	Korea, Rep.	East Asia & Pacific	High income: OECD	43.17
8	Australia	East Asia & Pacific	High income: OECD	43.17
9	Russian Federation	Europe & Central Asia	High income: nonOECD	42.83
10	Netherlands	Europe & Central Asia	High income: OECD	41.50

recommandation des pays en fonction des indicateurs

Indicateur économique

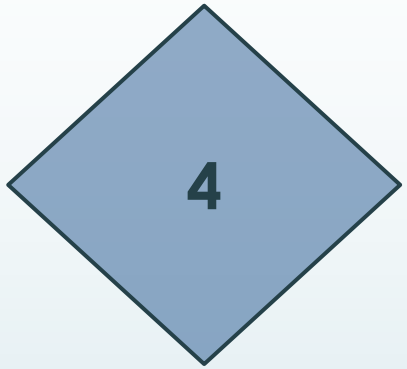
Pays	Score_tot	Place
United States	45.29	1
Japan	44.00	2
United Kingdom	43.86	3
Germany	43.86	4
France	43.43	5
Australia	43.43	6
Spain	43.29	7
Korea, Rep.	43.00	8
Netherlands	42.14	9
Saudi Arabia	41.71	10

Indicateur population entre 15 et 24 ans

Pays	Score_tot	Place
United States	45.57	1
Japan	44.29	2
United Kingdom	43.86	3
Germany	43.57	4
Russian Federation ↑4	43.43	5
France	43.14	6
Korea, Rep.	42.86	7
Spain	42.43	8
Turkey	42.14	9
Australia	41.71	10

Indicateur de taux de scolarisation

Pays	Score_tot	Place
Spain	44.88 ↑4	1
Australia	44.75 ↑6	2
United States	43.88	3
United Kingdom	43.50	4
France	43.00	5
Korea, Rep.	43.00	6
Russian Federation	43.00	7
Germany	42.88	8
Netherlands	42.88	9
Japan	42.75 ↓7	10



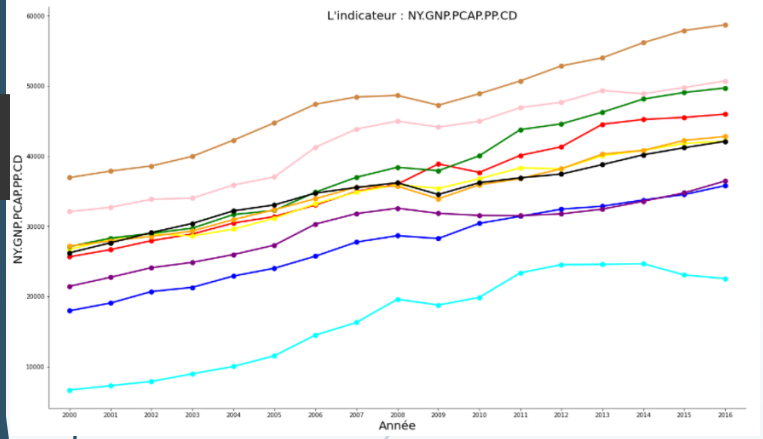
l’évolution de potentiel de clients

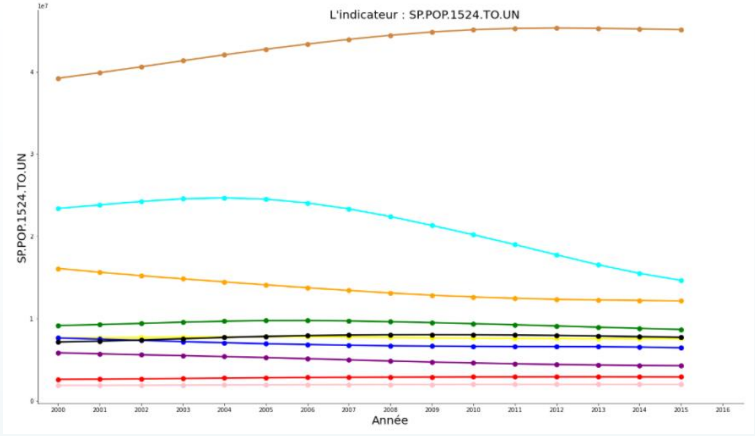
```python
# Fonction pour filtrer et melt le DF
def df_melt (df,ind):
    df=df[df['Indicator Name'].isin(ind)]
    df_evol=df.drop(columns=['Country Code','Indicator Name', 'Indicator Code'])
    df_evol=df_evol.set_index('Country Name').T #Transposer le df
    df_evol= df_evol.rename_axis('Year').reset_index()# renomer la premiere colonne
    df_evol= df_evol.melt('Year', var_name='Country',  value_name='vals')
    return df_evol
```

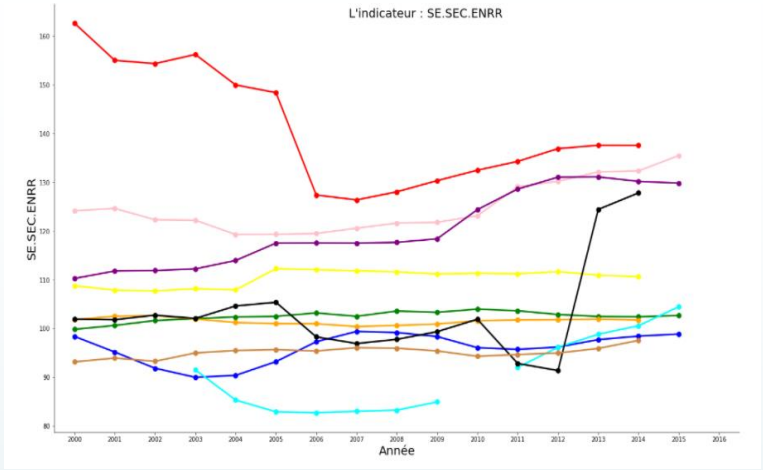
	Year	Country	vals
0	2000	Australia	25640.0
1	2001	Australia	26660.0
2	2002	Australia	27940.0
3	2003	Australia	28890.0
4	2004	Australia	30450.0
...	...	...	...
165	2012	United States	52850.0
166	2013	United States	54000.0
167	2014	United States	56160.0
168	2015	United States	57900.0
169	2016	United States	58700.0

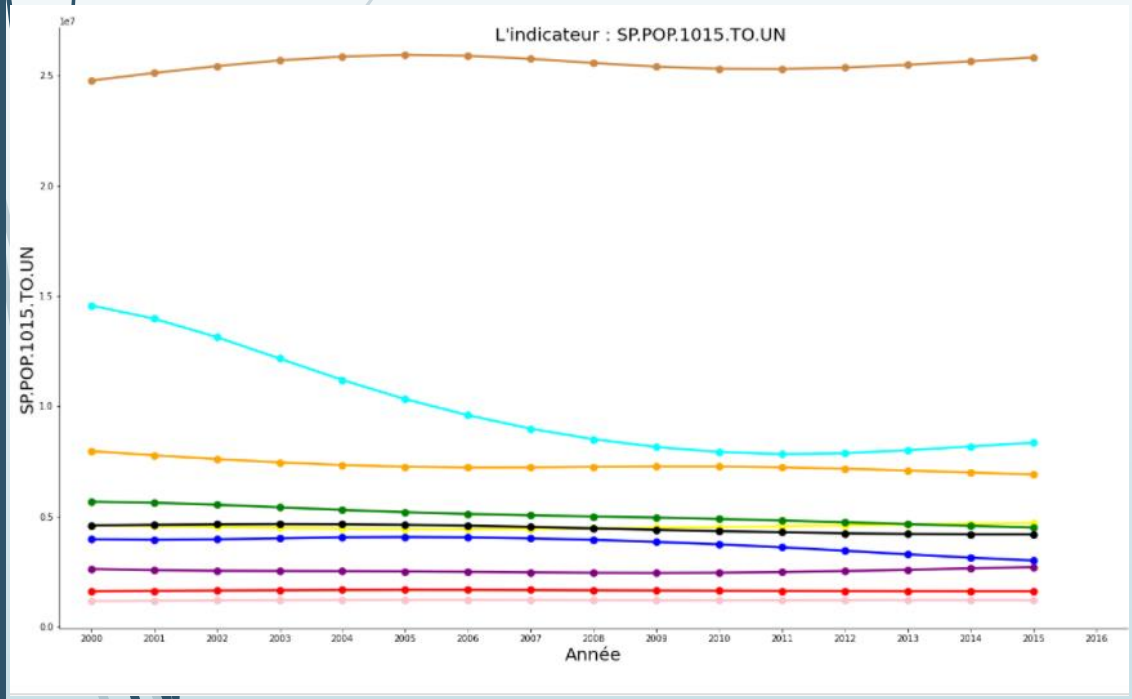
170 rows × 3 columns

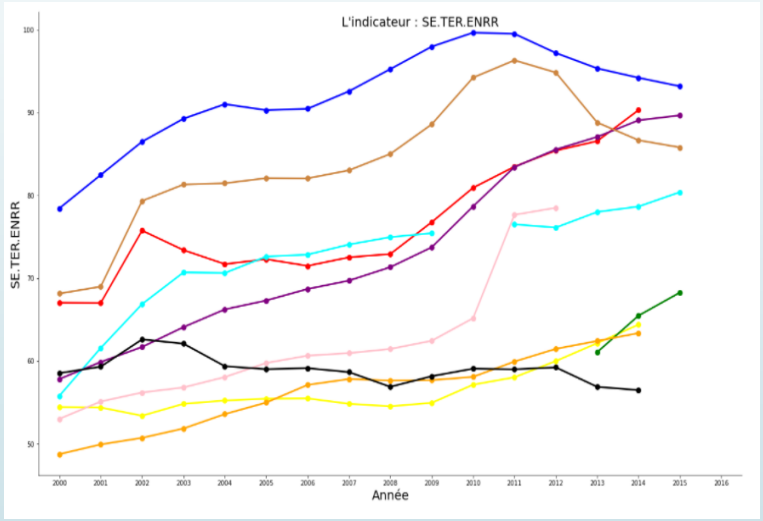
```python
# fonction pour tracer les graphes d'evolution
def evolution_indic(df,ind_name):
    #Year= ['2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '201
    year_palette = ["red","yellow","green","orange","blue","pink","cyan","purple","black","Peru"]
    figsize=(30,15)
    g=sns.factorplot(x="Year", y="vals", hue='Country', data=df, height=15,ci=None, palette= year_palette)
    g.fig.suptitle("L'indicateur : "+ ind_name , fontsize= 20)
    g.fig.set_size_inches(20, 10)
    g.set_xlabels('Année', fontsize= 20)
    g.set_ylabels(ind_name, fontsize= 20)
```











Pays à garder :
- Les états Unis
- Allemagne
- Japon

Conclusion

- Le pays dont l'entreprise doit-elle opérer en priorité c’est **l’USA**
- Le jeu des données contient tous les pays du monde mais beaucoup de données manquantes pour comparer
- Jeu de données plus récent, plus de données éducatives
- Manque d’informations sur la société Academy Stratégie d’entreprise : langue, marché cible, proximité géographique d’implantation

Merci !