Présentation du Projet 2

Analyse des données des systèmes éducatifs.

Parcours Data Scientist.

Mentor : Mr Hassan Amoud.

Présenté par Mr Dai TENSAOUT

Plan

Contexte

Présentation des Data Sets

Présentation des Indicateurs

Opérations réalisées

Scoring

Conclusion

Contexte

academy

Analyse exploratoire des données de la banque mondiale.

Analyse pré-exploratoire des données

Les données permettent-elles d'informer le projet d'expansion ?

LA BANQUE MONDIALE
BIRD • IDA

Les pays : fort potentiel client ?
L'évolution de ce potentiel ?

Priorité : Pays ?

Présentation des data sets

LA BANQUE MONDIALE
BIRD • IDA

EdStatsSeries :	3665 lignes et 70 colonnes
EdStatsCountry :	613 lignes et 32 colonnes
EdStatsFootNote :	239 lignes et 4 colonnes
EdStatsCountry-Series :	613 lignes et 3 colonnes
EdStatsData :	886 930 lignes et 70 colonnes

Présentation des data sets

Attainment

EMIS

Policy

1 – Cinq fichiers de données.

2 – 3665 Indicateurs répartis en différentes thématiques.

3 – 241 Pays/Régions.

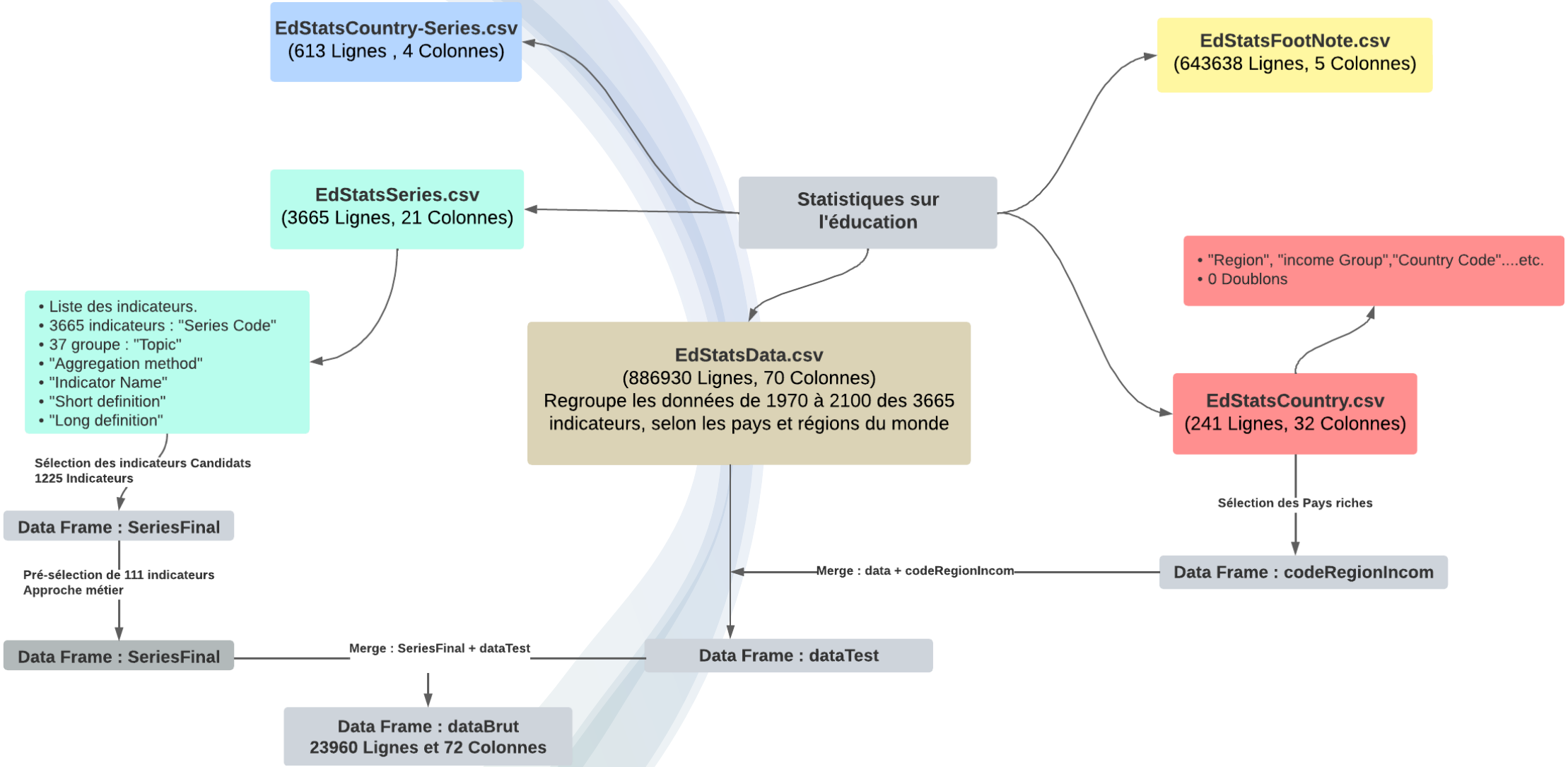
4 - Une période de 1970 à 2100.

Expenditures

Learning Outcomes

Equity

Présentation des data sets



Présentation des data sets

Data Frame : dataBrut
23960 Lignes et 72 Colonnes

```
# Regardons le taux de remplissage de notre data frame

df_nan_annee = (dataBrut.loc[:,"1970":"2100"].isna().sum()*100/dataBrut.shape[0]).to_frame(name='% NaN')
```

```
df_nan_annee
```

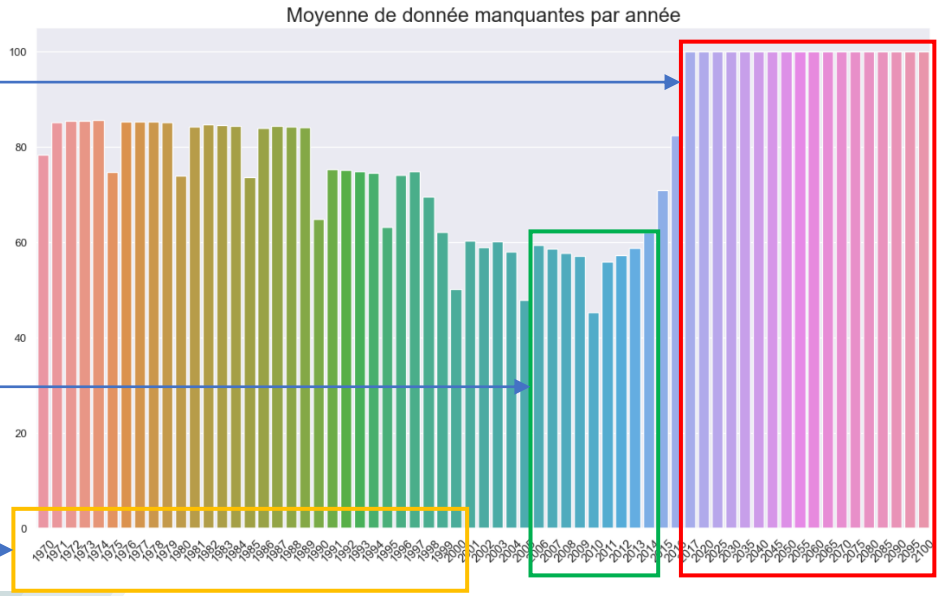
	% NaN
1970	78.351419
1971	85.175292
1972	85.471619
1973	85.479967
1974	85.638564
...	...
2080	100.000000
2085	100.000000
2090	100.000000
2095	100.000000
2100	100.000000

65 rows × 1 columns

Taux de NaN : 100%

Taux de NaN « 2010 » : 45,31%

Taux de NaN 1970-1990 : 82,02%



Présentation des data sets

Data Frame : dataBrut
23960 Lignes et 72 Colonnes

```
1  # Taux de valeurs manquantes  de notre dataBrut
2  tauxNan = round(dataBrut.loc[:,'1970':'2100'].isna().mean().mean()*100, 2)
3  print('Le Taux de valeurs manquantes de l\'ensemble des données est de : ',tauxNan, '%')
```

```
Le Taux de valeurs manquantes de l'ensemble des données est de :  79.48 %
```

- Le Taux de valeurs manquantes de l'ensemble des données est de : 79.48 %
- Le Taux de valeurs manquantes entre 2000 et 2010 : 55.73 %
- Le Taux de valeurs manquantes entre 1970 à 1990 : 82.02 %
- Le Taux de valeurs manquantes de l'année 2010 : 45.31 %
- Le Nombre de Valeurs en double du "dataBrut" est égal à : 0

Présentation des Indicateurs

```
#Nombre d'indicateurs total :
series['Indicator Name'].unique().shape
```

```
(3665,)
```

```
# Nombre de total des thématiques.
series['Topic'].unique().shape
```

```
(37,)
```

```
#Indicateurs selon : Gap-filled total
series.loc[series['Aggregation method'] == 'Gap-filled total', 'Indicator Name']
```

```
1658                              GDP (current US$)
1659                        GDP (constant 2010 US$)
1660              GDP, PPP (current international $)
1661        GDP, PPP (constant 2011 international $)
1666                              GNI (current US$)
1667              GNI, PPP (current international $)
Name: Indicator Name, dtype: object
```

Attainment

Learning Outcomes

Expenditures

Policy

	Series Code	Topic	Indicator Name
30	BAR.POP.1519	Attainment	Barro-Lee: Population in thousands, age 15-19,...
32	BAR.POP.15UP	Attainment	Barro-Lee: Population in thousands, age 15+, t...
34	BAR.POP.2024	Attainment	Barro-Lee: Population in thousands, age 20-24,...
36	BAR.POP.2529	Attainment	Barro-Lee: Population in thousands, age 25-29,...
38	BAR.POP.25UP	Attainment	Barro-Lee: Population in thousands, age 25+, t...

Opérations réalisées

Les Indicateurs

1. Choix des Indicateurs candidats

```
1  # 1225 indicateurs candidats :
2  seriesFinal = seriesFinal.reset_index()
3  seriesFinal.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1225 entries, 0 to 1224
```

2. Pré-sélection

```
1  seriesFinal.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 101 entries, 181 to 3664
Data columns (total 3 columns):
```

3. Choix des Indicateur finaux

```
1  #Liste d'indicateurs intéressants pour notre étude : Approche métier
2  indicateurs_finaux = ['IT.NET.USER.P2','IT.CMP.PCMP.P2','NY.GDP.MKTP.CD',
3                        'NY.GDP.PCAP.CD','NY.GNP.MKTP.CD','SP.SEC.TOTL.IN',
4                        'SP.TER.TOTL.IN','SP.POP.GROW','SP.POP.TOTL']
```

Opérations réalisées

Les Indicateurs finaux

IT.NET.USER.P2 : Personal computers (per 100 people)

NY.GDP.MKTP.CD : GDP (current US$)
NY.GDP.PCAP.CD : GDP per capita (current US$)
NY.GNP.MKTP.CD : GNI (current US$)

SP.SEC.TOTL.IN : Population of the official age for secondary education
SP.TER.TOTL.IN : Population of the official age for tertiary education

SP.POP.GROW : Population growth (annual %)
SP.POP.TOTL : Population, total

Opérations réalisées

Sélection des Pays : approche métier

```
# Les différentes catégories des groupes de revenus :

dataBrut["Income Group"].unique()
```

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

```
# Choix des Pays Hight Income :
high_OCDE = dataBrut["Income Group"] == 'High income: OECD'
high_nonOCDE = dataBrut["Income Group"] == 'High income: nonOECD'
```

```
# data frame pays High Income : OCDE et Non OCDE
paysRiches = dataBrut[high_OCDE|high_nonOCDE]
```

```
#Sélection des Pays :
dataBrut = dataBrut[dataBrut['Country Code'].isin(paysRiches['Country Code'])]
```

```
# Nous garderons 75 Pays pour l'analyse.
len(dataBrut['Country Code'].unique())
```

```
75
```

Opérations réalisées

Sélection de la période d'analyse

```
# Taux de remplissage des années :
Remplissage_Annee = pd.DataFrame({'Année':df_annee.mean(axis=0).index, 'Mean_NotNull':df_annee.mean(axis=0).values})
Remplissage_Annee = Remplissage_Annee.set_index('Année')
Remplissage_Annee.head()
```

Année	Mean_NotNull
1970	20.606061
1971	13.979798
1972	13.468013
1973	13.791246
1974	13.750842

```
# Suppression des années dont Mean_NotNull = 0
annee_supp = Remplissage_Annee.loc[Remplissage_Annee['Mean_NotNull'] ==  0].index
```

```
# Les années à supprimer :
annee_supp
```

```
Index(['2017', '2020', '2025', '2030', '2035', '2040', '2045', '2050', '2055',
       '2060', '2065', '2070', '2075', '2080', '2085', '2090', '2095', '2100'],
      dtype='object', name='Année')
```

```
# Création de 6 période d'analyse :
```

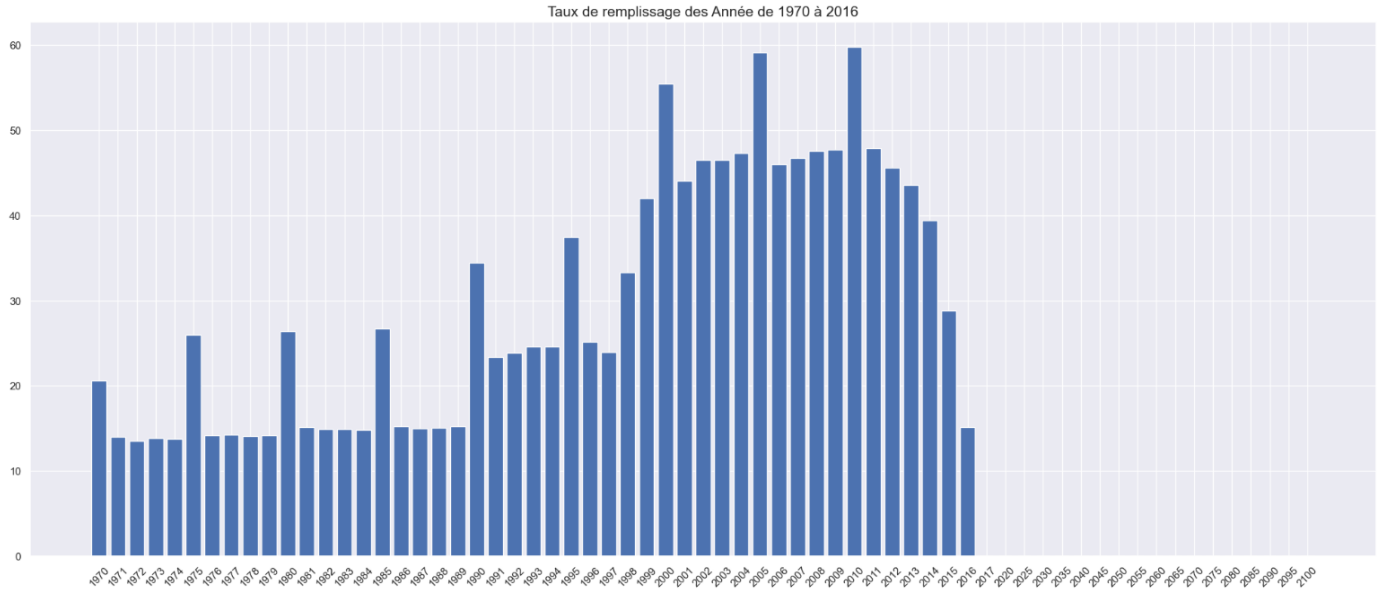
```
# 1970 - 1989
dataBrut['1970-1989'] = dataBrut.iloc[:,2:22].mean(axis = 1)
# 1990-1995
dataBrut['1990-1995'] = dataBrut.iloc[:,22:28].mean(axis = 1)
# 1996-2000
dataBrut['1996-2000'] = dataBrut.iloc[:,28:33].mean(axis = 1)
# 2001-2005
dataBrut['2001-2005'] = dataBrut.iloc[:,33:38].mean(axis = 1)
#2006-2010
dataBrut['2006-2010'] = dataBrut.iloc[:,38:43].mean(axis = 1)
#2011-2016
dataBrut['2011-2016'] = dataBrut.iloc[:,43:49].mean(axis = 1)
```

Opérations réalisées

Sélection de la période d'analyse

```
1  # Taux de remplissage des années de 1970 à 2016 :
2  Remplissage_Annee = pd.DataFrame({'Année':df_annee.mean(axis=0).index, 'Mean_NotNull':df_annee.mean(axis=0).values})
3  Remplissage_Annee = Remplissage_Annee.set_index('Année')
4  Remplissage_Annee.head()
```

Année	Mean_NotNull
1970	20.606061
1971	13.979798
1972	13.468013
1973	13.791246
1974	13.750842



Opérations réalisées

Choix des pays attractifs

```python
# # Taux de remplissage par Pays :
pays = list(df_pays.index)
mean_not_null_pays = []
for c in pays:
    mean_not_null_pays.append(df_pays.loc[c].mean())
```

```python
#Taux de remplissage par Pays :
Remplissage_Pays = pd.DataFrame({'Pays':df_pays.index, 'Mean_NotNull':mean_not_null_pays})
Remplissage_Pays = Remplissage_Pays.set_index('Pays')
Remplissage_Pays.head()
```

Pays	Mean_NotNull
ABW	12.416472
AND	11.857032
ARE	18.601399
ATG	14.560995
AUS	29.883450
....	

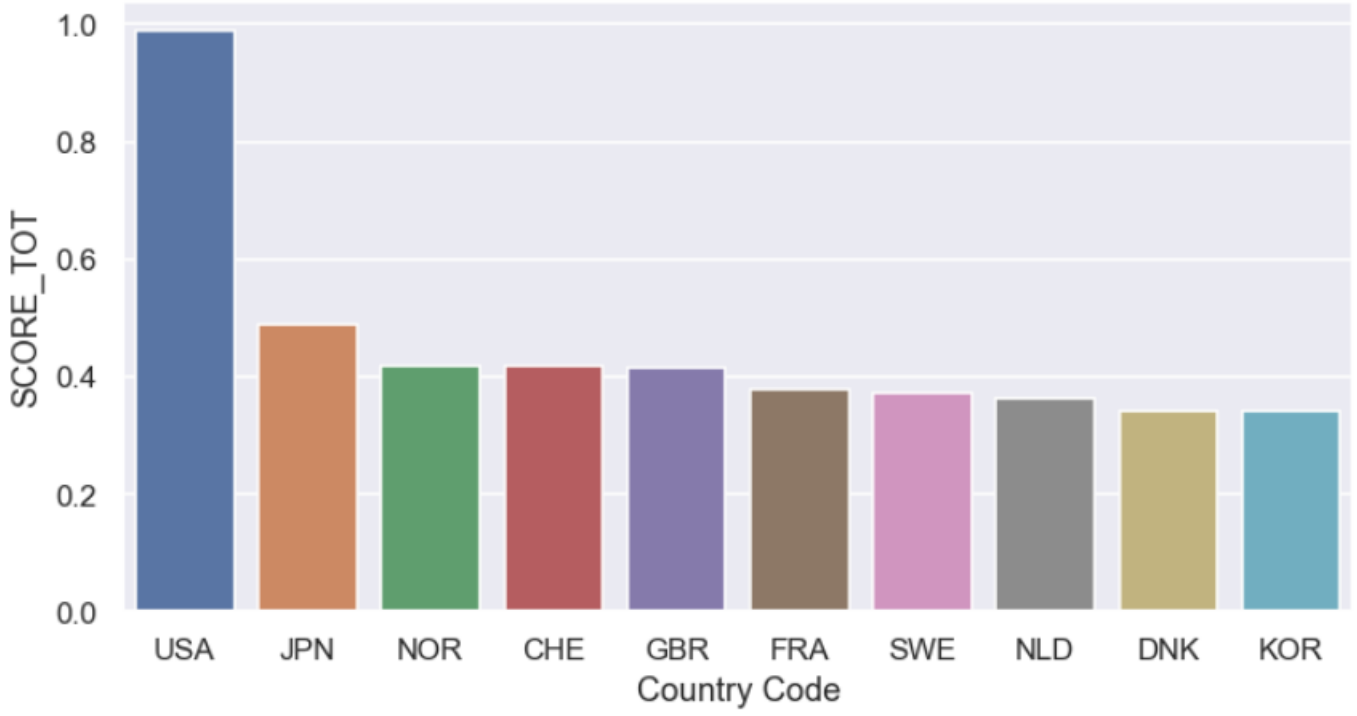
```python
# Suppression des pays dont Mean_NotNull < 30%
pays_supp = Remplissage_Pays.loc[Remplissage_Pays['Mean_NotNull'] < 30].index
```

```python
# Les Pays à supprimer :
pays_supp
```

```
Index(['ABW', 'AND', 'ARE', 'ATG', 'AUS', 'BHR', 'BHS', 'BMU', 'BRB', 'BRN',
       'CAN', 'CHI', 'CHL', 'CUW', 'CYM', 'DEU', 'EST', 'FRO', 'GNQ', 'GRC',
       'GRL', 'GUM', 'HKG', 'HRV', 'IMN', 'ISL', 'KNA', 'KWT', 'LIE', 'LTU',
       'LUX', 'LVA', 'MAC', 'MAF', 'MCO', 'MNP', 'NCL', 'OMN', 'POL', 'PRI',
       'PYF', 'QAT', 'RUS', 'SAU', 'SGP', 'SMR', 'SVK', 'SVN', 'SXM', 'TCA',
       'TTO', 'URY', 'VIR'],
      dtype='object', name='Pays')
```

Scoring

```
# Présentation des pays selon le score total
plt.figure(figsize=(8,4),dpi=110)
sns.barplot(data=top_ten, x='Country Code', y='SCORE_TOT')
sns.set_palette('dark')
sns.set_theme()
```

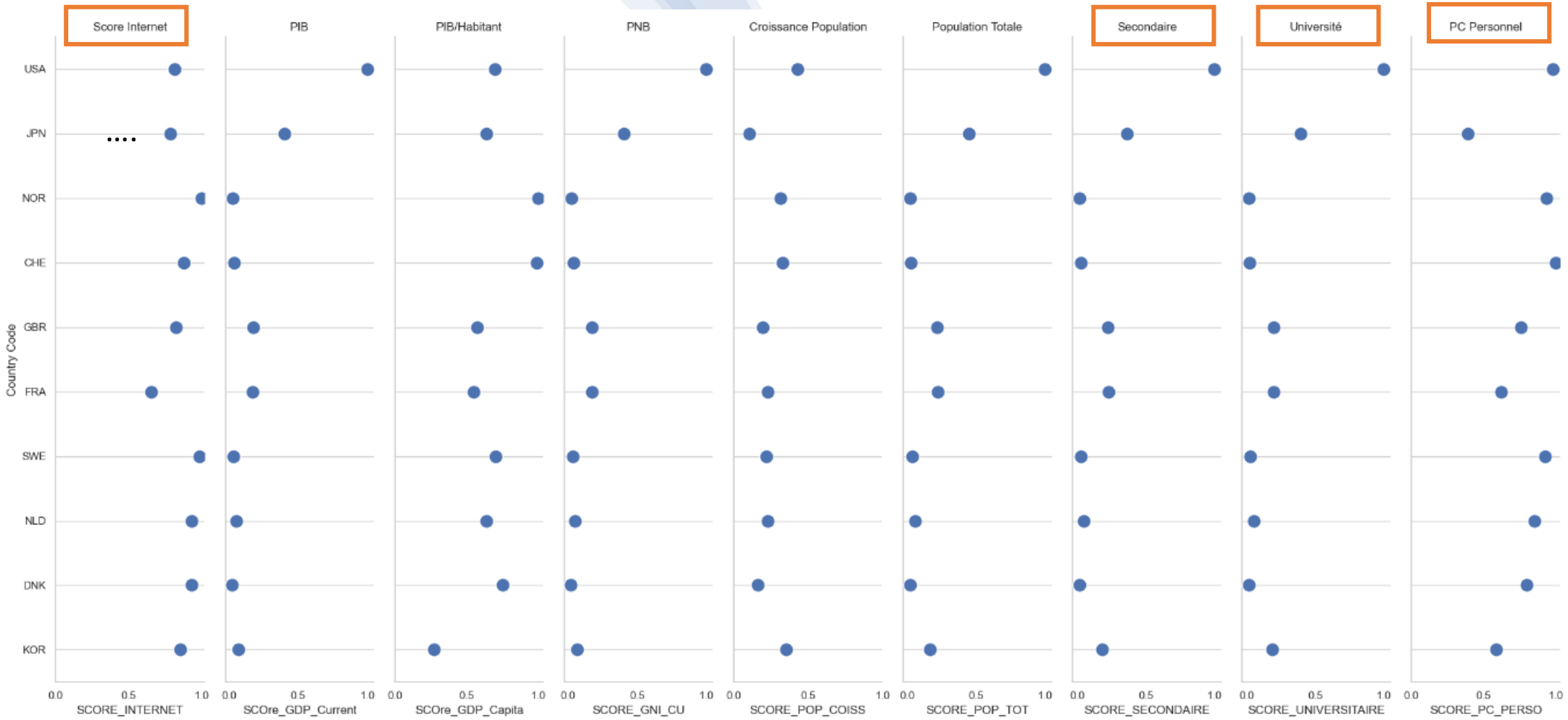


Les États-Unis, enregistre les meilleurs scores sur la majorité des indicateurs, hormis la croissance de la population et le produit intérieur brut par habitant.

En matière de population étudiante en âge d'entrée dans le secondaire, nous remarquons que la plupart des pays n'enregistre pas un bon score, c'est le même constat pour la population en âge d'entrée au Lycée, hormis les deux pays États-Unis et Japon.

Scoring

Scores des pays TOP10



Conclusion

- Les data frames utilisés sont : **EdStatsSeries** (pour la sélection des indicateurs pertinents), **EdStatsCountry** (pour la sélection des pays à haut revenu) et le **EdStatsData** (pour finaliser les analyses).

- Le **taux de remplissage** du data frame **EdStatsData** au départ est de **20,52 %.**

- **L'année** la plus fournie en matière de données est l'an **2010**, on retrouve le moins de données manquantes.

- Après avoir filtrer le data frame **EdStatsData** plusieurs fois, nous sommes arrivés à un taux de remplissage de **87,11**%.

- Les **10 Pays** les plus attractifs sont donc : les **États-Unis**, le **Japon**, la **Norvège** , la **Suisse**, la **Grande-Bretagne**, la **France**, la **Suède**, les **Pays-Bas**, le **Danemark** et la **Corée**.