Projet 2 - Analyse des données de systèmes éducatifs

1. Introduction à la problématique

Objectives

- Valider la qualité du jeu de données
- Décrire les informations contenues dans le jeu de données
  - regarder les informations sur les données, indicateurs
- Selectionner les information pertinentes pour répondre à la problématique
  - identifier les bonnes indicateurs
- Déterminer des ordres de grandeurs des indicateurs statistiques classiques pour différents groupes
  - identifier les groupes pertinents
  - définier les indicateurs et leurs ordres de grandeurs
  - analysez les données
- Présentez l'analyse de données / le travail fait
  - faires des graphiques pertinents et lisible
  - formuler une synthèse

Contexte

Une start-up de la EdTech propose des contenus de formation en ligne pour un public de niveau lycée et université et considère une expansion à l'international

Questions générales

- Quels sont les pays avec un fort potentiel de clients pour les services de la start-up ?
- Pour chacun de ces pays, quel sera l'évolution de ce potentiel de clients ?
- Dans quel pays l'enterprise doit-elle opérer en priorité ?

Questions spécifiques

- 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)

Ressources

Les données de la Banque mondiale :

- https://datacatalog.worldbank.org/dataset/education-statistics
- http://datatopics.worldbank.org/education/

2. Tour des Données

2.1. Préparation des Outils

- imports des libraries Python spécialisées : numpy, matplotlib, pandas, seaborn, ... après les avoir installer avec pip
- définir la configuration des modules graphiques

In [1]:

```
#imports: regular expresssions and operating system
import re,os, math

#data modules
import numpy as np
import pandas as pd
import scipy as sp

#graphic modules
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
sns.set() #sets the theme of seaborn
#creates static png images of the plots within the notebook (other option: 'notebook
%matplotlib inline
```

2.2 Import des données

In [3]:

```
#import data locally
#(could do it remotely)
PATH_TO_DATA = os.getcwd() + '\\Edstats_csv'

RAW_DATA = pd.read_csv(PATH_TO_DATA +'\\EdStatsData.csv')
COUNTRY_DATA = pd.read_csv(PATH_TO_DATA +'\\EdStatsCountry.csv')
COUNTRYSERIES_DATA = pd.read_csv(PATH_TO_DATA +'\\EdStatsCountry-Series.csv')
FOOTNOTE_DATA = pd.read_csv(PATH_TO_DATA +'\\EdStatsFootNote.csv')
SERIES_DATA = pd.read_csv(PATH_TO_DATA +'\\EdStatsSeries.csv')


DATA_ALL_DIC = {"Data" : RAW_DATA, "Country Data" : COUNTRY_DATA, "Country Series" :
            "Series Data" : SERIES_DATA, "Footnote Data" : FOOTNOTE_DATA}
```

2.3 Comprendre les bloques des données

- premiers tentatives de regarder les données de plus près
  - info(), head(), describe(), shape()
- comprendre l'organisation des données, les reférences entre les differents fichiers / données
- utiliser l'aide de visulisation de la Banque mondiale : https://databank.worldbank.org/source/education-statistics-%5E-all-indicators/Type/TABLE/preview/on

- lire la documentation sur les données

In [4]:
```
def show_data_info(dataname, data):
    print(50*'*')
    print(10*'*',dataname,10*'*')
    print(30*'-')
    print(f'The dataframe has the dimensions: {data.shape}')
    print(30*'-')
    print(10*'*','head()',10*'*')
    print(data.head())
    print(30*'-')
    print(10*'*','info()',10*'*')
    print(data.info())
    print(30*'-')
    print(10*'*','describe()',10*'*')
    print(data.describe())
    print(50*'_')
```

In [5]:
```
#have a quick first look at the data
for k,v in DATA_ALL_DIC.items():
    show_data_info(k,v)
```

```
**************************************************
********** Data **********
------------------------------
The dataframe has the dimensions: (886930, 70)
------------------------------
********** head() **********
  Country Name Country Code  \
0   Arab World          ARB
1   Arab World          ARB
2   Arab World          ARB
3   Arab World          ARB
4   Arab World          ARB

                                     Indicator Name  Indicator Code  \
0  Adjusted net enrolment rate, lower secondary, ...      UIS.NERA.2
1  Adjusted net enrolment rate, lower secondary, ...    UIS.NERA.2.F
2  Adjusted net enrolment rate, lower secondary, ...  UIS.NERA.2.GPI
3  Adjusted net enrolment rate, lower secondary, ...    UIS.NERA.2.M
4  Adjusted net enrolment rate, primary, both sex...     SE.PRM.TENR

        1970       1971       1972       1973       1974      1975  ...  2060  \
0        NaN        NaN        NaN        NaN        NaN       NaN  ...   NaN
1        NaN        NaN        NaN        NaN        NaN       NaN  ...   NaN
2        NaN        NaN        NaN        NaN        NaN       NaN  ...   NaN
3        NaN        NaN        NaN        NaN        NaN       NaN  ...   NaN
4  54.822121  54.894138  56.209438  57.267109  57.991138  59.36554  ...   NaN

   2065  2070  2075  2080  2085  2090  2095  2100  Unnamed: 69
0   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN          NaN
1   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN          NaN
2   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN          NaN
3   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN          NaN
4   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN          NaN

[5 rows x 70 columns]
------------------------------
********** info() **********
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 886930 entries, 0 to 886929
Data columns (total 70 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Country Name    886930 non-null  object
```

```
 1   Country Code    886930 non-null  object
 2   Indicator Name  886930 non-null  object
 3   Indicator Code  886930 non-null  object
 4   1970            72288 non-null   float64
 5   1971            35537 non-null   float64
 6   1972            35619 non-null   float64
 7   1973            35545 non-null   float64
 8   1974            35730 non-null   float64
 9   1975            87306 non-null   float64
 10  1976            37483 non-null   float64
 11  1977            37574 non-null   float64
 12  1978            37576 non-null   float64
 13  1979            36809 non-null   float64
 14  1980            89122 non-null   float64
 15  1981            38777 non-null   float64
 16  1982            37511 non-null   float64
 17  1983            38460 non-null   float64
 18  1984            38606 non-null   float64
 19  1985            90296 non-null   float64
 20  1986            39372 non-null   float64
 21  1987            38641 non-null   float64
 22  1988            38552 non-null   float64
 23  1989            37540 non-null   float64
 24  1990            124405 non-null  float64
 25  1991            74437 non-null   float64
 26  1992            75543 non-null   float64
 27  1993            75793 non-null   float64
 28  1994            77462 non-null   float64
 29  1995            131361 non-null  float64
 30  1996            76807 non-null   float64
 31  1997            73453 non-null   float64
 32  1998            84914 non-null   float64
 33  1999            118839 non-null  float64
 34  2000            176676 non-null  float64
 35  2001            123509 non-null  float64
 36  2002            124205 non-null  float64
 37  2003            130363 non-null  float64
 38  2004            128814 non-null  float64
 39  2005            184108 non-null  float64
 40  2006            140312 non-null  float64
 41  2007            137272 non-null  float64
 42  2008            134387 non-null  float64
 43  2009            142108 non-null  float64
 44  2010            242442 non-null  float64
 45  2011            146012 non-null  float64
 46  2012            147264 non-null  float64
 47  2013            137509 non-null  float64
 48  2014            113789 non-null  float64
 49  2015            131058 non-null  float64
 50  2016            16460 non-null   float64
 51  2017            143 non-null     float64
 52  2020            51436 non-null   float64
 53  2025            51436 non-null   float64
 54  2030            51436 non-null   float64
 55  2035            51436 non-null   float64
 56  2040            51436 non-null   float64
 57  2045            51436 non-null   float64
 58  2050            51436 non-null   float64
 59  2055            51436 non-null   float64
 60  2060            51436 non-null   float64
 61  2065            51436 non-null   float64
 62  2070            51436 non-null   float64
 63  2075            51436 non-null   float64
 64  2080            51436 non-null   float64
 65  2085            51436 non-null   float64
 66  2090            51436 non-null   float64
 67  2095            51436 non-null   float64
 68  2100            51436 non-null   float64
 69  Unnamed: 69     0 non-null       float64
```

```
dtypes: float64(66), object(4)
memory usage: 473.7+ MB
None
------------------------------
********** describe() **********
               1970          1971          1972          1973          1974  \
count  7.228800e+04  3.553700e+04  3.561900e+04  3.554500e+04  3.573000e+04
mean   1.974772e+09  4.253638e+09  4.592365e+09  5.105006e+09  5.401493e+09
std    1.211687e+11  1.804814e+11  1.914083e+11  2.059170e+11  2.112150e+11
min   -1.435564e+00 -1.594625e+00 -3.056522e+00 -4.032582e+00 -4.213563e+00
25%    8.900000e-01  8.853210e+00  9.240920e+00  9.595200e+00  9.861595e+00
50%    6.317724e+00  6.316240e+01  6.655139e+01  6.969595e+01  7.087760e+01
75%    6.251250e+01  5.655200e+04  5.863650e+04  6.202900e+04  6.383675e+04
max    1.903929e+13  1.986457e+13  2.100916e+13  2.238367e+13  2.282991e+13

               1975          1976          1977          1978          1979  \
count  8.730600e+04  3.748300e+04  3.757400e+04  3.757600e+04  3.680900e+04
mean   2.314288e+09  5.731808e+09  6.124437e+09  6.671489e+09  7.436724e+09
std    1.375059e+11  2.215546e+11  2.325489e+11  2.473986e+11  2.660957e+11
min   -3.658569e+00 -2.950945e+00 -3.174870e+00 -3.558749e+00 -2.973612e+00
25%    1.400000e+00  9.312615e+00  9.519913e+00  1.000000e+01  1.000000e+01
50%    9.677420e+00  7.101590e+01  7.133326e+01  7.290512e+01  7.510173e+01
75%    7.854163e+01  5.682800e+04  5.739175e+04  5.940425e+04  6.411500e+04
max    2.300634e+13  2.424128e+13  2.521383e+13  2.622101e+13  2.730873e+13

       ...          2060          2065          2070          2075  \
count  ...  5.143600e+04  5.143600e+04  5.143600e+04  5.143600e+04
mean   ...  7.224868e+02  7.271290e+02  7.283779e+02  7.266484e+02
std    ...  2.215845e+04  2.287990e+04  2.352338e+04  2.408149e+04
min    ... -1.630000e+00 -1.440000e+00 -1.260000e+00 -1.090000e+00
25%    ...  3.000000e-02  3.000000e-02  2.000000e-02  2.000000e-02
50%    ...  2.300000e-01  2.300000e-01  2.300000e-01  2.300000e-01
75%    ...  7.505000e+00  7.500000e+00  7.300000e+00  7.100000e+00
max    ...  2.951569e+06  3.070879e+06  3.169711e+06  3.246239e+06

               2080          2085          2090          2095          2100  \
count  5.143600e+04  5.143600e+04  5.143600e+04  5.143600e+04  5.143600e+04
mean   7.228327e+02  7.176899e+02  7.113072e+02  7.034274e+02  6.940296e+02
std    2.455897e+04  2.496587e+04  2.530183e+04  2.556069e+04  2.574189e+04
min   -9.200000e-01 -7.800000e-01 -6.500000e-01 -5.500000e-01 -4.500000e-01
25%    1.000000e-02  1.000000e-02  1.000000e-02  1.000000e-02  1.000000e-02
50%    2.300000e-01  2.300000e-01  2.300000e-01  2.300000e-01  2.200000e-01
75%    6.722500e+00  6.080000e+00  5.462500e+00  4.680000e+00  4.032500e+00
max    3.301586e+06  3.337871e+06  3.354746e+06  3.351887e+06  3.330484e+06

       Unnamed: 69
count          0.0
mean           NaN
std            NaN
min            NaN
25%            NaN
50%            NaN
75%            NaN
max            NaN

[8 rows x 66 columns]
_____________________________________________________
**************************************************
********** Country Data **********
------------------------------
The dataframe has the dimensions: (241, 32)
------------------------------
********** head() **********
  Country Code   Short Name   Table Name                     Long Name  \
0          ABW        Aruba        Aruba                         Aruba
1          AFG  Afghanistan  Afghanistan  Islamic State of Afghanistan
2          AGO       Angola       Angola   People's Republic of Angola
3          ALB      Albania      Albania           Republic of Albania
4          AND      Andorra      Andorra       Principality of Andorra
```

```
  2-alpha code   Currency Unit  \
0           AW   Aruban florin
1           AF  Afghan afghani
2           AO  Angolan kwanza
3           AL    Albanian lek
4           AD            Euro

                                      Special Notes  \
0  SNA data for 2000-2011 are updated from offici...
1  Fiscal year end: March 20; reporting period fo...
2  April 2013 database update: Based on IMF data,...
3                                                NaN
4                                                NaN

                     Region          Income Group WB-2 code  ...  \
0  Latin America & Caribbean  High income: nonOECD        AW  ...
1                South Asia            Low income        AF  ...
2        Sub-Saharan Africa   Upper middle income        AO  ...
3     Europe & Central Asia   Upper middle income        AL  ...
4     Europe & Central Asia  High income: nonOECD        AD  ...

           IMF data dissemination standard  \
0                                      NaN
1  General Data Dissemination System (GDDS)
2  General Data Dissemination System (GDDS)
3  General Data Dissemination System (GDDS)
4                                      NaN

                           Latest population census  \
0                                               2010
1                                               1979
2                                               1970
3                                               2011
4  2011. Population figures compiled from adminis...

                            Latest household survey  \
0                                                NaN
1  Multiple Indicator Cluster Survey (MICS), 2010/11
2              Malaria Indicator Survey (MIS), 2011
3      Demographic and Health Survey (DHS), 2008/09
4                                                NaN

   Source of most recent Income and expenditure data  \
0                                                NaN
1           Integrated household survey (IHS), 2008
2           Integrated household survey (IHS), 2008
3  Living Standards Measurement Study Survey (LSM...
4                                                NaN

  Vital registration complete Latest agricultural census  \
0                        Yes                        NaN
1                        NaN                    2013/14
2                        NaN                       2015
3                        Yes                       2012
4                        Yes                        NaN

  Latest industrial data Latest trade data Latest water withdrawal data  \
0                    NaN            2012.0                          NaN
1                    NaN            2012.0                         2000
2                    NaN               NaN                         2005
3                 2010.0            2012.0                         2006
4                    NaN            2006.0                          NaN

  Unnamed: 31
0         NaN
1         NaN
2         NaN
3         NaN
```

```
4         NaN

[5 rows x 32 columns]
------------------------------
********** info() **********
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241 entries, 0 to 240
Data columns (total 32 columns):
 #   Column                                             Non-Null Count  Dtype
---  ------                                             --------------  -----
 0   Country Code                                       241 non-null    object
 1   Short Name                                         241 non-null    object
 2   Table Name                                         241 non-null    object
 3   Long Name                                          241 non-null    object
 4   2-alpha code                                       238 non-null    object
 5   Currency Unit                                      215 non-null    object
 6   Special Notes                                      145 non-null    object
 7   Region                                             214 non-null    object
 8   Income Group                                       214 non-null    object
 9   WB-2 code                                          240 non-null    object
 10  National accounts base year                        205 non-null    object
 11  National accounts reference year                   32 non-null     float64
 12  SNA price valuation                                197 non-null    object
 13  Lending category                                   144 non-null    object
 14  Other groups                                       58 non-null     object
 15  System of National Accounts                        215 non-null    object
 16  Alternative conversion factor                      47 non-null     object
 17  PPP survey year                                    145 non-null    object
 18  Balance of Payments Manual in use                  181 non-null    object
 19  External debt Reporting status                     124 non-null    object
 20  System of trade                                    200 non-null    object
 21  Government Accounting concept                      161 non-null    object
 22  IMF data dissemination standard                    181 non-null    object
 23  Latest population census                           213 non-null    object
 24  Latest household survey                            141 non-null    object
 25  Source of most recent Income and expenditure data  160 non-null    object
 26  Vital registration complete                        111 non-null    object
 27  Latest agricultural census                         142 non-null    object
 28  Latest industrial data                             107 non-null    float64
 29  Latest trade data                                  185 non-null    float64
 30  Latest water withdrawal data                       179 non-null    object
 31  Unnamed: 31                                        0 non-null      float64
dtypes: float64(4), object(28)
memory usage: 60.4+ KB
None
------------------------------
********** describe() **********
       National accounts reference year  Latest industrial data  \
count                          32.00000              107.000000
mean                         2001.53125             2008.102804
std                             5.24856                2.616834
min                          1987.00000             2000.000000
25%                          1996.75000             2007.500000
50%                          2002.00000             2009.000000
75%                          2005.00000             2010.000000
max                          2012.00000             2010.000000

       Latest trade data  Unnamed: 31
count         185.000000          0.0
mean         2010.994595          NaN
std             2.569675          NaN
min          1995.000000          NaN
25%          2011.000000          NaN
50%          2012.000000          NaN
75%          2012.000000          NaN
max          2012.000000          NaN
_____________________________________________________
**************************************************
********** Country Series **********
```

```
------------------------------
The dataframe has the dimensions: (613, 4)
------------------------------
********** head() **********
  CountryCode         SeriesCode  \
0         ABW        SP.POP.TOTL
1         ABW        SP.POP.GROW
2         AFG        SP.POP.GROW
3         AFG  NY.GDP.PCAP.PP.CD
4         AFG        SP.POP.TOTL

                                        DESCRIPTION  Unnamed: 3
0  Data sources : United Nations World Population...         NaN
1  Data sources: United Nations World Population ...         NaN
2  Data sources: United Nations World Population ...         NaN
3                 Estimates are based on regression.         NaN
4  Data sources : United Nations World Population...         NaN
------------------------------
********** info() **********
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 613 entries, 0 to 612
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   CountryCode  613 non-null    object
 1   SeriesCode   613 non-null    object
 2   DESCRIPTION  613 non-null    object
 3   Unnamed: 3   0 non-null      float64
dtypes: float64(1), object(3)
memory usage: 19.3+ KB
None
------------------------------
********** describe() **********
       Unnamed: 3
count         0.0
mean          NaN
std           NaN
min           NaN
25%           NaN
50%           NaN
75%           NaN
max           NaN
____________________________________________________
****************************************************
********** Series Data **********
------------------------------
The dataframe has the dimensions: (3665, 21)
------------------------------
********** head() **********
           Series Code       Topic  \
0  BAR.NOED.1519.FE.ZS  Attainment
1     BAR.NOED.1519.ZS  Attainment
2  BAR.NOED.15UP.FE.ZS  Attainment
3     BAR.NOED.15UP.ZS  Attainment
4  BAR.NOED.2024.FE.ZS  Attainment

                                     Indicator Name  \
0  Barro-Lee: Percentage of female population age...
1  Barro-Lee: Percentage of population age 15-19 ...
2  Barro-Lee: Percentage of female population age...
3  Barro-Lee: Percentage of population age 15+ wi...
4  Barro-Lee: Percentage of female population age...

                                   Short definition  \
0  Percentage of female population age 15-19 with...
1  Percentage of population age 15-19 with no edu...
2  Percentage of female population age 15+ with n...
3  Percentage of population age 15+ with no educa...
4  Percentage of female population age 20-24 with...
```

```
                                     Long definition  Unit of measure  \
0  Percentage of female population age 15-19 with...              NaN
1  Percentage of population age 15-19 with no edu...              NaN
2  Percentage of female population age 15+ with n...              NaN
3  Percentage of population age 15+ with no educa...              NaN
4  Percentage of female population age 20-24 with...              NaN

  Periodicity Base Period Other notes Aggregation method  ...  \
0         NaN         NaN         NaN                NaN  ...
1         NaN         NaN         NaN                NaN  ...
2         NaN         NaN         NaN                NaN  ...
3         NaN         NaN         NaN                NaN  ...
4         NaN         NaN         NaN                NaN  ...

  Notes from original source  General comments  \
0                        NaN               NaN
1                        NaN               NaN
2                        NaN               NaN
3                        NaN               NaN
4                        NaN               NaN

                                            Source  \
0  Robert J. Barro and Jong-Wha Lee: http://www.b...
1  Robert J. Barro and Jong-Wha Lee: http://www.b...
2  Robert J. Barro and Jong-Wha Lee: http://www.b...
3  Robert J. Barro and Jong-Wha Lee: http://www.b...
4  Robert J. Barro and Jong-Wha Lee: http://www.b...

  Statistical concept and methodology Development relevance  \
0                                NaN                   NaN
1                                NaN                   NaN
2                                NaN                   NaN
3                                NaN                   NaN
4                                NaN                   NaN

  Related source links Other web links  Related indicators  License Type  \
0                  NaN             NaN                 NaN           NaN
1                  NaN             NaN                 NaN           NaN
2                  NaN             NaN                 NaN           NaN
3                  NaN             NaN                 NaN           NaN
4                  NaN             NaN                 NaN           NaN

   Unnamed: 20
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN

[5 rows x 21 columns]
------------------------------
********** info() **********
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3665 entries, 0 to 3664
Data columns (total 21 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Series Code                         3665 non-null   object
 1   Topic                               3665 non-null   object
 2   Indicator Name                      3665 non-null   object
 3   Short definition                    2156 non-null   object
 4   Long definition                     3665 non-null   object
 5   Unit of measure                     0 non-null      float64
 6   Periodicity                         99 non-null     object
 7   Base Period                         314 non-null    object
 8   Other notes                         552 non-null    object
 9   Aggregation method                  47 non-null     object
 10  Limitations and exceptions          14 non-null     object
```

```
 11  Notes from original source           0 non-null      float64
 12  General comments                     14 non-null     object
 13  Source                               3665 non-null   object
 14  Statistical concept and methodology  23 non-null     object
 15  Development relevance                3 non-null      object
 16  Related source links                 215 non-null    object
 17  Other web links                      0 non-null      float64
 18  Related indicators                   0 non-null      float64
 19  License Type                         0 non-null      float64
 20  Unnamed: 20                          0 non-null      float64
dtypes: float64(6), object(15)
memory usage: 601.4+ KB
None
------------------------------
********** describe() **********
       Unit of measure  Notes from original source  Other web links  \
count              0.0                         0.0              0.0
mean               NaN                         NaN              NaN
std                NaN                         NaN              NaN
min                NaN                         NaN              NaN
25%                NaN                         NaN              NaN
50%                NaN                         NaN              NaN
75%                NaN                         NaN              NaN
max                NaN                         NaN              NaN

       Related indicators  License Type  Unnamed: 20
count                 0.0           0.0          0.0
mean                  NaN           NaN          NaN
std                   NaN           NaN          NaN
min                   NaN           NaN          NaN
25%                   NaN           NaN          NaN
50%                   NaN           NaN          NaN
75%                   NaN           NaN          NaN
max                   NaN           NaN          NaN
_____________________________________________________
**************************************************
********** Footnote Data **********
------------------------------
The dataframe has the dimensions: (643638, 5)
------------------------------
********** 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
------------------------------
********** info() **********
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 643638 entries, 0 to 643637
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   CountryCode  643638 non-null  object
 1   SeriesCode   643638 non-null  object
 2   Year         643638 non-null  object
 3   DESCRIPTION  643638 non-null  object
 4   Unnamed: 4   0 non-null       float64
dtypes: float64(1), object(4)
memory usage: 24.6+ MB
None
------------------------------
********** describe() **********
       Unnamed: 4
count         0.0
mean          NaN
std           NaN
min           NaN
```

```
25%           NaN
50%           NaN
75%           NaN
max           NaN
```

Résumé: Data Series - EdstatsData.csv

- Il y a **886930 lignes** et **70 colonnes**.

- **Les colonnes 0 à 3 sont des identificateurs** de la data (pays (2x) et data series (2x))
  - La data est de type 'object'.
  - La data est complète : tous les 886930 lignes sont non-null.
  - Ce sont les pays et les variables. (nom et code)
- **Les colonnes 4 à 68 sont des années**
  - Ce sont 65 colonnes.
  - La data est de type 'float64' : soit quantitatif (discrète ou continue) soit ordinal qualitatif.
  - La data n'est pas très complète. (Une estimation rapide: moins que 10% sont non-null)
  - Les années vont de 1970 à 2100.
    - Les dates dans le future doivent être des projections.
- **La colonne 69 est vide**.

- **Ici, on trouve la data sur les variables éducatives des différents pays**.

Résumé: Country Data - EdstatsCountry.csv

- Il y a **241 lignes** et **32 colonnes**.

- **Les colonnes 0 à 3 sont des identificateurs**
  - La data est de type 'object'.
  - La data est complète: tous les 271 lignes sont non-null.
  - Ce sont le code et noms de pay / région.
- **Les colonnes 4 à 30 donnent plus des informations sur les pays**
  - Ce sont 27 colonnes.
  - La plus part de la data est de type 'object'
    - sauf les colonnes 11, 28 et 29. Elles sont des 'float64' et donnent des années.
  - La data est plutôt complète. (Une estimation rapide: pour la plus part, plus que 60% des entrées sont non-null.)
- **La colonne 31 est vide**.

- **Ce tableau relie le 'code pays' à plus d'informations sur le pays/la région, c'est-à-dire le nom court et long, la description de la région et donne des infos caractèristiques sur les pays (devise, région, classment, dernière data d'enquète, etc.)**
- Il y a **241 différents classifications de pays / région.**

Résumé: Country Series - EdstatsCountry-Series.csv

- Il y a **613 lignes** et **4 colonnes**.

- **Les colonnes 0 à 1 sont des identificateurs**.
  - La data est de type 'object'.
  - La data est complète: tous les 613 lignes sont non-null.

    - Ce sont les codes de pays et de la variable.
- **La colonnes 3 donne la déscription de la variable du pays**.
    - La la data est de type 'object' : qualitatif nominale.
    - La data est complète.
- **La colonne 3 est vide**.

- **Ce tableau donne des déscriptions de certaines variables d'un pays**.

Résumé: Series Data - Edstats-Series.csv

- Il y a **3665 lignes** et **21 colonnes**.

- **La colonnes 0 est l'identificateur**.
    - La data est de type 'object'.
    - C'est le code de la variable.
- **Les colonnes 1 à 4 donnent la définition de la variable**.
    - La data est de type 'object' : qualitatif nominale.
    - La data est presque complète.
        - sauf la colonne 3, pas tous les variable ont une définition courte ('Short Description).
- **Les colonnes 5 à 19 donnent plus des détails sur la variable**.
    - La data est de type 'object'.
    - Ces colonnes ne sont pas bien remplies à l'exception de la colonne 13.
        - **Les colonnes 5, 11, 17, 18, 19 sont complètement vides**.
        - **La colonne 13 est complète. Elle donne la source de la data.**
- **La colonne 20 est vide**.

- **Ce tableau relie le code d'une variable ('series code') avec son nom, sa définition et sa source**.

Résumé: Footnote Data - EdstatsFootnote.csv

- Il y a **643638 lignes** et **5 colonnes**.

- **Les colonnes 0 à 2 sont des identificateurs**
    - La data est de type 'object'.
    - La data est complète.
    - Ce sont les pays,les variables et l'année.
- **La colonne 3 est une description**
    - La data est de type 'object' .
    - La data est complète.
- **La colonne 4 est vide**.

- **Ce tableau des notes de bas de page sur les datas**.
    - les variables éducatives des différents pays pour un année.

Résumé:

Pour l'analyse de données on va principalement avoir besoin de:

- EdstatsData.csv, qui contient les données à analyser
    - un point de données est donné par les parametres: variable, pays/régions, année

- EdstatsSeries.csv, qui contient les définitions des variables
- EdstataCountry.csv, qui contient les définitions des pays, régions et groupes de pays

2.2 La qualité du jeu de données

- Re-formater le jeu de données
  - laisser tomber les colonnes indices dupliquées
  - travailler avec les codes de pays/région et de variable
  - re-nommer les indices/colonnes des parametres ('indicator', 'country', 'year')
- Regarder l'exhaustivité de l'ensemble de données
  - données manquantes, dupliquées
  - Nombre d'unique variables, pays (lignes) et nombre d'années (colonnes)

In [6]:

```
def empty_rows(data):
    '''return a list with the indices of the empty rows'''
    tmp = data.notna().T.sum()
    return [tmp.index[i] for i,s in enumerate(tmp) if s == 0 ]

def empty_columns(data):
    '''returns a list with the column names of the empty columns'''
    tmp = data.notna().sum()
    return [tmp.index[i] for i,s in enumerate(tmp) if s == 0 ]
```

In [7]:

```
ALL_COUNTRIES = RAW_DATA['Country Code'].unique()
ALL_INDICATORS = RAW_DATA['Indicator Code'].unique()
#the first four columns are ID columns and the last column is empty.
data = RAW_DATA.iloc[:,4:-1]
ALL_YEARS = data.columns.values
num_cc = len(ALL_COUNTRIES)
num_ind = len(ALL_INDICATORS)
num_yyyy = len(ALL_YEARS)
print("Unique countries :", num_cc)
print("Unique indicators:", num_ind)
print("Number of unique country-indicator indices :", num_cc*num_ind)
print("Number of years for which there is data:", num_yyyy)
print("Total number of data points :", num_cc*num_ind*num_yyyy)
print(f"{data.isna().sum().sum()} data points are empty = {100*data.isna().sum().sum

print(f"There are {len(data.index.values)} rows.")
print(f"There are {len(empty_rows(data))} empty rows = {100*len(empty_rows(data))/le
print(f"There are {len(data.columns.values)} columns.")
print(f"There are {len(empty_columns(data))} empty columns = {100*len(empty_columns(
```

```
Unique countries : 242
Unique indicators: 3665
Number of unique country-indicator indices : 886930
Number of years for which there is data: 65
Total number of data points : 57650450
52568249 data points are empty = 91.18445562870714%.
There are 886930 rows.
There are 529525 empty rows = 59.703133279965726%.
There are 65 columns.
There are 0 empty columns = 0.0%
```

Résume : données en générale

- **242 uniques pays/régions**
- **3665 uniques variables**
- **65 années**

- **886930 combinisons de pays-variable**
  - C'est égal au nombre des lignes. **Il n'y a pas des duplicates dans les donnés**. (sauf s'il y a des erreur de mise en forme.)
  - Mais 529525 lignes sont vides. Il n'y a pas tous les variables pour tous les pays.
  - **approx. 60% des lignes sont vides**.

- **65 années**
  - Il n'y aucune année pour laquelle il n'y a pas aucune entrée.

- **57650450 combinisons de pays-variable-année**
  - **Moins de 10% des données sont complètes.**

3. Nettoyer les données

- regarder les pays, les variables et les années de plus près
  - choisir des pays et des régions / zones géographiques, écarter les groupes de pays non-géographiques
  - réduire les années aux 2000 à 2050
  - choisir les variables pertinents à la problématique

3.1 Les pays et les régions

- choisir les pays et les régions
- écarter les groupes de pays non-géographiques
- diviser en pays et régions geographiques
- utiliser EdstatsCountry.csv' / 'Country Data' / COUNTRY_DATA

In [8]:

```
ALL_COUNTRIES_COUNTRY = COUNTRY_DATA['Country Code'].unique()
print("Countries in Country Data :\t", len(ALL_COUNTRIES_COUNTRY))
print("Countries in the Data Series :\t", len(ALL_COUNTRIES))
```

```
Countries in Country Data :      241
Countries in the Data Series :   242
```

In [9]:

```
#look at the set differences to see if the Country codes in the two data set are the
#See what the extra country in the Data Set.
print("Set difference of the countries in Data Series and Country Data: ",set(ALL_CO
```

```
Set difference of the countries in Data Series and Country Data:  {'VGB'}
```

In [10]:

```
cc = 'VGB'
print(f'The country name of {cc} is\n{ RAW_DATA[RAW_DATA["Country Code"] == cc]["Cou
```

```
The country name of VGB is
190580    British Virgin Islands
190581    British Virgin Islands
190582    British Virgin Islands
190583    British Virgin Islands
190584    British Virgin Islands
                   ...
194240    British Virgin Islands
194241    British Virgin Islands
194242    British Virgin Islands
194243    British Virgin Islands
194244    British Virgin Islands
Name: Country Name, Length: 3665, dtype: object.
```

In [11]:

```
#see if there is actual data for VBG = British Virgin Islands,
# and if there is potentially another code, i.e. if there is a formatting error
RAW_DATA[RAW_DATA['Country Code']=='VGB'].info()
print()
print(COUNTRY_DATA[COUNTRY_DATA['Long Name'].str.contains('Brit|United Kingdom|Islan
print(COUNTRY_DATA[COUNTRY_DATA['Short Name'].str.contains('Brit|United Kingdom|Isla
print(COUNTRY_DATA[COUNTRY_DATA['Table Name'].str.contains('Brit|United Kingdom|Isla
print()
print(COUNTRY_DATA[COUNTRY_DATA['Short Name']=='Virgin Islands'])
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3665 entries, 190580 to 194244
Data columns (total 70 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country Name    3665 non-null   object
 1   Country Code    3665 non-null   object
 2   Indicator Name  3665 non-null   object
 3   Indicator Code  3665 non-null   object
 4   1970            10 non-null     float64
 5   1971            29 non-null     float64
 6   1972            25 non-null     float64
 7   1973            31 non-null     float64
 8   1974            22 non-null     float64
 9   1975            38 non-null     float64
 10  1976            38 non-null     float64
 11  1977            12 non-null     float64
 12  1978            12 non-null     float64
 13  1979            38 non-null     float64
 14  1980            13 non-null     float64
 15  1981            39 non-null     float64
 16  1982            7 non-null      float64
 17  1983            32 non-null     float64
 18  1984            42 non-null     float64
 19  1985            31 non-null     float64
 20  1986            7 non-null      float64
 21  1987            7 non-null      float64
 22  1988            7 non-null      float64
 23  1989            7 non-null      float64
 24  1990            33 non-null     float64
 25  1991            42 non-null     float64
 26  1992            33 non-null     float64
 27  1993            8 non-null      float64
 28  1994            36 non-null     float64
 29  1995            21 non-null     float64
 30  1996            7 non-null      float64
 31  1997            7 non-null      float64
 32  1998            8 non-null      float64
 33  1999            66 non-null     float64
 34  2000            51 non-null     float64
 35  2001            51 non-null     float64
 36  2002            93 non-null     float64
 37  2003            77 non-null     float64
 38  2004            77 non-null     float64
 39  2005            72 non-null     float64
 40  2006            69 non-null     float64
 41  2007            71 non-null     float64
 42  2008            15 non-null     float64
 43  2009            104 non-null    float64
 44  2010            60 non-null     float64
 45  2011            65 non-null     float64
 46  2012            48 non-null     float64
 47  2013            50 non-null     float64
 48  2014            69 non-null     float64
 49  2015            92 non-null     float64
 50  2016            14 non-null     float64
 51  2017            0 non-null      float64
```

```
 52  2020            0 non-null      float64
 53  2025            0 non-null      float64
 54  2030            0 non-null      float64
 55  2035            0 non-null      float64
 56  2040            0 non-null      float64
 57  2045            0 non-null      float64
 58  2050            0 non-null      float64
 59  2055            0 non-null      float64
 60  2060            0 non-null      float64
 61  2065            0 non-null      float64
 62  2070            0 non-null      float64
 63  2075            0 non-null      float64
 64  2080            0 non-null      float64
 65  2085            0 non-null      float64
 66  2090            0 non-null      float64
 67  2095            0 non-null      float64
 68  2100            0 non-null      float64
 69  Unnamed: 69     0 non-null      float64
dtypes: float64(66), object(4)
memory usage: 2.0+ MB

['Channel Islands' 'Cayman Islands' 'Faeroe Islands'
 'United Kingdom of Great Britain and Northern Ireland'
 'Republic of the Marshall Islands'
 'Commonwealth of the Northern Mariana Islands' 'Solomon Islands'
 'Turks and Caicos Islands' 'Virgin Islands of the United States']
['Channel Islands' 'Cayman Islands' 'Faeroe Islands' 'United Kingdom'
 'Marshall Islands' 'Northern Mariana Islands' 'Solomon Islands'
 'Turks and Caicos Islands' 'Virgin Islands']
['Channel Islands' 'Cayman Islands' 'Faeroe Islands' 'United Kingdom'
 'Marshall Islands' 'Northern Mariana Islands' 'Solomon Islands'
 'Turks and Caicos Islands' 'Virgin Islands (U.S.)']

    Country Code      Short Name             Table Name  \
231          VIR  Virgin Islands  Virgin Islands (U.S.)

                               Long Name 2-alpha code Currency Unit  \
231  Virgin Islands of the United States           VI   U.S. dollar

    Special Notes                     Region          Income Group WB-2 code  \
231           NaN  Latin America & Caribbean  High income: nonOECD        VI

     ... IMF data dissemination standard  Latest population census  \
231  ...                             NaN                      2010

    Latest household survey Source of most recent Income and expenditure data  \
231                     NaN                                                NaN

    Vital registration complete Latest agricultural census  \
231                         Yes                       2007

    Latest industrial data Latest trade data Latest water withdrawal data  \
231                    NaN               NaN                          NaN

    Unnamed: 31
231         NaN

[1 rows x 32 columns]
```

Résumé: 'Country Code'

- Les 241 'Country Code' dans la 'Country Data' (EdstatsCountry.csv) sont les même que dans le 'Data Series' (EdstatsData.csv).
- Il y a un pays en plus dans le 'Data Series' : Les îles vierges britanniques avec le code 'VGB'.

In [12]:

```
for cc in ALL_COUNTRIES:
    print(COUNTRY_DATA[COUNTRY_DATA['Country Code'] == cc].loc[:,['Country Code',
```

```
                                                                    'Table Name',
                                                                    'Long Name',
                                                                    'Region',
                                                                    'Income Group']].v
```

```
[['ARB' 'Arab World' 'Arab World' nan nan]]
[['EAS' 'East Asia & Pacific (all income levels)'
  'East Asia & Pacific (all income levels)' nan nan]]
[['EAP' 'East Asia & Pacific' 'East Asia & Pacific (developing only)' nan
  nan]]
[['EMU' 'Euro area' 'Euro area' nan nan]]
[['ECS' 'Europe & Central Asia (all income levels)'
  'Europe & Central Asia (all income levels)' nan nan]]
[['ECA' 'Europe & Central Asia' 'Europe & Central Asia (developing only)'
  nan nan]]
[['EUU' 'European Union' 'European Union' nan nan]]
[['HPC' 'Heavily indebted poor countries (HIPC)'
  'Heavily indebted poor countries (HIPC)' nan nan]]
[['HIC' 'High income' 'High income' nan nan]]
[['LCN' 'Latin America & Caribbean (all income levels)'
  'Latin America & Caribbean (all income levels)' nan nan]]
[['LAC' 'Latin America & Caribbean'
  'Latin America & Caribbean (developing only)' nan nan]]
[['LDC' 'Least developed countries: UN classification'
  'Least developed countries: UN classification' nan nan]]
[['LMY' 'Low & middle income' 'Low & middle income' nan nan]]
[['LIC' 'Low income' 'Low income' nan nan]]
[['LMC' 'Lower middle income' 'Lower middle income' nan nan]]
[['MEA' 'Middle East & North Africa (all income levels)'
  'Middle East & North Africa (all income levels)' nan nan]]
[['MNA' 'Middle East & North Africa'
  'Middle East & North Africa (developing only)' nan nan]]
[['MIC' 'Middle income' 'Middle income' nan nan]]
[['NAC' 'North America' 'North America' nan nan]]
[['OED' 'OECD members' 'OECD members' nan nan]]
[['SAS' 'South Asia' 'South Asia' nan nan]]
[['SSF' 'Sub-Saharan Africa (all income levels)'
  'Sub-Saharan Africa (all income levels)' nan nan]]
[['SSA' 'Sub-Saharan Africa' 'Sub-Saharan Africa (developing only)' nan
  nan]]
[['UMC' 'Upper middle income' 'Upper middle income' nan nan]]
[['WLD' 'World' 'World' nan nan]]
[['AFG' 'Afghanistan' 'Islamic State of Afghanistan' 'South Asia'
  'Low income']]
[['ALB' 'Albania' 'Republic of Albania' 'Europe & Central Asia'
  'Upper middle income']]
[['DZA' 'Algeria' "People's Democratic Republic of Algeria"
  'Middle East & North Africa' 'Upper middle income']]
[['ASM' 'American Samoa' 'American Samoa' 'East Asia & Pacific'
  'Upper middle income']]
[['AND' 'Andorra' 'Principality of Andorra' 'Europe & Central Asia'
  'High income: nonOECD']]
[['AGO' 'Angola' "People's Republic of Angola" 'Sub-Saharan Africa'
  'Upper middle income']]
[['ATG' 'Antigua and Barbuda' 'Antigua and Barbuda'
  'Latin America & Caribbean' 'High income: nonOECD']]
[['ARG' 'Argentina' 'Argentine Republic' 'Latin America & Caribbean'
  'Upper middle income']]
[['ARM' 'Armenia' 'Republic of Armenia' 'Europe & Central Asia'
  'Lower middle income']]
[['ABW' 'Aruba' 'Aruba' 'Latin America & Caribbean'
  'High income: nonOECD']]
[['AUS' 'Australia' 'Commonwealth of Australia' 'East Asia & Pacific'
  'High income: OECD']]
[['AUT' 'Austria' 'Republic of Austria' 'Europe & Central Asia'
  'High income: OECD']]
[['AZE' 'Azerbaijan' 'Republic of Azerbaijan' 'Europe & Central Asia'
  'Upper middle income']]
[['BHS' 'Bahamas, The' 'Commonwealth of The Bahamas'
```

```
  'Latin America & Caribbean' 'High income: nonOECD']]
[['BHR' 'Bahrain' 'Kingdom of Bahrain' 'Middle East & North Africa'
  'High income: nonOECD']]
[['BGD' 'Bangladesh' "People's Republic of Bangladesh" 'South Asia'
  'Low income']]
[['BRB' 'Barbados' 'Barbados' 'Latin America & Caribbean'
  'High income: nonOECD']]
[['BLR' 'Belarus' 'Republic of Belarus' 'Europe & Central Asia'
  'Upper middle income']]
[['BEL' 'Belgium' 'Kingdom of Belgium' 'Europe & Central Asia'
  'High income: OECD']]
[['BLZ' 'Belize' 'Belize' 'Latin America & Caribbean'
  'Upper middle income']]
[['BEN' 'Benin' 'Republic of Benin' 'Sub-Saharan Africa' 'Low income']]
[['BMU' 'Bermuda' 'The Bermudas' 'North America' 'High income: nonOECD']]
[['BTN' 'Bhutan' 'Kingdom of Bhutan' 'South Asia' 'Lower middle income']]
[['BOL' 'Bolivia' 'Plurinational State of Bolivia'
  'Latin America & Caribbean' 'Lower middle income']]
[['BIH' 'Bosnia and Herzegovina' 'Bosnia and Herzegovina'
  'Europe & Central Asia' 'Upper middle income']]
[['BWA' 'Botswana' 'Republic of Botswana' 'Sub-Saharan Africa'
  'Upper middle income']]
[['BRA' 'Brazil' 'Federative Republic of Brazil'
  'Latin America & Caribbean' 'Upper middle income']]
[]
[['BRN' 'Brunei Darussalam' 'Brunei Darussalam' 'East Asia & Pacific'
  'High income: nonOECD']]
[['BGR' 'Bulgaria' 'Republic of Bulgaria' 'Europe & Central Asia'
  'Upper middle income']]
[['BFA' 'Burkina Faso' 'Burkina Faso' 'Sub-Saharan Africa' 'Low income']]
[['BDI' 'Burundi' 'Republic of Burundi' 'Sub-Saharan Africa' 'Low income']]
[['CPV' 'Cabo Verde' 'Republic of Cabo Verde' 'Sub-Saharan Africa'
  'Lower middle income']]
[['KHM' 'Cambodia' 'Kingdom of Cambodia' 'East Asia & Pacific'
  'Low income']]
[['CMR' 'Cameroon' 'Republic of Cameroon' 'Sub-Saharan Africa'
  'Lower middle income']]
[['CAN' 'Canada' 'Canada' 'North America' 'High income: OECD']]
[['CYM' 'Cayman Islands' 'Cayman Islands' 'Latin America & Caribbean'
  'High income: nonOECD']]
[['CAF' 'Central African Republic' 'Central African Republic'
  'Sub-Saharan Africa' 'Low income']]
[['TCD' 'Chad' 'Republic of Chad' 'Sub-Saharan Africa' 'Low income']]
[['CHI' 'Channel Islands' 'Channel Islands' 'Europe & Central Asia'
  'High income: nonOECD']]
[['CHL' 'Chile' 'Republic of Chile' 'Latin America & Caribbean'
  'High income: OECD']]
[['CHN' 'China' "People's Republic of China" 'East Asia & Pacific'
  'Upper middle income']]
[['COL' 'Colombia' 'Republic of Colombia' 'Latin America & Caribbean'
  'Upper middle income']]
[['COM' 'Comoros' 'Union of the Comoros' 'Sub-Saharan Africa'
  'Low income']]
[['COD' 'Congo, Dem. Rep.' 'Democratic Republic of the Congo'
  'Sub-Saharan Africa' 'Low income']]
[['COG' 'Congo, Rep.' 'Republic of Congo' 'Sub-Saharan Africa'
  'Lower middle income']]
[['CRI' 'Costa Rica' 'Republic of Costa Rica' 'Latin America & Caribbean'
  'Upper middle income']]
[['CIV' "Côte d'Ivoire" "Republic of Côte d'Ivoire" 'Sub-Saharan Africa'
  'Lower middle income']]
[['HRV' 'Croatia' 'Republic of Croatia' 'Europe & Central Asia'
  'High income: nonOECD']]
[['CUB' 'Cuba' 'Republic of Cuba' 'Latin America & Caribbean'
  'Upper middle income']]
[['CUW' 'Curaçao' 'Curaçao' 'Latin America & Caribbean'
  'High income: nonOECD']]
[['CYP' 'Cyprus' 'Republic of Cyprus' 'Europe & Central Asia'
  'High income: nonOECD']]
```

```
[['CZE' 'Czech Republic' 'Czech Republic' 'Europe & Central Asia'
  'High income: OECD']]
[['DNK' 'Denmark' 'Kingdom of Denmark' 'Europe & Central Asia'
  'High income: OECD']]
[['DJI' 'Djibouti' 'Republic of Djibouti' 'Middle East & North Africa'
  'Lower middle income']]
[['DMA' 'Dominica' 'Commonwealth of Dominica' 'Latin America & Caribbean'
  'Upper middle income']]
[['DOM' 'Dominican Republic' 'Dominican Republic'
  'Latin America & Caribbean' 'Upper middle income']]
[['ECU' 'Ecuador' 'Republic of Ecuador' 'Latin America & Caribbean'
  'Upper middle income']]
[['EGY' 'Egypt, Arab Rep.' 'Arab Republic of Egypt'
  'Middle East & North Africa' 'Lower middle income']]
[['SLV' 'El Salvador' 'Republic of El Salvador'
  'Latin America & Caribbean' 'Lower middle income']]
[['GNQ' 'Equatorial Guinea' 'Republic of Equatorial Guinea'
  'Sub-Saharan Africa' 'High income: nonOECD']]
[['ERI' 'Eritrea' 'State of Eritrea' 'Sub-Saharan Africa' 'Low income']]
[['EST' 'Estonia' 'Republic of Estonia' 'Europe & Central Asia'
  'High income: OECD']]
[['ETH' 'Ethiopia' 'Federal Democratic Republic of Ethiopia'
  'Sub-Saharan Africa' 'Low income']]
[['FRO' 'Faeroe Islands' 'Faeroe Islands' 'Europe & Central Asia'
  'High income: nonOECD']]
[['FJI' 'Fiji' 'Republic of Fiji' 'East Asia & Pacific'
  'Upper middle income']]
[['FIN' 'Finland' 'Republic of Finland' 'Europe & Central Asia'
  'High income: OECD']]
[['FRA' 'France' 'French Republic' 'Europe & Central Asia'
  'High income: OECD']]
[['PYF' 'French Polynesia' 'French Polynesia' 'East Asia & Pacific'
  'High income: nonOECD']]
[['GAB' 'Gabon' 'Gabonese Republic' 'Sub-Saharan Africa'
  'Upper middle income']]
[['GMB' 'Gambia, The' 'Republic of The Gambia' 'Sub-Saharan Africa'
  'Low income']]
[['GEO' 'Georgia' 'Georgia' 'Europe & Central Asia' 'Lower middle income']]
[['DEU' 'Germany' 'Federal Republic of Germany' 'Europe & Central Asia'
  'High income: OECD']]
[['GHA' 'Ghana' 'Republic of Ghana' 'Sub-Saharan Africa'
  'Lower middle income']]
[['GIB' 'Gibraltar' 'Gibraltar' nan nan]]
[['GRC' 'Greece' 'Hellenic Republic' 'Europe & Central Asia'
  'High income: OECD']]
[['GRL' 'Greenland' 'Greenland' 'Europe & Central Asia'
  'High income: nonOECD']]
[['GRD' 'Grenada' 'Grenada' 'Latin America & Caribbean'
  'Upper middle income']]
[['GUM' 'Guam' 'Guam' 'East Asia & Pacific' 'High income: nonOECD']]
[['GTM' 'Guatemala' 'Republic of Guatemala' 'Latin America & Caribbean'
  'Lower middle income']]
[['GIN' 'Guinea' 'Republic of Guinea' 'Sub-Saharan Africa' 'Low income']]
[['GNB' 'Guinea-Bissau' 'Republic of Guinea-Bissau' 'Sub-Saharan Africa'
  'Low income']]
[['GUY' 'Guyana' 'Co-operative Republic of Guyana'
  'Latin America & Caribbean' 'Lower middle income']]
[['HTI' 'Haiti' 'Republic of Haiti' 'Latin America & Caribbean'
  'Low income']]
[['HND' 'Honduras' 'Republic of Honduras' 'Latin America & Caribbean'
  'Lower middle income']]
[['HKG' 'Hong Kong SAR, China'
  "Hong Kong Special Administrative Region of the People's Republic of China"
  'East Asia & Pacific' 'High income: nonOECD']]
[['HUN' 'Hungary' 'Hungary' 'Europe & Central Asia' 'Upper middle income']]
[['ISL' 'Iceland' 'Republic of Iceland' 'Europe & Central Asia'
  'High income: OECD']]
[['IND' 'India' 'Republic of India' 'South Asia' 'Lower middle income']]
[['IDN' 'Indonesia' 'Republic of Indonesia' 'East Asia & Pacific'
```

```
  'Lower middle income']]
 [['IRN' 'Iran, Islamic Rep.' 'Islamic Republic of Iran'
  'Middle East & North Africa' 'Upper middle income']]
 [['IRQ' 'Iraq' 'Republic of Iraq' 'Middle East & North Africa'
  'Upper middle income']]
 [['IRL' 'Ireland' 'Ireland' 'Europe & Central Asia' 'High income: OECD']]
 [['IMN' 'Isle of Man' 'Isle of Man' 'Europe & Central Asia'
  'High income: nonOECD']]
 [['ISR' 'Israel' 'State of Israel' 'Middle East & North Africa'
  'High income: OECD']]
 [['ITA' 'Italy' 'Italian Republic' 'Europe & Central Asia'
  'High income: OECD']]
 [['JAM' 'Jamaica' 'Jamaica' 'Latin America & Caribbean'
  'Upper middle income']]
 [['JPN' 'Japan' 'Japan' 'East Asia & Pacific' 'High income: OECD']]
 [['JOR' 'Jordan' 'Hashemite Kingdom of Jordan'
  'Middle East & North Africa' 'Upper middle income']]
 [['KAZ' 'Kazakhstan' 'Republic of Kazakhstan' 'Europe & Central Asia'
  'Upper middle income']]
 [['KEN' 'Kenya' 'Republic of Kenya' 'Sub-Saharan Africa' 'Low income']]
 [['KIR' 'Kiribati' 'Republic of Kiribati' 'East Asia & Pacific'
  'Lower middle income']]
 [['PRK' 'Korea, Dem. Rep.' "Democratic People's Republic of Korea"
  'East Asia & Pacific' 'Low income']]
 [['KOR' 'Korea, Rep.' 'Republic of Korea' 'East Asia & Pacific'
  'High income: OECD']]
 [['XKX' 'Kosovo' 'Republic of Kosovo' 'Europe & Central Asia'
  'Lower middle income']]
 [['KWT' 'Kuwait' 'State of Kuwait' 'Middle East & North Africa'
  'High income: nonOECD']]
 [['KGZ' 'Kyrgyz Republic' 'Kyrgyz Republic' 'Europe & Central Asia'
  'Lower middle income']]
 [['LAO' 'Lao PDR' "Lao People's Democratic Republic"
  'East Asia & Pacific' 'Lower middle income']]
 [['LVA' 'Latvia' 'Republic of Latvia' 'Europe & Central Asia'
  'High income: nonOECD']]
 [['LBN' 'Lebanon' 'Lebanese Republic' 'Middle East & North Africa'
  'Upper middle income']]
 [['LSO' 'Lesotho' 'Kingdom of Lesotho' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['LBR' 'Liberia' 'Republic of Liberia' 'Sub-Saharan Africa' 'Low income']]
 [['LBY' 'Libya' "Socialist People's Libyan Arab Jamahiriya"
  'Middle East & North Africa' 'Upper middle income']]
 [['LIE' 'Liechtenstein' 'Principality of Liechtenstein'
  'Europe & Central Asia' 'High income: nonOECD']]
 [['LTU' 'Lithuania' 'Republic of Lithuania' 'Europe & Central Asia'
  'High income: nonOECD']]
 [['LUX' 'Luxembourg' 'Grand Duchy of Luxembourg' 'Europe & Central Asia'
  'High income: OECD']]
 [['MAC' 'Macao SAR, China'
  "Macao Special Administrative Region of the People's Republic of China"
  'East Asia & Pacific' 'High income: nonOECD']]
 [['MKD' 'Macedonia, FYR' 'Former Yugoslav Republic of Macedonia'
  'Europe & Central Asia' 'Upper middle income']]
 [['MDG' 'Madagascar' 'Republic of Madagascar' 'Sub-Saharan Africa'
  'Low income']]
 [['MWI' 'Malawi' 'Republic of Malawi' 'Sub-Saharan Africa' 'Low income']]
 [['MYS' 'Malaysia' 'Malaysia' 'East Asia & Pacific' 'Upper middle income']]
 [['MDV' 'Maldives' 'Republic of Maldives' 'South Asia'
  'Upper middle income']]
 [['MLI' 'Mali' 'Republic of Mali' 'Sub-Saharan Africa' 'Low income']]
 [['MLT' 'Malta' 'Republic of Malta' 'Middle East & North Africa'
  'High income: nonOECD']]
 [['MHL' 'Marshall Islands' 'Republic of the Marshall Islands'
  'East Asia & Pacific' 'Upper middle income']]
 [['MRT' 'Mauritania' 'Islamic Republic of Mauritania'
  'Sub-Saharan Africa' 'Lower middle income']]
 [['MUS' 'Mauritius' 'Republic of Mauritius' 'Sub-Saharan Africa'
  'Upper middle income']]
```

```
[['MEX' 'Mexico' 'United Mexican States' 'Latin America & Caribbean'
  'Upper middle income']]
[['FSM' 'Micronesia, Fed. Sts.' 'Federated States of Micronesia'
  'East Asia & Pacific' 'Lower middle income']]
[['MDA' 'Moldova' 'Republic of Moldova' 'Europe & Central Asia'
  'Lower middle income']]
[['MCO' 'Monaco' 'Principality of Monaco' 'Europe & Central Asia'
  'High income: nonOECD']]
[['MNG' 'Mongolia' 'Mongolia' 'East Asia & Pacific' 'Lower middle income']]
[['MNE' 'Montenegro' 'Montenegro' 'Europe & Central Asia'
  'Upper middle income']]
[['MAR' 'Morocco' 'Kingdom of Morocco' 'Middle East & North Africa'
  'Lower middle income']]
[['MOZ' 'Mozambique' 'Republic of Mozambique' 'Sub-Saharan Africa'
  'Low income']]
[['MMR' 'Myanmar' 'Republic of the Union of Myanmar'
  'East Asia & Pacific' 'Low income']]
[['NAM' 'Namibia' 'Republic of Namibia' 'Sub-Saharan Africa'
  'Upper middle income']]
[['NRU' 'Nauru' 'Nauru' nan nan]]
[['NPL' 'Nepal' 'Nepal' 'South Asia' 'Low income']]
[['NLD' 'Netherlands' 'Kingdom of the Netherlands'
  'Europe & Central Asia' 'High income: OECD']]
[['NCL' 'New Caledonia' 'New Caledonia' 'East Asia & Pacific'
  'High income: nonOECD']]
[['NZL' 'New Zealand' 'New Zealand' 'East Asia & Pacific'
  'High income: OECD']]
[['NIC' 'Nicaragua' 'Republic of Nicaragua' 'Latin America & Caribbean'
  'Lower middle income']]
[['NER' 'Niger' 'Republic of Niger' 'Sub-Saharan Africa' 'Low income']]
[['NGA' 'Nigeria' 'Federal Republic of Nigeria' 'Sub-Saharan Africa'
  'Lower middle income']]
[['MNP' 'Northern Mariana Islands'
  'Commonwealth of the Northern Mariana Islands' 'East Asia & Pacific'
  'High income: nonOECD']]
[['NOR' 'Norway' 'Kingdom of Norway' 'Europe & Central Asia'
  'High income: OECD']]
[['OMN' 'Oman' 'Sultanate of Oman' 'Middle East & North Africa'
  'High income: nonOECD']]
[['PAK' 'Pakistan' 'Islamic Republic of Pakistan' 'South Asia'
  'Lower middle income']]
[['PLW' 'Palau' 'Republic of Palau' 'East Asia & Pacific'
  'Upper middle income']]
[['PAN' 'Panama' 'Republic of Panama' 'Latin America & Caribbean'
  'Upper middle income']]
[['PNG' 'Papua New Guinea' 'The Independent State of Papua New Guinea'
  'East Asia & Pacific' 'Lower middle income']]
[['PRY' 'Paraguay' 'Republic of Paraguay' 'Latin America & Caribbean'
  'Lower middle income']]
[['PER' 'Peru' 'Republic of Peru' 'Latin America & Caribbean'
  'Upper middle income']]
[['PHL' 'Philippines' 'Republic of the Philippines' 'East Asia & Pacific'
  'Lower middle income']]
[['POL' 'Poland' 'Republic of Poland' 'Europe & Central Asia'
  'High income: OECD']]
[['PRT' 'Portugal' 'Portuguese Republic' 'Europe & Central Asia'
  'High income: OECD']]
[['PRI' 'Puerto Rico' 'Puerto Rico' 'Latin America & Caribbean'
  'High income: nonOECD']]
[['QAT' 'Qatar' 'State of Qatar' 'Middle East & North Africa'
  'High income: nonOECD']]
[['ROU' 'Romania' 'Romania' 'Europe & Central Asia' 'Upper middle income']]
[['RUS' 'Russian Federation' 'Russian Federation' 'Europe & Central Asia'
  'High income: nonOECD']]
[['RWA' 'Rwanda' 'Republic of Rwanda' 'Sub-Saharan Africa' 'Low income']]
[['WSM' 'Samoa' 'Samoa' 'East Asia & Pacific' 'Lower middle income']]
[['SMR' 'San Marino' 'Republic of San Marino' 'Europe & Central Asia'
  'High income: nonOECD']]
[['STP' 'São Tomé and Principe'
```

```
  'Democratic Republic of São Tomé and Principe' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['SAU' 'Saudi Arabia' 'Kingdom of Saudi Arabia'
  'Middle East & North Africa' 'High income: nonOECD']]
 [['SEN' 'Senegal' 'Republic of Senegal' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['SRB' 'Serbia' 'Republic of Serbia' 'Europe & Central Asia'
  'Upper middle income']]
 [['SYC' 'Seychelles' 'Republic of Seychelles' 'Sub-Saharan Africa'
  'Upper middle income']]
 [['SLE' 'Sierra Leone' 'Republic of Sierra Leone' 'Sub-Saharan Africa'
  'Low income']]
 [['SGP' 'Singapore' 'Republic of Singapore' 'East Asia & Pacific'
  'High income: nonOECD']]
 [['SXM' 'Sint Maarten (Dutch part)' 'Sint Maarten (Dutch part)'
  'Latin America & Caribbean' 'High income: nonOECD']]
 [['SVK' 'Slovak Republic' 'Slovak Republic' 'Europe & Central Asia'
  'High income: OECD']]
 [['SVN' 'Slovenia' 'Republic of Slovenia' 'Europe & Central Asia'
  'High income: OECD']]
 [['SLB' 'Solomon Islands' 'Solomon Islands' 'East Asia & Pacific'
  'Lower middle income']]
 [['SOM' 'Somalia' 'Somali Democratic Republic' 'Sub-Saharan Africa'
  'Low income']]
 [['ZAF' 'South Africa' 'Republic of South Africa' 'Sub-Saharan Africa'
  'Upper middle income']]
 [['SSD' 'South Sudan' 'Republic of South Sudan' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['ESP' 'Spain' 'Kingdom of Spain' 'Europe & Central Asia'
  'High income: OECD']]
 [['LKA' 'Sri Lanka' 'Democratic Socialist Republic of Sri Lanka'
  'South Asia' 'Lower middle income']]
 [['KNA' 'St. Kitts and Nevis' 'St. Kitts and Nevis'
  'Latin America & Caribbean' 'High income: nonOECD']]
 [['LCA' 'St. Lucia' 'St. Lucia' 'Latin America & Caribbean'
  'Upper middle income']]
 [['MAF' 'St. Martin (French part)' 'St. Martin (French part)'
  'Latin America & Caribbean' 'High income: nonOECD']]
 [['VCT' 'St. Vincent and the Grenadines' 'St. Vincent and the Grenadines'
  'Latin America & Caribbean' 'Upper middle income']]
 [['SDN' 'Sudan' 'Republic of the Sudan' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['SUR' 'Suriname' 'Republic of Suriname' 'Latin America & Caribbean'
  'Upper middle income']]
 [['SWZ' 'Swaziland' 'Kingdom of Swaziland' 'Sub-Saharan Africa'
  'Lower middle income']]
 [['SWE' 'Sweden' 'Kingdom of Sweden' 'Europe & Central Asia'
  'High income: OECD']]
 [['CHE' 'Switzerland' 'Switzerland' 'Europe & Central Asia'
  'High income: OECD']]
 [['SYR' 'Syrian Arab Republic' 'Syrian Arab Republic'
  'Middle East & North Africa' 'Lower middle income']]
 [['TJK' 'Tajikistan' 'Republic of Tajikistan' 'Europe & Central Asia'
  'Low income']]
 [['TZA' 'Tanzania' 'United Republic of Tanzania' 'Sub-Saharan Africa'
  'Low income']]
 [['THA' 'Thailand' 'Kingdom of Thailand' 'East Asia & Pacific'
  'Upper middle income']]
 [['TLS' 'Timor-Leste' 'Democratic Republic of Timor-Leste'
  'East Asia & Pacific' 'Lower middle income']]
 [['TGO' 'Togo' 'Republic of Togo' 'Sub-Saharan Africa' 'Low income']]
 [['TON' 'Tonga' 'Kingdom of Tonga' 'East Asia & Pacific'
  'Upper middle income']]
 [['TTO' 'Trinidad and Tobago' 'Republic of Trinidad and Tobago'
  'Latin America & Caribbean' 'High income: nonOECD']]
 [['TUN' 'Tunisia' 'Republic of Tunisia' 'Middle East & North Africa'
  'Upper middle income']]
 [['TUR' 'Turkey' 'Republic of Turkey' 'Europe & Central Asia'
  'Upper middle income']]
```

```
[['TKM' 'Turkmenistan' 'Turkmenistan' 'Europe & Central Asia'
  'Upper middle income']]
[['TCA' 'Turks and Caicos Islands' 'Turks and Caicos Islands'
  'Latin America & Caribbean' 'High income: nonOECD']]
[['TUV' 'Tuvalu' 'Tuvalu' 'East Asia & Pacific' 'Upper middle income']]
[['UGA' 'Uganda' 'Republic of Uganda' 'Sub-Saharan Africa' 'Low income']]
[['UKR' 'Ukraine' 'Ukraine' 'Europe & Central Asia' 'Lower middle income']]
[['ARE' 'United Arab Emirates' 'United Arab Emirates'
  'Middle East & North Africa' 'High income: nonOECD']]
[['GBR' 'United Kingdom'
  'United Kingdom of Great Britain and Northern Ireland'
  'Europe & Central Asia' 'High income: OECD']]
[['USA' 'United States' 'United States of America' 'North America'
  'High income: OECD']]
[['URY' 'Uruguay' 'Oriental Republic of Uruguay'
  'Latin America & Caribbean' 'High income: nonOECD']]
[['UZB' 'Uzbekistan' 'Republic of Uzbekistan' 'Europe & Central Asia'
  'Lower middle income']]
[['VUT' 'Vanuatu' 'Republic of Vanuatu' 'East Asia & Pacific'
  'Lower middle income']]
[['VEN' 'Venezuela, RB' 'República Bolivariana de Venezuela'
  'Latin America & Caribbean' 'Upper middle income']]
[['VNM' 'Vietnam' 'Socialist Republic of Vietnam' 'East Asia & Pacific'
  'Lower middle income']]
[['VIR' 'Virgin Islands (U.S.)' 'Virgin Islands of the United States'
  'Latin America & Caribbean' 'High income: nonOECD']]
[['PSE' 'West Bank and Gaza' 'West Bank and Gaza'
  'Middle East & North Africa' 'Lower middle income']]
[['YEM' 'Yemen, Rep.' 'Republic of Yemen' 'Middle East & North Africa'
  'Lower middle income']]
[['ZMB' 'Zambia' 'Republic of Zambia' 'Sub-Saharan Africa'
  'Lower middle income']]
[['ZWE' 'Zimbabwe' 'Republic of Zimbabwe' 'Sub-Saharan Africa'
  'Low income']]
```

In [13]:

```
COUNTRY_DATA[COUNTRY_DATA['Country Code'] == 'ARB']['Special Notes'].values
```

Out[13]:

```
array(['Arab World aggregate. Arab World is composed of members of the League of Ara
b States.'],
      dtype=object)
```

In [14]:

```
print(np.where(ALL_COUNTRIES == 'WLD'))
print(ALL_COUNTRIES[:25])
print(ALL_COUNTRIES[25:])

print(COUNTRY_DATA['Region'].unique())

droplist = ['ARB','EAP','ECA','EMU','EUU','HPC','HIC','LAC','LDC','LMY','LIC','LMC',

def drop_elements(elementarray, droplist):
    return np.array([el for el in elementarray if el not in droplist])

ALL_COUNTRIES = drop_elements(ALL_COUNTRIES, droplist)
REGIONS = ALL_COUNTRIES[:7]
COUNTRIES = ALL_COUNTRIES[7:]
#reduce the Data Series
DATA = RAW_DATA[RAW_DATA['Country Code'].isin(ALL_COUNTRIES)]
```

```
(array([24], dtype=int64),)
['ARB' 'EAS' 'EAP' 'EMU' 'ECS' 'ECA' 'EUU' 'HPC' 'HIC' 'LCN' 'LAC' 'LDC'
 'LMY' 'LIC' 'LMC' 'MEA' 'MNA' 'MIC' 'NAC' 'OED' 'SAS' 'SSF' 'SSA' 'UMC'
 'WLD']
['AFG' 'ALB' 'DZA' 'ASM' 'AND' 'AGO' 'ATG' 'ARG' 'ARM' 'ABW' 'AUS' 'AUT'
 'AZE' 'BHS' 'BHR' 'BGD' 'BRB' 'BLR' 'BEL' 'BLZ' 'BEN' 'BMU' 'BTN' 'BOL'
 'BIH' 'BWA' 'BRA' 'VGB' 'BRN' 'BGR' 'BFA' 'BDI' 'CPV' 'KHM' 'CMR' 'CAN'
 'CYM' 'CAF' 'TCD' 'CHI' 'CHL' 'CHN' 'COL' 'COM' 'COD' 'COG' 'CRI' 'CIV'
```

```
 'HRV' 'CUB' 'CUW' 'CYP' 'CZE' 'DNK' 'DJI' 'DMA' 'DOM' 'ECU' 'EGY' 'SLV'
 'GNQ' 'ERI' 'EST' 'ETH' 'FRO' 'FJI' 'FIN' 'FRA' 'PYF' 'GAB' 'GMB' 'GEO'
 'DEU' 'GHA' 'GIB' 'GRC' 'GRL' 'GRD' 'GUM' 'GTM' 'GIN' 'GNB' 'GUY' 'HTI'
 'HND' 'HKG' 'HUN' 'ISL' 'IND' 'IDN' 'IRN' 'IRQ' 'IRL' 'IMN' 'ISR' 'ITA'
 'JAM' 'JPN' 'JOR' 'KAZ' 'KEN' 'KIR' 'PRK' 'KOR' 'XKX' 'KWT' 'KGZ' 'LAO'
 'LVA' 'LBN' 'LSO' 'LBR' 'LBY' 'LIE' 'LTU' 'LUX' 'MAC' 'MKD' 'MDG' 'MWI'
 'MYS' 'MDV' 'MLI' 'MLT' 'MHL' 'MRT' 'MUS' 'MEX' 'FSM' 'MDA' 'MCO' 'MNG'
 'MNE' 'MAR' 'MOZ' 'MMR' 'NAM' 'NRU' 'NPL' 'NLD' 'NCL' 'NZL' 'NIC' 'NER'
 'NGA' 'MNP' 'NOR' 'OMN' 'PAK' 'PLW' 'PAN' 'PNG' 'PRY' 'PER' 'PHL' 'POL'
 'PRT' 'PRI' 'QAT' 'ROU' 'RUS' 'RWA' 'WSM' 'SMR' 'STP' 'SAU' 'SEN' 'SRB'
 'SYC' 'SLE' 'SGP' 'SXM' 'SVK' 'SVN' 'SLB' 'SOM' 'ZAF' 'SSD' 'ESP' 'LKA'
 'KNA' 'LCA' 'MAF' 'VCT' 'SDN' 'SUR' 'SWZ' 'SWE' 'CHE' 'SYR' 'TJK' 'TZA'
 'THA' 'TLS' 'TGO' 'TON' 'TTO' 'TUN' 'TUR' 'TKM' 'TCA' 'TUV' 'UGA' 'UKR'
 'ARE' 'GBR' 'USA' 'URY' 'UZB' 'VUT' 'VEN' 'VNM' 'VIR' 'PSE' 'YEM' 'ZMB'
 'ZWE']
['Latin America & Caribbean' 'South Asia' 'Sub-Saharan Africa'
 'Europe & Central Asia' nan 'Middle East & North Africa'
 'East Asia & Pacific' 'North America']
```

Résumé : pays et régions

- Il y a 25 régions / groupements de pays dont 18 sont des groupement économiques ou politiques
- **Garde 7 régions** et **217 pays** = **224 pays et régions**

3.2 Les années

- réduire les nombre des années à une plage pertinente : 2000 à 2050.
  - les dernières 20 années et les futures 30 années pour voir l'évolution.
- diviser les années and deux groups : dates passées et dates futures.
  - La base de données était mise à jour en 2020.

In [15]:

```
ALL_YEARS = np.array([y for y in ALL_YEARS if int(y) >= 2000 and int(y) <= 2050])

l = len(RAW_DATA.index.values)
for y in ALL_YEARS:
    d = RAW_DATA.loc[:,y].count()
    print(f"For {y}, there are {d} entries. {round(100*d/l,1)}% of the rows are fill

YEARS = np.array([y for y in ALL_YEARS if int(y) <= 2017])
YEARS_PROJ = np.array([y for y in ALL_YEARS if int(y) >= 2020])

print(YEARS)
print(YEARS_PROJ)
```

```
For 2000, there are 176676 entries. 19.9% of the rows are filled.
For 2001, there are 123509 entries. 13.9% of the rows are filled.
For 2002, there are 124205 entries. 14.0% of the rows are filled.
For 2003, there are 130363 entries. 14.7% of the rows are filled.
For 2004, there are 128814 entries. 14.5% of the rows are filled.
For 2005, there are 184108 entries. 20.8% of the rows are filled.
For 2006, there are 140312 entries. 15.8% of the rows are filled.
For 2007, there are 137272 entries. 15.5% of the rows are filled.
For 2008, there are 134387 entries. 15.2% of the rows are filled.
For 2009, there are 142108 entries. 16.0% of the rows are filled.
For 2010, there are 242442 entries. 27.3% of the rows are filled.
For 2011, there are 146012 entries. 16.5% of the rows are filled.
For 2012, there are 147264 entries. 16.6% of the rows are filled.
For 2013, there are 137509 entries. 15.5% of the rows are filled.
For 2014, there are 113789 entries. 12.8% of the rows are filled.
For 2015, there are 131058 entries. 14.8% of the rows are filled.
For 2016, there are 16460 entries. 1.9% of the rows are filled.
For 2017, there are 143 entries. 0.0% of the rows are filled.
```

```
For 2020, there are 51436 entries. 5.8% of the rows are filled.
For 2025, there are 51436 entries. 5.8% of the rows are filled.
For 2030, there are 51436 entries. 5.8% of the rows are filled.
For 2035, there are 51436 entries. 5.8% of the rows are filled.
For 2040, there are 51436 entries. 5.8% of the rows are filled.
For 2045, there are 51436 entries. 5.8% of the rows are filled.
For 2050, there are 51436 entries. 5.8% of the rows are filled.
['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']
```

In [16]:

```python
#reduce the years
print(DATA.columns.values[0:3])
print(ALL_YEARS)
DATA = DATA.filter(list(DATA.columns.values[0:4]) + list(ALL_YEARS))
DATA.info()
```

```
['Country Name' 'Country Code' 'Indicator Name']
['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']
<class 'pandas.core.frame.DataFrame'>
Int64Index: 820960 entries, 3665 to 886929
Data columns (total 29 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Country Name    820960 non-null  object
 1   Country Code    820960 non-null  object
 2   Indicator Name  820960 non-null  object
 3   Indicator Code  820960 non-null  object
 4   2000            170693 non-null  float64
 5   2001            117498 non-null  float64
 6   2002            118168 non-null  float64
 7   2003            124321 non-null  float64
 8   2004            122758 non-null  float64
 9   2005            177999 non-null  float64
 10  2006            134198 non-null  float64
 11  2007            131149 non-null  float64
 12  2008            128240 non-null  float64
 13  2009            135933 non-null  float64
 14  2010            235967 non-null  float64
 15  2011            139830 non-null  float64
 16  2012            140898 non-null  float64
 17  2013            131149 non-null  float64
 18  2014            107911 non-null  float64
 19  2015            129518 non-null  float64
 20  2016            15955 non-null   float64
 21  2017            143 non-null     float64
 22  2020            51128 non-null   float64
 23  2025            51128 non-null   float64
 24  2030            51128 non-null   float64
 25  2035            51128 non-null   float64
 26  2040            51128 non-null   float64
 27  2045            51128 non-null   float64
 28  2050            51128 non-null   float64
dtypes: float64(25), object(4)
memory usage: 187.9+ MB
```

Résumé : années

- Les nombre des entrées pour les années 2020 à 2050 sont égal. Ce sont des projections qui sont fait pour des intervalles de 5 ans.
- Il semble que **les données les plus actuelles viennent de 2017**. Après, ce sont de projections.

3.3 Les variables

- regarder les déscriptions des variables pour voir si elles sont pertinentes.
- choisir les variables pertinents:
  - infrastructure : internet, usage ordinateur, ...
  - cible : niveau education = BAC+ / universitaire
  - marché : nombre des potentiels clients, moyennes financières, ...
- utiliser 'EdstatsSeries.csv' / 'Series Data' / SERIES_DATA

In [17]:

```
ALL_INDICATORS_SERIES = SERIES_DATA['Series Code'].unique()
print("Indicators in Series Data :\t", len(ALL_INDICATORS_SERIES))
print("Indicators in the Data Series :\t", len(ALL_INDICATORS))
```

```
Indicators in Series Data :      3665
Indicators in the Data Series :  3665
```

In [18]:

```
#Verify that the 'Indicator Code' and the 'Series Code' are the indeed the same in t
#even if they have the same number of unique elements.

print("Set difference of the indicators in the Data Series and Series Data:\n\n",set
print()
print("Set difference of the indicators in the Series Data and Data Series:\n\n",set
print()

ALL_INDICATORS = np.array([i.upper().strip() for i in ALL_INDICATORS])
ALL_INDICATORS_SERIES = np.array([i.upper().strip() for i in ALL_INDICATORS_SERIES])

print("Set difference of the indicators in the Data Series and Series Data:\n\n",set
print()
print("Set difference of the indicators in the Series Data and Data Series:\n\n",set
print()
```

```
Set difference of the indicators in the Data Series and Series Data:

 {'UIS.LP.AG15T99', 'UIS.THAGE.3.A.GPV', 'UIS.THAGE.0', 'UIS.LP.AG65.M', 'UIS.SLE.1T
6.GPI', 'UIS.LP.AG15T99.F', 'UIS.AIR.1.GLAST.GPI', 'UIS.E.4.PU', 'UIS.XGDP.FSGOV.FDI
NSTADM.FFD', 'UIS.NIR.1.AGM1.F', 'UIS.NIR.1.AGP1.M', 'UIS.NIR.1.AGP1.F', 'UIS.E.3.P
U.F', 'UIS.LPP.AG65', 'UIS.LP.AG15T24', 'UIS.E.23.PU.F', 'UIS.LP.AG15T24.M', 'UIS.E.
1.GUK.F', 'UIS.XPUBP.0', 'UIS.LR.AG15T99.GPI', 'UIS.LR.AG65.F', 'UIS.NIR.1.AGP1', 'U
IS.NIR.1.AGM1.M', 'UIS.LPP.AG15T99', 'UIS.E.2.PU.F', 'UIS.NIR.1.AGM1', 'UIS.XGDP.0.F
SGOV.FDINSTADM.FFD', 'UIS.E.1.GUK', 'UIS.OE.56.40510', 'UIS.THDUR.0', 'UIS.XSPENDP.5
6.FDPUB.FNCAP', 'UIS.E.0.PU.F', 'UIS.XPUBP.4', 'UIS.FOSEP.56.FUK.F', 'UIS.TRANRA.23.
GPV.GPI', 'UIS.E.1.PU.F', 'UIS.GER.1T6.M', 'UIS.THAGE.4.A.GPV', 'UIS.XGDP.1.FSGOV.FD
INSTADM.FFD', 'UIS.GER.1T6.GPI', 'UIS.GER.1T6.F', 'UIS.E.1.PU', 'UIS.XSPENDP.56.FDPU
B.FNNONS', 'UIS.E.4.PU.F', 'UIS.LR.AG65.M', 'UIS.E.23.PU', 'UIS.LP.AG65', 'UIS.SR.1.
GLAST.GPI', 'UIS.LP.AG15T24.F', 'SE.SEC.DURS.LO', 'UIS.CEAGE.1', 'SE.SEC.ENRR.UP.F
E', 'UIS.THDUR.4.A.GPV'}

Set difference of the indicators in the Series Data and Data Series:

 {'UIS.LP.Ag15t99.F', 'UIS.CEAge.1', 'UIS.LP.Ag15t24', 'UIS.GER.1t6.GPI', 'UIS.XPub
P.0', 'UIS.XSpendP.56.Fdpub.Fncap', 'UIS.OE.56.40510 ', 'UIS.LR.Ag15t99.GPI', 'UIS.t
hAge.4.A.GPV', 'UIS.LP.Ag65.M', 'UIS.NIR.1.AgP1', 'UIS.NIR.1.AgP1.M', 'UIS.E.4.Pu.
F', 'UIS.XPubP.4', 'UIS.E.1.Guk.F', 'UIS.LPP.Ag65', 'UIS.E.23.Pu', 'UIS.NIR.1.AgP1.
F', 'UIS.AIR.1.Glast.GPI', 'UIS.GER.1t6.F', 'UIS.thDur.0', 'UIS.E.1.Pu', 'UIS.FOSEP.
56.Fuk.F', 'UIS.LR.Ag65.F', 'UIS.thDur.4.A.GPV', 'UIS.LPP.Ag15t99', 'UIS.LP.Ag15t24.
F', 'SE.SEC.DURS.LO ', 'UIS.NIR.1.AgM1.F', 'UIS.TranRA.23.GPV.GPI', 'UIS.LP.Ag65',
'UIS.E.23.Pu.F', 'SE.SEC.ENRR.UP.FE ', 'UIS.XGDP.1.FSgov.FDinstadm.FFd', 'UIS.E.3.P
u.F', 'UIS.E.1.Guk', 'UIS.E.0.Pu.F', 'UIS.E.4.Pu', 'UIS.XGDP.FSgov.FDinstadm.FFd',
'UIS.LP.Ag15t99', 'UIS.NIR.1.AgM1.M', 'UIS.thAge.0', 'UIS.XGDP.0.FSgov.FDinstadm.FF
d', 'UIS.E.2.Pu.F', 'UIS.SR.1.Glast.GPI', 'UIS.XSpendP.56.Fdpub.Fnnons', 'UIS.SLE.1t
6.GPI', 'UIS.LR.Ag65.M', 'UIS.E.1.Pu.F', 'UIS.NIR.1.AgM1', 'UIS.GER.1t6.M', 'UIS.thA
ge.3.A.GPV', 'UIS.LP.Ag15t24.M'}

Set difference of the indicators in the Data Series and Series Data:
```

```
 set()
```

Set difference of the indicators in the Series Data and Data Series:

```
 set()
```

In [19]:

```
#look more closely at the topics and choose the indicators that are relevant.
#Have a look at the topics to see which might be potentially interesting
TOPICS = SERIES_DATA['Topic'].unique()
print(f"There are {len(TOPICS)} different topics:\n")
data = SERIES_DATA['Topic']
print(data.value_counts())
```

```
There are 37 different topics:

Learning Outcomes
1046
Attainment
733
Education Equality
426
Secondary
256
Primary
248
Population
213
Tertiary
158
Teachers
137
Expenditures
93
Engaging the Private Sector (SABER)
51
Literacy
32
Teachers (SABER)
29
School Autonomy and Accountability (SABER)
29
Early Childhood Education
24
Education Management Information Systems (SABER)
23
Post-Secondary/Non-Tertiary
19
School Finance (SABER)
18
Tertiary Education (SABER)
17
Student Assessment (SABER)
15
Pre-Primary
14
Health: Population: Structure
13
Workforce Development (SABER)
12
Early Child Development (SABER)
12
Social Protection & Labor: Labor force structure
11
School Health and School Feeding (SABER)
9
Economic Policy & Debt: Purchasing power parity
```

```
6
Laber
3
Social Protection & Labor: Unemployment
3
Economic Policy & Debt: National accounts: US$ at current prices: Aggregate indicato
rs             3
Economic Policy & Debt: National accounts: US$ at constant 2010 prices: Aggregate in
dicators       2
Background
2
Infrastructure: Communications
2
EMIS
2
Economic Policy & Debt: National accounts: Atlas GNI & GNI per capita
1
Health: Risk factors
1
Health: Mortality
1
Health: Population: Dynamics
1
Name: Topic, dtype: int64
```

Résumé: 'Indicator Code'

- Il y a des erreurs de mise en forme entre les deux fichiers 'EdstatsSeries.csv' ('Series Data' / SERIES_DATA) et 'EdstatsData.csv' ('Data Series' / RAW_DATA).
  - miniscule/majoscule et des espaces qui traînnent.
- Les 'Indicator Code' dans le Data Series et équivalent au 'Series Code' dans le Series Data.
- Il y **3665 unique indicateurs et 37 thématiques**.

In [20]:

```
#Look at little more closely at the indicators that are provided - especially under

interesting_topics = ['Attainment']
interesting_topics = ['Pre-Primary','Primary', 'Early Childhood Education']
interesting_topics = ['Secondary', 'Tertiary', 'Post-Secondary/Non-Tertiary']
#interesting_topics = ['Infrastructure: Communications', 'Expenditures']
#interesting_topics = ['EMIS','Education Management Information Systems (SABER)','Te
#interesting_topics = ['Learning Outcomes']
#interesting_topics = ['Teachers']
#interesting_topics = ['Population','Health: Population: Structure', 'Health: Popula
#interesting_topics = ['Education Equality']
#interesting_topics = ['Expenditures']
#interesting_topics = ['Laber','Background']
#interesting_topics = []
#interesting_topics = ['Economic Policy & Debt: Purchasing power parity',
#                      'Economic Policy & Debt: National accounts: US$ at current pr
#                      'Economic Policy & Debt: National accounts: US$ at constant 2
#                      'Economic Policy & Debt: National accounts: Atlas GNI & GNI p
interesting_topics = ['Secondary', 'Tertiary']

for topic in interesting_topics:
    print(topic)
    print(50*'-')
    data = SERIES_DATA[SERIES_DATA['Topic'] == topic].loc[:,['Series Code','Indicato
    for name in data.values:
        print(name[0], name[1],sep="\t--")
    print(50*'_')
```

```
Secondary
--------------------------------------------------
SE.ENR.PRSC.FM.ZS       --Gross enrolment ratio, primary and secondary, gender parit
```

```
y index (GPI)
SE.ENR.SECO.FM.ZS       --Gross enrolment ratio, secondary, gender parity index (GP
I)
SE.SEC.AGES     --Official entrance age to lower secondary education (years)
SE.SEC.CMPT.LO.FE.ZS    --Lower secondary completion rate, female (%)
SE.SEC.CMPT.LO.MA.ZS    --Lower secondary completion rate, male (%)
SE.SEC.CMPT.LO.ZS       --Lower secondary completion rate, both sexes (%)
SE.SEC.DURS     --Theoretical duration of secondary education (years)
SE.SEC.DURS.LO  --Theoretical duration of lower secondary education (years)
SE.SEC.DURS.UP  --Theoretical duration of upper secondary education (years)
SE.SEC.ENRL     --Enrolment in secondary education, both sexes (number)
SE.SEC.ENRL.FE  --Enrolment in secondary education, female (number)
SE.SEC.ENRL.FE.VO.ZS    --Percentage of female students in secondary education enrol
led in vocational programmes, female (%)
SE.SEC.ENRL.FE.ZS       --Percentage of students in secondary education who are fema
le (%)
SE.SEC.ENRL.GC  --Enrolment in secondary general, both sexes (number)
SE.SEC.ENRL.GC.FE       --Enrolment in secondary general, female (number)
SE.SEC.ENRL.GC.FE.ZS    --Percentage of students in secondary general education who
are female (%)
SE.SEC.ENRL.MA.VO.ZS    --Percentage of male students in secondary education enrolle
d in vocational programmes, male (%)
SE.SEC.ENRL.VO  --Enrolment in secondary vocational, both sexes (number)
SE.SEC.ENRL.VO.FE       --Enrolment in secondary vocational, female (number)
SE.SEC.ENRL.VO.FE.ZS    --Percentage of students in secondary vocational education w
ho are female (%)
SE.SEC.ENRL.VO.ZS       --Percentage of students in secondary education enrolled in
vocational programmes, both sexes (%)
SE.SEC.ENRR     --Gross enrolment ratio, secondary, both sexes (%)
SE.SEC.ENRR.FE  --Gross enrolment ratio, secondary, female (%)
SE.SEC.ENRR.LO  --Gross enrolment ratio, lower secondary, both sexes (%)
SE.SEC.ENRR.LO.FE       --Gross enrolment ratio, lower secondary, female (%)
SE.SEC.ENRR.LO.MA       --Gross enrolment ratio, lower secondary, male (%)
SE.SEC.ENRR.MA  --Gross enrolment ratio, secondary, male (%)
SE.SEC.ENRR.UP  --Gross enrolment ratio, upper secondary, both sexes (%)
SE.SEC.ENRR.UP.FE       --Gross enrolment ratio, upper secondary, female (%)
SE.SEC.ENRR.UP.MA       --Gross enrolment ratio, upper secondary, male (%)
SE.SEC.NENR     --Net enrolment rate, secondary, both sexes (%)
SE.SEC.NENR.FE  --Net enrolment rate, secondary, female (%)
SE.SEC.NENR.MA  --Net enrolment rate, secondary, male (%)
SE.SEC.PRIV.ZS  --Percentage of enrolment in secondary education in private institut
ions (%)
SE.SEC.PROG.FE.ZS       --Effective transition rate from primary to lower secondary
general education, female (%)
SE.SEC.PROG.MA.ZS       --Effective transition rate from primary to lower secondary
general education, male (%)
SE.SEC.PROG.ZS  --Effective transition rate from primary to lower secondary general
education, both sexes (%)
UIS.AFR.SCHBSP.2.PU.MIXTOIL     --Africa Dataset: Percentage of lower secondary scho
ols with mixed-sex toilets (%)
UIS.AFR.SCHBSP.2.PU.WELEC       --Africa Dataset: Percentage of lower secondary scho
ols with access to electricity (%)
UIS.AFR.SCHBSP.2.PU.WNIELEC     --Africa Dataset: Percentage of lower secondary scho
ols with no information on electricity (%)
UIS.AFR.SCHBSP.2.PU.WNIPOWAT    --Africa Dataset: Percentage of lower secondary scho
ols with no information on potable water (%)
UIS.AFR.SCHBSP.2.PU.WNITOIL     --Africa Dataset: Percentage of lower secondary scho
ols with no information on toilets (%)
UIS.AFR.SCHBSP.2.PU.WOELEC      --Africa Dataset: Percentage of lower secondary scho
ols without access to electricity (%)
UIS.AFR.SCHBSP.2.PU.WOPOWAT     --Africa Dataset: Percentage of lower secondary scho
ols without access to potable water (%)
UIS.AFR.SCHBSP.2.PU.WOTOIL      --Africa Dataset: Percentage of lower secondary scho
ols without toilets (%)
UIS.AFR.SCHBSP.2.PU.WPOWAT      --Africa Dataset: Percentage of lower secondary scho
ols with access to potable water (%)
UIS.AFR.SCHBSP.2.PU.WSTOIL      --Africa Dataset: Percentage of lower secondary scho
ols with single-sex toilets (%)
UIS.AFR.SCHBSP.2.PU.WTOIL       --Africa Dataset: Percentage of lower secondary scho
```

```
ols with toilets (%)
UIS.AIR.2.GPV   --Gross intake ratio to Grade 1 of lower secondary general educatio
n, both sexes (%)
UIS.AIR.2.GPV.F --Gross intake ratio to Grade 1 of lower secondary general educatio
n, female (%)
UIS.AIR.2.GPV.GLAST.GPI --Lower secondary completion rate, gender parity index (GPI)
UIS.AIR.2.GPV.GPI       --Gross intake ratio to Grade 1 of lower secondary general e
ducation, gender parity index (GPI)
UIS.AIR.2.GPV.M --Gross intake ratio to Grade 1 of lower secondary general educatio
n, male (%)
UIS.DR.2.GPV.F  --Cumulative drop-out rate to the last grade of lower secondary gene
ral education, female (%)
UIS.DR.2.GPV.G1.F       --Drop-out rate from Grade 1 of lower secondary general educ
ation, female (%)
UIS.DR.2.GPV.G1.M       --Drop-out rate from Grade 1 of lower secondary general educ
ation, male (%)
UIS.DR.2.GPV.G1.T       --Drop-out rate from Grade 1 of lower secondary general educ
ation, both sexes (%)
UIS.DR.2.GPV.G2.F       --Drop-out rate from Grade 2 of lower secondary general educ
ation, female (%)
UIS.DR.2.GPV.G2.M       --Drop-out rate from Grade 2 of lower secondary general educ
ation, male (%)
UIS.DR.2.GPV.G2.T       --Drop-out rate from Grade 2 of lower secondary general educ
ation, both sexes (%)
UIS.DR.2.GPV.G3.F       --Drop-out rate from Grade 3 of lower secondary general educ
ation, female (%)
UIS.DR.2.GPV.G3.M       --Drop-out rate from Grade 3 of lower secondary general educ
ation, male (%)
UIS.DR.2.GPV.G3.T       --Drop-out rate from Grade 3 of lower secondary general educ
ation, both sexes (%)
UIS.DR.2.GPV.G4.F       --Drop-out rate from Grade 4 of lower secondary general educ
ation, female (%)
UIS.DR.2.GPV.G4.M       --Drop-out rate from Grade 4 of lower secondary general educ
ation, male (%)
UIS.DR.2.GPV.G4.T       --Drop-out rate from Grade 4 of lower secondary general educ
ation, both sexes (%)
UIS.DR.2.GPV.G5.F       --Drop-out rate from Grade 5 of lower secondary general educ
ation, female (%)
UIS.DR.2.GPV.G5.M       --Drop-out rate from Grade 5 of lower secondary general educ
ation, male (%)
UIS.DR.2.GPV.G5.T       --Drop-out rate from Grade 5 of lower secondary general educ
ation, both sexes (%)
UIS.DR.2.GPV.M  --Cumulative drop-out rate to the last grade of lower secondary gene
ral education, male (%)
UIS.DR.2.GPV.T  --Cumulative drop-out rate to the last grade of lower secondary gene
ral education, both sexes (%)
UIS.E.2 --Enrolment in lower secondary education, both sexes (number)
UIS.E.2.F       --Enrolment in lower secondary education, female (number)
UIS.E.2.GPV     --Enrolment in lower secondary general, both sexes (number)
UIS.E.2.GPV.F   --Enrolment in lower secondary general, female (number)
UIS.E.2.GPV.G1.F        --Enrolment in Grade 1 of lower secondary general education,
female (number)
UIS.E.2.GPV.G1.T        --Enrolment in Grade 1 of lower secondary general education,
both sexes (number)
UIS.E.2.GPV.G2.F        --Enrolment in Grade 2 of lower secondary general education,
female (number)
UIS.E.2.GPV.G2.T        --Enrolment in Grade 2 of lower secondary general education,
both sexes (number)
UIS.E.2.GPV.G3.F        --Enrolment in Grade 3 of lower secondary general education,
female (number)
UIS.E.2.GPV.G3.T        --Enrolment in Grade 3 of lower secondary general education,
both sexes (number)
UIS.E.2.GPV.G4.F        --Enrolment in Grade 4 of lower secondary general education,
female (number)
UIS.E.2.GPV.G4.T        --Enrolment in Grade 4 of lower secondary general education,
both sexes (number)
UIS.E.2.GPV.G5.F        --Enrolment in Grade 5 of lower secondary general education,
female (number)
UIS.E.2.GPV.G5.T        --Enrolment in Grade 5 of lower secondary general education,
```

```
both sexes (number)
UIS.E.2.GPV.G6.F        --Enrolment in Grade 6 of lower secondary general education,
female (number)
UIS.E.2.GPV.G6.T        --Enrolment in Grade 6 of lower secondary general education,
both sexes (number)
UIS.E.2.GPV.GUK.F       --Enrolment in lower secondary general education, Grade unsp
ecified, female (number)
UIS.E.2.GPV.GUK.T       --Enrolment in lower secondary general education, Grade unsp
ecified, both sexes (number)
UIS.E.2.PR      --Enrolment in lower secondary education, private institutions, both
sexes (number)
UIS.E.2.PR.F    --Enrolment in lower secondary education, private institutions, fema
le (number)
UIS.E.2.PU      --Enrolment in lower secondary education, public institutions, both
sexes (number)
UIS.E.2.Pu.F    --Enrolment in lower secondary education, public institutions, femal
e (number)
UIS.E.2.V       --Enrolment in lower secondary vocational, both sexes (number)
UIS.E.2.V.F     --Enrolment in lower secondary vocational, female (number)
UIS.E.23.PR     --Enrolment in secondary education, private institutions, both sexes
(number)
UIS.E.23.PR.F   --Enrolment in secondary education, private institutions, female (nu
mber)
UIS.E.23.Pu     --Enrolment in secondary education, public institutions, both sexes
(number)
UIS.E.23.Pu.F   --Enrolment in secondary education, public institutions, female (num
ber)
UIS.E.3 --Enrolment in upper secondary education, both sexes (number)
UIS.E.3.F       --Enrolment in upper secondary education, female (number)
UIS.E.3.GPV     --Enrolment in upper secondary general, both sexes (number)
UIS.E.3.GPV.F   --Enrolment in upper secondary general, female (number)
UIS.E.3.PR      --Enrolment in upper secondary education, private institutions, both
sexes (number)
UIS.E.3.PR.F    --Enrolment in upper secondary education, private institutions, fema
le (number)
UIS.E.3.PU      --Enrolment in upper secondary education, public institutions, both
sexes (number)
UIS.E.3.Pu.F    --Enrolment in upper secondary education, public institutions, femal
e (number)
UIS.E.3.V       --Enrolment in upper secondary vocational, both sexes (number)
UIS.E.3.V.F     --Enrolment in upper secondary vocational, female (number)
UIS.FEP.2       --Percentage of students in lower secondary education who are female
(%)
UIS.FEP.2.GPV   --Percentage of students in lower secondary general education who ar
e female (%)
UIS.FEP.2.V     --Percentage of students in lower secondary vocational education who
are female (%)
UIS.FEP.3       --Percentage of students in upper secondary education who are female
(%)
UIS.FEP.3.GPV   --Percentage of students in upper secondary general education who ar
e female (%)
UIS.FEP.3.V     --Percentage of students in upper secondary vocational education who
are female (%)
UIS.FOFSTP.2    --Out-of-school adolescents of lower secondary school age, percentag
e female (%)
UIS.FOFSTP.3    --Out-of-school youth of upper secondary school age, percentage fema
le (%)
UIS.GER.12      --Gross enrolment ratio, primary and lower secondary, both sexes (%)
UIS.GER.12.F    --Gross enrolment ratio, primary and lower secondary, female (%)
UIS.GER.12.GPI  --Gross enrolment ratio, primary and lower secondary, gender parity
index (GPI)
UIS.GER.12.M    --Gross enrolment ratio, primary and lower secondary, male (%)
UIS.GER.123     --Gross enrolment ratio, primary and secondary, both sexes (%)
UIS.GER.123.F   --Gross enrolment ratio, primary and secondary, female (%)
UIS.GER.123.M   --Gross enrolment ratio, primary and secondary, male (%)
UIS.GER.2.GPI   --Gross enrolment ratio, lower secondary, gender parity index (GPI)
UIS.GER.3.GPI   --Gross enrolment ratio, upper secondary, gender parity index (GPI)
UIS.GGR.2       --Gross graduation ratio from lower secondary education, both sexes
(%)
```

```
UIS.GGR.2.F     --Gross graduation ratio from lower secondary education, female (%)
UIS.GGR.2.GPI   --Gross graduation ratio from lower secondary education, gender pari
ty index (GPI)
UIS.GGR.2.M     --Gross graduation ratio from lower secondary education, male (%)
UIS.GTVP.2.GPV  --Percentage of students in lower secondary education enrolled in ge
neral programmes, both sexes (%)
UIS.GTVP.2.GPV.F        --Percentage of female students in lower secondary education
enrolled in general programmes, female (%)
UIS.GTVP.2.GPV.M        --Percentage of male students in lower secondary education e
nrolled in general programmes, male (%)
UIS.GTVP.2.V    --Percentage of students in lower secondary education enrolled in vo
cational programmes, both sexes (%)
UIS.GTVP.2.V.F  --Percentage of female students in lower secondary education enrolle
d in vocational programmes, female (%)
UIS.GTVP.2.V.M  --Percentage of male students in lower secondary education enrolled
in vocational programmes, male (%)
UIS.GTVP.23.GPV --Percentage of students in secondary education enrolled in general
programmes, both sexes (%)
UIS.GTVP.23.GPV.F       --Percentage of female students in secondary education enrol
led in general programmes, female (%)
UIS.GTVP.23.GPV.M       --Percentage of male students in secondary education enrolle
d in general programmes, male (%)
UIS.GTVP.3.GPV  --Percentage of students in upper secondary education enrolled in ge
neral programmes, both sexes (%)
UIS.GTVP.3.GPV.F        --Percentage of female students in upper secondary education
enrolled in general programmes, female (%)
UIS.GTVP.3.GPV.M        --Percentage of male students in upper secondary education e
nrolled in general programmes, male (%)
UIS.GTVP.3.V    --Percentage of students in upper secondary education enrolled in vo
cational programmes, both sexes (%)
UIS.GTVP.3.V.F  --Percentage of female students in upper secondary education enrolle
d in vocational programmes, female (%)
UIS.GTVP.3.V.M  --Percentage of male students in upper secondary education enrolled
in vocational programmes, male (%)
UIS.NER.2       --Net enrolment rate, lower secondary, both sexes (%)
UIS.NER.2.F     --Net enrolment rate, lower secondary, female (%)
UIS.NER.2.M     --Net enrolment rate, lower secondary, male (%)
UIS.NER.23.GPI  --Net enrolment rate, secondary, gender parity index (GPI)
UIS.NER.3       --Net enrolment rate, upper secondary, both sexes (%)
UIS.NER.3.F     --Net enrolment rate, upper secondary, female (%)
UIS.NER.3.M     --Net enrolment rate, upper secondary, male (%)
UIS.NERA.2      --Adjusted net enrolment rate, lower secondary, both sexes (%)
UIS.NERA.2.F    --Adjusted net enrolment rate, lower secondary, female (%)
UIS.NERA.2.GPI  --Adjusted net enrolment rate, lower secondary, gender parity index
(GPI)
UIS.NERA.2.M    --Adjusted net enrolment rate, lower secondary, male (%)
UIS.NERA.3      --Adjusted net enrolment rate, upper secondary, both sexes (%)
UIS.NERA.3.F    --Adjusted net enrolment rate, upper secondary, female (%)
UIS.NERA.3.GPI  --Adjusted net enrolment rate, upper secondary, gender parity index
(GPI)
UIS.NERA.3.M    --Adjusted net enrolment rate, upper secondary, male (%)
UIS.NERT.2      --Total net enrolment rate, lower secondary, both sexes (%)
UIS.NERT.2.F    --Total net enrolment rate, lower secondary, female (%)
UIS.NERT.2.GPI  --Total net enrolment rate, lower secondary, gender parity index (GP
I)
UIS.NERT.2.M    --Total net enrolment rate, lower secondary, male (%)
UIS.OFST.2      --Out-of-school adolescents of lower secondary school age, both sexe
s (number)
UIS.OFST.2.F    --Out-of-school adolescents of lower secondary school age, female (n
umber)
UIS.OFST.2.M    --Out-of-school adolescents of lower secondary school age, male (num
ber)
UIS.OFST.3.CP   --Out-of-school youth of upper secondary school age, both sexes (num
ber)
UIS.OFST.3.F.CP --Out-of-school youth of upper secondary school age, female (number)
UIS.OFST.3.M.CP --Out-of-school youth of upper secondary school age, male (number)
UIS.PRP.2       --Percentage of enrolment in lower secondary education in private in
stitutions (%)
UIS.PRP.3       --Percentage of enrolment in upper secondary education in private in
```

```
stitutions (%)
UIS.R.2.GPV     --Repeaters in lower secondary general education, all grades, both s
exes (number)
UIS.R.2.GPV.F   --Repeaters in lower secondary general education, all grades, female
(number)
UIS.R.2.GPV.G1  --Repeaters in Grade 1 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G1.F        --Repeaters in Grade 1 of lower secondary general education,
female (number)
UIS.R.2.GPV.G2  --Repeaters in Grade 2 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G2.F        --Repeaters in Grade 2 of lower secondary general education,
female (number)
UIS.R.2.GPV.G3  --Repeaters in Grade 3 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G3.F        --Repeaters in Grade 3 of lower secondary general education,
female (number)
UIS.R.2.GPV.G4  --Repeaters in Grade 4 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G4.F        --Repeaters in Grade 4 of lower secondary general education,
female (number)
UIS.R.2.GPV.G5  --Repeaters in Grade 5 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G5.F        --Repeaters in Grade 5 of lower secondary general education,
female (number)
UIS.R.2.GPV.G6  --Repeaters in Grade 6 of lower secondary general education, both se
xes (number)
UIS.R.2.GPV.G6.F        --Repeaters in Grade 6 of lower secondary general education,
female (number)
UIS.REPP.2.GPV  --Percentage of repeaters in lower secondary general education, all
grades, both sexes (%)
UIS.REPP.2.GPV.F        --Percentage of repeaters in lower secondary general educati
on, all grades, female (%)
UIS.REPP.2.GPV.G1       --Percentage of repeaters in Grade 1 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G1.F     --Percentage of repeaters in Grade 1 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G1.M     --Percentage of repeaters in Grade 1 of lower secondary gene
ral education, male (%)
UIS.REPP.2.GPV.G2       --Percentage of repeaters in Grade 2 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G2.F     --Percentage of repeaters in Grade 2 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G2.M     --Percentage of repeaters in Grade 2 of lower secondary gene
ral education, male (%)
UIS.REPP.2.GPV.G3       --Percentage of repeaters in Grade 3 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G3.F     --Percentage of repeaters in Grade 3 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G3.M     --Percentage of repeaters in Grade 3 of lower secondary gene
ral education, male (%)
UIS.REPP.2.GPV.G4       --Percentage of repeaters in Grade 4 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G4.F     --Percentage of repeaters in Grade 4 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G4.M     --Percentage of repeaters in Grade 4 of lower secondary gene
ral education, male (%)
UIS.REPP.2.GPV.G5       --Percentage of repeaters in Grade 5 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G5.F     --Percentage of repeaters in Grade 5 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G5.M     --Percentage of repeaters in Grade 5 of lower secondary gene
ral education, male (%)
UIS.REPP.2.GPV.G6       --Percentage of repeaters in Grade 6 of lower secondary gene
ral education, both sexes (%)
UIS.REPP.2.GPV.G6.F     --Percentage of repeaters in Grade 6 of lower secondary gene
ral education, female (%)
UIS.REPP.2.GPV.G6.M     --Percentage of repeaters in Grade 6 of lower secondary gene
ral education, male (%)
```

```
UIS.REPP.2.GPV.GPI      --Percentage of repeaters in lower secondary education, all
grades, gender parity index (GPI)
UIS.REPP.2.GPV.M        --Percentage of repeaters in lower secondary general educati
on, all grades, male (%)
UIS.REPR.2.GPV  --Repetition rate in lower secondary general education (all grades),
both sexes (%)
UIS.REPR.2.GPV.F        --Repetition rate in lower secondary general education (all
grades), female (%)
UIS.REPR.2.GPV.G1       --Repetition rate in Grade 1 of lower secondary general educ
ation, both sexes (%)
UIS.REPR.2.GPV.G1.F     --Repetition rate in Grade 1 of lower secondary general educ
ation, female (%)
UIS.REPR.2.GPV.G1.M     --Repetition rate in Grade 1 of lower secondary general educ
ation, male (%)
UIS.REPR.2.GPV.G2       --Repetition rate in Grade 2 of lower secondary general educ
ation, both sexes (%)
UIS.REPR.2.GPV.G2.F     --Repetition rate in Grade 2 of lower secondary general educ
ation, female (%)
UIS.REPR.2.GPV.G2.M     --Repetition rate in Grade 2 of lower secondary general educ
ation, male (%)
UIS.REPR.2.GPV.G3       --Repetition rate in Grade 3 of lower secondary general educ
ation, both sexes (%)
UIS.REPR.2.GPV.G3.F     --Repetition rate in Grade 3 of lower secondary general educ
ation, female (%)
UIS.REPR.2.GPV.G3.M     --Repetition rate in Grade 3 of lower secondary general educ
ation, male (%)
UIS.REPR.2.GPV.G4       --Repetition rate in Grade 4 of lower secondary general educ
ation, both sexes (%)
UIS.REPR.2.GPV.G4.F     --Repetition rate in Grade 4 of lower secondary general educ
ation, female (%)
UIS.REPR.2.GPV.G4.M     --Repetition rate in Grade 4 of lower secondary general educ
ation, male (%)
UIS.REPR.2.GPV.G5       --Repetition rate in Grade 5 of lower secondary general educ
ation, both sexes (%)
UIS.REPR.2.GPV.G5.F     --Repetition rate in Grade 5 of lower secondary general educ
ation, female (%)
UIS.REPR.2.GPV.G5.M     --Repetition rate in Grade 5 of lower secondary general educ
ation, male (%)
UIS.REPR.2.GPV.M        --Repetition rate in lower secondary general education (all
grades), male (%)
UIS.ROFST.2     --Rate of out-of-school adolescents of lower secondary school age, b
oth sexes (%)
UIS.ROFST.2.F   --Rate of out-of-school adolescents of lower secondary school age, f
emale (%)
UIS.ROFST.2.M   --Rate of out-of-school adolescents of lower secondary school age, m
ale (%)
UIS.ROFST.3.CP  --Rate of out-of-school youth of upper secondary school age, both se
xes (%)
UIS.ROFST.3.F.CP        --Rate of out-of-school youth of upper secondary school age,
female (%)
UIS.ROFST.3.M.CP        --Rate of out-of-school youth of upper secondary school age,
male (%)
UIS.SLE.12      --School life expectancy, primary and lower secondary, both sexes (y
ears)
UIS.SLE.12.F    --School life expectancy, primary and lower secondary, female (year
s)
UIS.SLE.12.M    --School life expectancy, primary and lower secondary, male (years)
UIS.SLE.123     --School life expectancy, primary and secondary, both sexes (years)
UIS.SLE.123.F   --School life expectancy, primary and secondary, female (years)
UIS.SLE.123.GPI --School life expectancy, primary and secondary, gender parity index
(GPI)
UIS.SLE.123.M   --School life expectancy, primary and secondary, male (years)
UIS.SLE.23      --School life expectancy, secondary, both sexes (years)
UIS.SLE.23.F    --School life expectancy, secondary, female (years)
UIS.SLE.23.GPI  --School life expectancy, secondary, gender parity index (GPI)
UIS.SLE.23.M    --School life expectancy, secondary, male (years)
UIS.SLEN.12.F   --School life expectancy, primary and lower secondary (excluding rep
etition), female (years)
UIS.SLEN.12.GPI --School life expectancy, primary and lower secondary (excluding rep
```

```
etition), gender parity index (GPI)
UIS.SLEN.12.M   --School life expectancy, primary and lower secondary (excluding rep
etition), male (years)
UIS.SLEN.12.T   --School life expectancy, primary and lower secondary (excluding rep
etition), both sexes (years)
UIS.SR.2.GPV.GLAST.CP.F --Survival rate to the last grade of lower secondary general
education, female (%)
UIS.SR.2.GPV.GLAST.CP.M --Survival rate to the last grade of lower secondary general
education, male (%)
UIS.SR.2.GPV.GLAST.CP.T --Survival rate to the last grade of lower secondary general
education, both sexes (%)
UIS.SR.2.GPV.GLAST.GPI  --Survival rate to the last grade of lower secondary general
education, gender parity index (GPI)
UIS.thAge.3.A.GPV       --Official entrance age to upper secondary education (years)
UIS.TranRA.23.GPV.GPI   --Effective transition rate from primary to lower secondary
general education, gender parity index (GPI)
UIS.UAPP.23     --Under-age enrolment ratio in secondary education, both sexes (%)
UIS.UAPP.23.F   --Under-age enrolment ratio in secondary education, female (%)
UIS.UAPP.23.M   --Under-age enrolment ratio in secondary education, male (%)
____________________________________________________
Tertiary
----------------------------------------------------
SE.ENR.TERT.FM.ZS       --Gross enrolment ratio, tertiary, gender parity index (GPI)
SE.SCH.LIFE     --School life expectancy, primary to tertiary, both sexes (years)
SE.SCH.LIFE.FE  --School life expectancy, primary to tertiary, female (years)
SE.SCH.LIFE.MA  --School life expectancy, primary to tertiary, male (years)
SE.TER.CMPL.FE.ZS       --Gross graduation ratio from first degree programmes (ISCED
6 and 7) in tertiary education, female (%)
SE.TER.CMPL.MA.ZS       --Gross graduation ratio from first degree programmes (ISCED
6 and 7) in tertiary education, male (%)
SE.TER.CMPL.ZS  --Gross graduation ratio from first degree programmes (ISCED 6 and
7) in tertiary education, both sexes (%)
SE.TER.ENRL     --Enrolment in tertiary education, all programmes, both sexes (numbe
r)
SE.TER.ENRL.FE  --Enrolment in tertiary education, all programmes, female (number)
SE.TER.ENRL.FE.ZS       --Percentage of students in tertiary education who are femal
e (%)
SE.TER.ENRR     --Gross enrolment ratio, tertiary, both sexes (%)
SE.TER.ENRR.FE  --Gross enrolment ratio, tertiary, female (%)
SE.TER.ENRR.MA  --Gross enrolment ratio, tertiary, male (%)
SE.TER.GRAD     --Graduates from tertiary education, both sexes (number)
SE.TER.GRAD.AG.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Agriculture programmes, female (%)
SE.TER.GRAD.AG.ZS       --Percentage of graduates from tertiary education graduating
from Agriculture programmes, both sexes (%)
SE.TER.GRAD.ED.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Education programmes, female (%)
SE.TER.GRAD.ED.ZS       --Percentage of graduates from tertiary education graduating
from Education programmes, both sexes (%)
SE.TER.GRAD.EN.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Engineering, Manufacturing and Construction programmes, female (%)
SE.TER.GRAD.EN.ZS       --Percentage of graduates from tertiary education graduating
from Engineering, Manufacturing and Construction programmes, both sexes (%)
SE.TER.GRAD.FE  --Graduates from tertiary education, female (number)
SE.TER.GRAD.FE.AG.ZS    --Percentage of graduates from Agriculture programmes in ter
tiary education who are female (%)
SE.TER.GRAD.FE.ED.ZS    --Percentage of graduates from Education programmes in terti
ary education who are female (%)
SE.TER.GRAD.FE.EN.ZS    --Percentage of graduates from Engineering, Manufacturing an
d Construction programmes in tertiary education who are female (%)
SE.TER.GRAD.FE.HL.ZS    --Percentage of graduates from Health and Welfare programmes
in tertiary education who are female (%)
SE.TER.GRAD.FE.HU.ZS    --Percentage of graduates from Humanities and Arts programme
s in tertiary education who are female (%)
SE.TER.GRAD.FE.OT.ZS    --Percentage of graduates from programmes in unspecified fie
lds in tertiary education who are female (%)
SE.TER.GRAD.FE.SC.ZS    --Percentage of graduates from Science programmes in tertiar
y education who are female (%)
SE.TER.GRAD.FE.SS.ZS    --Percentage of graduates from Social Sciences, Business and
```

```
Law programmes in tertiary education who are female (%)
SE.TER.GRAD.FE.SV.ZS    --Percentage of graduates from Services programmes in tertia
ry education who are female (%)
SE.TER.GRAD.FE.ZS       --Percentage of graduates from tertiary education who are fe
male (%)
SE.TER.GRAD.HL.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Health and Welfare programmes, female (%)
SE.TER.GRAD.HL.ZS       --Percentage of graduates from tertiary education graduating
from Health and Welfare programmes, both sexes (%)
SE.TER.GRAD.HU.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Humanities and Arts programmes, female (%)
SE.TER.GRAD.HU.ZS       --Percentage of graduates from tertiary education graduating
from Humanities and Arts programmes, both sexes (%)
SE.TER.GRAD.OT.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from programmes in unspecified fields, female (%)
SE.TER.GRAD.OT.ZS       --Percentage of graduates from tertiary education graduating
from programmes in unspecified fields, both sexes (%)
SE.TER.GRAD.SC.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Science programmes, female (%)
SE.TER.GRAD.SC.ZS       --Percentage of graduates from tertiary education graduating
from Science programmes, both sexes (%)
SE.TER.GRAD.SS.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Social Sciences, Business and Law programmes, female (%)
SE.TER.GRAD.SS.ZS       --Percentage of graduates from tertiary education graduating
from Social Sciences, Business and Law programmes, both sexes (%)
SE.TER.GRAD.SV.FE.ZS    --Percentage of female graduates from tertiary education gra
duating from Services programmes, female (%)
SE.TER.GRAD.SV.ZS       --Percentage of graduates from tertiary education graduating
from Services programmes, both sexes (%)
SE.TER.PRIV.ZS  --Percentage of enrolment in tertiary education in private instituti
ons (%)
SE.TOT.ENRR     --Gross enrolment ratio, primary to tertiary, both sexes (%)
UIS.E.4 --Enrolment in post-secondary non-tertiary education, both sexes (number)
UIS.E.4.F       --Enrolment in post-secondary non-tertiary education, female (numbe
r)
UIS.E.4.Pu      --Enrolment in post-secondary non-tertiary education, public institu
tions, both sexes (number)
UIS.E.4.Pu.F    --Enrolment in post-secondary non-tertiary education, public institu
tions, female (number)
UIS.E.5.B       --Enrolment in tertiary education, ISCED 5 programmes, both sexes (n
umber)
UIS.E.5.B.F     --Enrolment in tertiary education, ISCED 5 programmes, female (numbe
r)
UIS.E.6 --Enrolment in tertiary education, ISCED 6 programmes, both sexes (number)
UIS.E.6.F       --Enrolment in tertiary education, ISCED 6 programmes, female (numbe
r)
UIS.E.7 --Enrolment in tertiary education, ISCED 7 programmes, both sexes (number)
UIS.E.7.F       --Enrolment in tertiary education, ISCED 7 programmes, female (numbe
r)
UIS.E.8 --Enrolment in tertiary education, ISCED 8 programmes, both sexes (number)
UIS.E.8.F       --Enrolment in tertiary education, ISCED 8 programmes, female (numbe
r)
UIS.FEP.4       --Percentage of students in post-secondary non-tertiary education wh
o are female (%)
UIS.FEP.5.B     --Percentage of students in tertiary ISCED 5 programmes who are fema
le (%)
UIS.FEP.56.F140 --Percentage of students enrolled in Education programmes in tertiar
y education who are female (%)
UIS.FEP.56.F200 --Percentage of students enrolled in Humanities and Arts programmes
in tertiary education who are female (%)
UIS.FEP.56.F300 --Percentage of students enrolled in Social Sciences, Business and L
aw programmes in tertiary education who are female (%)
UIS.FEP.56.F400 --Percentage of students enrolled in Science programmes in tertiary
education who are female (%)
UIS.FEP.56.F500 --Percentage of students enrolled in Engineering, Manufacturing and
Construction programmes in tertiary education who are female (%)
UIS.FEP.56.F600 --Percentage of students enrolled in Agriculture programmes in terti
ary education who are female (%)
UIS.FEP.56.F700 --Percentage of students enrolled in Health and Welfare programmes i
```

```
n tertiary education who are female (%)
UIS.FEP.56.F800 --Percentage of students enrolled in Services programmes in tertiary
education who are female (%)
UIS.FEP.56.FOREIGN      --Percentage of all inbound internationally mobile students
in tertiary education in the host country who are female, (%)
UIS.FEP.56.FUK  --Percentage of students enrolled in programmes in unspecified field
s in tertiary education who are female (%)
UIS.FEP.6       --Percentage of students in tertiary ISCED 6 programmes who are fema
le (%)
UIS.FEP.7       --Percentage of students in tertiary ISCED 7 programmes who are fema
le (%)
UIS.FEP.8       --Percentage of students in tertiary ISCED 8 programmes who are fema
le (%)
UIS.FGP.5.B     --Percentage of graduates from tertiary ISCED 5 programmes who are f
emale (%)
UIS.FGP.56.F400500      --Percentage of graduates from Science and Technology progra
mmes in tertiary education who are female (%)
UIS.FGP.56.FNON400500   --Percentage of graduates from non-science and non-technolog
y fields in tertiary education who are female (%)
UIS.FGP.6       --Percentage of graduates from tertiary ISCED 6 programmes who are f
emale (%)
UIS.FGP.7       --Percentage of graduates from tertiary ISCED 7 programmes who are f
emale (%)
UIS.FGP.8       --Percentage of graduates from tertiary ISCED 8 programmes who are f
emale (%)
UIS.FOSEP.56.F140       --Percentage of students in tertiary education enrolled in E
ducation programmes, both sexes (%)
UIS.FOSEP.56.F140.F     --Percentage of female students in tertiary education enroll
ed in Education programmes, female (%)
UIS.FOSEP.56.F140.M     --Percentage of male students in tertiary education enrolled
in Education programmes, male (%)
UIS.FOSEP.56.F200       --Percentage of students in tertiary education enrolled in H
umanities and Arts programmes, both sexes (%)
UIS.FOSEP.56.F200.F     --Percentage of female students in tertiary education enroll
ed in Humanities and Arts programmes, female (%)
UIS.FOSEP.56.F200.M     --Percentage of male students in tertiary education enrolled
in Humanities and Arts programmes, male (%)
UIS.FOSEP.56.F300       --Percentage of students in tertiary education enrolled in S
ocial Sciences, Business and Law programmes, both sexes (%)
UIS.FOSEP.56.F300.F     --Percentage of female students in tertiary education enroll
ed in Social Sciences, Business and Law programmes, female (%)
UIS.FOSEP.56.F300.M     --Percentage of male students in tertiary education enrolled
in Social Sciences, Business and Law programmes, male (%)
UIS.FOSEP.56.F400       --Percentage of students in tertiary education enrolled in S
cience programmes, both sexes (%)
UIS.FOSEP.56.F400.F     --Percentage of female students in tertiary education enroll
ed in Science programmes, female (%)
UIS.FOSEP.56.F400.M     --Percentage of male students in tertiary education enrolled
in Science programmes, male (%)
UIS.FOSEP.56.F500       --Percentage of students in tertiary education enrolled in E
ngineering, Manufacturing and Construction programmes, both sexes (%)
UIS.FOSEP.56.F500.F     --Percentage of female students in tertiary education enroll
ed in Engineering, Manufacturing and Construction programmes, female (%)
UIS.FOSEP.56.F500.M     --Percentage of male students in tertiary education enrolled
in Engineering, Manufacturing and Construction programmes, male (%)
UIS.FOSEP.56.F600       --Percentage of students in tertiary education enrolled in A
griculture programmes, both sexes (%)
UIS.FOSEP.56.F600.F     --Percentage of female students in tertiary education enroll
ed in Agriculture programmes, female (%)
UIS.FOSEP.56.F600.M     --Percentage of male students in tertiary education enrolled
in Agriculture programmes, male (%)
UIS.FOSEP.56.F700       --Percentage of students in tertiary education enrolled in H
ealth and Welfare programmes, both sexes (%)
UIS.FOSEP.56.F700.F     --Percentage of female students in tertiary education enroll
ed in Health and Welfare programmes, female (%)
UIS.FOSEP.56.F700.M     --Percentage of male students in tertiary education enrolled
in Health and Welfare programmes, male (%)
UIS.FOSEP.56.F800       --Percentage of students in tertiary education enrolled in S
ervices programmes, both sexes (%)
```

```
UIS.FOSEP.56.F800.F     --Percentage of female students in tertiary education enroll
ed in Services programmes, female (%)
UIS.FOSEP.56.F800.M     --Percentage of male students in tertiary education enrolled
in Services programmes, male (%)
UIS.FOSEP.56.FUK        --Percentage of students in tertiary education enrolled in p
rogrammes in unspecified fields, both sexes (%)
UIS.FOSEP.56.Fuk.F      --Percentage of female students in tertiary education enroll
ed in programmes in unspecified fields, female (%)
UIS.FOSEP.56.FUK.M      --Percentage of male students in tertiary education enrolled
in programmes in unspecified fields, male (%)
UIS.FOSGP.56.F140.M     --Percentage of male graduates from tertiary education gradu
ating from Education programmes, male (%)
UIS.FOSGP.56.F200.M     --Percentage of male graduates from tertiary education gradu
ating from Humanities and Arts programmes, male (%)
UIS.FOSGP.56.F300.M     --Percentage of male graduates from tertiary education gradu
ating from Social Sciences, Business and Law programmes, male (%)
UIS.FOSGP.56.F400.M     --Percentage of male graduates from tertiary education gradu
ating from Science programmes, male (%)
UIS.FOSGP.56.F500.M     --Percentage of male graduates from tertiary education gradu
ating from Engineering, Manufacturing and Construction programmes, male (%)
UIS.FOSGP.56.F600.M     --Percentage of male graduates from tertiary education gradu
ating from Agriculture programmes, male (%)
UIS.FOSGP.56.F700.M     --Percentage of male graduates from tertiary education gradu
ating from Health and Welfare programmes, male (%)
UIS.FOSGP.56.F800.M     --Percentage of male graduates from tertiary education gradu
ating from Services programmes, male (%)
UIS.FOSGP.56.FUK.M      --Percentage of male graduates from tertiary education gradu
ating from programmes in unspecified fields, male (%)
UIS.G.5.B.F     --Graduates from ISCED 5 programmes in tertiary education, female (n
umber)
UIS.G.5.B.T     --Graduates from ISCED 5 programmes in tertiary education, both sexe
s (number)
UIS.G.6.F       --Graduates from ISCED 6 programmes in tertiary education, female (n
umber)
UIS.G.6.T       --Graduates from ISCED 6 programmes in tertiary education, both sexe
s (number)
UIS.G.7.F       --Graduates from ISCED 7 programmes in tertiary education, female (n
umber)
UIS.G.7.T       --Graduates from ISCED 7 programmes in tertiary education, both sexe
s (number)
UIS.G.8.F       --Graduates from ISCED 8 programmes in tertiary education, female (n
umber)
UIS.G.8.T       --Graduates from ISCED 8 programmes in tertiary education, both sexe
s (number)
UIS.GER.1t6.F   --Gross enrolment ratio, primary to tertiary, female (%)
UIS.GER.1t6.GPI --Gross enrolment ratio, primary to tertiary, gender parity index (G
PI)
UIS.GER.1t6.M   --Gross enrolment ratio, primary to tertiary, male (%)
UIS.GGR.5.A.GPI --Gross graduation ratio from first degree programmes (ISCED 6 and
7) in tertiary education, gender parity index (GPI)
UIS.GOER.56     --Gross outbound enrolment ratio, all regions, both sexes (%)
UIS.MENF.56     --Net flow of internationally mobile students (inbound - outbound),
both sexes (number)
UIS.MENFR.56    --Net flow ratio of internationally mobile students (inbound - outbo
und), both sexes (%)
UIS.MS.56.F     --Total inbound internationally mobile students, female (number)
UIS.MS.56.T     --Total inbound internationally mobile students, both sexes (number)
UIS.MSEP.56     --Inbound mobility rate, both sexes (%)
UIS.MSEP.56.F   --Inbound mobility rate, female (%)
UIS.MSEP.56.M   --Inbound mobility rate, male (%)
UIS.OE.56.40510         --Total outbound internationally mobile tertiary students st
udying abroad, all countries, both sexes (number)
UIS.OMR.56      --Outbound mobility ratio, all regions, both sexes (%)
UIS.SLE.1t6.GPI --School life expectancy, primary to tertiary, gender parity index
(GPI)
UIS.SLE.56      --School life expectancy, tertiary, both sexes (years)
UIS.SLE.56.F    --School life expectancy, tertiary, female (years)
UIS.SLE.56.GPI  --School life expectancy, tertiary, gender parity index (GPI)
UIS.SLE.56.M    --School life expectancy, tertiary, male (years)
```

```
UIS.TE_100000.56        --Enrolment in tertiary education per 100,000 inhabitants, b
oth sexes
UIS.TE_100000.56.F      --Enrolment in tertiary education per 100,000 inhabitants, f
emale
UIS.TE_100000.56.M      --Enrolment in tertiary education per 100,000 inhabitants, m
ale
UIS.TEP.5.B     --Percentage of all students in tertiary education enrolled in ISCED
5, both sexes (%)
UIS.TEP.5.B.F   --Percentage of female students in tertiary education enrolled in IS
CED 5
UIS.TEP.5.B.M   --Percentage of male students in tertiary education enrolled in ISCE
D 5
UIS.TEP.6       --Percentage of all students in tertiary education enrolled in ISCED
6, both sexes (%)
UIS.TEP.6.F     --Percentage of female students in tertiary education enrolled in IS
CED 6
UIS.TEP.6.M     --Percentage of male students in tertiary education enrolled in ISCE
D 6
UIS.TEP.7       --Percentage of all students in tertiary education enrolled in ISCED
7, both sexes (%)
UIS.TEP.7.F     --Percentage of female students in tertiary education enrolled in IS
CED 7
UIS.TEP.7.M     --Percentage of male students in tertiary education enrolled in ISCE
D 7
UIS.TEP.8       --Percentage of all students in tertiary education enrolled in ISCED
8, both sexes (%)
UIS.TEP.8.F     --Percentage of female students in tertiary education enrolled in IS
CED 8
UIS.TEP.8.M     --Percentage of male students in tertiary education enrolled in ISCE
D 8
UIS.thAge.4.A.GPV       --Official entrance age to post-secondary non-tertiary educa
tion (years)
UIS.thDur.4.A.GPV       --Theoretical duration of post-secondary non-tertiary educat
ion (years)
```

Notes: les indicators, leurs definitions, et les thematiques

Les indicators pertinents

'Attainment'

- Les études mesures le niveau scolaire de la population.
- Il y 3 études / sources :
  - Barro-Lee
  - Projection
  - UIS ("The UNESCO Institute for Statistics")
- ISCED = International Standard Classification of Education
- https://en.wikipedia.org/wiki/International_Standard_Classification_of_Education
  - rédefiné en 2011 (ancienne definition date de 1997)
- ISCED 3 = Upper Secondary Education = Lycée
- ISCED 6+ = Bachelors and higher / Tertiary Education = Université

'Secondary', 'Tertiary'

- Ces thèmes ont des indicateurs interessantes:
  - complètion de niveau 'upper secondary' et 'tertiary'
- information sur taux d'inscription, nombre total d'inscriptions, hors-école, achèvement, système, infrastructures, ...

'Infrastructure: Communications'

- il y a deux indicators:
  - usage des ordinateurs : "Personal computers (per 100 people)"
  - usage de l'internet : "Internet users (per 100 people)"
- très important pour une service en ligne

'Population', 'Health: Population: Structure', 'Health: Population: Dynamics'

- Donne informations sur la démographique et sa croissance

'Learning outcomes'

- Différentes études évaluent la qualité / les résultats d'apprentissage des systèmes éducatifs dans les différents pays.
- Les thèmes abordés entre autres sont : la lecture, les différentes langues, les sciences, les mathématiques.
- La plus part des études regardent les compétences de l'enseignement primaire. On les ignore.
- PIAAC regarde les compétences génerals des adults
  - "The Programme for the International Assessment of Adult Competencies (PIAAC) is a programme of assessment and analysis of adult skills."
  - Il y a un indicateur qui montre le percentage des adultes qui on échoué dans le test basique ICT:
    - "the ICT core test, which assesses basic ICT skills, such as the capacity to use a mouse or scroll through a web page, needed to take the computer-based assessment."
- PISA regarde les compétences des jeunes de l'âge de 15 ans.
- TIMSS regarde les compétences en sciences et mathématiques pour '4th' et '8th' grade.

'Education Equality'

- Les études regardent le niveau scolaire de la population, mais aussi la distribution entre homme/femme, rich/pauvre et urban/rural
- Il y a études / sources:
  - DHS (The Demographic and Health Surveys (DHS) Program): regarde 'primary', 'secondary' et 'post secondary' scolarité.
  - MICS ("Multiple Indicator Cluster Surveys" de UNICEF): regarde 'primary', 'secondary', et le moyen nombre d'années de scolarité 15+ et 15-19.
  - UIS : ne regarde que la scolarité justqu'au niveau 'lower secondary' = collège.

'Economic Policy & Debt: Purchasing power parity', 'Economic Policy & Debt: National accounts: US at current prices: Aggregate indicators', Economic Policy & Debt: National accounts: US at constant 2010 prices: Aggregate indicators', 'Economic Policy & Debt: National accounts: Atlas GNI & GNI per capita'

- Le GDP et GNI: donne information sur la taille de l'economie du pays.
- "Purchasing power parity (PPP) is a measurement of prices in different countries that uses the prices of specific goods to compare the absolute purchasing power of the countries' currencies."

Les thématiques pas pertinents

Certaines pourraient peut-être être intéressantes pour une analyse plus poussée.

'SABER', 'EMIS'

- Ce sont des indicateurs liés aux politiques. Pourrait potentiellement montrer les efforts des pays pour améliorer leurs systèmes éducatifs. Mais sont difficiles à interpréter à ce stade. SABER = Systems Approach for Better Education Results

- https://www.worldbank.org/en/topic/education/brief/systems-approach-for-better-education-results-saber

- "SABER is an initiative to produce comparative data and knowledge on education policies and institutions, with the aim of helping countries systematically strengthen their education systems and the ultimate goal of promoting Learning for All."
- The indicators labeled SABER refer to policiy goals and the evaluation towards achieving these goals.

EMIS = Education Management Information Systems

- http://wbgfiles.worldbank.org/documents/hdn/ed/saber/supporting_doc/Background/EMIS/SAI
- "An education management information system is responsible for producing and monitoring education statistics within an education system."
- "SABER's EMIS policy domain aims to help countries identify areas for improvement in data collection, data and system management, and data use in decision making and improving the different elements of the education system."

'Pre-Primary', 'Primary', 'Early Childhood Education', 'Post-Secondary/Non-Tertiary'

'Teachers'

- Information sur le nombre, la salaire et l'éducation des enseignants pour les différents niveaux.

'Expenditures'

- Informations sur les dépenses des pays pour leurs systhèmes éducatifs.

'Literacy'

- Informations sur l'alphabétisation de la population. Ces indicateurs ne sont pas intéressants, car on peut assumer que quelqu'un avec un Baccalauréat sait lire.

'Laber'

- Inforamtion sur la pourcentage des jeunes sans occupation (pas scolarisés, pas en formation et sans travaille).

'Background'

- Informations sur la durée de la scolarisation obligatoire et l'âge d'entrée.

'Social Protection & Labor: Labor force structure', 'Social Protection & Labor: Unemployment'

'Health: Risk factors', 'Health: Mortality'

Notes: unités

- percentage de la population
- population / population en milliers
- nombre moyen des années (e.g. de scolarité)
- taux, ratios (scolarisation, abandon, fem/masc, ...)
- nombre moyen d'étudiants par X (class, livres, ...)

3.3.1 Selectionner les données pertinentes

Liste des indicateurs potentiellement pertinents :

- Internet users (per 100 people)(IT.NET.USER.P2)
- Personal computers (per 100 people)(IT.CMP.PCMP.P2)

- Barro-Lee: Percentage of population age 15+ with secondary schooling. Completed Secondary (BAR.SEC.CMPT.15UP.ZS)
- Barro-Lee: Percentage of population age 15+ with tertiary schooling. Completed Tertiary (BAR.TER.CMPT.15UP.ZS)
- Barro-Lee: Percentage of population age 25+ with secondary schooling. Completed Secondary (BAR.SEC.CMPT.25UP.ZS)
- Barro-Lee: Percentage of population age 25+ with tertiary schooling. Completed Tertiary (BAR.TER.CMPT.25UP.ZS)
- Barro-Lee: Population in thousands, age 15+, total (BAR.POP.15UP)
- Barro-Lee: Population in thousands, age 25+, total (BAR.POP.25UP)

- Wittgenstein Projection: Percentage of the population age 15+ by highest level of educational attainment. Upper Secondary. Total (PRJ.ATT.15UP.3.MF)
- Wittgenstein Projection: Percentage of the population age 15+ by highest level of educational attainment. Post Secondary. Total (PRJ.ATT.15UP.4.MF)
- Wittgenstein Projection: Percentage of the population age 25+ by highest level of educational attainment. Upper Secondary. Total (PRJ.ATT.25UP.3.MF)
- Wittgenstein Projection: Percentage of the population age 25+ by highest level of educational attainment. Post Secondary. Total (PRJ.ATT.25UP.4.MF)
- Projection: Percentage of the population age 20-24 by highest level of educational attainment. Upper Secondary. Total (PRJ.ATT.2024.3.MF)
- Projection: Percentage of the population age 20-24 by highest level of educational attainment. Post Secondary. Total (PRJ.ATT.2024.4.MF)
- Projection: Population age 20-24 in thousands by highest level of educational attainment. Upper Secondary. Total (PRJ.POP.2024.3.MF)
- Projection: Population age 20-24 in thousands by highest level of educational attainment. Post Secondary. Total (PRJ.POP.2024.4.MF)
- Wittgenstein Projection: Population in thousands by highest level of educational attainment. Upper Secondary. Total (PRJ.POP.ALL.3.MF)

- Wittgenstein Projection: Population in thousands by highest level of educational attainment. Post Secondary. Total (PRJ.POP.ALL.4.MF)
- Wittgenstein Projection: Percentage of the total population by highest level of educational attainment. Upper Secondary. Total (PRJ.ATT.ALL.3.MF)
- Wittgenstein Projection: Percentage of the total population by highest level of educational attainment. Post Secondary. Total (PRJ.ATT.ALL.4.MF)

- UIS: Percentage of population age 25+ with completed upper secondary education. Total (UIS.EA.3.AG25T99)
- UIS: Percentage of population age 25+ with at least completed upper secondary education (ISCED 3 or higher). Total (UIS.EA.3T6.AG25T99)
- UIS: Percentage of population age 25+ with at least a completed short-cycle tertiary degree (ISCED 5 or higher). Total (UIS.EA.5T8.AG25T99)
- UIS: Percentage of population age 25+ with at least a completed bachelor's or equivalent degree (ISCED 6 or higher). Total (UIS.EA.6T8.AG25T99)

- Enrolement in upper secondary education, both sexes (number) (UIS.E.3)
- Enrolment in upper secondary general, both sexes (number) (UIS.E.3.GPV)
- Adjusted net enrolment rate, upper secondary, both sexes (%) (UIS.NERA.3)
- Enrolment in tertiary education, ISCED 5 programmes, both sexes (number) (UIS.E.5.B)
- Enrolment in tertiary education, ISCED 6 programmes, both sexes (number) (UIS.E.6)
- Enrolment in tertiary education, ISCED 7 programmes, both sexes (number) (UIS.E.7)
- Enrolment in tertiary education, ISCED 8 programmes, both sexes (number) (UIS.E.8)
- Graduates from ISCED 5 programmes in tertiary education, both sexes (number) (UIS.G.5.B.T)
- Graduates from ISCED 6 programmes in tertiary education, both sexes (number) (UIS.G.6.T)
- Graduates from ISCED 7 programmes in tertiary education, both sexes (number) (UIS.G.7.T)
- Graduates from ISCED 8 programmes in tertiary education, both sexes (number) (UIS.G.8.T)

- Enrolment in secondary education, both sexes (number) (SE.SEC.ENRL)
- Enrolment in secondary general, both sexes (number) (SE.SEC.ENRL.GC)
- Gross enrolment ratio, upper secondary, both sexes (%)(SE.SEC.ENRR.UP)

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

- Gross enrolment ratio for tertiary education, both sexes (%) (SE.TER.ENRR)
- Graduates from tertiary education, both sexes (number) (SE.TER.GRAD)
- Gross graduation ratio from first degree programmes (ISCED 6 and 7) in tertiary education, both sexes (%)(SE.TER.CMPL.ZS)

- Population, total (SP.POP.TOTL)
- Population growth (annual %) (SP.POP.GROW)
- Population, ages 0-14, total(SP.POP.0014.TO)
- Population, ages 15-24, total(SP.POP.1524.TO.UN)
- Population, ages 15-64, total (SP.POP.1564.TO)
- School age population, upper secondary education, both sexes (number)(SP.SEC.UTOT.IN)
- School age population, tertiary education, both sexes (number)(SP.TER.TOTL.IN)

- PIAAC: Young adults by proficiency level in problem solving in technology-rich environments (%). No computer experience or failed the ICT core test

(LO.PIAAC.TEC.YOU.FAILNO)
- PIAAC: Adults by proficiency level in problem solving in technology-rich environments (%). Failed the ICT Core Test (LO.PIAAC.TEC.FAIL)
- PIAAC: Adults by proficiency level in problem solving in technology-rich environments (%). No computer experience (LO.PIAAC.TEC.NO)

- GDP, PPP (current international US) (NY.GDP.MKTP.PP.CD)
- GDP per capita, PPP (current international US) (NY.GDP.PCAP.PP.CD)
- GDP (current US) (NY.GDP.MKTP.CD)
- GDP per capita (current US) (NY.GDP.PCAP.CD)
- GNI, PPP (current international US) (NY.GNP.MKTP.PP.CD)
- GNI per capita, PPP (current international US) (NY.GNP.PCAP.PP.CD)
- GNI (current US) (NY.GNP.MKTP.CD)
- GNI per capita, Atlas method (current US) (NY.GNP.PCAP.CD)

- HH.DHS.GAR.456 --DHS: Gross attendance rate. Post Secondary
- HH.DHS.GAR.456.F --DHS: Gross attendance rate. Post Secondary. Female
- HH.DHS.GAR.456.M --DHS: Gross attendance rate. Post Secondary. Male
- HH.DHS.GAR.456.Q1 --DHS: Gross attendance rate. Post Secondary. Quintile 1
- HH.DHS.GAR.456.Q2 --DHS: Gross attendance rate. Post Secondary. Quintile 2
- HH.DHS.GAR.456.Q3 --DHS: Gross attendance rate. Post Secondary. Quintile 3
- HH.DHS.GAR.456.Q4 --DHS: Gross attendance rate. Post Secondary. Quintile 4
- HH.DHS.GAR.456.Q5 --DHS: Gross attendance rate. Post Secondary. Quintile 5
- HH.DHS.GAR.456.R --DHS: Gross attendance rate. Post Secondary. Rural
- HH.DHS.GAR.456.U --DHS: Gross attendance rate. Post Secondary. Urban

- HH.MICS.GAR.456 --MICS: Gross attendance rate. Post Secondary
- HH.MICS.GAR.456.F --MICS: Gross attendance rate. Post Secondary. Female
- HH.MICS.GAR.456.M --MICS: Gross attendance rate. Post Secondary. Male
- HH.MICS.GAR.456.Q1 --MICS: Gross attendance rate. Post Secondary. Quintile 1
- HH.MICS.GAR.456.Q2 --MICS: Gross attendance rate. Post Secondary. Quintile 2
- HH.MICS.GAR.456.Q3 --MICS: Gross attendance rate. Post Secondary. Quintile 3
- HH.MICS.GAR.456.Q4 --MICS: Gross attendance rate. Post Secondary. Quintile 4
- HH.MICS.GAR.456.Q5 --MICS: Gross attendance rate. Post Secondary. Quintile 5
- HH.MICS.GAR.456.R --MICS: Gross attendance rate. Post Secondary. Rural
- HH.MICS.GAR.456.U --MICS: Gross attendance rate. Post Secondary. Urban

- HH.MICS.YRS.15UP.GIN --MICS: Gini coefficient of average years of schooling. Age 15+
- HH.MICS.YRS.15UP.GIN.F --MICS: Gini coefficient of average years of schooling. Age 15+. Female
- HH.MICS.YRS.15UP.GIN.M --MICS: Gini coefficient of average years of schooling. Age 15+. Male
- HH.MICS.YRS.15UP.GIN.Q1 --MICS: Gini coefficient of average years of schooling. Age 15+. Quintile 1
- HH.MICS.YRS.15UP.GIN.Q2 --MICS: Gini coefficient of average years of schooling. Age 15+. Quintile 2
- HH.MICS.YRS.15UP.GIN.Q3 --MICS: Gini coefficient of average years of schooling. Age 15+. Quintile 3

- HH.MICS.YRS.15UP.GIN.Q4 --MICS: Gini coefficient of average years of schooling. Age 15+. Quintile 4
- HH.MICS.YRS.15UP.GIN.Q5 --MICS: Gini coefficient of average years of schooling. Age 15+. Quintile 5
- HH.MICS.YRS.15UP.GIN.R --MICS: Gini coefficient of average years of schooling. Age 15+. Rural
- HH.MICS.YRS.15UP.GIN.U --MICS: Gini coefficient of average years of schooling. Age 15+. Urban

In [21]:

```
#look at the groups of indicators and see if there is sufficient data
def num_entries_per_year(data, indicatorlist, output = False):
    years = [str(y) for y in range(2010, 2026)]
    data = data[data['Indicator Code'].isin(indicatorlist)].filter(['Indicator Code'
    droplist = []
    for i in indicatorlist:
        data2 = data[data['Indicator Code'] == i]
        countlist = {y : data[data['Indicator Code'] == i][y].count() for y in data.
        n = sum(countlist.values())
        num_cc = len(data2.iloc[:,2:].dropna(how = 'all').index.values)

        if output:
            print(i, "Number of countries: ", num_cc)
            for y,c in countlist.items():
                print(y, c)
        if n == 0:
            droplist.append(i)
    return droplist
```

In [22]:

```
my_indicators = ['IT.NET.USER.P2','IT.CMP.PCMP.P2',
                 'BAR.SEC.CMPT.15UP.ZS', 'BAR.TER.CMPT.15UP.ZS', 'BAR.SEC.CMPT.25UP.
                 'PRJ.ATT.15UP.3.MF','PRJ.ATT.15UP.4.MF','PRJ.ATT.25UP.3.MF', 'PRJ.A
                 'PRJ.ATT.2024.3.MF','PRJ.ATT.2024.4.MF', 'PRJ.POP.2024.3.MF', 'PRJ.
                 'PRJ.POP.ALL.3.MF', 'PRJ.POP.ALL.4.MF', 'PRJ.ATT.ALL.3.MF', 'PRJ.AT
                 'UIS.EA.3T6.AG25T99', 'UIS.EA.6T8.AG25T99', 'UIS.EA.3.AG25T99', 'UI
                 'UIS.E.3','UIS.E.3.GPV','UIS.NERA.3','UIS.E.5.B','UIS.E.6','UIS.E.7
                 'UIS.G.5.B.T', 'UIS.G.6.T', 'UIS.G.7.T', 'UIS.G.8.T',
                 'SE.SEC.ENRL', 'SE.SEC.ENRL.GC', 'SE.SEC.ENRR.UP','SE.TER.ENRL', 'S
                 'SP.POP.TOTL', 'SP.POP.GROW', 'SP.POP.0014.TO', 'SP.POP.1524.TO.UN'
                 'SP.SEC.UTOT.IN', 'SP.TER.TOTL.IN',
                 'LO.PIAAC.TEC.YOU.FAILNO', 'LO.PIAAC.TEC.FAIL', 'LO.PIAAC.TEC.NO',
                 'NY.GDP.MKTP.PP.CD','NY.GDP.PCAP.PP.CD', 'NY.GDP.MKTP.CD', 'NY.GDP.
                 'NY.GNP.MKTP.PP.CD', 'NY.GNP.PCAP.PP.CD', 'NY.GNP.MKTP.CD', 'NY.GNP
                 'HH.DHS.GAR.456', 'HH.MICS.GAR.456', 'HH.MICS.YRS.15UP.GIN']
droplist = num_entries_per_year(DATA, my_indicators, output = True)
print(droplist)
```

```
IT.NET.USER.P2 Number of countries:  211
2010 209
2011 211
2012 209
2013 208
2014 208
2015 208
2016 208
2017 0
2020 0
2025 0
IT.CMP.PCMP.P2 Number of countries:  0
2010 0
2011 0
2012 0
```

```
2013 0
2014 0
2015 0
2016 0
2017 0
2020 0
2025 0
BAR.SEC.CMPT.15UP.ZS Number of countries:  144
2010 144
2011 0
2012 0
2013 0
2014 0
2015 0
2016 0
2017 0
2020 0
2025 0
BAR.TER.CMPT.15UP.ZS Number of countries:  144
2010 144
2011 0
2012 0
2013 0
2014 0
2015 0
2016 0
2017 0
2020 0
2025 0
BAR.SEC.CMPT.25UP.ZS Number of countries:  144
2010 144
2011 0
2012 0
2013 0
2014 0
2015 0
2016 0
2017 0
2020 0
2025 0
BAR.TER.CMPT.25UP.ZS Number of countries:  144
2010 144
2011 0
2012 0
2013 0
2014 0
2015 0
2016 0
2017 0
2020 0
2025 0
PRJ.ATT.15UP.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.15UP.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
```

```
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.25UP.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.25UP.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.2024.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.2024.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.POP.2024.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.POP.2024.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
```

```
2025 166
PRJ.POP.ALL.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.POP.ALL.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.ALL.3.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
PRJ.ATT.ALL.4.MF Number of countries:  166
2010 166
2011 0
2012 0
2013 0
2014 0
2015 166
2016 0
2017 0
2020 166
2025 166
UIS.EA.3T6.AG25T99 Number of countries:  120
2010 74
2011 70
2012 62
2013 55
2014 65
2015 56
2016 2
2017 0
2020 0
2025 0
UIS.EA.6T8.AG25T99 Number of countries:  89
2010 15
2011 12
2012 17
2013 37
2014 51
2015 46
2016 2
2017 0
2020 0
2025 0
UIS.EA.3.AG25T99 Number of countries:  113
2010 72
```

```
2011 69
2012 61
2013 51
2014 53
2015 1
2016 0
2017 0
2020 0
2025 0
UIS.EA.5T8.AG25T99 Number of countries:  113
2010 72
2011 67
2012 59
2013 51
2014 59
2015 51
2016 2
2017 0
2020 0
2025 0
UIS.E.3 Number of countries:  188
2010 156
2011 161
2012 159
2013 147
2014 125
2015 7
2016 0
2017 0
2020 0
2025 0
UIS.E.3.GPV Number of countries:  200
2010 164
2011 168
2012 167
2013 160
2014 138
2015 8
2016 0
2017 0
2020 0
2025 0
UIS.NERA.3 Number of countries:  140
2010 57
2011 57
2012 53
2013 100
2014 90
2015 3
2016 0
2017 0
2020 0
2025 0
UIS.E.5.B Number of countries:  143
2010 112
2011 113
2012 115
2013 99
2014 87
2015 4
2016 0
2017 0
2020 0
2025 0
UIS.E.6 Number of countries:  125
2010 3
2011 4
2012 16
2013 109
```

```
2014 97
2015 3
2016 0
2017 0
2020 0
2025 0
UIS.E.7 Number of countries:  123
2010 2
2011 3
2012 16
2013 108
2014 94
2015 2
2016 0
2017 0
2020 0
2025 0
UIS.E.8 Number of countries:  136
2010 97
2011 102
2012 106
2013 109
2014 93
2015 3
2016 0
2017 0
2020 0
2025 0
UIS.G.5.B.T Number of countries:  121
2010 82
2011 81
2012 92
2013 62
2014 63
2015 5
2016 0
2017 0
2020 0
2025 0
UIS.G.6.T Number of countries:  100
2010 3
2011 3
2012 11
2013 80
2014 75
2015 4
2016 0
2017 0
2020 0
2025 0
UIS.G.7.T Number of countries:  98
2010 2
2011 2
2012 10
2013 75
2014 74
2015 3
2016 0
2017 0
2020 0
2025 0
UIS.G.8.T Number of countries:  108
2010 78
2011 81
2012 83
2013 68
2014 66
2015 4
2016 0
```

```
2017 0
2020 0
2025 0
SE.SEC.ENRL Number of countries:  191
2010 158
2011 163
2012 157
2013 148
2014 148
2015 123
2016 8
2017 0
2020 0
2025 0
SE.SEC.ENRL.GC Number of countries:  205
2010 164
2011 170
2012 170
2013 163
2014 163
2015 140
2016 12
2017 0
2020 0
2025 0
SE.SEC.ENRR.UP Number of countries:  184
2010 152
2011 157
2012 153
2013 144
2014 142
2015 115
2016 8
2017 0
2020 0
2025 0
SE.TER.ENRL Number of countries:  179
2010 147
2011 149
2012 148
2013 138
2014 131
2015 110
2016 7
2017 0
2020 0
2025 0
SE.TER.ENRR Number of countries:  172
2010 140
2011 145
2012 143
2013 132
2014 126
2015 100
2016 7
2017 0
2020 0
2025 0
SE.TER.GRAD Number of countries:  145
2010 94
2011 94
2012 107
2013 101
2014 105
2015 86
2016 7
2017 0
2020 0
2025 0
```

```
SE.TER.CMPL.ZS Number of countries:  115
2010 71
2011 74
2012 78
2013 72
2014 71
2015 48
2016 2
2017 0
2020 0
2025 0
SP.POP.TOTL Number of countries:  222
2010 222
2011 222
2012 221
2013 221
2014 217
2015 217
2016 217
2017 0
2020 0
2025 0
SP.POP.GROW Number of countries:  222
2010 222
2011 221
2012 221
2013 221
2014 217
2015 217
2016 217
2017 0
2020 0
2025 0
SP.POP.0014.TO Number of countries:  201
2010 201
2011 201
2012 200
2013 200
2014 198
2015 198
2016 198
2017 0
2020 0
2025 0
SP.POP.1524.TO.UN Number of countries:  181
2010 181
2011 181
2012 181
2013 181
2014 181
2015 181
2016 0
2017 0
2020 0
2025 0
SP.POP.1564.TO Number of countries:  201
2010 201
2011 201
2012 200
2013 200
2014 198
2015 198
2016 198
2017 0
2020 0
2025 0
SP.SEC.UTOT.IN Number of countries:  204
2010 202
2011 203
```

```
2012 203
2013 203
2014 204
2015 195
2016 158
2017 0
2020 0
2025 0
SP.TER.TOTL.IN Number of countries:  202
2010 199
2011 182
2012 176
2013 170
2014 202
2015 129
2016 7
2017 0
2020 0
2025 0
LO.PIAAC.TEC.YOU.FAILNO Number of countries:  30
2010 0
2011 0
2012 22
2013 0
2014 0
2015 8
2016 0
2017 0
2020 0
2025 0
LO.PIAAC.TEC.FAIL Number of countries:  30
2010 0
2011 0
2012 22
2013 0
2014 0
2015 8
2016 0
2017 0
2020 0
2025 0
LO.PIAAC.TEC.NO Number of countries:  30
2010 0
2011 0
2012 22
2013 0
2014 0
2015 8
2016 0
2017 0
2020 0
2025 0
NY.GDP.MKTP.PP.CD Number of countries:  199
2010 197
2011 199
2012 195
2013 195
2014 192
2015 191
2016 188
2017 0
2020 0
2025 0
NY.GDP.PCAP.PP.CD Number of countries:  199
2010 197
2011 199
2012 195
2013 195
2014 192
```

```
2015 191
2016 188
2017 0
2020 0
2025 0
NY.GDP.MKTP.CD Number of countries:  211
2010 210
2011 210
2012 206
2013 207
2014 204
2015 203
2016 196
2017 0
2020 0
2025 0
NY.GDP.PCAP.CD Number of countries:  211
2010 210
2011 210
2012 206
2013 207
2014 204
2015 203
2016 196
2017 0
2020 0
2025 0
NY.GNP.MKTP.PP.CD Number of countries:  195
2010 195
2011 195
2012 193
2013 193
2014 190
2015 188
2016 184
2017 0
2020 0
2025 0
NY.GNP.PCAP.PP.CD Number of countries:  195
2010 195
2011 195
2012 193
2013 193
2014 190
2015 188
2016 184
2017 0
2020 0
2025 0
NY.GNP.MKTP.CD Number of countries:  201
2010 200
2011 200
2012 198
2013 198
2014 194
2015 192
2016 187
2017 0
2020 0
2025 0
NY.GNP.PCAP.CD Number of countries:  198
2010 197
2011 197
2012 196
2013 196
2014 191
2015 190
2016 185
2017 0
```

```
2020 0
2025 0
HH.DHS.GAR.456 Number of countries:  39
2010 11
2011 7
2012 14
2013 7
2014 8
2015 5
2016 0
2017 0
2020 0
2025 0
HH.MICS.GAR.456 Number of countries:  19
2010 6
2011 6
2012 1
2013 0
2014 7
2015 0
2016 0
2017 0
2020 0
2025 0
HH.MICS.YRS.15UP.GIN Number of countries:  19
2010 6
2011 6
2012 1
2013 0
2014 7
2015 0
2016 0
2017 0
2020 0
2025 0
['IT.CMP.PCMP.P2']
```

Résumé: variables - un premier choix

Pour une première analyse garde les variables qui ont de la data recente et assez complète:

- IT.NET.USER.P2 -- information sur l'usage internet, récente (2016) et presque complète
- PRJ.ATT ... -- projection sur le niveau d'education de la population des pays / régions
- UIS.EA. ... -- information sur le niveau d'education de la population (pas très complète)
- UIS.E. ... -- information sur le nombre des personnes inscrites dans les différents programmes d'éducation (plus complète que la data de UIS.EA. ...)
- SE.SEC. / SE.TER. ... -- information sur le nombre des personnes inscrites dans les différentes programmes d'éducation (un petite peu plus récente (2015) et complète que les data UIS.E ...)
- SP.POP. / SP.SEC / SP.TER ... -- information sur la popluation et démographie, récente (2016) et presque complète
- NY.GDP. / NY.GNP. ... -- information sur l'économie des pays

In [23]:

```
ALL_MY_INDICATORS = ['IT.NET.USER.P2',
                     'PRJ.ATT.15UP.3.MF','PRJ.ATT.15UP.4.MF','PRJ.ATT.25UP.3.MF', 'P
                     'PRJ.ATT.2024.3.MF','PRJ.ATT.2024.4.MF', 'PRJ.POP.2024.3.MF', '
                     'PRJ.POP.ALL.3.MF', 'PRJ.POP.ALL.4.MF', 'PRJ.ATT.ALL.3.MF', 'PR
                     'UIS.EA.3T6.AG25T99', 'UIS.EA.6T8.AG25T99', 'UIS.EA.3.AG25T99',
                     'UIS.E.3','UIS.E.3.GPV','UIS.NERA.3','UIS.E.5.B','UIS.E.6','UIS
                     'UIS.G.5.B.T', 'UIS.G.6.T', 'UIS.G.7.T', 'UIS.G.8.T',
                     'SE.SEC.ENRL', 'SE.SEC.ENRL.GC', 'SE.SEC.ENRR.UP','SE.TER.ENRL'
                     'SP.POP.TOTL', 'SP.POP.GROW', 'SP.POP.0014.TO', 'SP.POP.1524.TO
```

```
                        'SP.SEC.UTOT.IN', 'SP.TER.TOTL.IN',
                        'NY.GDP.MKTP.PP.CD','NY.GDP.PCAP.PP.CD', 'NY.GDP.MKTP.CD', 'NY.
                        'NY.GNP.MKTP.PP.CD', 'NY.GNP.PCAP.PP.CD', 'NY.GNP.MKTP.CD', 'NY
```

In [24]:

```
DATA = DATA[DATA['Indicator Code'].isin(ALL_MY_INDICATORS)]
DATA.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10976 entries, 3673 to 886921
Data columns (total 29 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country Name    10976 non-null  object
 1   Country Code    10976 non-null  object
 2   Indicator Name  10976 non-null  object
 3   Indicator Code  10976 non-null  object
 4   2000            4700 non-null   float64
 5   2001            4744 non-null   float64
 6   2002            4804 non-null   float64
 7   2003            4784 non-null   float64
 8   2004            4903 non-null   float64
 9   2005            4907 non-null   float64
 10  2006            4909 non-null   float64
 11  2007            4983 non-null   float64
 12  2008            5011 non-null   float64
 13  2009            5036 non-null   float64
 14  2010            7074 non-null   float64
 15  2011            5098 non-null   float64
 16  2012            5098 non-null   float64
 17  2013            5325 non-null   float64
 18  2014            5227 non-null   float64
 19  2015            5955 non-null   float64
 20  2016            2766 non-null   float64
 21  2017            0 non-null      float64
 22  2020            1992 non-null   float64
 23  2025            1992 non-null   float64
 24  2030            1992 non-null   float64
 25  2035            1992 non-null   float64
 26  2040            1992 non-null   float64
 27  2045            1992 non-null   float64
 28  2050            1992 non-null   float64
dtypes: float64(25), object(4)
memory usage: 2.5+ MB
```

In [25]:

```
#drop the year 2017, since there is no data for that
DATA = DATA.drop(columns = ['2017'])
ALL_YEARS = list(ALL_YEARS)
ALL_YEARS.remove('2017')
ALL_YEAR = np.array(ALL_YEARS)
YEARS = np.array([y for y in ALL_YEARS if int(y) < 2020])
```

3.4 Reformatter la data

- Reformater la data:
  - donner des noms sans espace aux columns
  - travailler que avec les codes (laisser tomber les noms des indicateurs et pays)
- Ecrire des fonctions pour pouvoir travailler plus facilement avec la data :
  - sélectionner des tranches
  - sélectionner les axis

In [26]:

```
DATA = DATA.drop(columns = ['Country Name', 'Indicator Name'])
```

```
DATA = DATA.rename(columns = {'Indicator Code' : 'indicator', 'Country Code' : 'coun
INDEX_NAMES = ['indicator','country','year']
DATA.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10976 entries, 3673 to 886921
Data columns (total 26 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   country    10976 non-null  object
 1   indicator  10976 non-null  object
 2   2000       4700 non-null   float64
 3   2001       4744 non-null   float64
 4   2002       4804 non-null   float64
 5   2003       4784 non-null   float64
 6   2004       4903 non-null   float64
 7   2005       4907 non-null   float64
 8   2006       4909 non-null   float64
 9   2007       4983 non-null   float64
 10  2008       5011 non-null   float64
 11  2009       5036 non-null   float64
 12  2010       7074 non-null   float64
 13  2011       5098 non-null   float64
 14  2012       5098 non-null   float64
 15  2013       5325 non-null   float64
 16  2014       5227 non-null   float64
 17  2015       5955 non-null   float64
 18  2016       2766 non-null   float64
 19  2020       1992 non-null   float64
 20  2025       1992 non-null   float64
 21  2030       1992 non-null   float64
 22  2035       1992 non-null   float64
 23  2040       1992 non-null   float64
 24  2045       1992 non-null   float64
 25  2050       1992 non-null   float64
dtypes: float64(24), object(2)
memory usage: 2.3+ MB
```

In [27]:

```
def deconstruct_df(data):
    #could also use filter function
    data_columns = [p for p in data.columns.values if p not in INDEX_NAMES]
    index_column = [i for i in INDEX_NAMES if i not in data.columns.values][0]
    data_d = pd.DataFrame(columns = INDEX_NAMES)

    for p in data_columns :
        #print(p)
        y_data = data.filter(INDEX_NAMES + [p])
        y_data[index_column] = (p)
        y_data = y_data.rename(columns = {p : 'value'})
        data_d = data_d.append(y_data)

    data = data_d

    #swap the order of the columns
    cols = list(data.columns)
    a = cols.index('value')
    a, b = cols.index('value'), cols.index(index_column)
    cols[b], cols[a] = cols[a], cols[b]
    data = data[cols]
    return data_d
```

In [28]:

```
DATA = deconstruct_df(DATA)
print(DATA.count())
DATA.sample(n=20)
```

```
indicator    263424
country      263424
year         263424
value         99268
dtype: int64
```

Out[28]:

	indicator	country	year	value
831688	PRJ.ATT.15UP.4.MF	TUV	2004	NaN
199207	UIS.G.5.B.T	BGR	2000	5103.000000
331099	NY.GDP.MKTP.PP.CD	FJI	2045	NaN
217478	NY.GDP.MKTP.CD	CMR	2030	NaN
122165	UIS.E.3.GPV	ARM	2007	100185.000000
274824	PRJ.POP.2024.4.MF	CUB	2003	NaN
570608	SP.POP.TOTL	MCO	2014	38132.000000
271039	PRJ.ATT.25UP.3.MF	HRV	2014	NaN
576710	SE.TER.GRAD	MNE	2020	NaN
113528	PRJ.ATT.ALL.4.MF	AGO	2000	NaN
290726	SE.SEC.ENRL	DJI	2002	20516.000000
770897	NY.GDP.PCAP.PP.CD	SUR	2020	NaN
377414	PRJ.ATT.ALL.3.MF	GRD	2003	NaN
570407	SP.SEC.UTOT.IN	MCO	2000	NaN
221197	UIS.G.5.B.T	CAN	2006	NaN
696305	PRJ.POP.2024.3.MF	STP	2015	1.910000
520385	PRJ.POP.2024.3.MF	MAC	2003	NaN
543617	SE.SEC.ENRL.GC	MLT	2020	NaN
195492	NY.GDP.PCAP.PP.CD	BRN	2007	79616.870305
540056	UIS.G.7.T	MLI	2007	NaN

In [29]:

```
def reduce_df_d(df_d, indicators, countries, years):
    select = [a and b and c for a, b, c in zip([i for i in df_d.indicator.isin(indic
                                                [i for i in df_d.country.isin(countri
                                                [i for i in df_d.year.isin(years)])]
    return df_d[select]

#test
test = True
if test:
    test_indicators = ['PRJ.POP.ALL.4.MF']
    test_countries = ['FRA','DEU']
    test_years = ['2010','2015','2020']

    print(DATA.info())
    print(reduce_df_d(DATA, test_indicators, test_countries, test_years))
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 263424 entries, 3673 to 886921
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
```

```
 0   indicator  263424 non-null  object
 1   country    263424 non-null  object
 2   year       263424 non-null  object
 3   value      99268 non-null   float64
dtypes: float64(1), object(3)
memory usage: 10.0+ MB
None
               indicator country  year     value
340830  PRJ.POP.ALL.4.MF     FRA  2010  11625.61
359155  PRJ.POP.ALL.4.MF     DEU  2010  19311.89
340830  PRJ.POP.ALL.4.MF     FRA  2015  13539.27
359155  PRJ.POP.ALL.4.MF     DEU  2015  20957.98
340830  PRJ.POP.ALL.4.MF     FRA  2020  15539.72
359155  PRJ.POP.ALL.4.MF     DEU  2020  22515.63
```

In [30]:

```
def reconstruct_df(df_d, index_columns, datacolumn) :
    df_T = df_d.set_index(index_columns).pivot(columns = datacolumn)
    df_T.columns = pd.Index([v[1] for v in df_T.columns], name = datacolumn)
    return df_T.reset_index()

reconstruct_df(DATA,['indicator','country'], 'year').info()
reconstruct_df(DATA,['indicator','year'], 'country').info()
reconstruct_df(DATA,['country','year'], 'indicator').info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10976 entries, 0 to 10975
Data columns (total 26 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   indicator  10976 non-null  object
 1   country    10976 non-null  object
 2   2000       4700 non-null   float64
 3   2001       4744 non-null   float64
 4   2002       4804 non-null   float64
 5   2003       4784 non-null   float64
 6   2004       4903 non-null   float64
 7   2005       4907 non-null   float64
 8   2006       4909 non-null   float64
 9   2007       4983 non-null   float64
 10  2008       5011 non-null   float64
 11  2009       5036 non-null   float64
 12  2010       7074 non-null   float64
 13  2011       5098 non-null   float64
 14  2012       5098 non-null   float64
 15  2013       5325 non-null   float64
 16  2014       5227 non-null   float64
 17  2015       5955 non-null   float64
 18  2016       2766 non-null   float64
 19  2020       1992 non-null   float64
 20  2025       1992 non-null   float64
 21  2030       1992 non-null   float64
 22  2035       1992 non-null   float64
 23  2040       1992 non-null   float64
 24  2045       1992 non-null   float64
 25  2050       1992 non-null   float64
dtypes: float64(24), object(2)
memory usage: 2.2+ MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1176 entries, 0 to 1175
Columns: 226 entries, indicator to ZWE
dtypes: float64(224), object(2)
memory usage: 2.0+ MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5376 entries, 0 to 5375
Data columns (total 51 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   country             5376 non-null   object
```

```
 1   year                 5376 non-null   object
 2   IT.NET.USER.P2       3516 non-null   float64
 3   NY.GDP.MKTP.CD       3532 non-null   float64
 4   NY.GDP.MKTP.PP.CD    3315 non-null   float64
 5   NY.GDP.PCAP.CD       3532 non-null   float64
 6   NY.GDP.PCAP.PP.CD    3315 non-null   float64
 7   NY.GNP.MKTP.CD       3374 non-null   float64
 8   NY.GNP.MKTP.PP.CD    3261 non-null   float64
 9   NY.GNP.PCAP.CD       3329 non-null   float64
 10  NY.GNP.PCAP.PP.CD    3261 non-null   float64
 11  PRJ.ATT.15UP.3.MF    1494 non-null   float64
 12  PRJ.ATT.15UP.4.MF    1494 non-null   float64
 13  PRJ.ATT.2024.3.MF    1494 non-null   float64
 14  PRJ.ATT.2024.4.MF    1494 non-null   float64
 15  PRJ.ATT.25UP.3.MF    1494 non-null   float64
 16  PRJ.ATT.25UP.4.MF    1494 non-null   float64
 17  PRJ.ATT.ALL.3.MF     1494 non-null   float64
 18  PRJ.ATT.ALL.4.MF     1494 non-null   float64
 19  PRJ.POP.2024.3.MF    1494 non-null   float64
 20  PRJ.POP.2024.4.MF    1494 non-null   float64
 21  PRJ.POP.ALL.3.MF     1494 non-null   float64
 22  PRJ.POP.ALL.4.MF     1494 non-null   float64
 23  SE.SEC.ENRL          2500 non-null   float64
 24  SE.SEC.ENRL.GC       2689 non-null   float64
 25  SE.SEC.ENRR.UP       2402 non-null   float64
 26  SE.TER.ENRL          2206 non-null   float64
 27  SE.TER.ENRR          2127 non-null   float64
 28  SE.TER.GRAD          1440 non-null   float64
 29  SP.POP.0014.TO       3406 non-null   float64
 30  SP.POP.1524.TO.UN    2963 non-null   float64
 31  SP.POP.1564.TO       3406 non-null   float64
 32  SP.POP.GROW          3756 non-null   float64
 33  SP.POP.TOTL          3757 non-null   float64
 34  SP.SEC.UTOT.IN       3376 non-null   float64
 35  SP.TER.TOTL.IN       3079 non-null   float64
 36  UIS.E.3              2341 non-null   float64
 37  UIS.E.3.GPV          2497 non-null   float64
 38  UIS.E.5.B            1529 non-null   float64
 39  UIS.E.6              243 non-null    float64
 40  UIS.E.7              226 non-null    float64
 41  UIS.E.8              1397 non-null   float64
 42  UIS.EA.3.AG25T99     664 non-null    float64
 43  UIS.EA.3T6.AG25T99   733 non-null    float64
 44  UIS.EA.5T8.AG25T99   718 non-null    float64
 45  UIS.EA.6T8.AG25T99   190 non-null    float64
 46  UIS.G.5.B.T          1110 non-null   float64
 47  UIS.G.6.T            184 non-null    float64
 48  UIS.G.7.T            166 non-null    float64
 49  UIS.G.8.T            1057 non-null   float64
 50  UIS.NERA.3           743 non-null    float64
dtypes: float64(49), object(2)
memory usage: 2.1+ MB
```

In [31]:

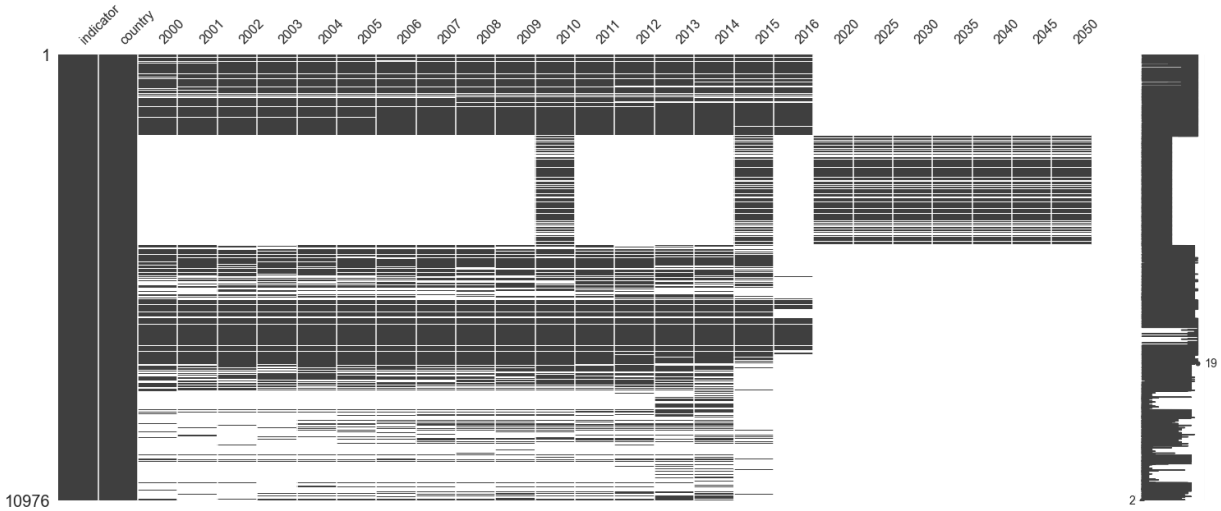
```
#try missingno package that is given as ressource
from missingno import matrix as msno_matrix
from missingno import heatmap as msno_heatmap

for i in range(3) :
    j = (i+1) % 3
    k = (i+2) % 3
    data = reconstruct_df(DATA,[INDEX_NAMES[i],INDEX_NAMES[j]],INDEX_NAMES[k])
    print(data.info())
    print(data.count())
    msno_matrix(data)
    msno_heatmap(data)
```

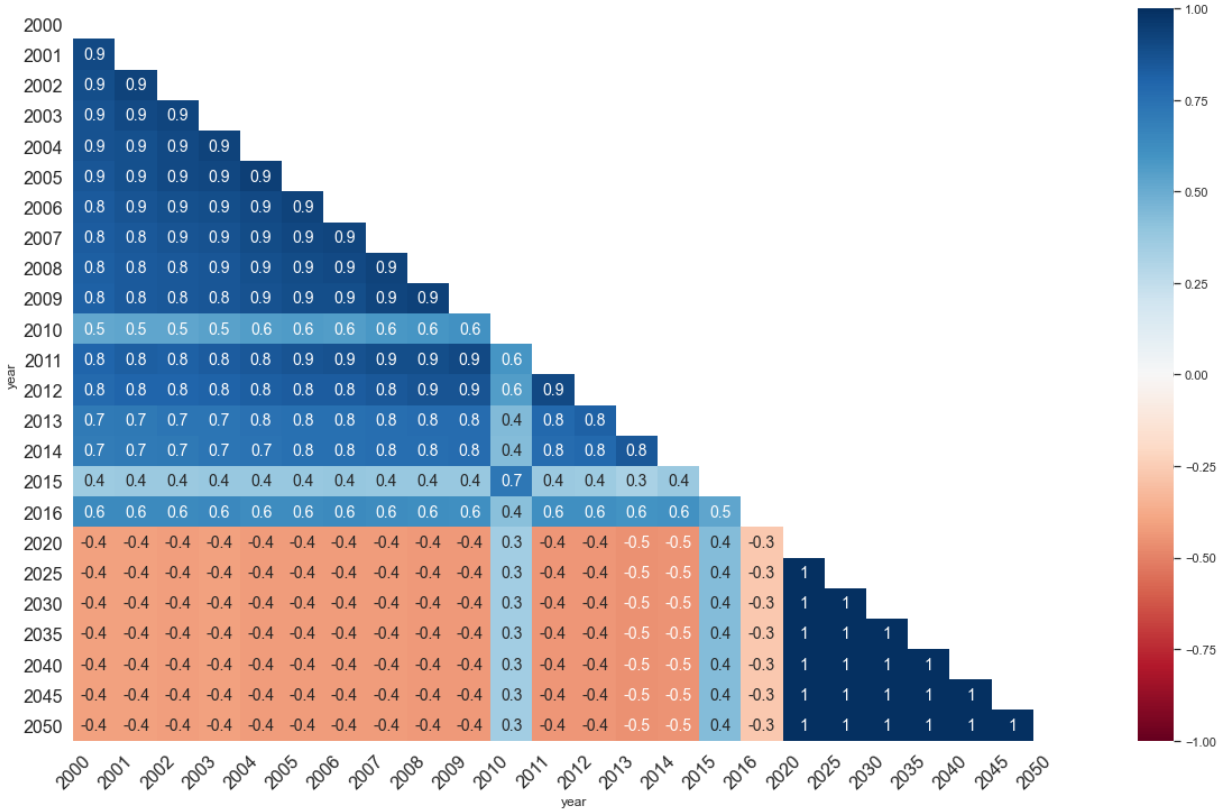
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10976 entries, 0 to 10975
```

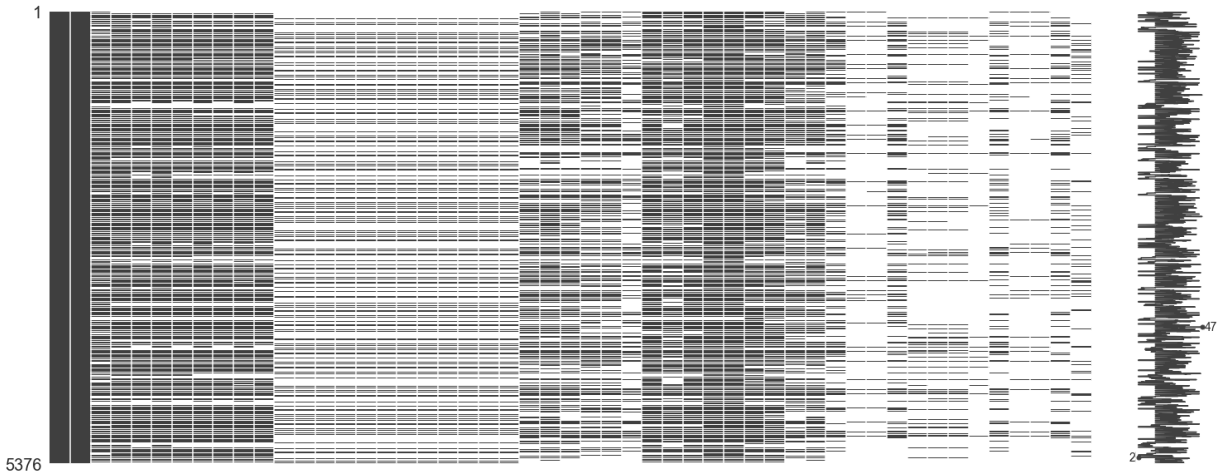
```
Data columns (total 26 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   indicator  10976 non-null  object
 1   country    10976 non-null  object
 2   2000       4700 non-null   float64
 3   2001       4744 non-null   float64
 4   2002       4804 non-null   float64
 5   2003       4784 non-null   float64
 6   2004       4903 non-null   float64
 7   2005       4907 non-null   float64
 8   2006       4909 non-null   float64
 9   2007       4983 non-null   float64
 10  2008       5011 non-null   float64
 11  2009       5036 non-null   float64
 12  2010       7074 non-null   float64
 13  2011       5098 non-null   float64
 14  2012       5098 non-null   float64
 15  2013       5325 non-null   float64
 16  2014       5227 non-null   float64
 17  2015       5955 non-null   float64
 18  2016       2766 non-null   float64
 19  2020       1992 non-null   float64
 20  2025       1992 non-null   float64
 21  2030       1992 non-null   float64
 22  2035       1992 non-null   float64
 23  2040       1992 non-null   float64
 24  2045       1992 non-null   float64
 25  2050       1992 non-null   float64
dtypes: float64(24), object(2)
memory usage: 2.2+ MB
None
year
indicator    10976
country      10976
2000          4700
2001          4744
2002          4804
2003          4784
2004          4903
2005          4907
2006          4909
2007          4983
2008          5011
2009          5036
2010          7074
2011          5098
2012          5098
2013          5325
2014          5227
2015          5955
2016          2766
2020          1992
2025          1992
2030          1992
2035          1992
2040          1992
2045          1992
2050          1992
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5376 entries, 0 to 5375
Data columns (total 51 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   country             5376 non-null   object
 1   year                5376 non-null   object
 2   IT.NET.USER.P2      3516 non-null   float64
 3   NY.GDP.MKTP.CD      3532 non-null   float64
```

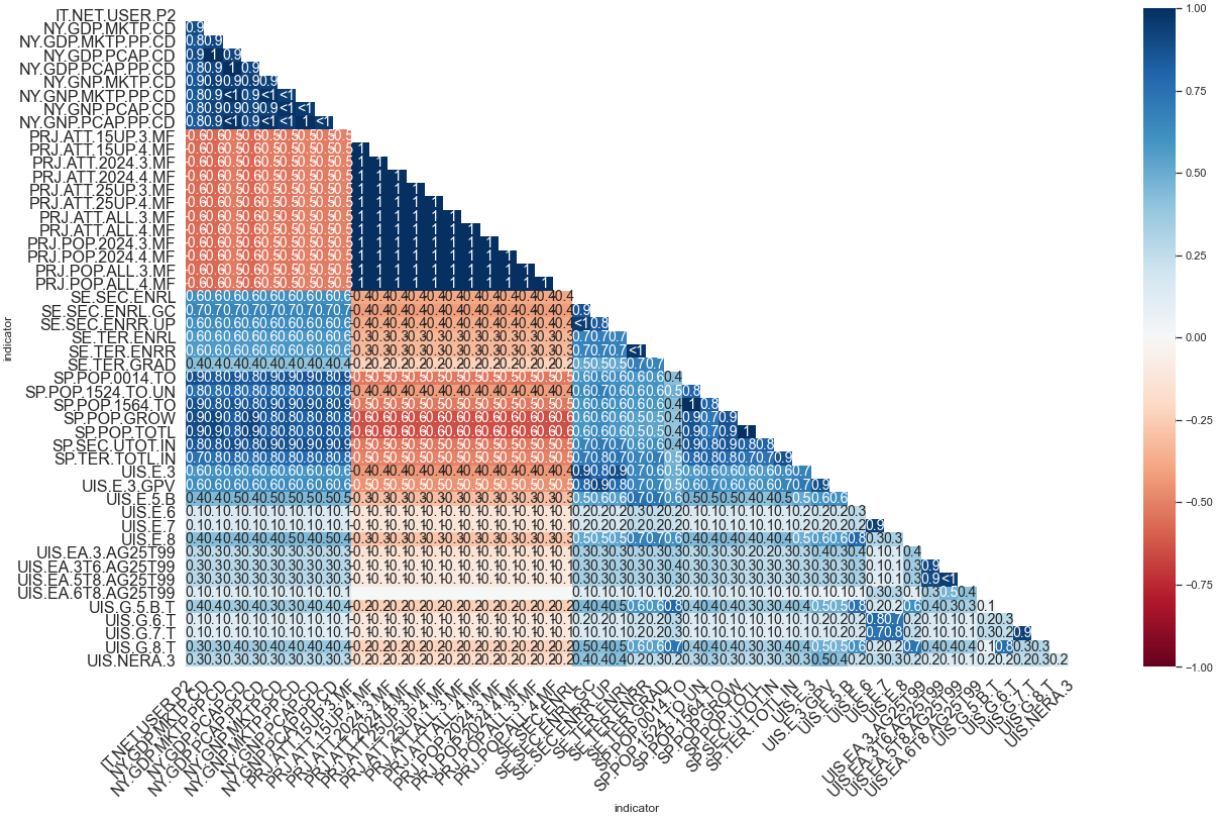
```
 4   NY.GDP.MKTP.PP.CD   3315 non-null   float64
 5   NY.GDP.PCAP.CD      3532 non-null   float64
 6   NY.GDP.PCAP.PP.CD   3315 non-null   float64
 7   NY.GNP.MKTP.CD      3374 non-null   float64
 8   NY.GNP.MKTP.PP.CD   3261 non-null   float64
 9   NY.GNP.PCAP.CD      3329 non-null   float64
 10  NY.GNP.PCAP.PP.CD   3261 non-null   float64
 11  PRJ.ATT.15UP.3.MF   1494 non-null   float64
 12  PRJ.ATT.15UP.4.MF   1494 non-null   float64
 13  PRJ.ATT.2024.3.MF   1494 non-null   float64
 14  PRJ.ATT.2024.4.MF   1494 non-null   float64
 15  PRJ.ATT.25UP.3.MF   1494 non-null   float64
 16  PRJ.ATT.25UP.4.MF   1494 non-null   float64
 17  PRJ.ATT.ALL.3.MF    1494 non-null   float64
 18  PRJ.ATT.ALL.4.MF    1494 non-null   float64
 19  PRJ.POP.2024.3.MF   1494 non-null   float64
 20  PRJ.POP.2024.4.MF   1494 non-null   float64
 21  PRJ.POP.ALL.3.MF    1494 non-null   float64
 22  PRJ.POP.ALL.4.MF    1494 non-null   float64
 23  SE.SEC.ENRL         2500 non-null   float64
 24  SE.SEC.ENRL.GC      2689 non-null   float64
 25  SE.SEC.ENRR.UP      2402 non-null   float64
 26  SE.TER.ENRL         2206 non-null   float64
 27  SE.TER.ENRR         2127 non-null   float64
 28  SE.TER.GRAD         1440 non-null   float64
 29  SP.POP.0014.TO      3406 non-null   float64
 30  SP.POP.1524.TO.UN   2963 non-null   float64
 31  SP.POP.1564.TO      3406 non-null   float64
 32  SP.POP.GROW         3756 non-null   float64
 33  SP.POP.TOTL         3757 non-null   float64
 34  SP.SEC.UTOT.IN      3376 non-null   float64
 35  SP.TER.TOTL.IN      3079 non-null   float64
 36  UIS.E.3             2341 non-null   float64
 37  UIS.E.3.GPV         2497 non-null   float64
 38  UIS.E.5.B           1529 non-null   float64
 39  UIS.E.6             243 non-null    float64
 40  UIS.E.7             226 non-null    float64
 41  UIS.E.8             1397 non-null   float64
 42  UIS.EA.3.AG25T99    664 non-null    float64
 43  UIS.EA.3T6.AG25T99  733 non-null    float64
 44  UIS.EA.5T8.AG25T99  718 non-null    float64
 45  UIS.EA.6T8.AG25T99  190 non-null    float64
 46  UIS.G.5.B.T         1110 non-null   float64
 47  UIS.G.6.T           184 non-null    float64
 48  UIS.G.7.T           166 non-null    float64
 49  UIS.G.8.T           1057 non-null   float64
 50  UIS.NERA.3          743 non-null    float64
dtypes: float64(49), object(2)
memory usage: 2.1+ MB
None
indicator
country               5376
year                  5376
IT.NET.USER.P2        3516
NY.GDP.MKTP.CD        3532
NY.GDP.MKTP.PP.CD     3315
NY.GDP.PCAP.CD        3532
NY.GDP.PCAP.PP.CD     3315
NY.GNP.MKTP.CD        3374
NY.GNP.MKTP.PP.CD     3261
NY.GNP.PCAP.CD        3329
NY.GNP.PCAP.PP.CD     3261
PRJ.ATT.15UP.3.MF     1494
PRJ.ATT.15UP.4.MF     1494
PRJ.ATT.2024.3.MF     1494
PRJ.ATT.2024.4.MF     1494
PRJ.ATT.25UP.3.MF     1494
PRJ.ATT.25UP.4.MF     1494
PRJ.ATT.ALL.3.MF      1494
```

```
PRJ.ATT.ALL.4.MF      1494
PRJ.POP.2024.3.MF     1494
PRJ.POP.2024.4.MF     1494
PRJ.POP.ALL.3.MF      1494
PRJ.POP.ALL.4.MF      1494
SE.SEC.ENRL           2500
SE.SEC.ENRL.GC        2689
SE.SEC.ENRR.UP        2402
SE.TER.ENRL           2206
SE.TER.ENRR           2127
SE.TER.GRAD           1440
SP.POP.0014.TO        3406
SP.POP.1524.TO.UN     2963
SP.POP.1564.TO        3406
SP.POP.GROW           3756
SP.POP.TOTL           3757
SP.SEC.UTOT.IN        3376
SP.TER.TOTL.IN        3079
UIS.E.3               2341
UIS.E.3.GPV           2497
UIS.E.5.B             1529
UIS.E.6                243
UIS.E.7                226
UIS.E.8               1397
UIS.EA.3.AG25T99       664
UIS.EA.3T6.AG25T99     733
UIS.EA.5T8.AG25T99     718
UIS.EA.6T8.AG25T99     190
UIS.G.5.B.T           1110
UIS.G.6.T              184
UIS.G.7.T              166
UIS.G.8.T             1057
UIS.NERA.3             743
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1176 entries, 0 to 1175
Columns: 226 entries, year to ZWE
dtypes: float64(224), object(2)
memory usage: 2.0+ MB
None
country
year         1176
indicator    1176
ABW           436
AFG           344
AGO           334
             ...
XKX           146
YEM           341
ZAF           519
ZMB           406
ZWE           443
Length: 226, dtype: int64
```

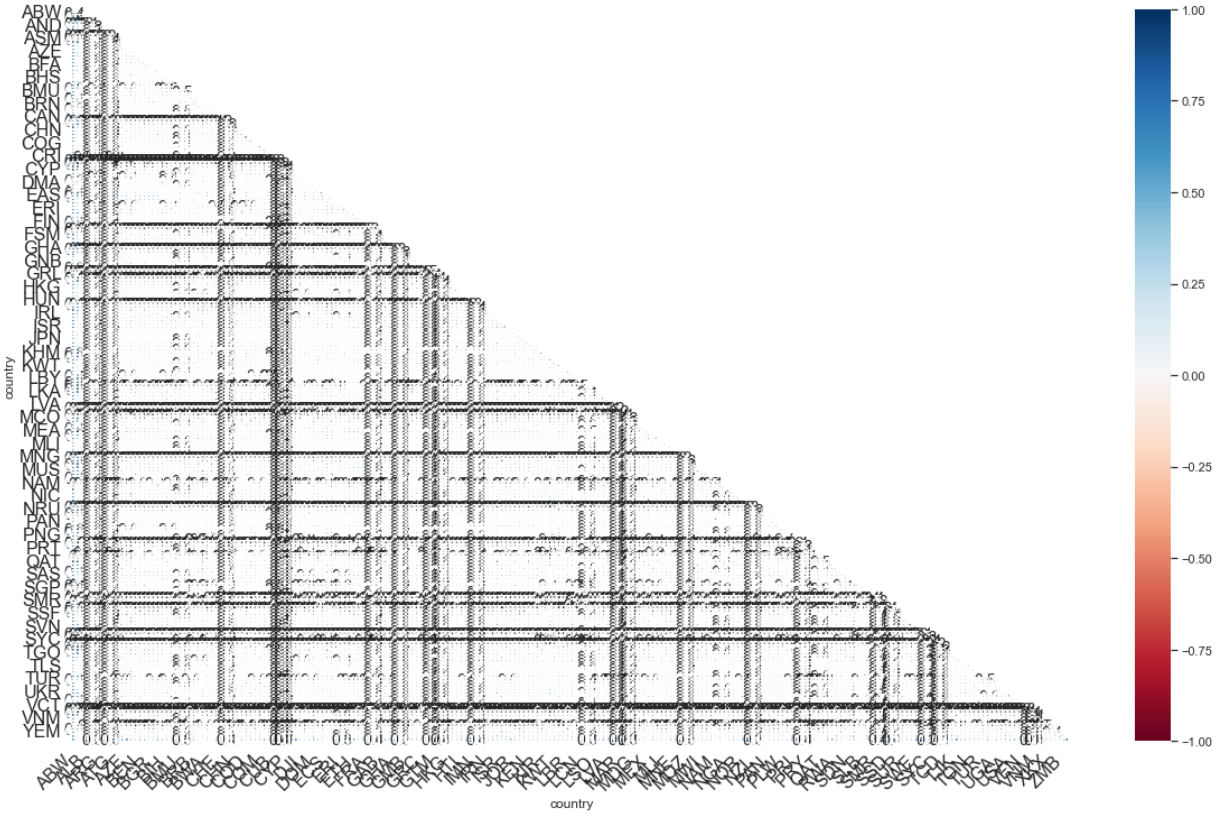

indicator
country
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2020 2025 2030 2035 2040 2045 2050
1
10976
19
2


year
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2020 2025 2030 2035 2040 2045 2050
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00


1
5376
47
2


indicator
indicator
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

1
1176
224
2



Résumé: dates

- La plus récente data vient de 2016, au delà la data est des projections.
- Les années plus proches sont correlées.
- Les dates de projection est anti-correlée avec les dates passées.
  - La data de projection est faite pour les intervales de 5 ans à partir de 2010.
  - La data de projection vient d'un dataset / un projection.

Résumé : variables

- Certaines variables appartiennent au même dataset ou sont measurées par la même organisation (proche de 100% correlation)
  - e.g. les projections: PRJ.
  - e.g. les GDP et GNP: NY.
  - e.g. la population : SP.
- Les projections sont anti-correlées avec les autres data. Il y a de la data pour differents dates. Des dates "futures" pour les projections et les dates "passées" pour la data actuelle.

4 Analyse des données

- Regarder les differents variables individuellement
  - Ecater des pays pour lesquels on n'a pas assez des infos / variables pour faire une analyse
  - Essayer à estimer les valeurs des variables manquantes
  - Regarder la moyenne/médiane/écart-type
- Définir un score pour évaluer l'attractivité d'un pays

4.1 Fonctions pour analyser les données

In [32]:

```
def make_2D_map(df_d, vv, indexcolumns, datacolumn):
    df = reconstruct_df(df_d,indexcolumns, datacolumn)
    return df[df[indexcolumns[0]] == vv]

ind = 'IT.NET.USER.P2'
make_2D_map(DATA, ind, ['indicator', 'year'] , 'country')
```

Out[32]:

country	indicator	year	ABW	AFG	AGO	ALB	AND	ARE	
0	IT.NET.USER.P2	2000	15.442823	NaN	0.105046	0.114097	10.538836	23.625301	7.
1	IT.NET.USER.P2	2001	17.100000	0.004723	0.136014	0.325798	NaN	26.271754	9.
2	IT.NET.USER.P2	2002	18.800000	0.004561	0.270377	0.390081	11.260469	28.316485	10.
3	IT.NET.USER.P2	2003	20.800000	0.087891	0.370682	0.971900	13.546413	29.477953	11.
4	IT.NET.USER.P2	2004	23.000000	0.105809	0.464815	2.420388	26.837954	30.131296	16.
5	IT.NET.USER.P2	2005	25.400000	1.224148	1.143367	6.043891	37.605766	40.000000	17.
6	IT.NET.USER.P2	2006	28.000000	2.107124	1.500000	9.609991	48.936847	52.000000	20.
7	IT.NET.USER.P2	2007	30.900000	1.900000	1.700000	15.036115	70.870000	61.000000	25.
8	IT.NET.USER.P2	2008	52.000000	1.840000	1.900000	23.860000	70.040000	63.000000	28.
9	IT.NET.USER.P2	2009	58.000000	3.550000	2.300000	41.200000	78.530000	64.000000	34.
10	IT.NET.USER.P2	2010	62.000000	4.000000	2.800000	45.000000	81.000000	68.000000	45.
11	IT.NET.USER.P2	2011	69.000000	5.000000	3.100000	49.000000	81.000000	78.000000	51.
12	IT.NET.USER.P2	2012	74.000000	5.454545	6.500000	54.655959	86.434425	84.999991	55.
13	IT.NET.USER.P2	2013	78.900000	5.900000	8.900000	57.200000	94.000000	88.000000	59.
14	IT.NET.USER.P2	2014	83.780000	7.000000	10.200000	60.100000	95.900000	90.400000	64.
15	IT.NET.USER.P2	2015	88.661227	8.260000	12.400000	63.252933	96.910000	90.500000	68.
16	IT.NET.USER.P2	2016	93.542454	10.595726	13.000000	66.363445	97.930637	90.600007	70.
17	IT.NET.USER.P2	2020	NaN	NaN	NaN	NaN	NaN	NaN	
18	IT.NET.USER.P2	2025	NaN	NaN	NaN	NaN	NaN	NaN	
19	IT.NET.USER.P2	2030	NaN	NaN	NaN	NaN	NaN	NaN	
20	IT.NET.USER.P2	2035	NaN	NaN	NaN	NaN	NaN	NaN	
21	IT.NET.USER.P2	2040	NaN	NaN	NaN	NaN	NaN	NaN	
22	IT.NET.USER.P2	2045	NaN	NaN	NaN	NaN	NaN	NaN	
23	IT.NET.USER.P2	2050	NaN	NaN	NaN	NaN	NaN	NaN	

24 rows × 226 columns

In [33]:

```
def make_1D_slice(df_d, valuelist, indexcolumns, datacolumn) :
    df = reconstruct_df(df_d,indexcolumns, datacolumn)
    select = [a and b for a, b in zip([a for a in df[indexcolumns[0]] == valuelist[0
                                       [a for a in df[indexcolumns[1]] == valuelist[1
    return df[select].set_index(indexcolumns).T

ind = 'IT.NET.USER.P2'
```

```
cc = 'FRA'
make_1D_slice(DATA, [ind, cc], ['indicator','country'],'year')
```

Out[33]:

indicator	IT.NET.USER.P2
country	FRA
year	
2000	14.307924
2001	26.325904
2002	30.180000
2003	36.140000
2004	39.150000
2005	42.870000
2006	46.870000
2007	66.090000
2008	70.680000
2009	71.580000
2010	77.280000
2011	77.819999
2012	81.440000
2013	81.919800
2014	83.750000
2015	84.694500
2016	85.622200
2020	NaN
2025	NaN
2030	NaN
2035	NaN
2040	NaN
2045	NaN
2050	NaN

In [34]:

```
def investigate_indicator(ind, df_d, is_rawdata = True, output = True) :
    '''return the 2D data without empty rows and columns and return the years and co

    if is_rawdata:
        data = make_2D_map(df_d, ind, ['indicator', 'year'], 'country').drop(columns
    else :
        data = df_d
    #drop columns / countries that have no data
    data = data.dropna(axis = 1, how='all').dropna(axis = 0, how='all')

    dy = data.index.values
    dc = data.columns.values
```

```
    if output:
        print(50*'*')
        print(ind)
        try:
            data.loc[:,np.isin(data.columns.values, COUNTRIES)].plot(legend = False)
            plt.show()
            data.loc[:,np.isin(data.columns.values, REGIONS)].plot(legend = False)
            plt.show()
        except:
            print("No data to plot.")
        print(f"We have data between {min(dy)} and {max(dy)}.\nThe exact years are :
        print()
        print(f"We have data for the {len(dc)} countries. \nThe countries are :\n{dc
        print()
        print(f"These countries have absolutely no data : {set(COUNTRIES)-set(dc)}")
        print(f"These regions have absolutely no data : {set(REGIONS)-set(dc)}")
        print(50*'*')

    return (data, dy, dc)
```

In [35]:

```
def initial_investigation(ind, is_projection_data = False, output = True, last = 3):

    data, dy, dc = investigate_indicator(ind, DATA, output=output)
    y1 = 2010
    y2 = 2016 if not is_projection_data else 2050
    years = [str(y) for y in range(y1,y2+1) if np.isin(str(y),data.index.values)]
    data2, dy2, dc2 = investigate_indicator(ind, data.loc[years, :], is_rawdata = Fa
    print(f"Completeness for the years {min(years)}-{max(years)}:", data2.count().su
    print()
    print(data2.T.describe())
    #if not is_projection_data :
     #   for y in years[-last:]:
      #      print(f"Number of countries for the year {y}:", data2.loc[str(y), :].co
    return (data2, dy2, dc2)
```

In [36]:

```
def lin_interpolate(xdata, ydata, x3):
    x1, x2 = xdata
    y1, y2 = ydata
    return (x3- x1)*(y2-y1)/(x2-x1) + y1
```

In [37]:

```
def check_nearest_neighbors(xdata, ydata, na_list):
    value_list = [np.nan]*len(na_list)
    for i, na in enumerate(na_list) :
        if na == xdata[0] :
            x1,x2 = xdata[1:3]
            #print(f"case 1 : {na}")
            #print(x1,x2)
        elif na == xdata[-1] :
            x1,x2 = xdata[-3:-1]
            #print(f"case 2 : {na}")
            #print(x1,x2)
        else :
            k = xdata.index(na)
            x1,x2 = xdata[k-1], xdata[k+1]
            #print(f"case 3 : {na}")
            #print(x1,x2)
        if x1 not in na_list and x2 not in na_list:
            y1 = ydata[xdata.index(x1)]
            y2 = ydata[xdata.index(x2)]
```

```
            value_list[i] = lin_interpolate((x1,x2),(y1,y2),na)
    return value_list
```

In [38]:

```
def drop_countries(data,last = 3):
    drop_country = []
    for c in data.columns.values :
        data_slice = data[c]
        na_list = [int(s) for s in data_slice.loc[data_slice.isna()].index.values]
        xrange = [int(y) for y in data.index.values]
        if len(na_list) > len(xrange)/2 and not len(set(xrange[-last:])-set(na_list)
            drop_country.append(c)
            data = data.drop(columns=[c])
    print("The following countries were dropped :")
    print(drop_country)
    return data
```

In [39]:

```
def fill_indicator_simple(data, ind='Indicator', output=False):
    '''Fill missing data points with a linear fit if they have two adjacent data poi
    The end points are lineraly extrapolated if there are two following / leading da

    #collect data for filling in holes in a DataFrame
    fill_data = data.copy()

    #iterate over all countries and keep all years
    xrange = [int(y) for y in data.index.values]
    for c in data.columns.values :
        data_slice = data[c] #note data_slice is a Series object and not longer a Da
        data_slice_clean = data_slice.dropna()
        na_list = [int(s) for s in data_slice.loc[data_slice.isna()].index.values]
        if len(na_list):
            yrange = data_slice.values
            fill_list = check_nearest_neighbors(xrange, yrange, na_list)
            for i,na in enumerate(na_list):
                fill_data[c][str(na)] = fill_list[i]
            if output:
                print(f"The country {c} has some missing data points.")
                xdata = [int(y) for y in data_slice_clean.index.values]
                ydata = data_slice_clean.values
                plt.plot(xdata, ydata,'o',alpha=0.5, color='b')
                plt.plot(na_list,fill_list,'o',alpha=0.5, color='r')
                plt.xlabel("Year")
                plt.ylabel(ind)
                plt.show()

    return data.fillna(value = fill_data)
```

In [40]:

```
def fill_indicator_linear(data):
    '''Fill the data holes using the dataframe interpolate function and extrapolate.
    If only one data point is given, fill the data set with this value.
    If no data points are given, an error will be raised.'''
    data.index = [int(y) for y in data.index.values]
    data_clean = data.copy()
    for c in data.columns.values :
        data_slice = data[c]
        num_datapoints = data_slice.notna().sum()
        if num_datapoints == 1 :
            data_clean[c] = data_slice.fillna(data_slice[data_slice.notna()].values[
        else :
            data_clean[c] = data_slice.interpolate(method = "slinear", fill_value =
    data_clean.index = [str(y) for y in data_clean.index]
```

```
    data.index = [str(y) for y in data.index.values]
    return data_clean
```

In [41]:

```
def fill_indicator(data, func, p0_func, bounds_func = lambda : (-np.inf,np.inf), ind
    '''Fill the data of the indicator for a given country using a fit to func1.'''
    #collect data for filling in holes in a DataFrame
    fill_data = data.copy()

    #iterate over all countries
    xrange = [int(y) for y in data.index.values]
    for c in data.columns.values :
        data_slice = data[c] #note data_slice is a Series object and not longer a Da
        data_slice_clean = data_slice.dropna()
        na_list = [int(s) for s in data_slice.loc[data_slice.isna()].index.values]
        if len(na_list):
            try :
                xdata = [int(y) for y in data_slice_clean.index.values]
                ydata = data_slice_clean.values
                if output :
                    plt.plot(xdata,ydata,'o',alpha=0.5)
                p, pcov = sp.optimize.curve_fit(func, xdata, ydata, p0 = p0_func(xda
                yrange = [func(x, *p) for x in xrange]
                if output :
                    print(f'For {c} the fit succeeded.')
                    print(p)
                    print(np.sqrt(np.diag(pcov)))
                    plt.plot(xrange,yrange,'-')
                    plt.xlabel("Year")
                    plt.ylabel(ind)
                    plt.show()
                fill_data[c] = yrange
            except :
                print(f'For {c} the fit failed.')
                if output:
                    yrange = [func(x, *p0_func(xdata, ydata)) for x in xrange]
                    plt.plot(xrange,yrange,'-')
            finally :
                plt.show()



    return data.fillna(value = fill_data)
```

In [42]:

```
def plot_boxplot(ind, data_clean, unit = "", years=None, **kwags) :

    for countries, titlename in zip([COUNTRIES, REGIONS], ["Pays","Régions"]) :

        if years == None:
            data_plot = data_clean.loc[:,np.isin(data_clean.columns.values, countrie
        else:
            data_plot = data_clean.loc[years,np.isin(data_clean.columns.values, coun

        fig1, ax1 = plt.subplots()
        data_years = [int(y) for y in data_plot.columns.values]
        ax1.boxplot(data_plot, positions = data_years, **kwags)
        #ax1.plot(data_years, data_plot.mean().values,'o', alpha=0.5, color = 'black
        ax1.set_xlabel("Années")
        ax1.set_ylabel(unit)
        title = " - ".join((ind, titlename))
        ax1.set_title(title)
        plt.show()
```

In [43]:
```
def print_info_year(y, data, unit="", ind = 'Indicator', **kwags):
    print(y)

    for countries, titlename in zip([COUNTRIES, REGIONS], ["Pays","Régions"]) :
        data_plot = data.loc[y,np.isin(data.columns.values, countries)].T
        print(titlename)
        print("Median: ", data_plot.median())
        print("Mean: ", data_plot.T.mean())
        print("Std: ", data_plot.std())

        fig, ax = plt.subplots()
        ax.hist(data_plot, **kwags)
            #ax1.plot(data_years, data_plot.mean().values,'o', alpha=0.5, color = 'b
        ax.set_xlabel(unit)
        ax.set_ylabel("Nombre d'occurences")
        title = " - ".join((ind, titlename))
        ax.set_title(title)
        plt.show()
```

In [44]:
```
def list_countrynames(clist):
    for c in clist:
        print(c, COUNTRY_DATA[COUNTRY_DATA['Country Code'] == c]['Short Name'].iloc[
```

4.2 Les variables

4.2.1 Usage de l'internet / ordinateurs

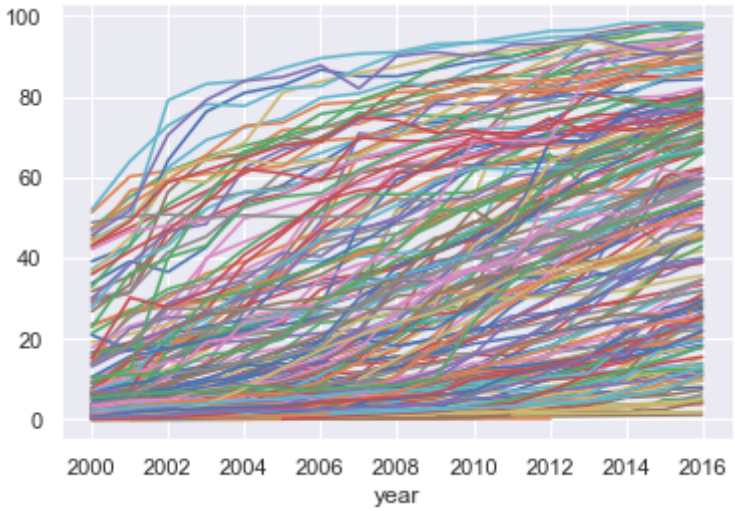
- Il n'y a qu'un indicator dans les données.

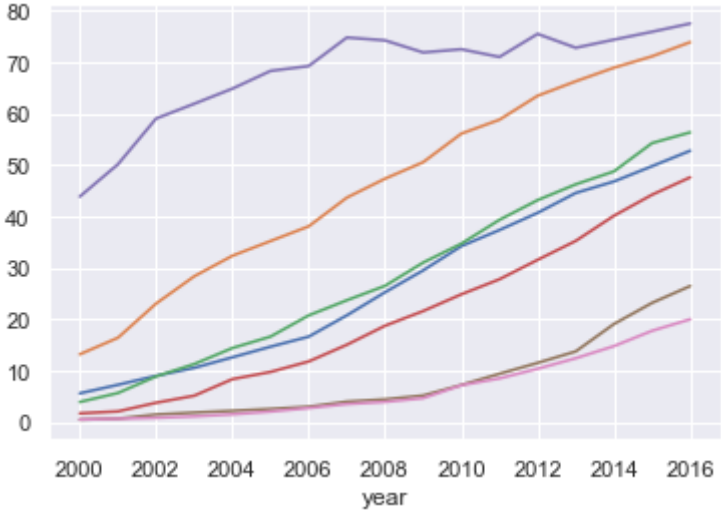
IT.NET.USER.P2

- Utilisateurs de l'Internet (per 100 personnes)
- Pourcentages de la population qui utilise l'internet.
- Un indicateur essentiel pour une service / application web.
- Cet indicateur est presque complèt pour les années 2010-2016.

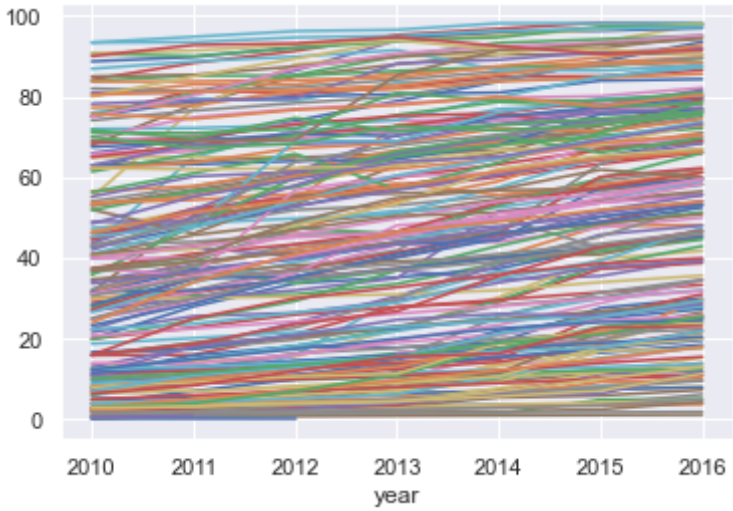
In [45]:
```
ind = 'IT.NET.USER.P2'
initial_investigation(ind)[0].T.info()
```

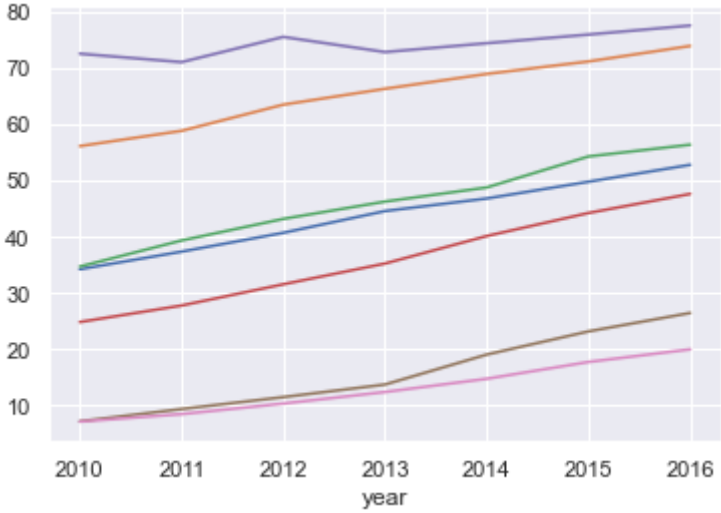
```
**************************************************
IT.NET.USER.P2
```






```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 212 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE'
 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BMU'
 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV'
 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CYM' 'CYP' 'CZE' 'DEU'
 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST'
 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIB' 'GIN'
 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY' 'HKG' 'HND' 'HRV'
 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR'
 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR'
 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO'
 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG'
 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NCL' 'NER' 'NGA' 'NIC' 'NLD'
 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI'
 'PRK' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN'
 'SEN' 'SGP' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSF' 'STP' 'SUR' 'SVK'
 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON'
 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN'
 'VIR' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'VGB', 'CHI', 'XKX', 'SXM', 'MNP', 'ASM',
'IMN', 'MAF', 'SSD', 'NRU', 'TCA', 'CUW'}
These regions have absolutely no data : set()
**************************************************
**************************************************
IT.NET.USER.P2
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 211 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE'
 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BMU'
 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV'
 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CYM' 'CYP' 'CZE' 'DEU'
 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST'
 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIB' 'GIN'
 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY' 'HKG' 'HND' 'HRV'
 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR'
 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR'
 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO'
 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG'
 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NCL' 'NER' 'NGA' 'NIC' 'NLD'
 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PNG' 'POL' 'PRI' 'PRK'
 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN'
 'SGP' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSF' 'STP' 'SUR' 'SVK' 'SVN'
 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO'
 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VIR'
 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'VGB', 'CHI', 'XKX', 'PLW', 'SXM', 'MNP',
'ASM', 'IMN', 'MAF', 'SSD', 'NRU', 'TCA', 'CUW'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.989167230873392

year         2010        2011        2012        2013        2014        2015  \
count  209.000000  211.000000  209.000000  208.000000  208.000000  208.000000
mean    34.245598   36.941067   40.006542   42.813243   45.754260   48.760984
std     27.267731   27.717388   28.398316   28.757671   28.606036   28.304541
min      0.000000    0.000000    0.000000    0.900000    0.990000    1.083733
25%      8.000000   10.035000   12.500000   15.000000   17.745000   21.845365
50%     30.650000   35.070000   38.200000   43.380000   46.217988   50.219659
75%     55.000000   58.312305   63.486253   66.342501   69.835000   72.949675
max     93.390000   94.819687   96.209800   96.546800   98.160000   98.323610

year         2016
count  208.000000
mean    51.389566
std     28.324695
min      1.177119
25%     25.471509
50%     53.613386
75%     76.234824
max     98.240016
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 211 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    209 non-null    float64
 1   2011    211 non-null    float64
 2   2012    209 non-null    float64
 3   2013    208 non-null    float64
 4   2014    208 non-null    float64
 5   2015    208 non-null    float64
 6   2016    208 non-null    float64
dtypes: float64(7)
memory usage: 13.2+ KB
```

Résumé: IT.NET.USER.P2

- Pour 204 pays, il y a de la data pour les années 2010-2016.
  - 13 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est presque complète pour les années 2010-2016. Après avoir écarter les pays, pour lequels il n'y a pas de data, presque 99% des données sont saisies.
- Garder que les pays pour les quels ont a soit plus la moité des données soit l'information d'un des dernières trois années.
- Utiliser d'abord un simple interpolation / extrapolation lineaire pour remplir les trous dans les données.
- Cette variable devrait suivre une courbe S / la fonction Sigmoïde.
  - utiliser cette fonction pour remplirer les données manquantes s'il y a plusieurs années manquantes

In [46]:

```
def SigmoidGrowth(x,b,c,x0):
    return c / (1 + np.exp(-b*(x-x0)))
```

In [47]:

```
def p0_SigmoidGrowth(xdata,ydata):
    return (0.5,max(ydata),2010)
```

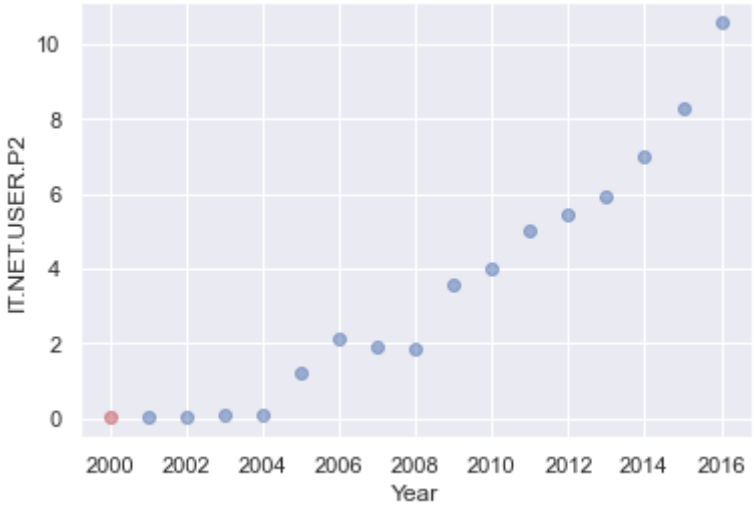
In [48]:

```
def bounds_SigmoidGrowth():
    bounds = ([0,0,1990],[np.inf,100,np.inf])
    return bounds
```

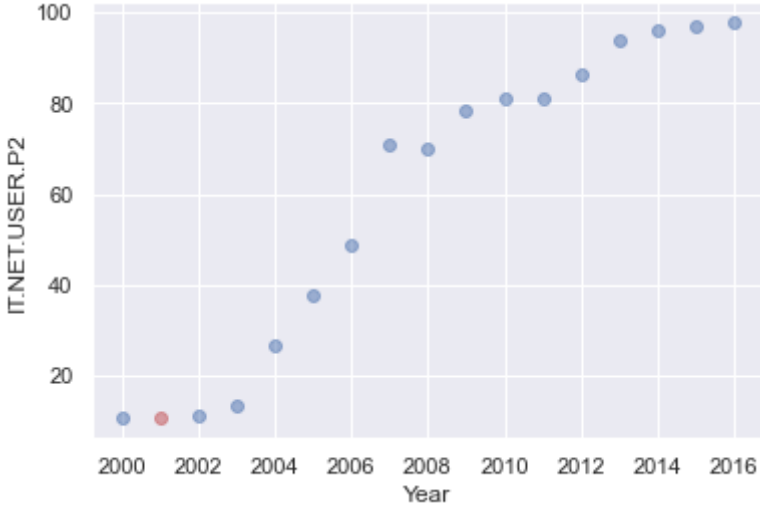
In [49]:

```
#keep this line just to check that every thing works
#print(ALL_YEARS)
ind = 'IT.NET.USER.P2'
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop countries for which there is less than half of the data points and none of the
data = drop_countries(data)
print()
#fill single holes that have nearest neighbors (including endpoints, if the two foll
data_holes = data.loc[: ,data.isna().sum() != 0]
data_holes_filled = fill_indicator_simple(data_holes, ind, output=True)
data_clean = data.fillna(value = data_holes_filled)
```

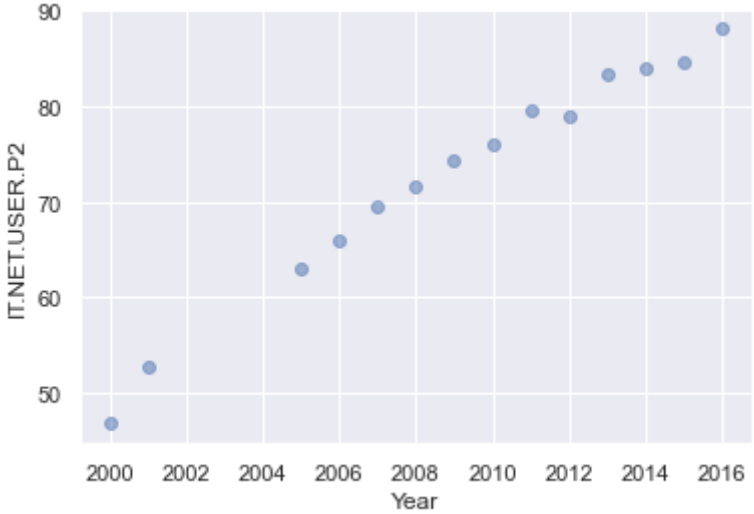
```
The following countries were dropped :
['ASM', 'CHI', 'CUW', 'IMN', 'MAF', 'MNP', 'NRU', 'PLW', 'SSD', 'SXM', 'TCA', 'VGB',
'XKX']

The country AFG has some missing data points.
```



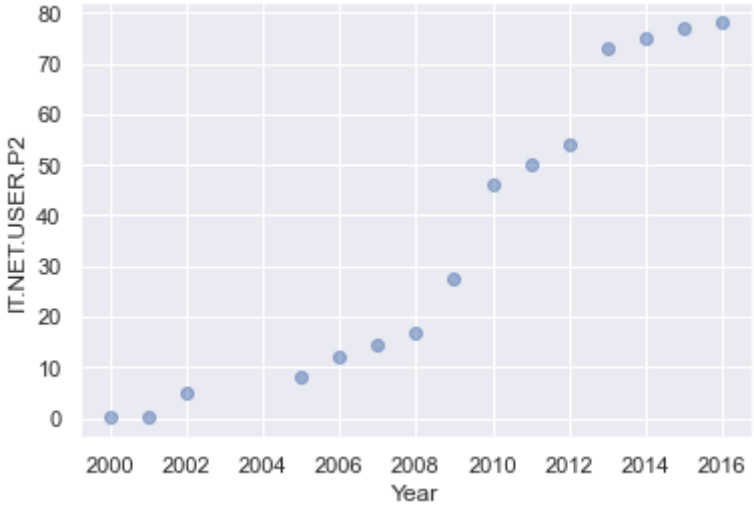

```
The country AND has some missing data points.
```



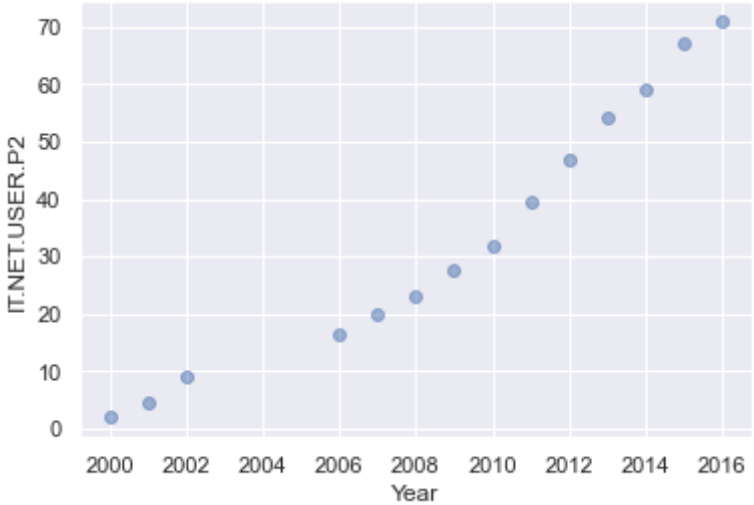

```
The country AUS has some missing data points.
```



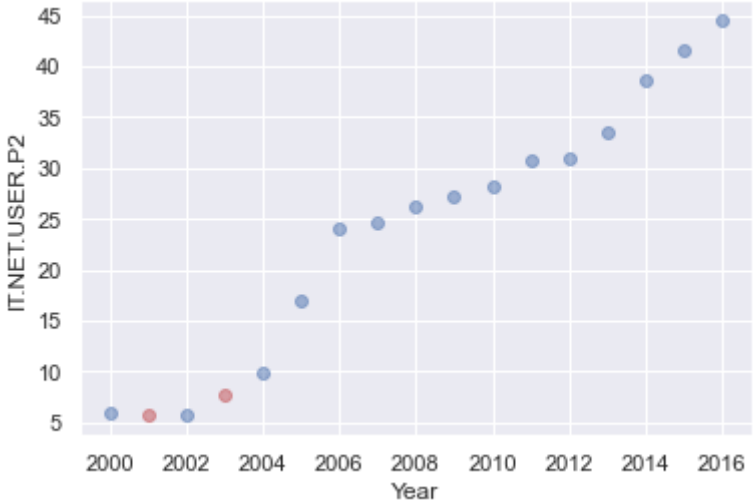

```
The country AZE has some missing data points.
```



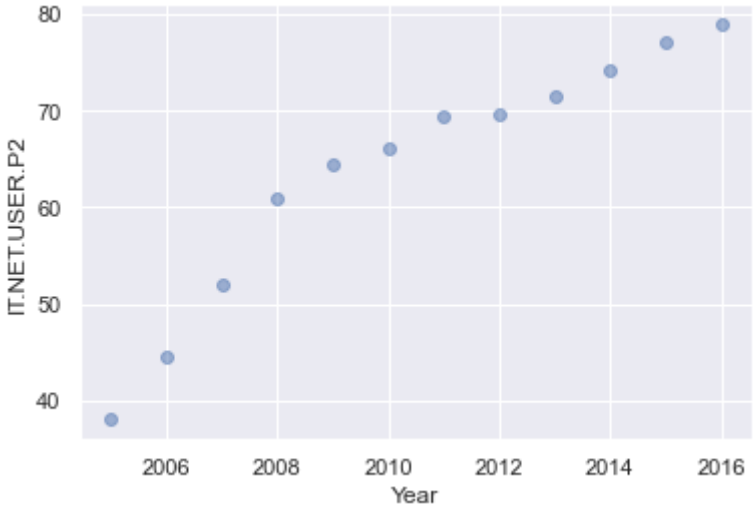

```
The country BLR has some missing data points.
```



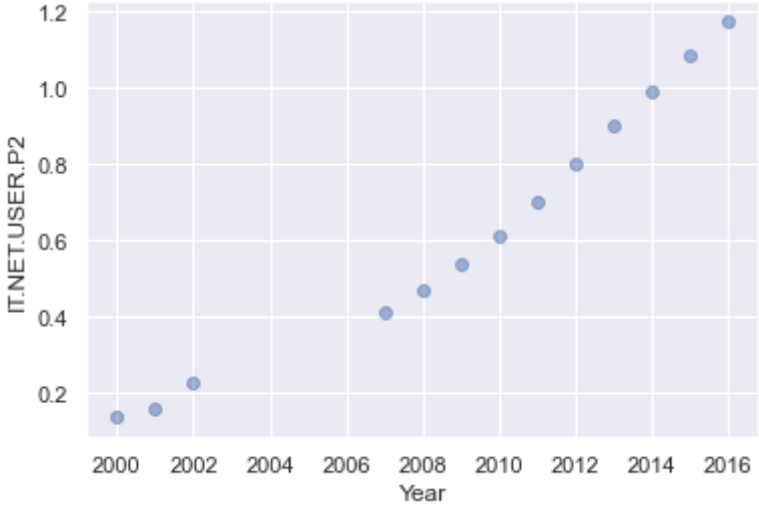

```
The country BLZ has some missing data points.
```



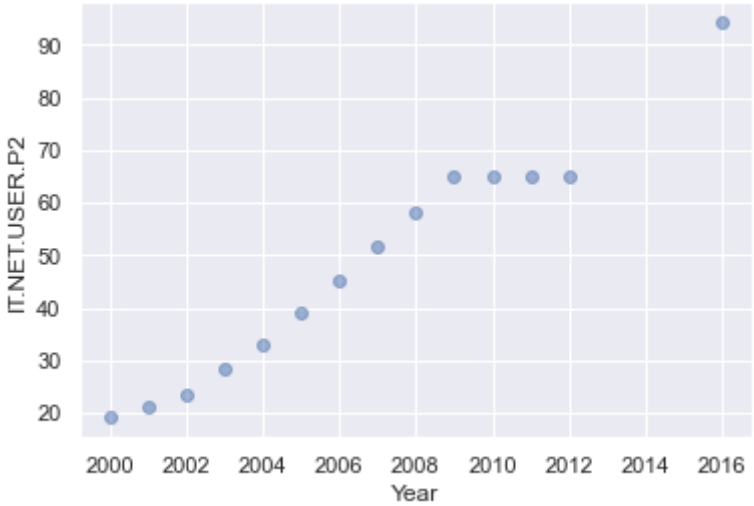

```
The country CYM has some missing data points.
```



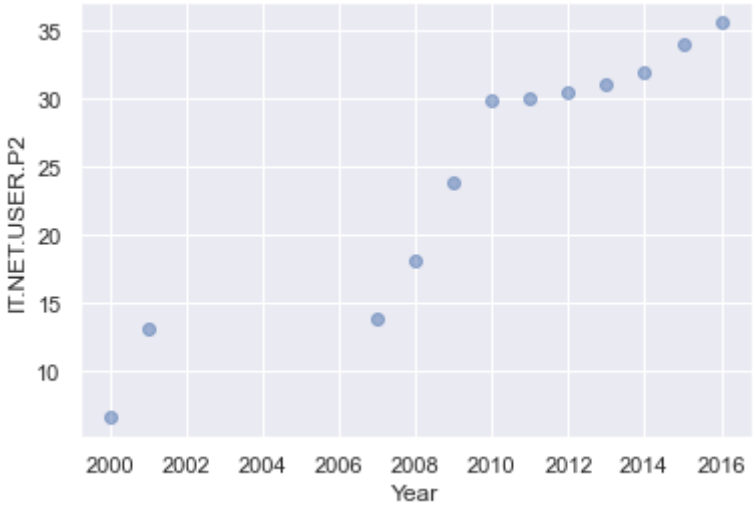

```
The country ERI has some missing data points.
```



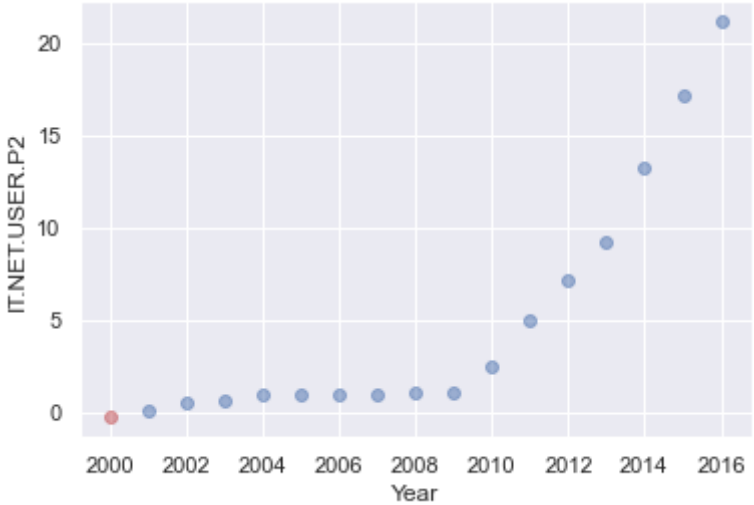

```
The country GIB has some missing data points.
```



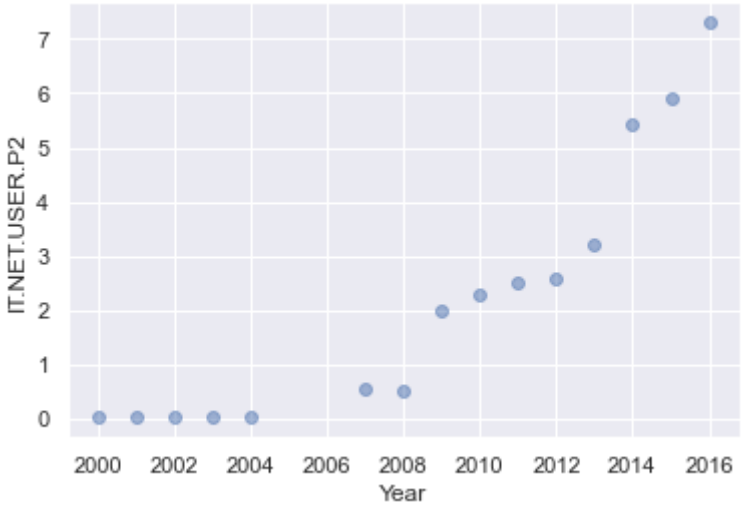

```
The country GUY has some missing data points.
```



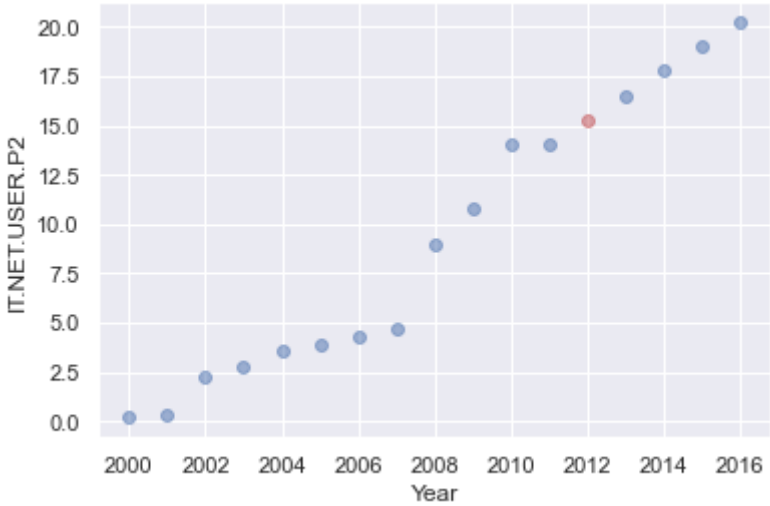

```
The country IRQ has some missing data points.
```



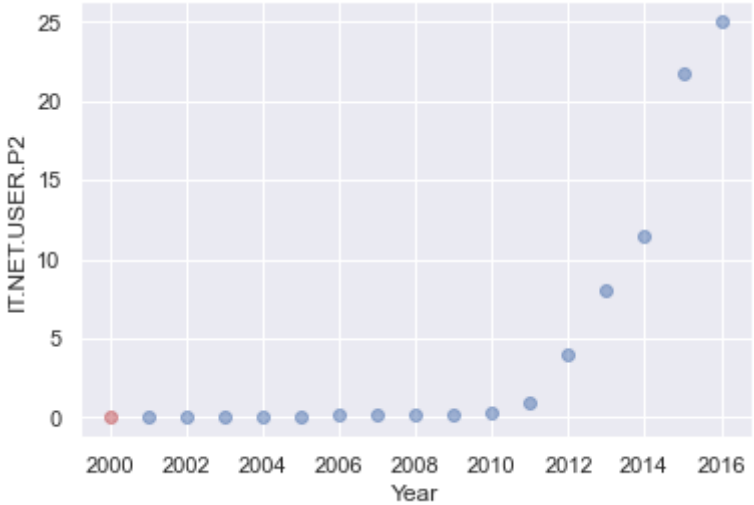

```
The country LBR has some missing data points.
```



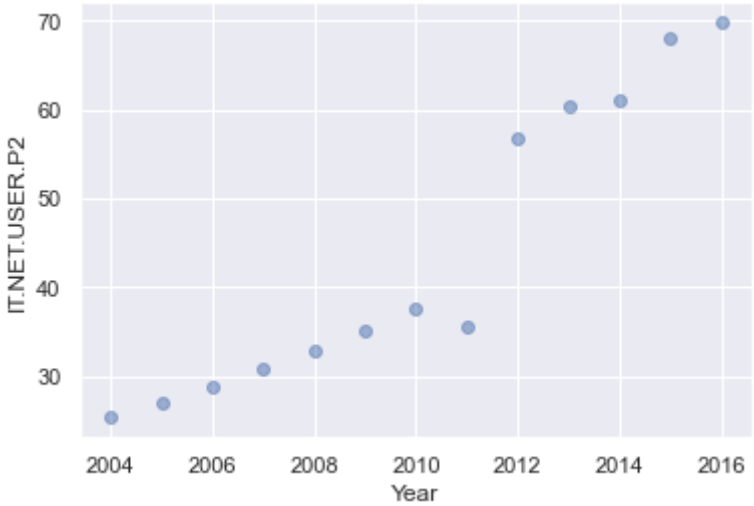

```
The country LBY has some missing data points.
```



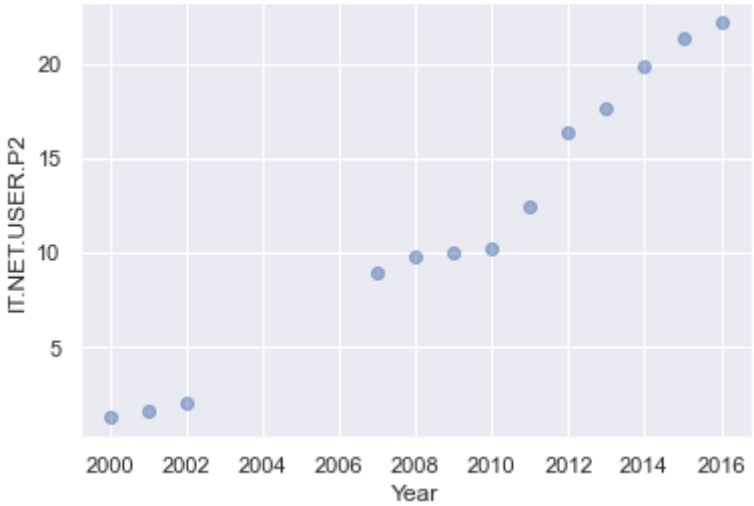

```
The country MMR has some missing data points.
```



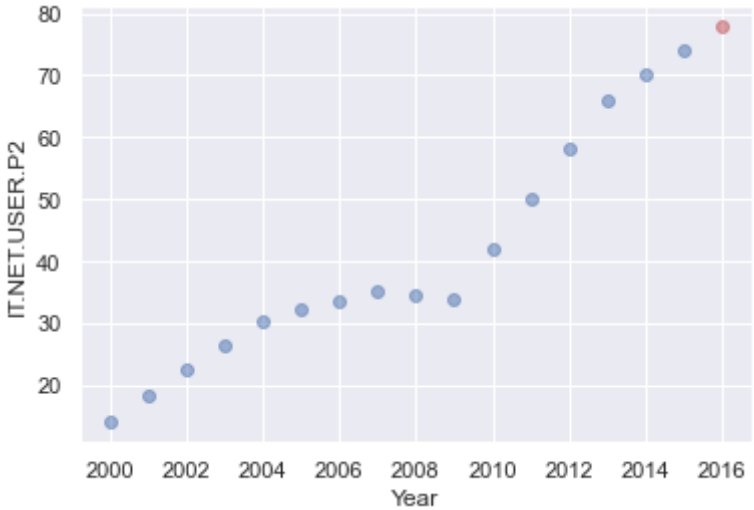

```
The country MNE has some missing data points.
```



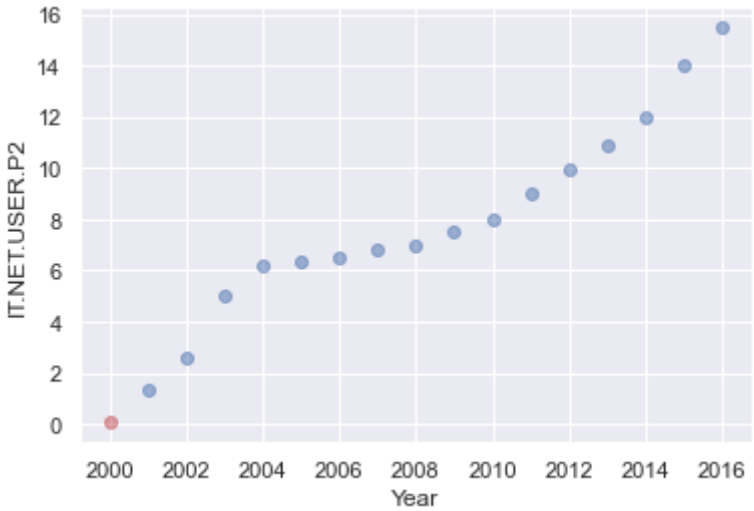

```
The country MNG has some missing data points.
```



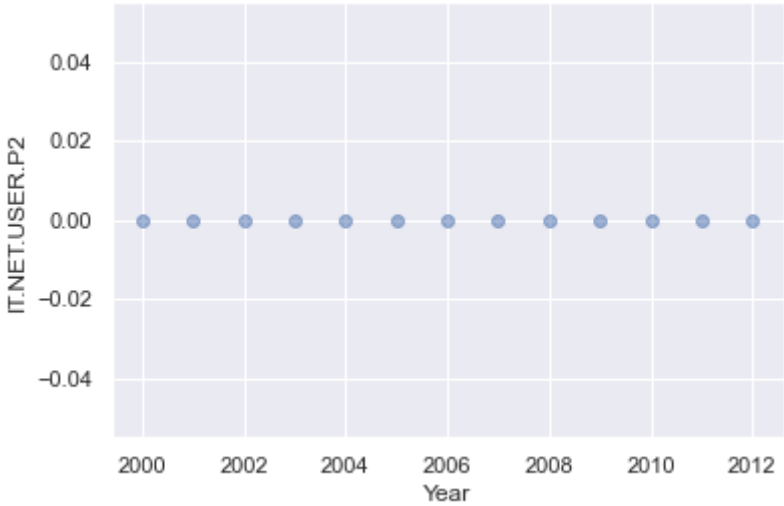

```
The country NCL has some missing data points.
```



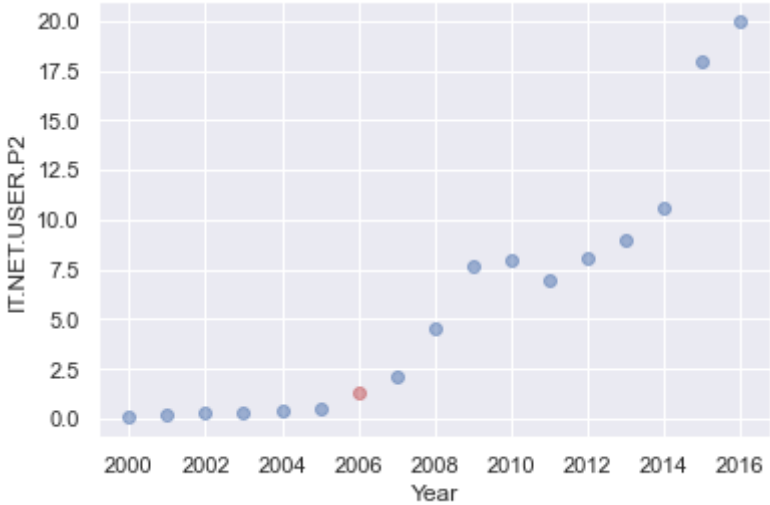

```
The country PAK has some missing data points.
```



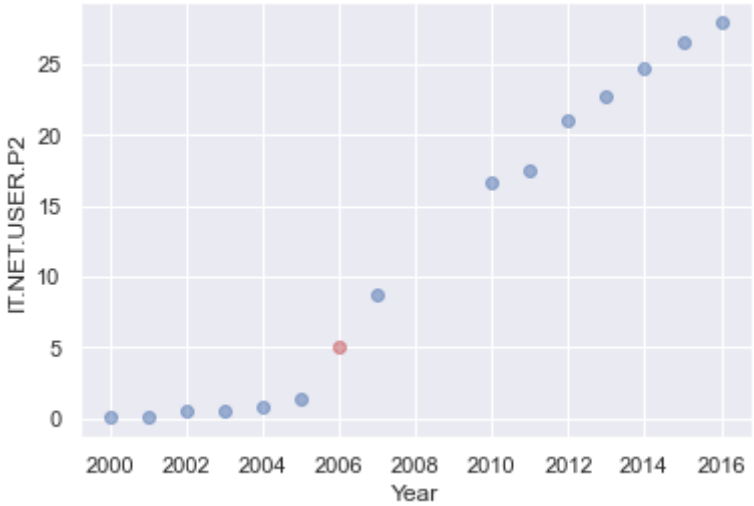

```
The country PRK has some missing data points.
```



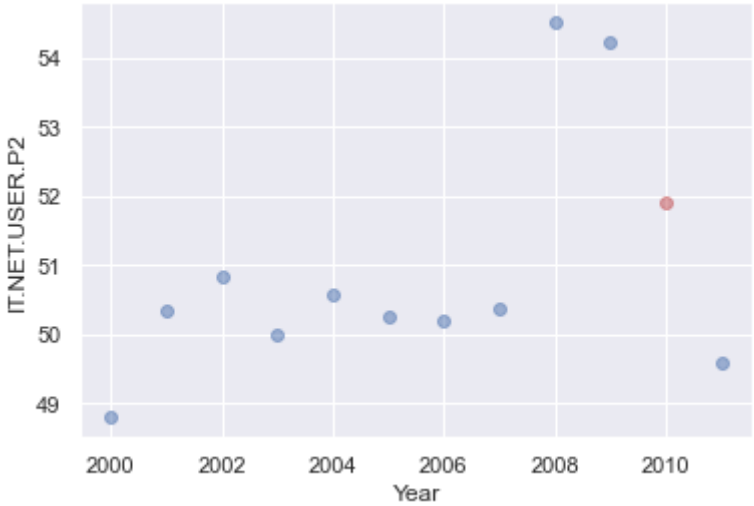

```
The country RWA has some missing data points.
```




```
The country SDN has some missing data points.
```

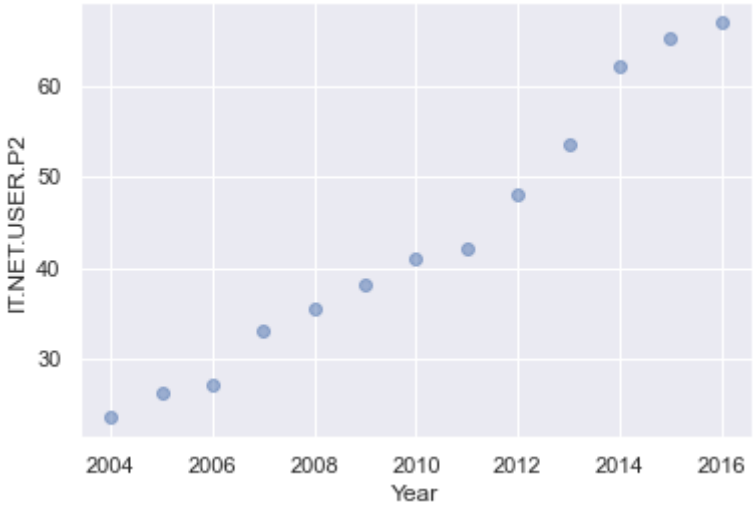



```
The country SMR has some missing data points.
```

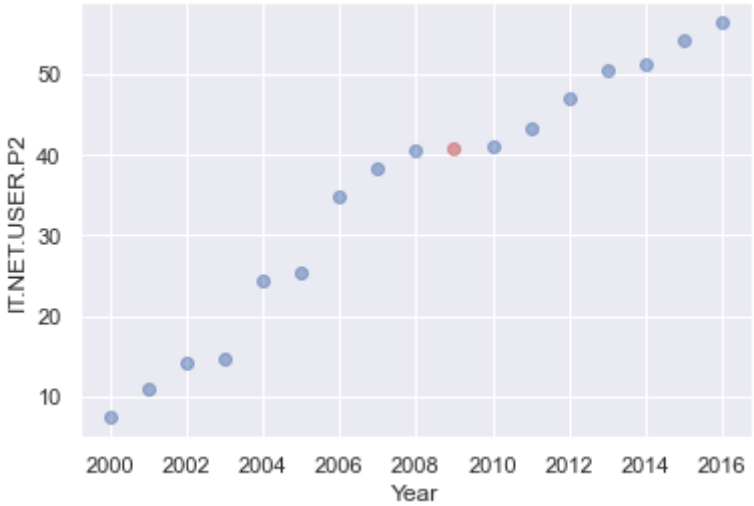



```
The country SOM has some missing data points.
```

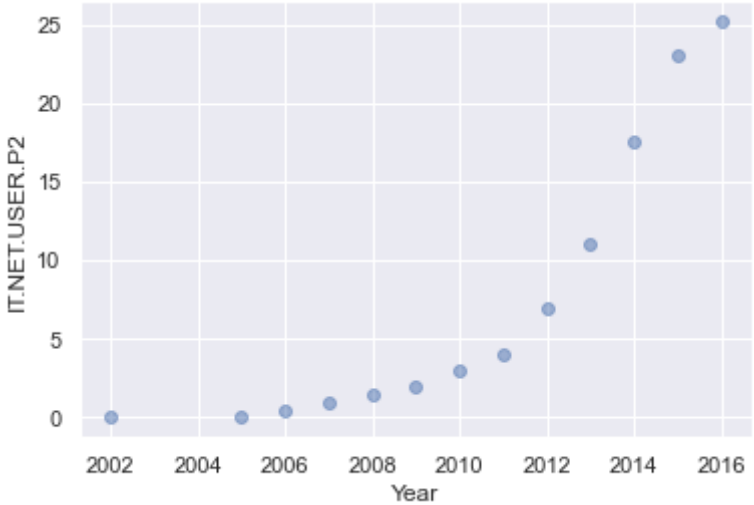
```
The country SRB has some missing data points.
```




```
The country SYC has some missing data points.
```




```
The country TLS has some missing data points.
```




```
The country TUV has some missing data points.
```

In [50]:

```
print(data_clean.T.describe())
```

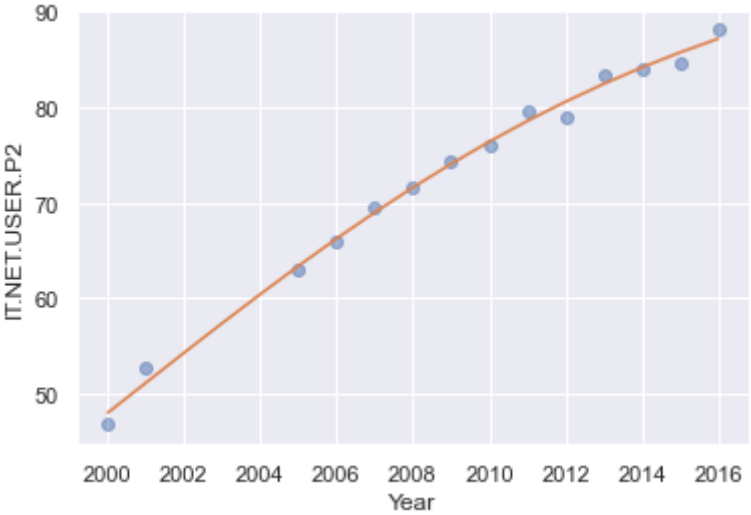
```
year         2000        2001        2002        2003        2004        2005  \
count  207.000000  206.000000  205.000000  200.000000  202.000000  205.000000
mean     8.630607   10.792301   13.706714   16.271223   18.734791   20.958863
std     13.278840   15.491234   18.530363   20.433342   21.852430   23.015649
min     -0.300000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.294900    0.603122    1.050423    1.618643    2.177737    2.851752
50%      2.210692    3.247668    4.626175    6.829578    8.697186   11.484009
75%      9.282820   14.207723   19.370000   25.020945   29.331181   34.862927
max     52.000000   64.000000   79.120000   83.140000   83.890000   87.000000
```

```
year         2006        2007        2008        2009        2010        2011  \
count  206.000000  211.000000  210.000000  210.000000  211.000000  211.000000
mean    23.317732   25.629998   28.433737   31.150500   34.172703   36.941067
std     24.070585   25.217963   26.188301   26.848486   27.260337   27.717388
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      3.715506    4.250000    5.612500    6.982500    8.000000   10.035000
50%     14.513474   16.680000   21.760000   26.445000   30.650000   35.070000
75%     38.005000   41.115000   46.030000   50.760507   54.520000   58.312305
max     89.510000   90.600000   91.000000   93.000000   93.390000   94.819687

year         2012        2013        2014        2015        2016
count  210.000000  208.000000  208.000000  208.000000  209.000000
mean    39.888654   42.813243   45.754260   48.760984   51.516905
std     28.381758   28.757671   28.606036   28.304541   28.316429
min      0.000000    0.900000    0.990000    1.083733    1.177119
25%     12.520015   15.000000   17.745000   21.845365   25.506579
50%     37.850000   43.380000   46.217988   50.219659   54.000000
75%     63.192189   66.342501   69.835000   72.949675   76.409085
max     96.209800   96.546800   98.160000   98.323610   98.240016
```

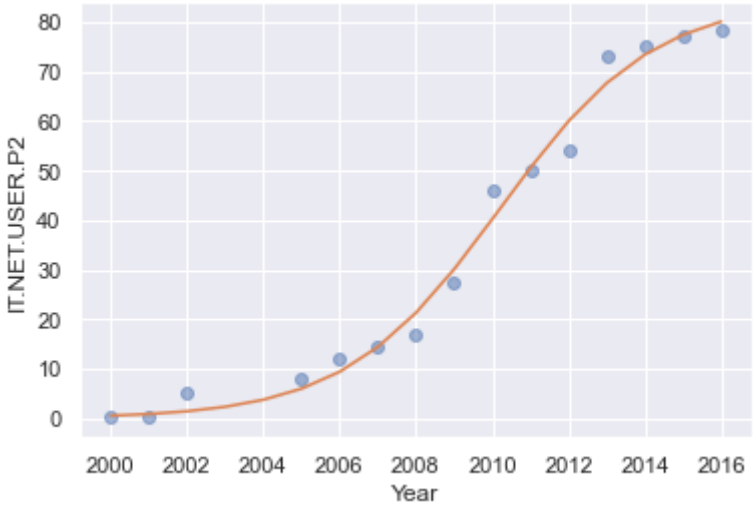
In [51]:

```
# fit the remaining data holes with a S-kurve function
data_holes = data_clean.loc[:,data_clean.isna().sum() != 0]
dc_holes = data_holes.columns.values
data_holes_filled = fill_indicator(data_holes, SigmoidGrowth, p0_SigmoidGrowth,
                                   bounds_func=bounds_SigmoidGrowth, ind = ind, outp

data_clean = data_clean.fillna(value = data_holes_filled)
```

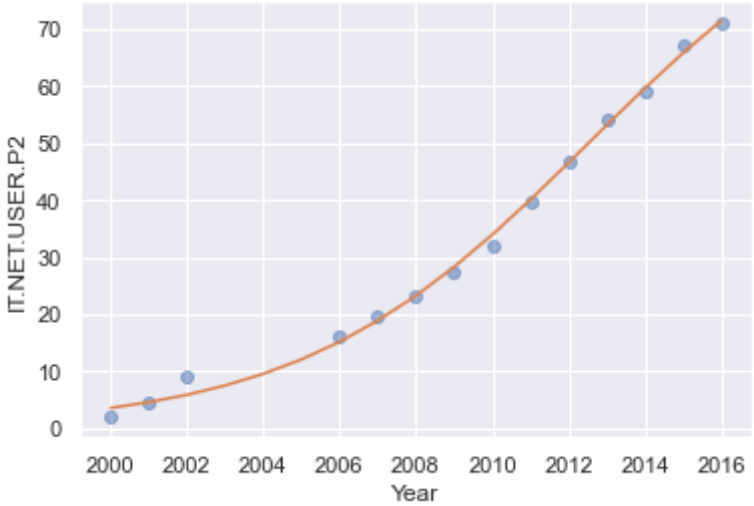
```
For AUS the fit succeeded.
[1.26528311e-01 9.95323672e+01 2.00055856e+03]
[0.01056357 3.15363551 0.44914493]
```



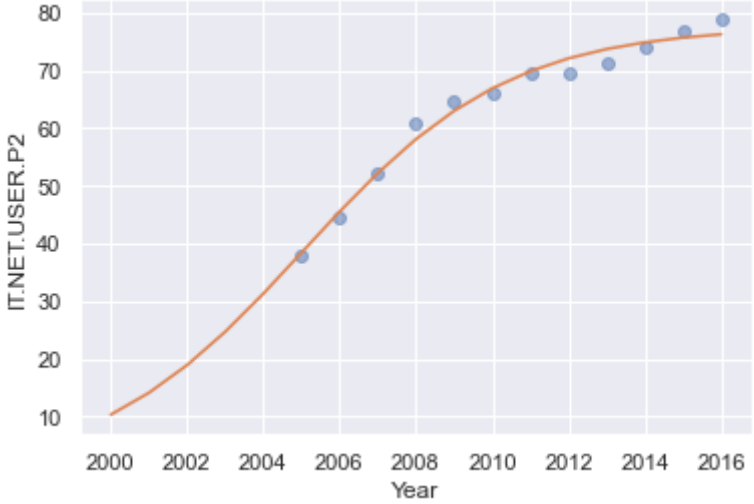

```
For AZE the fit succeeded.
[4.95095628e-01 8.46490091e+01 2.01019828e+03]
[0.05407184 4.17829862 0.31280689]
```



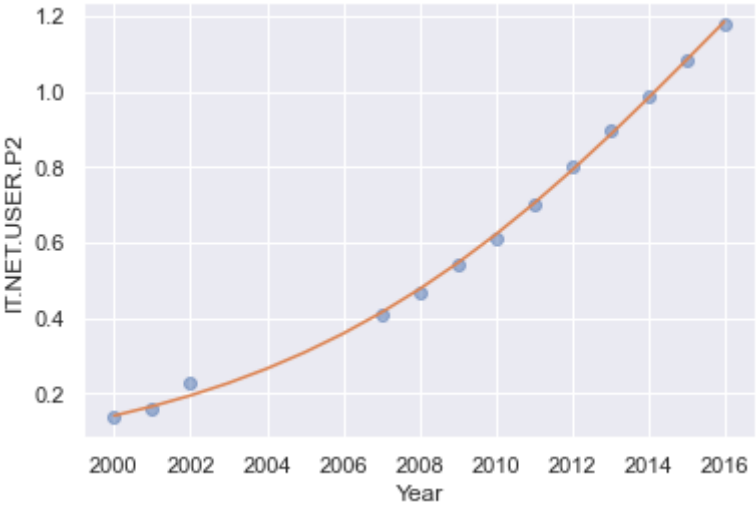

```
For BLR the fit succeeded.
[2.64335666e-01 1.00000000e+02 2.01251990e+03]
[0.01774382 7.67964554 0.6478775 ]
```




```
For CYM the fit succeeded.
[3.70740868e-01 7.75878116e+01 2.00505048e+03]
[0.043156   1.61732441 0.19366594]
```



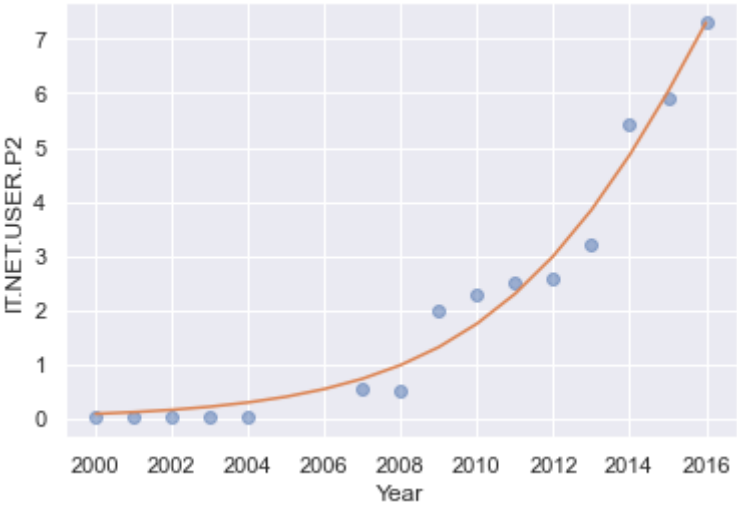

```
For ERI the fit succeeded.
[1.71704198e-01 2.39115957e+00 2.01609123e+03]
[0.00775465 0.23470446 1.0717966 ]
```



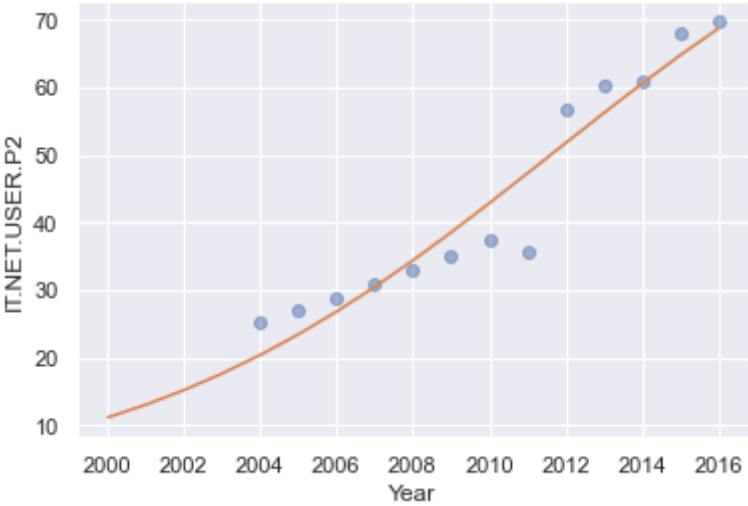

```
For GIB the fit succeeded.
[2.13667063e-01 1.00000000e+02 2.00707434e+03]
[ 0.03162917 10.35338622  1.15860539]
```

```
For GUY the fit succeeded.
[1.81822159e-01 4.57898276e+01 2.00876120e+03]
[ 0.05856755 11.34623258  3.13623344]
```

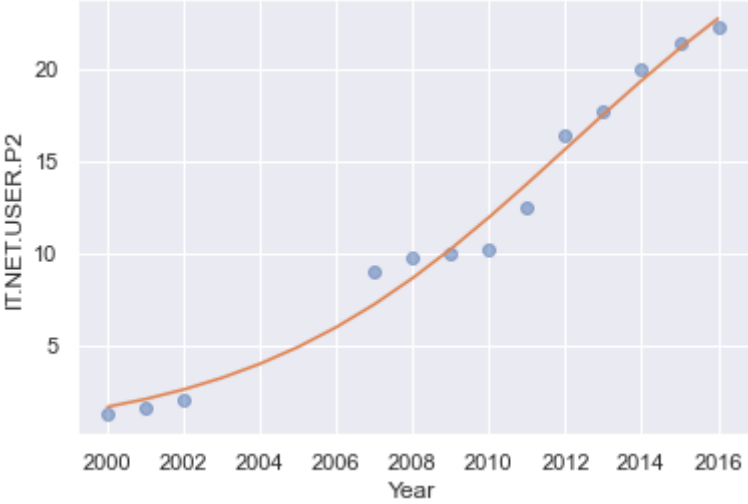
```
For LBR the fit succeeded.
[3.09890134e-01 1.75736788e+01 2.01710345e+03]
[ 0.06775366 11.98519719  3.827272  ]
```



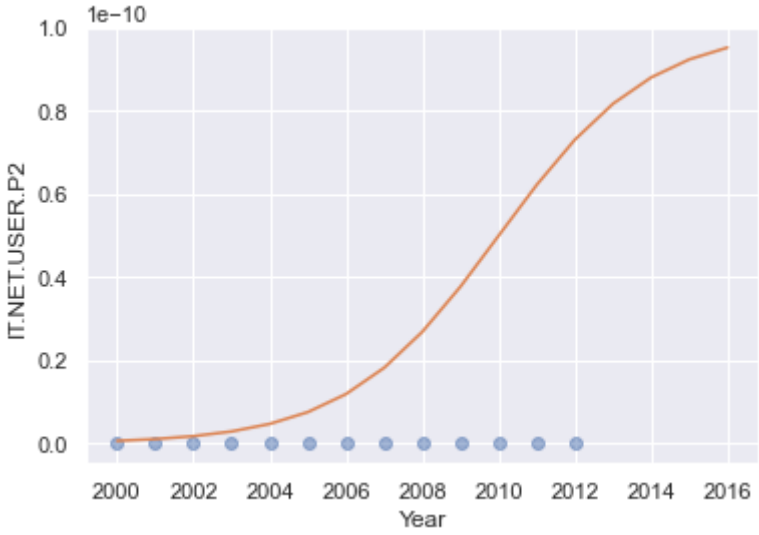

```
For MNE the fit succeeded.
[1.79074358e-01 1.00000000e+02 2.01160891e+03]
[ 0.06803982 41.08846697  4.95756362]
```



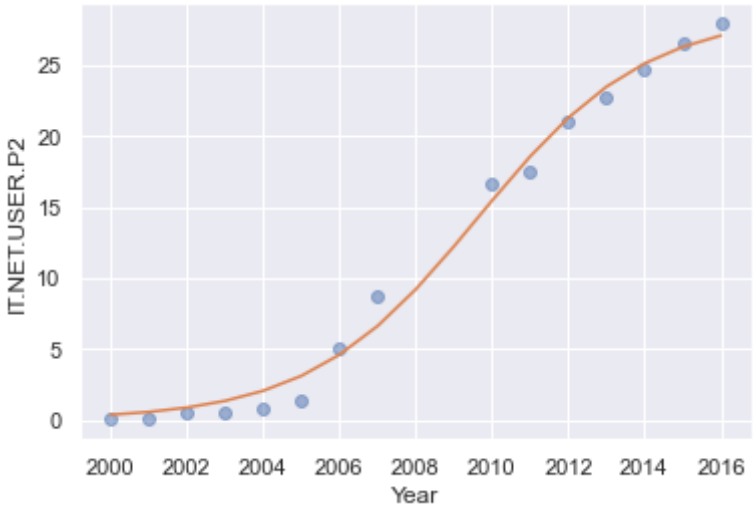

```
For MNG the fit succeeded.
[2.38152819e-01 3.17303919e+01 2.01211133e+03]
[0.03776345 5.46499433 1.6040223 ]
```




```
For PRK the fit succeeded.
[5.00e-01 1.00e-10 2.01e+03]
[8.54428008e-01 1.94886159e-10 9.02679116e+00]
```

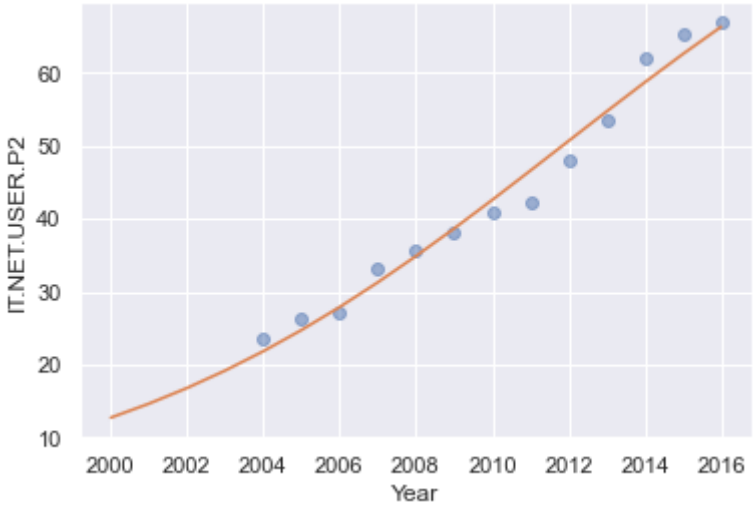



```
For SDN the fit succeeded.
[4.54160197e-01 2.86022460e+01 2.00964393e+03]
[0.04250495 1.25276318 0.33180517]
```

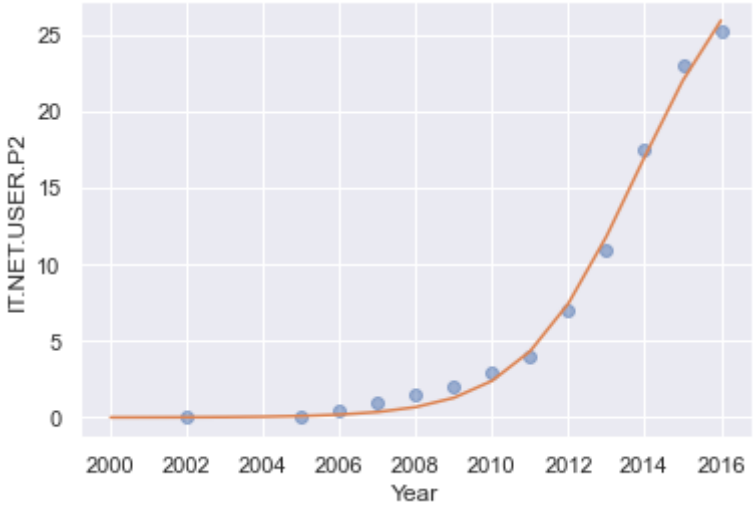



```
For SMR the fit succeeded.
[1.02164226e-02 9.44733586e+01 1.99000000e+03]
[3.83219344e-01 3.02346890e+03 6.22352887e+03]
```

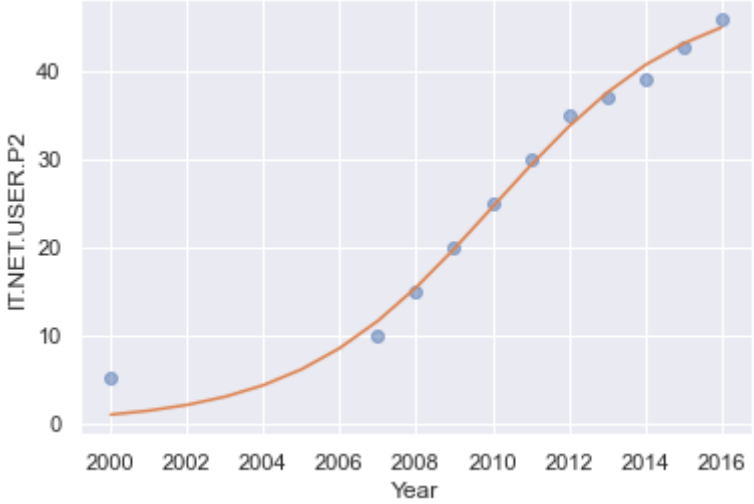
```
For SRB the fit succeeded.
[1.62864555e-01 1.00000000e+02 2.01183341e+03]
[ 0.03345673 23.18809474  3.02734361]
```




```
For TLS the fit succeeded.
[6.60543525e-01 3.20385386e+01 2.01380812e+03]
[0.05919294 2.42489603 0.29099718]
```




```
For TUV the fit succeeded.
[3.88096918e-01 4.94292407e+01 2.01003204e+03]
[0.05597513 3.35461792 0.45670838]
```



In [52]:

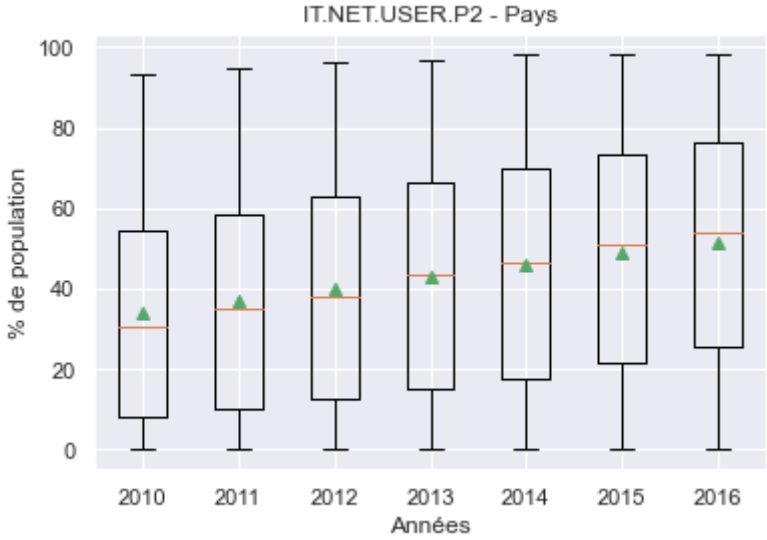
```
print(data_clean.T.describe())
```

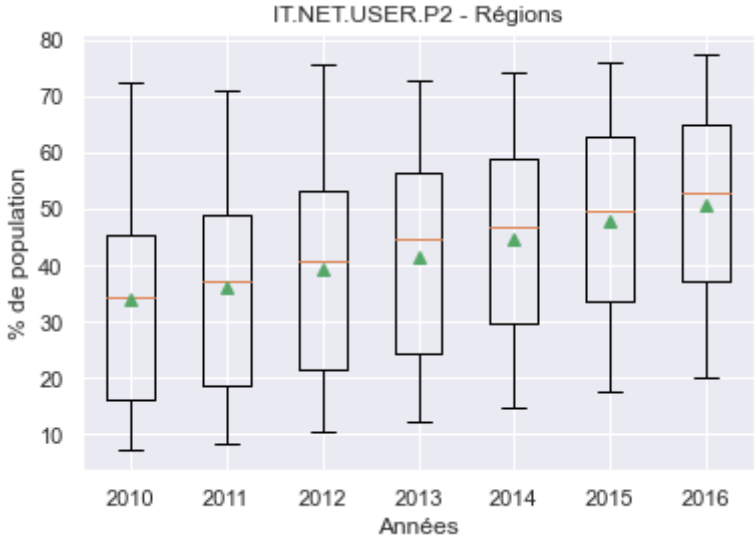
```
year         2000        2001        2002        2003        2004        2005  \
count  211.000000  211.000000  211.000000  211.000000  211.000000  211.000000
mean     8.628915   10.741395   13.874463   16.120339   18.538873   20.548607
std     13.169882   15.342062   18.500252   20.238065   21.747773   22.831016
min     -0.300000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.294900    0.608675    1.082941    1.666924    2.179950    2.715999
```

```
50%      2.434398    3.430887    4.999714    6.933722    8.528818   11.483198
75%     10.060011   14.331559   19.150239   24.170071   28.946757   33.570000
max     52.000000   64.000000   79.120000   83.140000   83.890000   87.000000

year         2006        2007        2008        2009        2010        2011  \
count  211.000000  211.000000  211.000000  211.000000  211.000000  211.000000
mean    22.920259   25.629998   28.342569   31.060805   34.172703   36.941067
std     23.938659   25.217963   26.159415   26.816155   27.260337   27.717388
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      3.648269    4.250000    5.675000    7.045000    8.000000   10.035000
50%     13.867109   16.680000   21.710000   26.340000   30.650000   35.070000
75%     37.985000   41.115000   46.020000   50.721014   54.520000   58.312305
max     89.510000   90.600000   91.000000   93.000000   93.390000   94.819687

year         2012          2013          2014          2015          2016
count  211.000000  2.110000e+02  2.110000e+02  2.110000e+02  2.110000e+02
mean    39.948532  4.282428e+01  4.574094e+01  4.872033e+01  5.128202e+01
std     28.327458  2.881471e+01  2.868628e+01  2.841107e+01  2.840395e+01
min      0.000000  8.175745e-11  8.807971e-11  9.241418e-11  9.525741e-11
25%     12.540030  1.500000e+01  1.773000e+01  2.180552e+01  2.543644e+01
50%     38.200000  4.346000e+01  4.623598e+01  5.030000e+01  5.347372e+01
75%     62.898125  6.637833e+01  6.989000e+01  7.299935e+01  7.629291e+01
max     96.209800  9.654680e+01  9.816000e+01  9.832361e+01  9.824002e+01
```

In [53]:

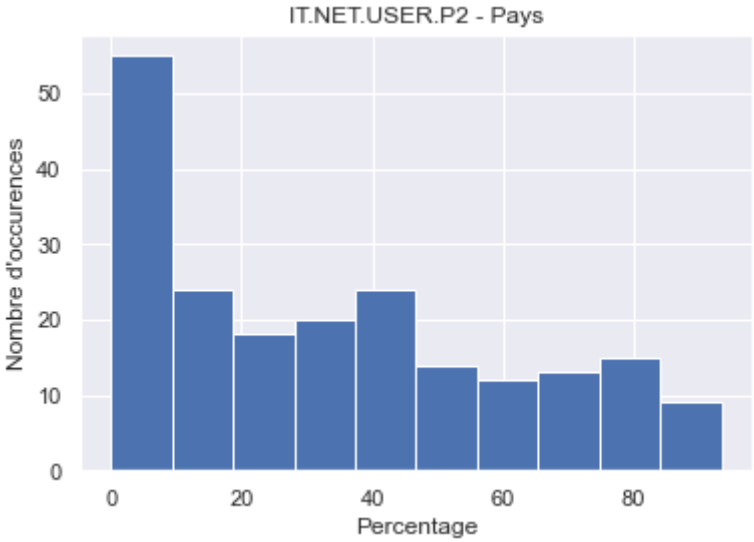
```
years = [str(y) for y in range(2010,2017)]
plot_boxplot(ind, data_clean, unit = "% de population", years = years, showmeans = T
print_info_year('2010', data_clean, unit = "Percentage", ind = ind)
print_info_year('2016', data_clean, unit = "Percentage", ind = ind)
```



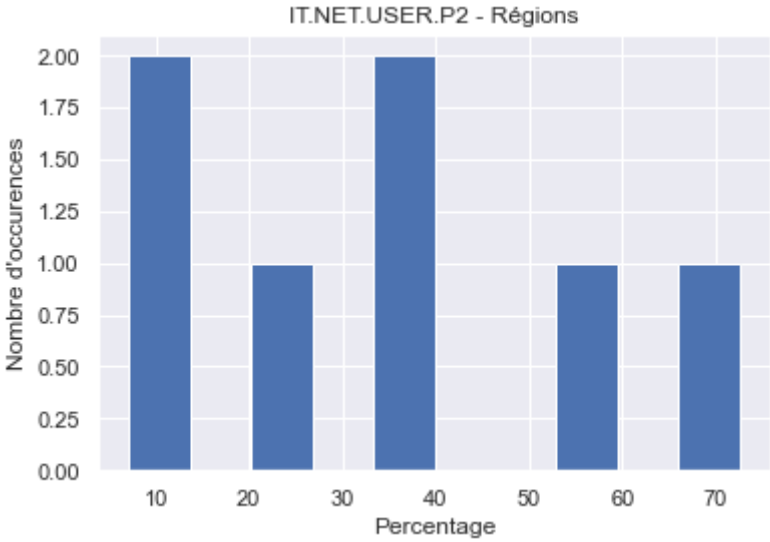



```
2010
```

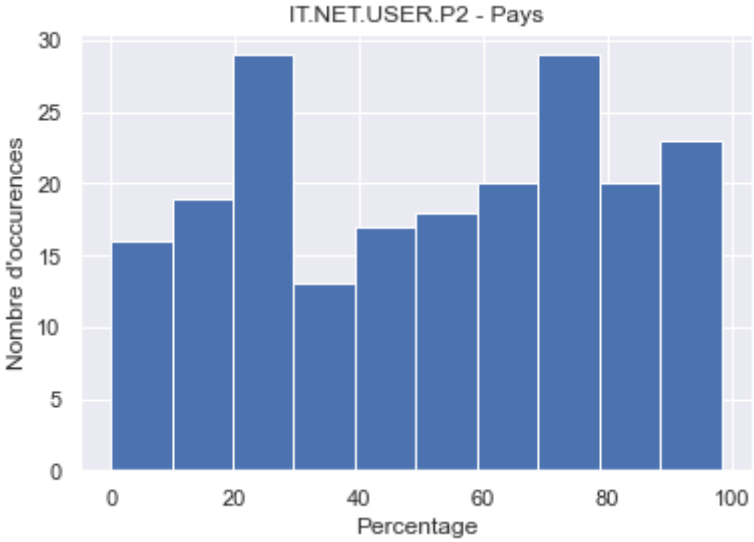
```
Pays
Median:  30.325
Mean:  34.1844416925049
Std:  27.413851660526802
```



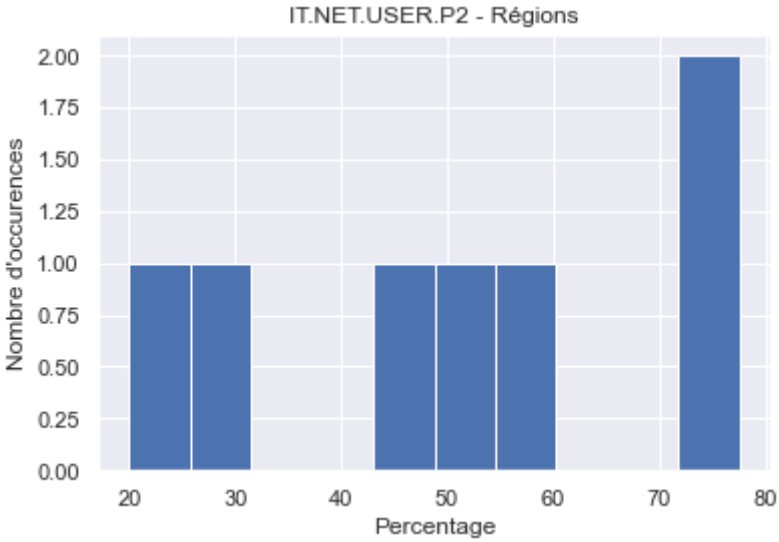

```
Régions
Median:  34.2308383100865
Mean:  33.830603724188364
Std:  24.142626104089167
```




```
2016
Pays
Median:  53.736859036515945
Mean:  51.30292298864768
Std:  28.646550367540545
```



```
Régions
Median:  52.7963207892303
Mean:  50.67294617784136
Std:  21.738012454402565
```



Résumé: IT.NET.USER.P2

- **Garde 204 pays et 7 régions**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 53.7 %, 51.3% et 28.6% pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 52.8 %, 50.7% et 21.7% pour les régions**
- Le score va être propotioniel au usage de l'internet.

In [54]:

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = deconstruct_df(data_clean.copy().reset_index())
DATA_CLEAN
```

Out[54]:

	indicator	country	year	value
0	IT.NET.USER.P2	ABW	2000	15.442823
1	IT.NET.USER.P2	ABW	2001	17.100000
2	IT.NET.USER.P2	ABW	2002	18.800000
3	IT.NET.USER.P2	ABW	2003	20.800000
4	IT.NET.USER.P2	ABW	2004	23.000000
...	...	...	...	...
12	IT.NET.USER.P2	ZWE	2012	12.000000
13	IT.NET.USER.P2	ZWE	2013	15.500000
14	IT.NET.USER.P2	ZWE	2014	16.364740
15	IT.NET.USER.P2	ZWE	2015	22.742818
16	IT.NET.USER.P2	ZWE	2016	23.119989

3587 rows × 4 columns

4.2.2 Niveau d'education du pays

- 'SE.SEC.ENRL', 'SE.SEC.ENRL.GC', 'SE.SEC.ENRR.UP','SE.TER.ENRL', 'SE.TER.ENRR', 'SE.TER.GRAD'
- 'UIS.E.3','UIS.E.3.GPV','UIS.NERA.3','UIS.E.5.B','UIS.E.6','UIS.E.7','UIS.E.8','UIS.G.5.B.T', 'UIS.G.6.T', 'UIS.G.7.T', 'UIS.G.8.T'
- 'UIS.EA.3T6.AG25T99', 'UIS.EA.6T8.AG25T99'
- 'SP.SEC.UTOT.IN', 'SP.TER.TOTL.IN'
- inscription dans l'enseignement secondaire et supérieur (nombre et %), population avec l'âge scolaire
- donne information sur le niveau d'education

Education tertiaire

In [55]:

```
#tertiary enrolment/graduation
my_indicators = ['SE.TER.ENRL', 'SE.TER.ENRR', 'SE.TER.GRAD',
                 'UIS.E.5.B','UIS.E.6','UIS.E.7','UIS.E.8','UIS.G.5.B.T', 'UIS.G.6.T
                 'UIS.EA.3T6.AG25T99', 'UIS.EA.6T8.AG25T99', 'UIS.EA.5T8.AG25T99',
                 'SP.TER.TOTL.IN']
```

In [56]:

```
num_c = []
for ind in my_indicators :
    print(50*'*')
    print(ind)
    data, dy, dc = initial_investigation(ind, output=False)
    num_c.append(len(dc))
    print()
    print(f"There are {len(dc)} countries for the indicator {ind}.")

print()
print(f"{my_indicators[num_c.index(max(num_c))]} has the most countries.")
```

```
**************************************************
SE.TER.ENRL
Completeness for the years 2010-2016: 0.6624102154828412

year           2010          2011          2012          2013          2014  \
count  1.470000e+02  1.490000e+02  1.480000e+02  1.380000e+02  1.310000e+02
mean   2.298180e+06  2.480601e+06  2.569238e+06  2.773283e+06  3.085502e+06
std    7.208971e+06  7.549402e+06  7.826913e+06  8.182494e+06  9.102269e+06
min    5.000000e+00  1.890000e+02  1.000000e+02  2.380000e+02  1.780000e+02
25%    5.886150e+04  7.068300e+04  6.710950e+04  9.721850e+04  1.130575e+05
50%    2.023450e+05  2.368910e+05  2.408850e+05  2.560121e+05  2.832940e+05
75%    8.522200e+05  1.021288e+06  1.023460e+06  1.232937e+06  1.629698e+06
max    5.523363e+07  5.658473e+07  5.887421e+07  6.091618e+07  6.909780e+07

year           2015           2016
count  1.100000e+02       7.000000
mean   1.573105e+06  197838.428571
std    5.494281e+06  210245.804130
min    1.940000e+02   20280.000000
25%    8.172150e+04   66778.000000
50%    2.329090e+05   94728.000000
75%    5.324260e+05  256385.500000
max    4.336739e+07  623534.000000

There are 179 countries for the indicator SE.TER.ENRL.
**************************************************
SE.TER.ENRR
Completeness for the years 2010-2016: 0.6586378737541528

year         2010        2011        2012        2013        2014        2015  \
count  140.000000  145.000000  143.000000  132.000000  126.000000  100.000000
```

```
mean    38.491967   38.647410   40.398885   42.749837   44.428420   44.863470
std     27.826331   28.307418   27.415387   26.960216   27.209240   25.503212
min      0.692570    0.797730    1.295340    3.142330    2.565380    5.621730
25%     13.860443   12.078200   13.474695   18.784398   19.309497   24.013793
50%     35.591730   34.827381   39.863510   40.103275   43.312885   43.405649
75%     61.536581   62.561241   61.562691   63.710171   66.002331   64.801764
max    102.730873  107.828552  110.263123  110.162682  113.871788   94.732300

year        2016
count   7.000000
mean   18.078432
std    15.503049
min     5.343410
25%     6.760450
50%     9.093000
75%    26.163945
max    46.263821

There are 172 countries for the indicator SE.TER.ENRR.
**************************************************
SE.TER.GRAD
Completeness for the years 2010-2016: 0.5852216748768473

year           2010          2011          2012          2013          2014  \
count  9.400000e+01  9.400000e+01  1.070000e+02  1.010000e+02  1.050000e+02
mean   2.416497e+05  2.819502e+05  2.844471e+05  3.680127e+05  3.745696e+05
std    8.745910e+05  9.793535e+05  9.613205e+05  1.327768e+06  1.379464e+06
min    9.500000e+01  6.000000e+01  4.900000e+01  7.800000e+01  3.000000e+01
25%    1.548925e+04  1.983100e+04  1.402550e+04  1.667500e+04  1.846200e+04
50%    4.515750e+04  4.775350e+04  5.329600e+04  5.287500e+04  5.880200e+04
75%    1.190952e+05  1.377350e+05  1.919850e+05  1.721860e+05  1.911410e+05
max    7.863663e+06  8.733298e+06  9.135720e+06  9.366202e+06  1.138050e+07

year           2015           2016
count  8.600000e+01       7.000000
mean   3.482576e+05   52358.285714
std    1.560819e+06   71554.942782
min    4.700000e+01    3035.000000
25%    1.478250e+04   14104.500000
50%    4.465950e+04   19969.000000
75%    8.566825e+04   54016.500000
max    1.192365e+07  207262.000000

There are 145 countries for the indicator SE.TER.GRAD.
**************************************************
UIS.E.5.B
Completeness for the years 2010-2015: 0.6177156177156177

year           2010          2011          2012          2013          2014  \
count  1.120000e+02  1.130000e+02  1.150000e+02  9.900000e+01  8.700000e+01
mean   6.189826e+05  6.464417e+05  6.534686e+05  7.581011e+05  8.833602e+05
std    2.347456e+06  2.320573e+06  2.314254e+06  2.621466e+06  3.283979e+06
min    5.000000e+00  6.900000e+01  7.400000e+01  3.000000e+01  2.100000e+01
25%    4.923500e+03  5.777000e+03  7.020500e+03  4.861000e+03  4.169000e+03
50%    2.754100e+04  2.941300e+04  2.807800e+04  3.169200e+04  2.371900e+04
75%    2.192280e+05  2.268020e+05  2.418405e+05  3.011167e+05  1.682465e+05
max    1.892907e+07  1.875695e+07  1.883137e+07  1.904951e+07  2.260251e+07

year            2015
count       4.000000
mean    51127.500000
std     69087.927698
min       145.000000
25%       190.750000
50%     28964.500000
75%     79901.250000
max    146436.000000

There are 143 countries for the indicator UIS.E.5.B.
```

```
**************************************************
UIS.E.6
Completeness for the years 2010-2015: 0.30933333333333335

year           2010          2011          2012          2013          2014  \
count  3.000000e+00  4.000000e+00  1.600000e+01  1.090000e+02  9.700000e+01
mean   1.038880e+06  8.013410e+05  8.493155e+06  2.215350e+06  2.164566e+06
std    1.229363e+06  1.226482e+06  1.121684e+07  5.981513e+06  6.326706e+06
min    9.975800e+04  7.185000e+03  3.518400e+04  4.650000e+02  3.700000e+01
25%    3.431575e+05  2.215425e+04  4.939142e+05  7.777200e+04  6.653000e+04
50%    5.865570e+05  2.997685e+05  2.181564e+06  1.906820e+05  1.879020e+05
75%    1.508441e+06  1.078955e+06  1.478619e+07  1.131757e+06  9.306750e+05
max    2.430325e+06  2.598642e+06  3.572179e+07  3.740795e+07  4.178894e+07

year            2015
count       3.000000
mean   165193.333333
std    270517.950241
min        80.000000
25%      9096.500000
50%     18113.000000
75%    247750.000000
max    477387.000000

There are 125 countries for the indicator UIS.E.6.
**************************************************
UIS.E.7
Completeness for the years 2010-2015: 0.3048780487804878

year           2010           2011          2012          2013          2014  \
count      2.000000       3.000000  1.600000e+01  1.080000e+02  9.400000e+01
mean   51938.500000   41041.333333  1.580176e+06  4.336309e+05  4.195071e+05
std    67718.909327   62745.222578  2.860704e+06  1.292884e+06  1.235094e+06
min     4054.000000     650.000000  6.420000e+02  2.800000e+01  2.600000e+01
25%    27996.250000    4898.500000  1.459350e+04  9.471500e+03  9.565000e+03
50%    51938.500000    9147.000000  1.186955e+05  4.208700e+04  5.457702e+04
75%    75880.750000   61237.000000  1.743003e+06  1.893428e+05  1.572657e+05
max    99823.000000  113327.000000  1.081846e+07  1.050313e+07  9.437299e+06

year           2015
count      2.000000
mean   17510.000000
std    21237.245066
min     2493.000000
25%    10001.500000
50%    17510.000000
75%    25018.500000
max    32527.000000

There are 123 countries for the indicator UIS.E.7.
**************************************************
UIS.E.8
Completeness for the years 2010-2015: 0.625

year           2010          2011          2012          2013          2014  \
count  9.700000e+01  1.020000e+02  1.060000e+02  1.090000e+02  9.300000e+01
mean   4.390104e+04  4.697211e+04  4.598408e+04  4.801587e+04  5.062366e+04
std    1.407704e+05  1.399228e+05  1.381984e+05  1.347383e+05  1.448214e+05
min    2.000000e+00  4.100000e+01  5.000000e+00  1.000000e+00  1.000000e+00
25%    7.910000e+02  1.072750e+03  1.086250e+03  8.290000e+02  7.090000e+02
50%    4.055000e+03  5.124500e+03  5.467000e+03  5.958000e+03  5.507000e+03
75%    2.592100e+04  2.627850e+04  2.549350e+04  2.519400e+04  2.390000e+04
max    1.049182e+06  1.050020e+06  1.054254e+06  1.067777e+06  1.057249e+06

year         2015
count     3.00000
mean   1283.00000
std    1081.03469
min      49.00000
```

```
25%     893.00000
50%    1737.00000
75%    1900.00000
max    2063.00000

There are 136 countries for the indicator UIS.E.8.
**************************************************
UIS.G.5.B.T
Completeness for the years 2010-2015: 0.5303030303030303

year           2010          2011          2012          2013          2014  \
count  8.200000e+01  8.100000e+01  9.200000e+01  6.200000e+01  6.300000e+01
mean   8.654068e+04  1.019805e+05  9.960107e+04  1.216790e+05  1.445040e+05
std    4.600162e+05  5.001188e+05  4.832398e+05  5.835863e+05  7.182693e+05
min    2.800000e+01  5.800000e+01  4.000000e+00  4.000000e+00  6.000000e+00
25%    1.023250e+03  1.387000e+03  1.322500e+03  1.112250e+03  1.122500e+03
50%    6.794000e+03  7.555000e+03  7.685500e+03  5.664000e+03  6.599000e+03
75%    2.855775e+04  3.412900e+04  4.844175e+04  3.026600e+04  5.736617e+04
max    4.144596e+06  4.452518e+06  4.584976e+06  4.513708e+06  5.647170e+06

year            2015
count       5.000000
mean    49356.200000
std     79315.349326
min         2.000000
25%        59.000000
50%     12379.000000
75%     47411.000000
max    186930.000000

There are 121 countries for the indicator UIS.G.5.B.T.
**************************************************
UIS.G.6.T
Completeness for the years 2010-2015: 0.29333333333333333

year            2010           2011           2012          2013  \
count       3.000000       3.000000      11.000000  8.000000e+01
mean   175643.666667  176379.666667  110790.181818  2.384864e+05
std    201810.176518  214175.017157  142017.036559  9.575364e+05
min     11638.000000    4438.000000     213.000000  7.200000e+01
25%     62959.000000   56423.500000   12774.000000  8.884500e+03
50%    114280.000000  108409.000000   55920.000000  2.800842e+04
75%    257646.500000  262350.500000  126160.000000  8.722775e+04
max    401013.000000  416292.000000  429006.000000  7.357287e+06

year           2014           2015
count  7.500000e+01       4.000000
mean   1.925121e+05  127443.500000
std    6.319973e+05  158263.726283
min    7.000000e+00      48.000000
25%    8.041500e+03    1965.000000
50%    3.177900e+04   90141.000000
75%    1.163515e+05  215619.500000
max    5.085064e+06  329444.000000

There are 100 countries for the indicator UIS.G.6.T.
**************************************************
UIS.G.7.T
Completeness for the years 2010-2015: 0.282312925170068

year          2010          2011          2012          2013           2014  \
count      2.00000      2.000000     10.000000  7.500000e+01      74.000000
mean    8387.50000   9512.000000  10266.200000  6.550743e+04   53852.078099
std    11320.07246  13282.293778  11974.217671  2.087019e+05  131925.172334
min      383.00000    120.000000     86.000000  6.000000e+00       9.000000
25%     4385.25000   4816.000000   1138.000000  1.626000e+03    2940.750000
50%     8387.50000   9512.000000   7444.000000  6.702000e+03   12466.000000
75%    12389.75000  14208.000000  11241.250000  2.737400e+04   36227.000000
max    16392.00000  18904.000000  37034.000000  1.464443e+06  870445.000000
```

```
year           2015
count      3.000000
mean   32747.333333
std    43548.971013
min      155.000000
25%     8017.500000
50%    15880.000000
75%    49043.500000
max    82207.000000

There are 98 countries for the indicator UIS.G.7.T.
**************************************************
UIS.G.8.T
Completeness for the years 2010-2015: 0.5864197530864198

year           2010          2011          2012          2013          2014  \
count     78.000000     81.000000     83.000000     68.000000     66.000000
mean    3573.692308   3921.049383   4085.879518   4232.904412   4855.071050
std     9465.797746   9727.879242   9982.908857  10843.269787  11497.281327
min        1.000000      1.000000      1.000000      1.000000      1.000000
25%      139.250000    208.000000    189.500000    136.500000    100.750000
50%      626.500000    917.000000   1062.000000    539.000000    825.500000
75%     2783.500000   3055.000000   2903.500000   3315.000000   3550.000000
max    69570.000000  73041.000000  76288.000000  65051.000000  67449.000000

year           2015
count      4.000000
mean    3456.500000
std     6319.835731
min        9.000000
25%      289.500000
50%      443.000000
75%     3610.000000
max    12931.000000

There are 108 countries for the indicator UIS.G.8.T.
**************************************************
UIS.EA.3T6.AG25T99
Completeness for the years 2010-2016: 0.45714285714285713

year        2010       2011       2012       2013       2014       2015  \
count  74.000000  70.000000  62.000000  55.000000  65.000000  56.000000
mean   54.967187  56.565261  55.052547  58.751273  58.748713  61.975301
std    22.951995  24.441709  25.849899  23.442239  24.078745  21.827091
min     5.169190   4.724750   3.395960   8.234610   2.805490   6.035220
25%    35.088583  36.402906  33.024071  36.493824  38.525700  41.598525
50%    57.533865  65.615959  62.364231  66.459259  64.215721  67.630615
75%    75.648539  76.318777  77.467764  78.392708  78.109840  79.620113
max    88.645401  90.237610  90.925499  91.540611  91.947632  92.079582

year        2016
count   2.000000
mean   29.969376
std     3.825866
min    27.264080
25%    28.616728
50%    29.969376
75%    31.322023
max    32.674671

There are 120 countries for the indicator UIS.EA.3T6.AG25T99.
**************************************************
UIS.EA.6T8.AG25T99
Completeness for the years 2010-2016: 0.28892455858747995

year        2010       2011       2012       2013       2014       2015  \
count  15.000000  12.000000  17.000000  37.000000  51.000000  46.000000
mean   14.028041  14.092196  11.452016  16.624016  18.583499  19.705454
```

```
std     8.477435   8.289203   8.906570   9.256448   9.000270   8.696821
min     0.000000   2.049400   1.698850   0.000000   0.000000   1.650730
25%     9.475630   9.426630   3.399720  10.075410  11.981315  12.679040
50%    12.858660  11.083140   9.464280  16.256241  18.217310  19.131975
75%    20.080959  20.029924  14.814900  24.887770  25.846106  27.145038
max    28.326151  26.837799  30.172710  31.660660  35.605011  36.949291

year       2016
count   2.00000
mean    9.14358
std     8.01350
min     3.47718
25%     6.31038
50%     9.14358
75%    11.97678
max    14.80998

There are 89 countries for the indicator UIS.EA.6T8.AG25T99.
**************************************************
UIS.EA.5T8.AG25T99
Completeness for the years 2010-2016: 0.4563843236409608

year        2010       2011       2012       2013       2014       2015  \
count  72.000000  67.000000  59.000000  51.000000  59.000000  51.000000
mean   20.802865  20.646568  21.506875  23.589514  23.912502  25.842116
std    11.586425  11.668515  11.040423  11.283258  11.969610  11.495134
min     0.000000   1.092990   1.928360   2.862670   0.000000   2.072470
25%    12.088948  11.833295  12.944825  15.378020  13.747650  16.918035
50%    20.019360  19.571461  21.010241  22.537451  23.495480  25.411600
75%    29.112878  29.546570  30.037390  31.746221  32.837669  33.675150
max    61.933189  47.674568  44.669971  51.932289  54.947639  55.303741

year        2016
count   2.000000
mean   14.092620
std     7.035401
min     9.117840
25%    11.605230
50%    14.092620
75%    16.580010
max    19.067400

There are 113 countries for the indicator UIS.EA.5T8.AG25T99.
**************************************************
SP.TER.TOTL.IN
Completeness for the years 2010-2016: 0.7531824611032532

year           2010          2011          2012          2013          2014  \
count  1.990000e+02  1.820000e+02  1.760000e+02  1.700000e+02  2.020000e+02
mean   6.113856e+06  6.695563e+06  6.857382e+06  6.996587e+06  5.955561e+06
std    2.306221e+07  2.384630e+07  2.389222e+07  2.389969e+07  2.172879e+07
min    1.041000e+03  1.448000e+03  1.457000e+03  1.395000e+03  8.680000e+02
25%    1.435380e+05  2.317242e+05  2.581600e+05  2.657060e+05  1.516418e+05
50%    6.242340e+05  7.627780e+05  7.601635e+05  8.630915e+05  6.267365e+05
75%    2.482447e+06  2.740180e+06  2.853672e+06  2.871583e+06  2.602514e+06
max    1.988264e+08  1.949116e+08  1.893518e+08  1.828877e+08  1.765053e+08

year           2015          2016
count  1.290000e+02  7.000000e+00
mean   3.672495e+06  1.208216e+06
std    1.393025e+07  9.147148e+05
min    1.986000e+03  2.200870e+05
25%    2.429840e+05  6.173370e+05
50%    6.695660e+05  1.030103e+06
75%    2.422484e+06  1.525426e+06
max    1.194700e+08  2.921796e+06

There are 202 countries for the indicator SP.TER.TOTL.IN.
```

SP.TER.TOTL.IN has the most countries.

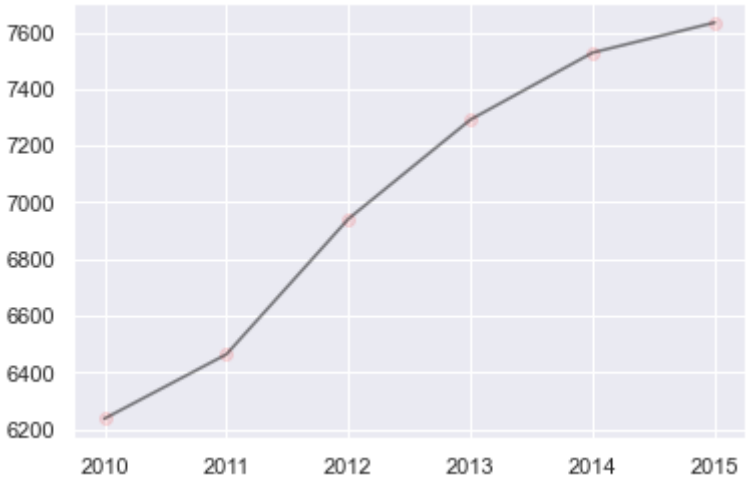
In [57]:

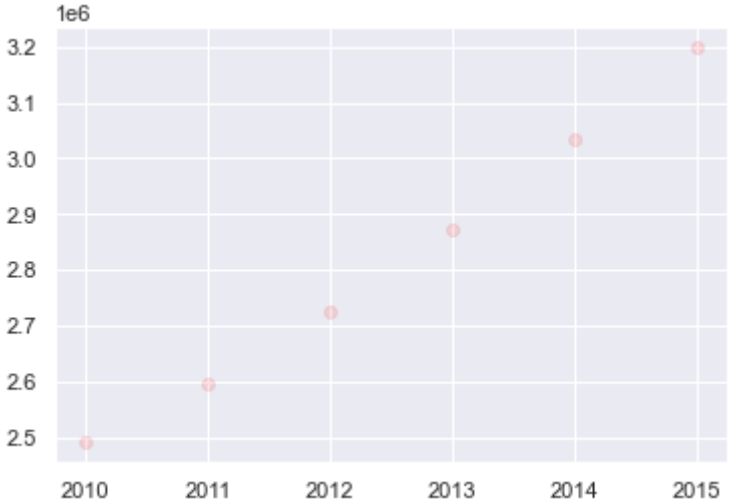
```
ind = 'SP.TER.TOTL.IN'
ind2 = 'SE.TER.ENRL'
ind3 = 'SE.TER.ENRR'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
data = drop_countries(data,len(years))

data2 = make_2D_map(DATA, ind2, ['indicator', 'year'], 'country').drop(columns = ['i
data3 = make_2D_map(DATA, ind3, ['indicator', 'year'], 'country').drop(columns = ['i
data4 = data2.div(data3).multiply(100)
```

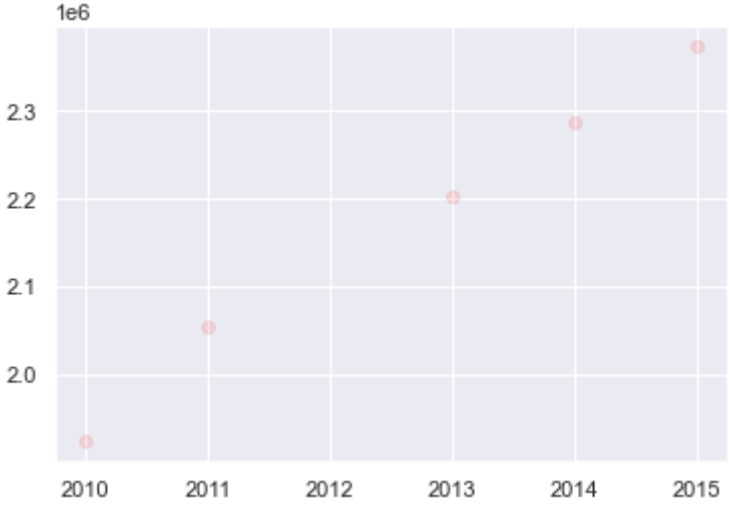
```
The following countries were dropped :
['AND', 'ARE', 'ASM', 'BIH', 'CHI', 'CYM', 'FRO', 'GIB', 'GRL', 'GUM', 'IMN', 'MAF',
'MCO', 'MNP', 'NCL', 'OMN', 'PYF', 'SGP', 'SXM', 'TCA', 'VIR', 'XKX']
```

In [58]:

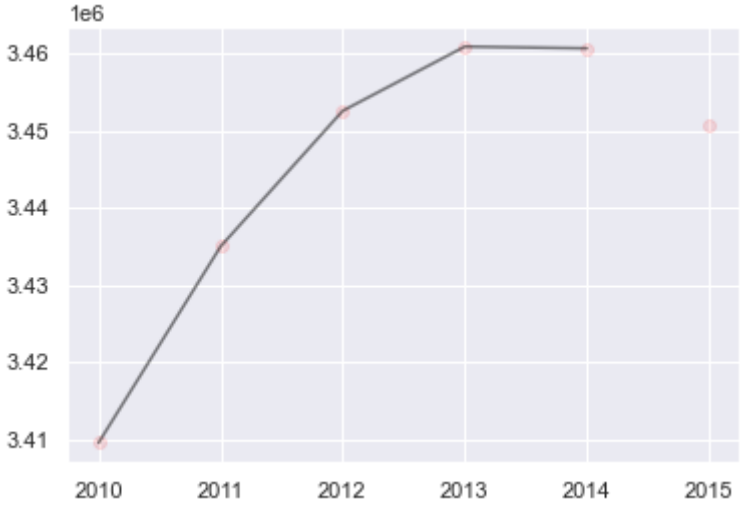
```
data_holes = data.loc[: ,data.isna().sum() != 0]
for c in data_holes.columns.values :
    plt.plot(data_holes[c], 'o', alpha=0.1, color = 'red')
    plt.plot(data4[c],'-', alpha=0.5, color = 'black')
    plt.show()
```

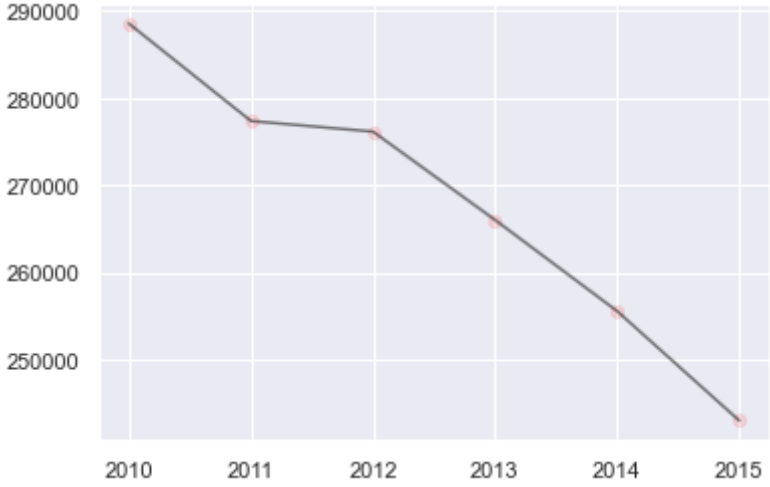


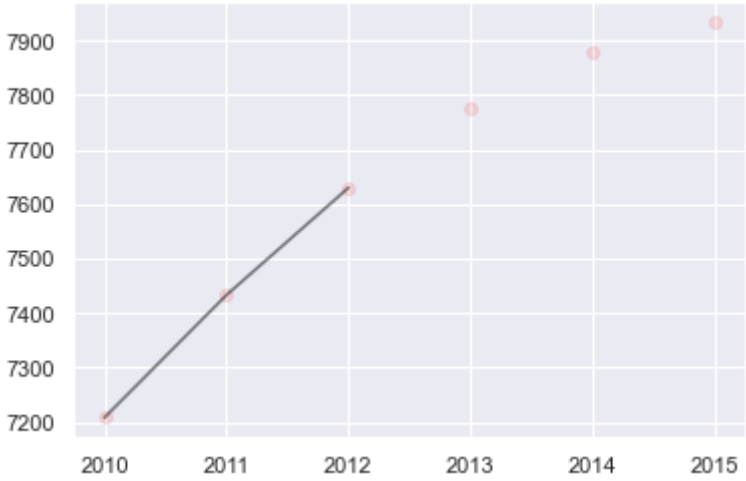


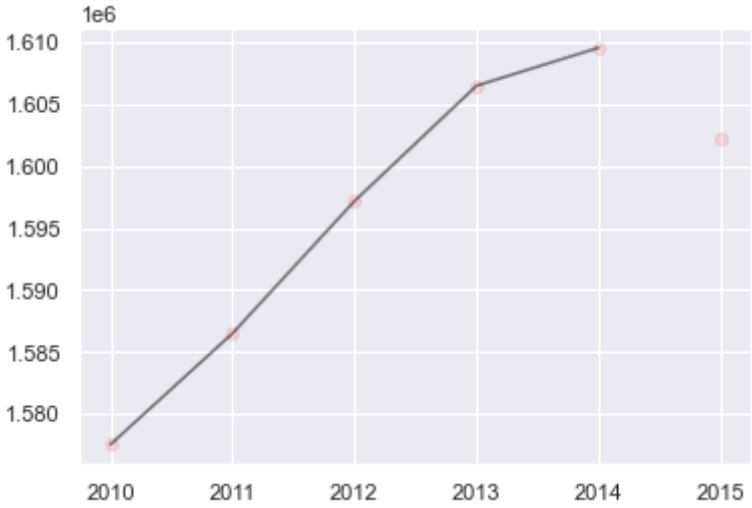

1e6
2.3
2.2
2.1
2.0
2010
2011
2012
2013
2014
2015

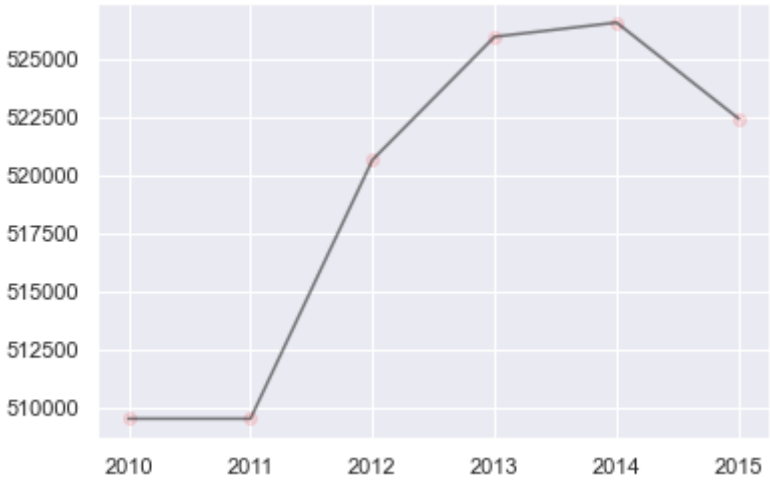
277000
276000
275000
274000
273000
272000
2010
2011
2012
2013
2014
2015

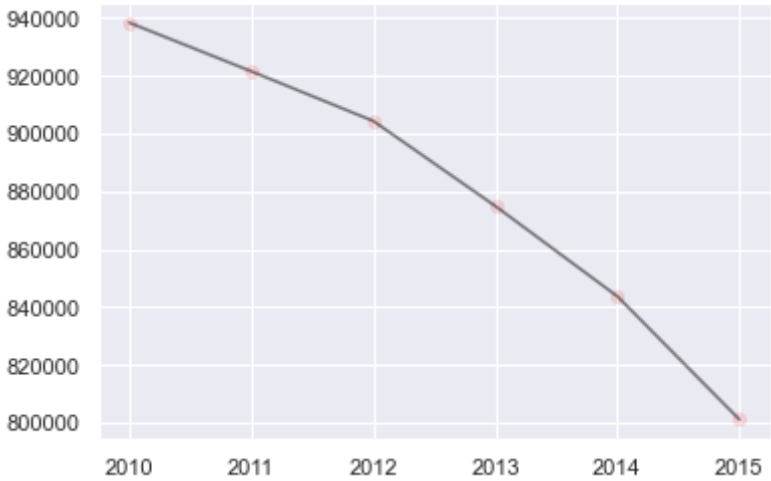

1e6
3.46
3.45
3.44
3.43
3.42
3.41
2010
2011
2012
2013
2014
2015

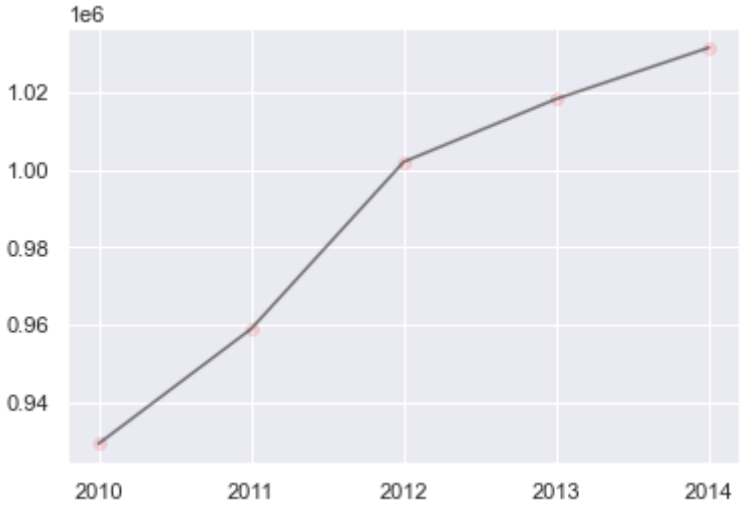

290000
280000
270000
260000
250000
2010
2011
2012
2013
2014
2015

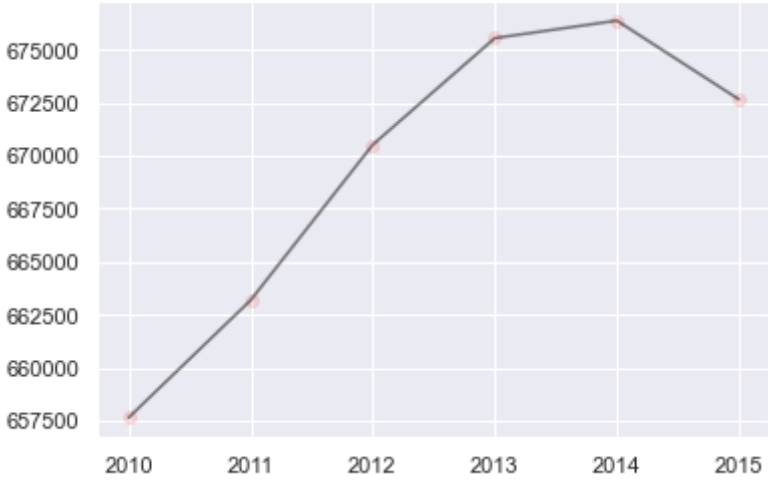

7900
7800
7700
7600
7500
7400
7300
7200
2010
2011
2012
2013
2014
2015

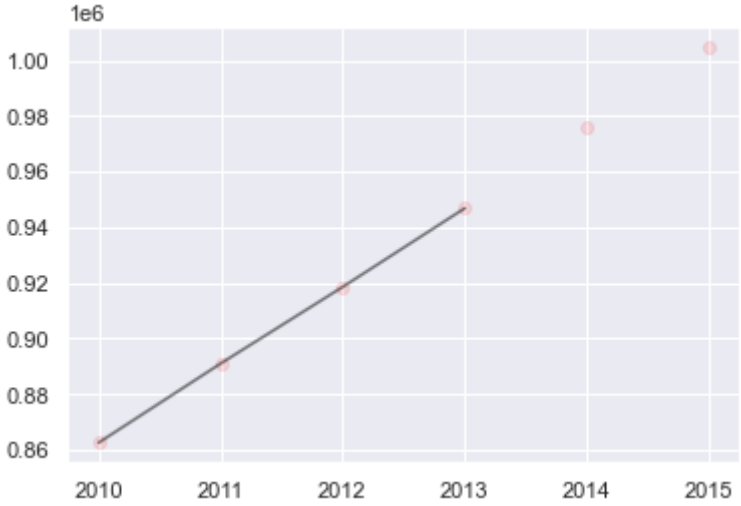

1e6
1.610
1.605
1.600
1.595
1.590
1.585
1.580
2010
2011
2012
2013
2014
2015

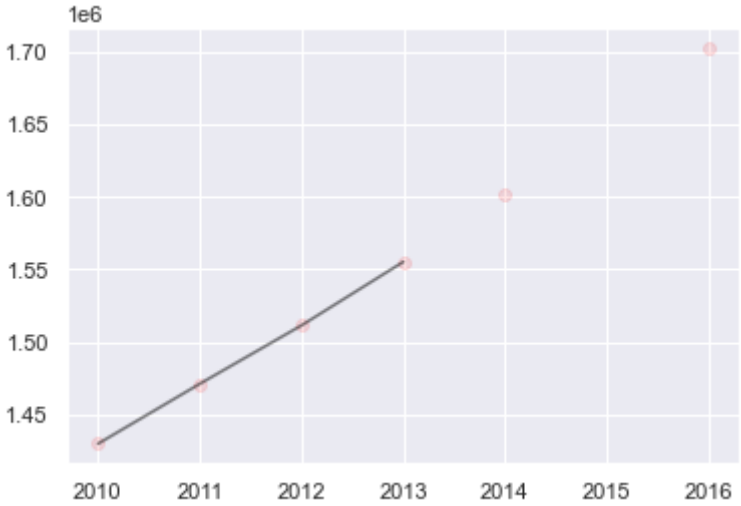

525000
522500
520000
517500
515000
512500
510000
2010
2011
2012
2013
2014
2015

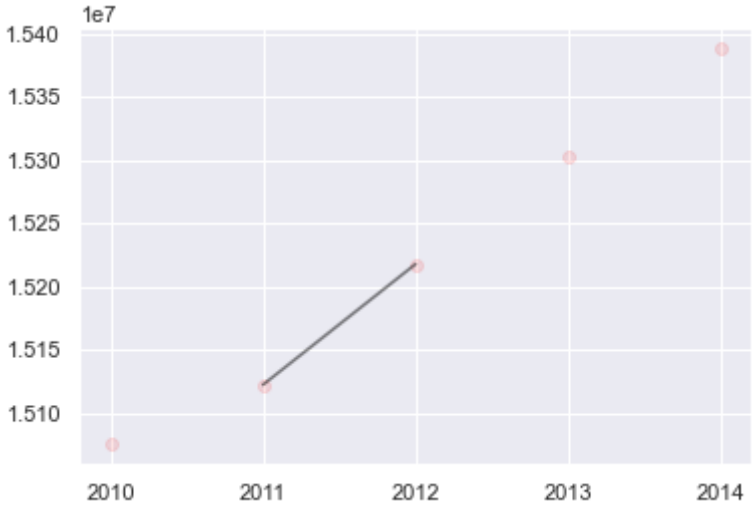

940000
920000
900000
880000
860000
840000
820000
800000
2010
2011
2012
2013
2014
2015

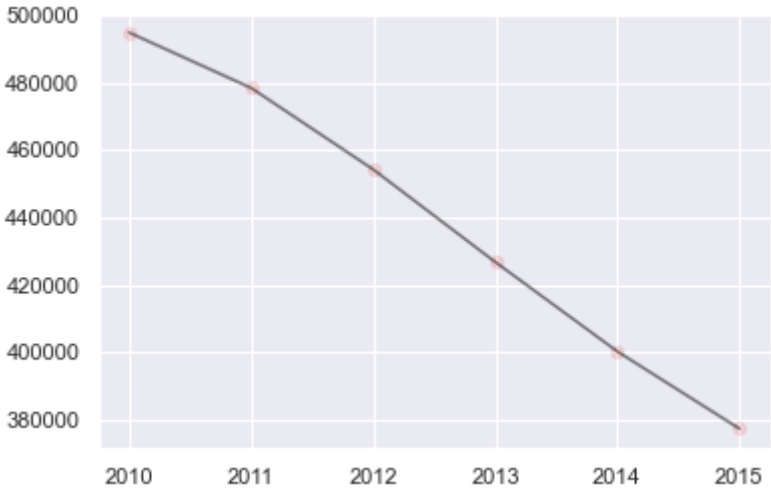

1e6
1.02
1.00
0.98
0.96
0.94
2010
2011
2012
2013
2014

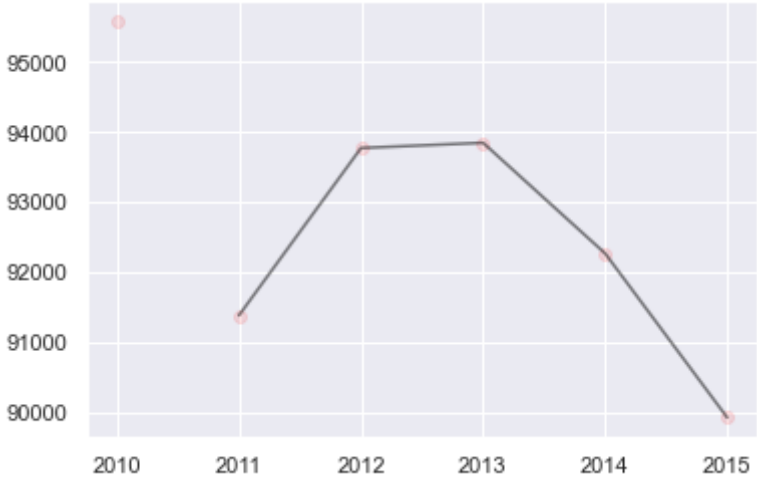

675000
672500
670000
667500
665000
662500
660000
657500
2010
2011
2012
2013
2014
2015

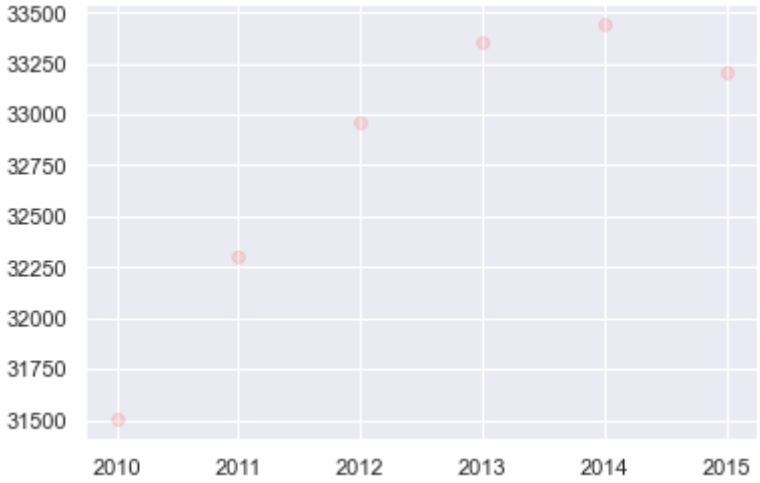

1e6
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
2010
2011
2012
2013
2014
2015


1e6
1.70
1.65
1.60
1.55
1.50
1.45
2010
2011
2012
2013
2014
2015
2016

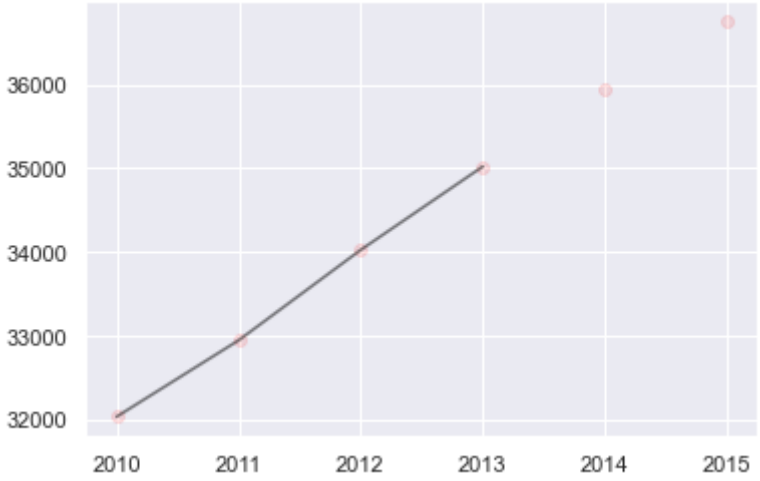

1e7
1.540
1.535
1.530
1.525
1.520
1.515
1.510
2010
2011
2012
2013
2014

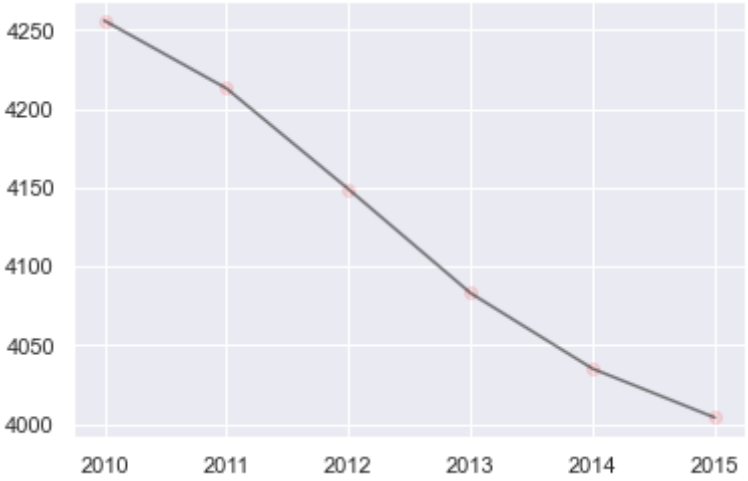

500000
480000
460000
440000
420000
400000
380000
2010
2011
2012
2013
2014
2015

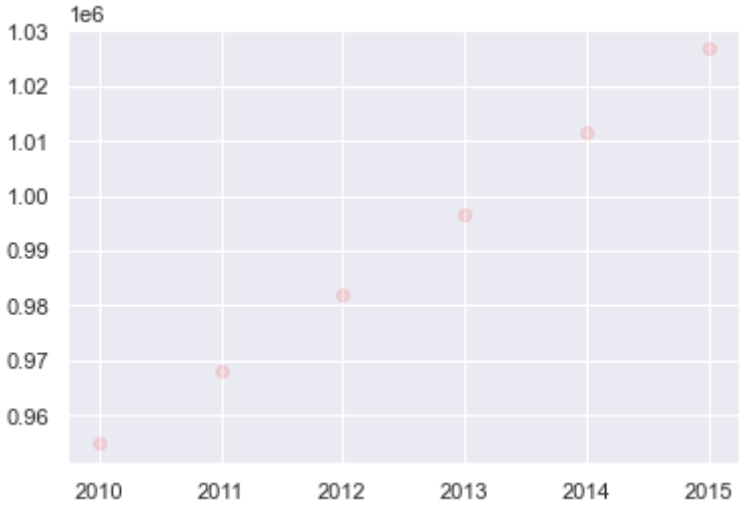

95000
94000
93000
92000
91000
90000
2010
2011
2012
2013
2014
2015

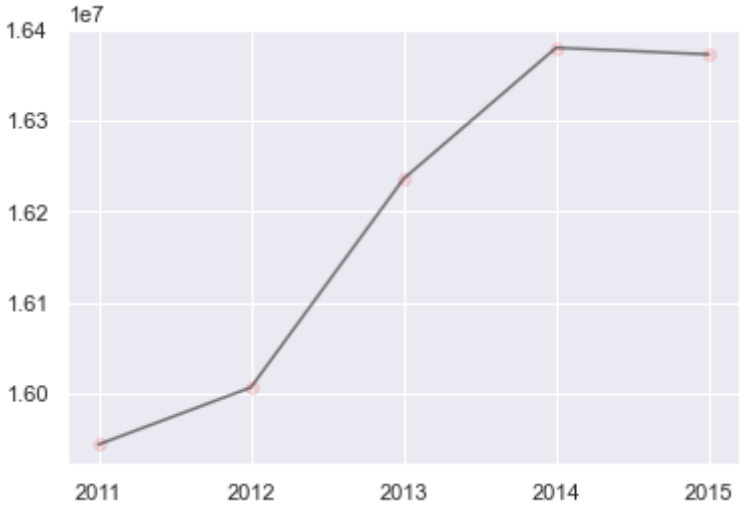

33500
33250
33000
32750
32500
32250
32000
31750
31500
2010
2011
2012
2013
2014
2015

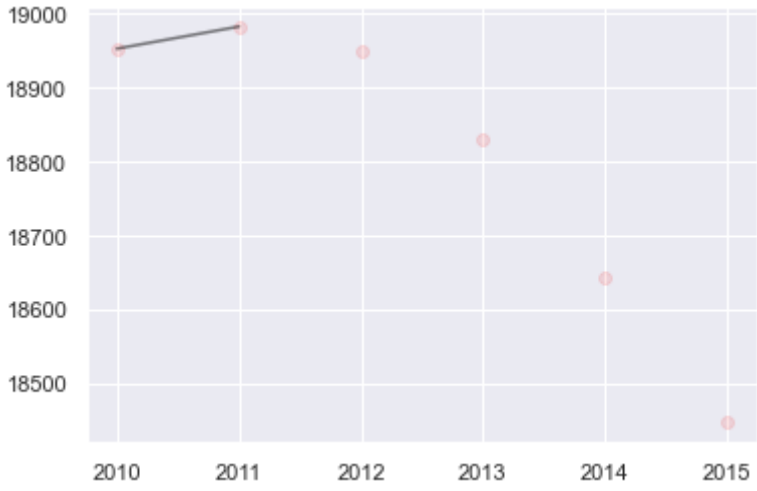
700000
675000
650000
625000
600000
575000
550000
2010
2011
2012
2013
2014
2015

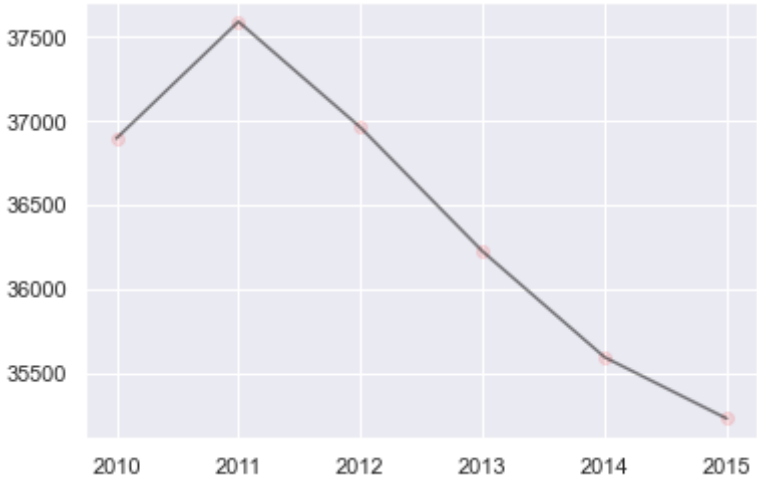

36000
35000
34000
33000
32000
2010
2011
2012
2013
2014
2015

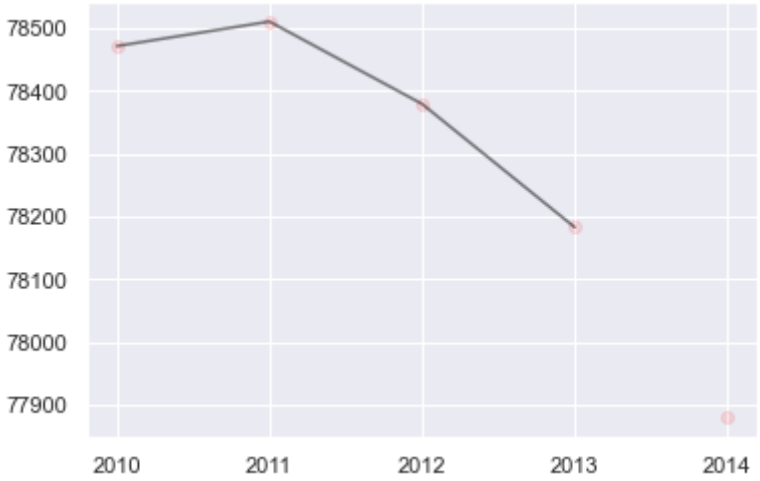

4250
4200
4150
4100
4050
4000
2010
2011
2012
2013
2014
2015

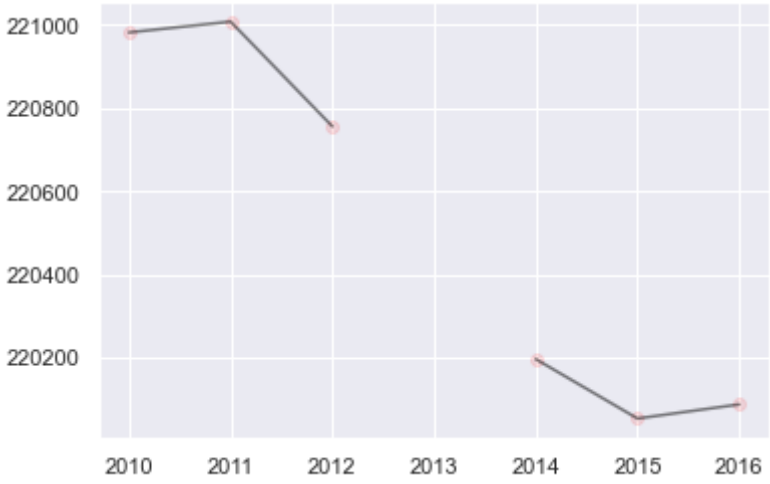

1e6
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.96
2010
2011
2012
2013
2014
2015

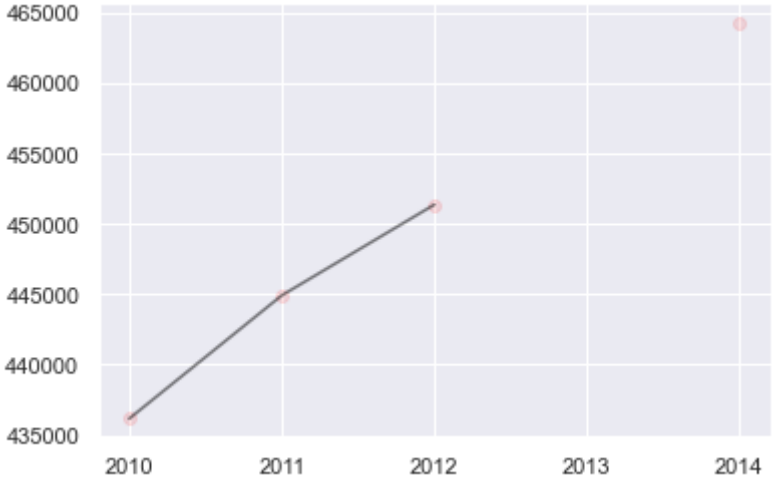

1e7
1.64
1.63
1.62
1.61
1.60
2011
2012
2013
2014
2015

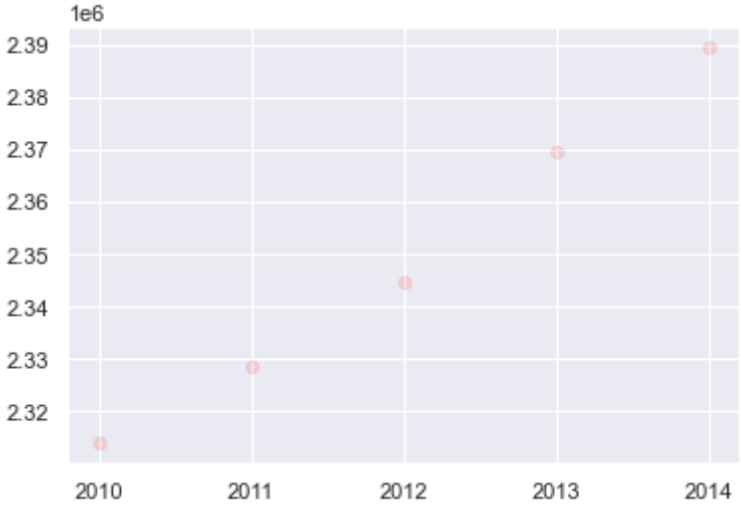

19000
18900
18800
18700
18600
18500
2010
2011
2012
2013
2014
2015

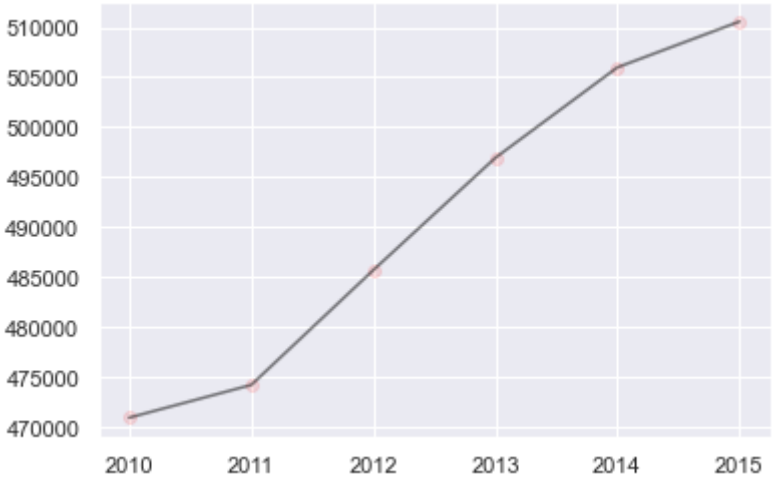

37500
37000
36500
36000
35500
2010
2011
2012
2013
2014
2015

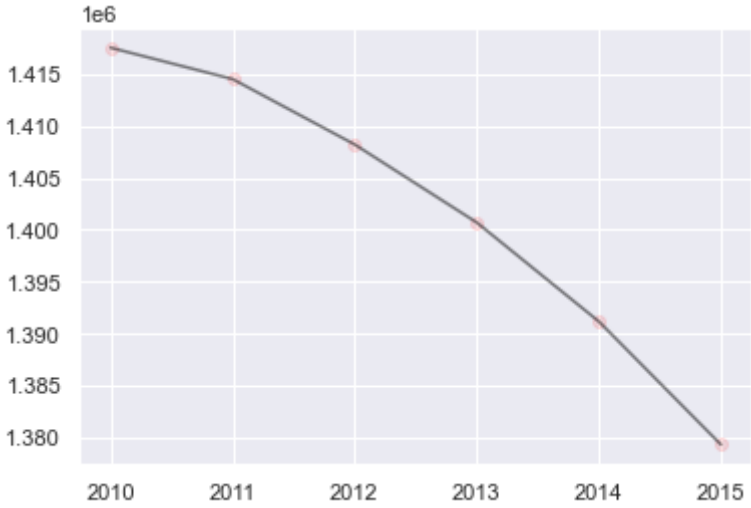

78500
78400
78300
78200
78100
78000
77900
2010
2011
2012
2013
2014

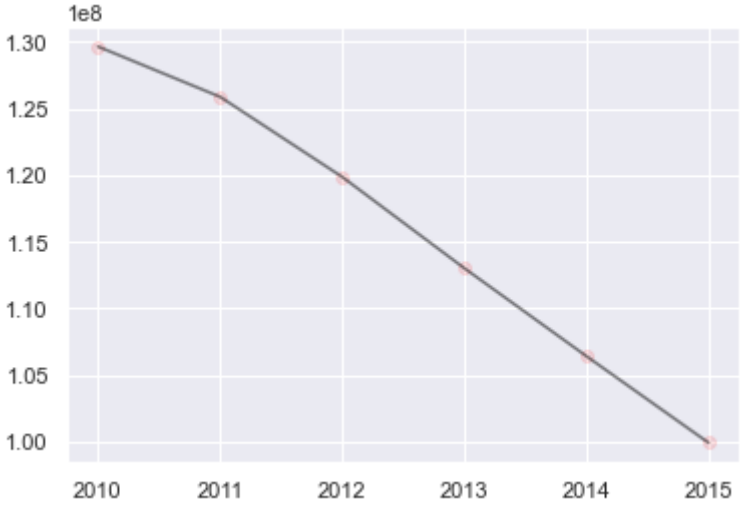

221000
220800
220600
220400
220200
2010
2011
2012
2013
2014
2015
2016

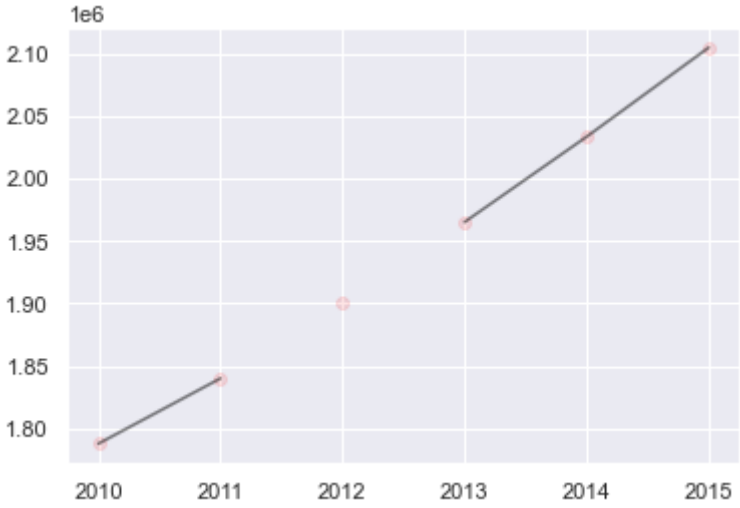

465000
460000
455000
450000
445000
440000
435000
2010
2011
2012
2013
2014

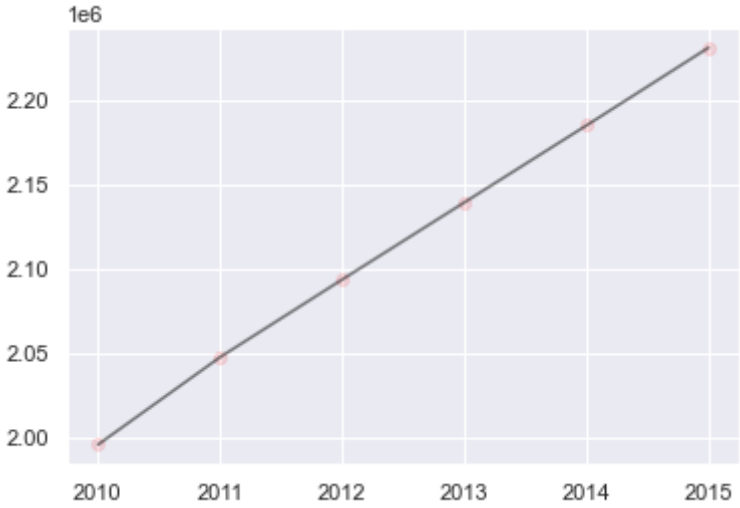

1e6
2.39
2.38
2.37
2.36
2.35
2.34
2.33
2.32
2010
2011
2012
2013
2014

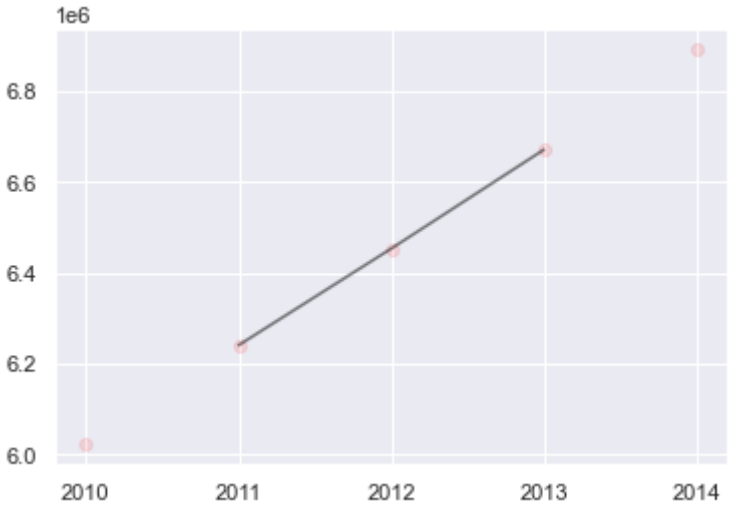

510000
505000
500000
495000
490000
485000
480000
475000
470000
2010
2011
2012
2013
2014
2015

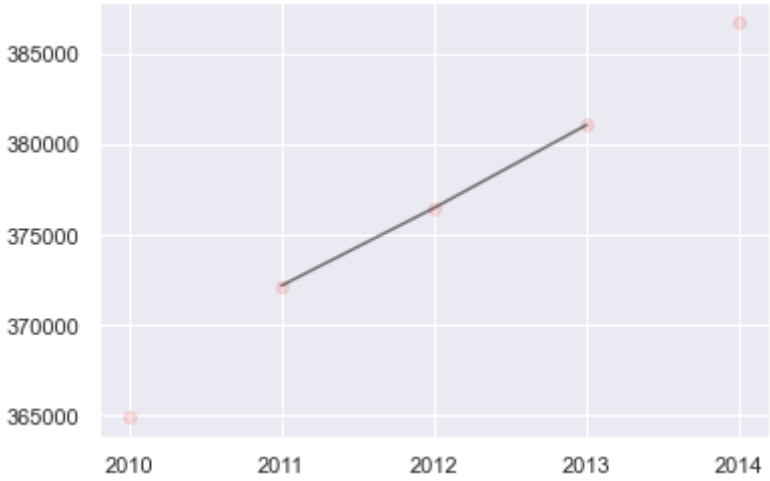

1e6
1.415
1.410
1.405
1.400
1.395
1.390
1.385
1.380
2010
2011
2012
2013
2014
2015

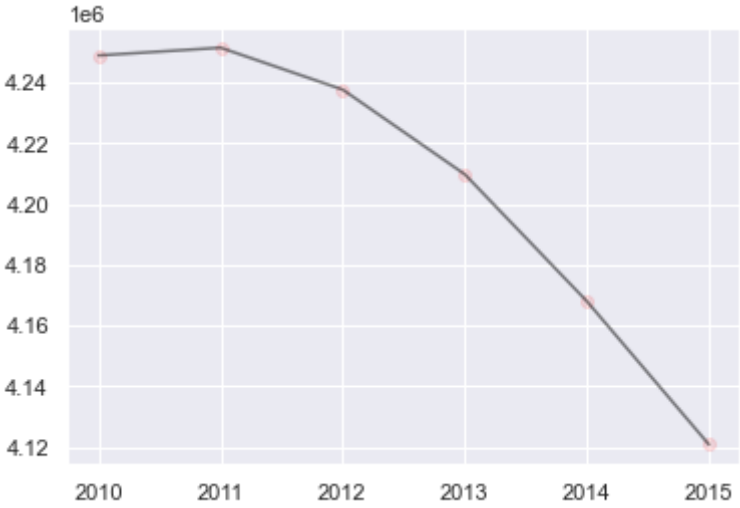

1e8
1.30
1.25
1.20
1.15
1.10
1.05
1.00
2010
2011
2012
2013
2014
2015

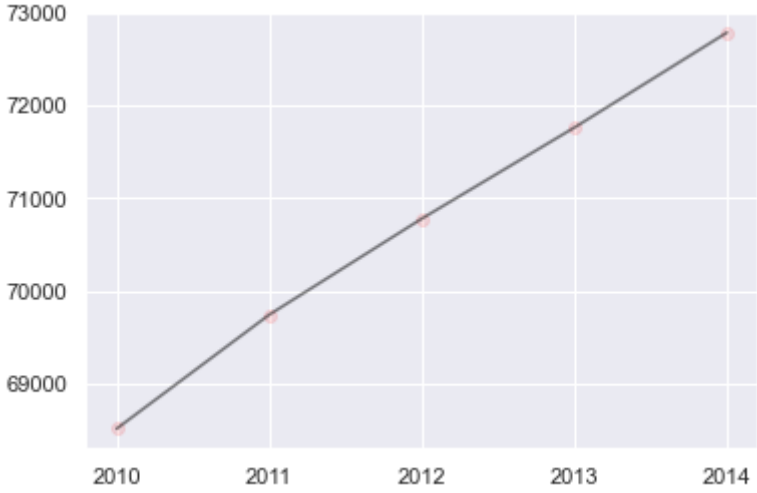

1e6
2.10
2.05
2.00
1.95
1.90
1.85
1.80
2010
2011
2012
2013
2014
2015


1e6
2.20
2.15
2.10
2.05
2.00
2010
2011
2012
2013
2014
2015

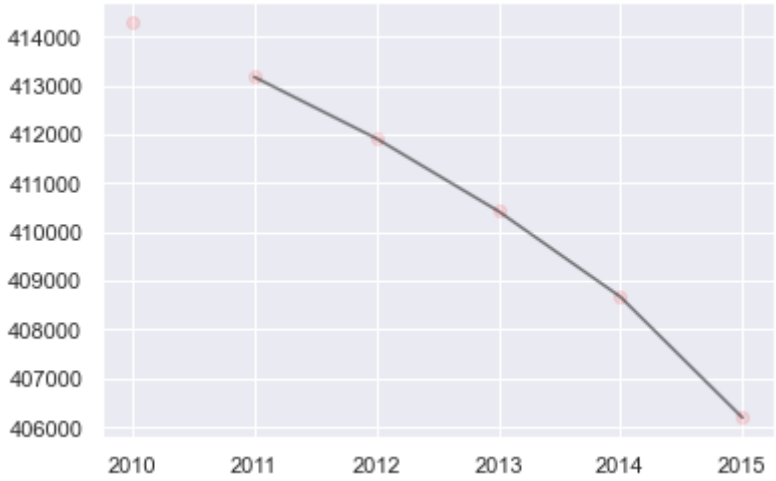

1e6
6.8
6.6
6.4
6.2
6.0
2010
2011
2012
2013
2014

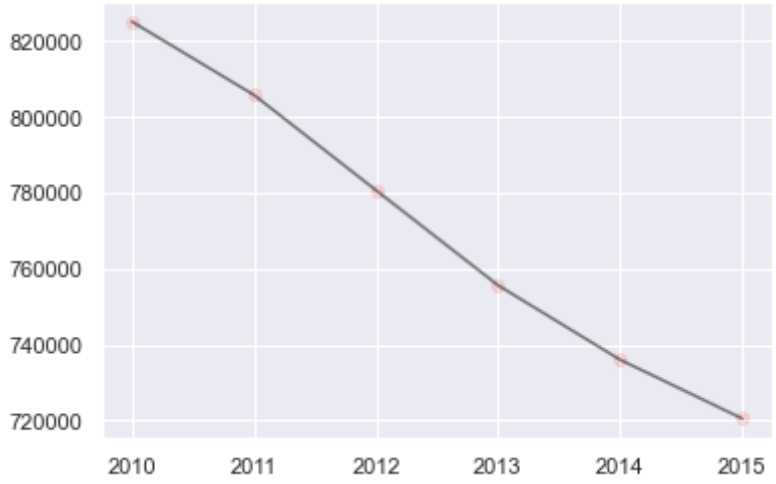

385000
380000
375000
370000
365000
2010
2011
2012
2013
2014

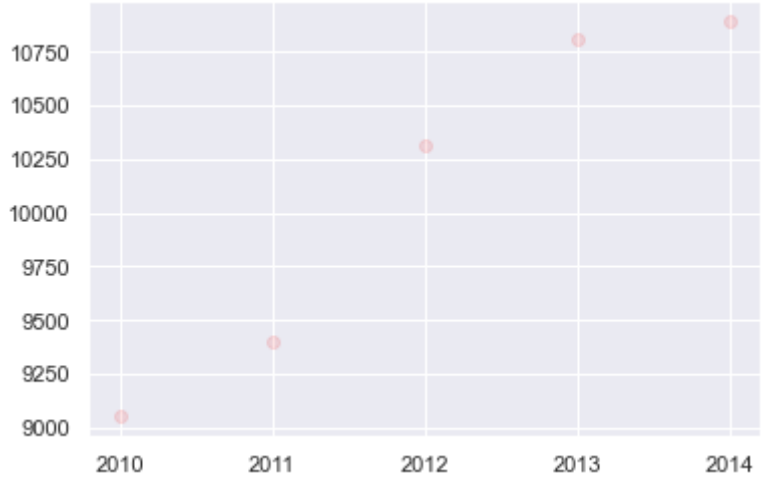

1e6
4.24
4.22
4.20
4.18
4.16
4.14
4.12
2010
2011
2012
2013
2014
2015

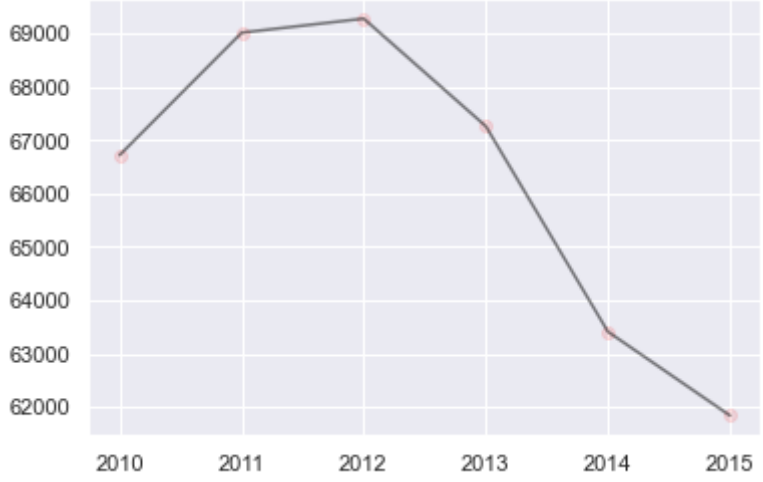

73000
72000
71000
70000
69000
2010
2011
2012
2013
2014

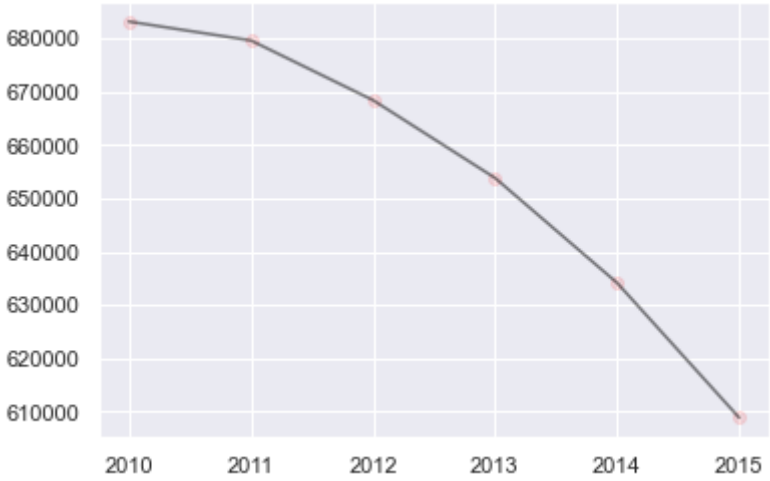
58600
58400
58200
58000
57800
2010
2011
2012
2013
2014
2015

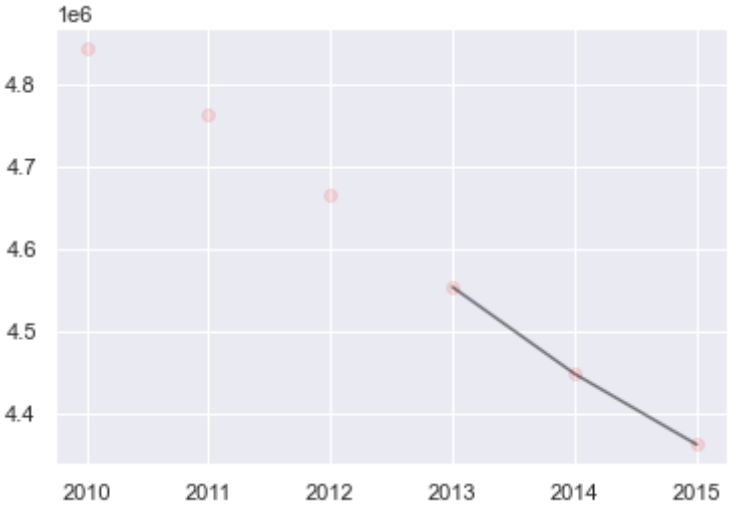

414000
413000
412000
411000
410000
409000
408000
407000
406000
2010
2011
2012
2013
2014
2015

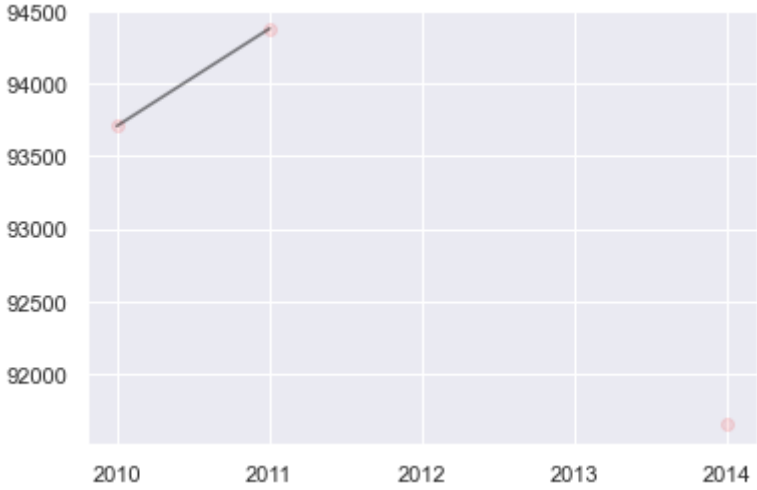

820000
800000
780000
760000
740000
720000
2010
2011
2012
2013
2014
2015

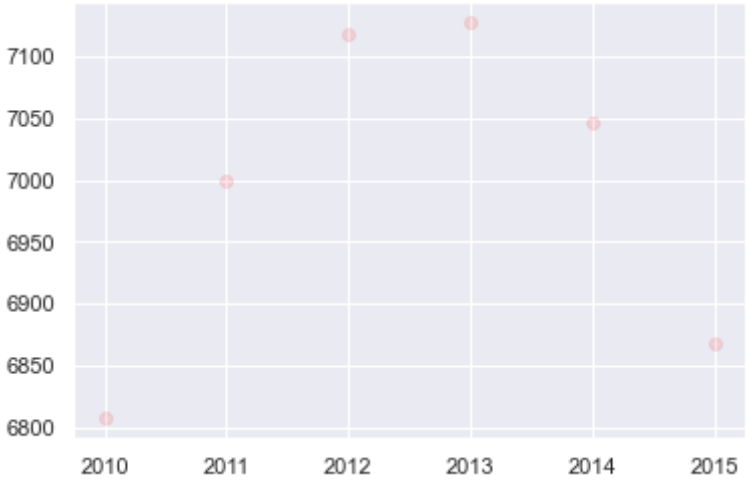

10750
10500
10250
10000
9750
9500
9250
9000
2010
2011
2012
2013
2014

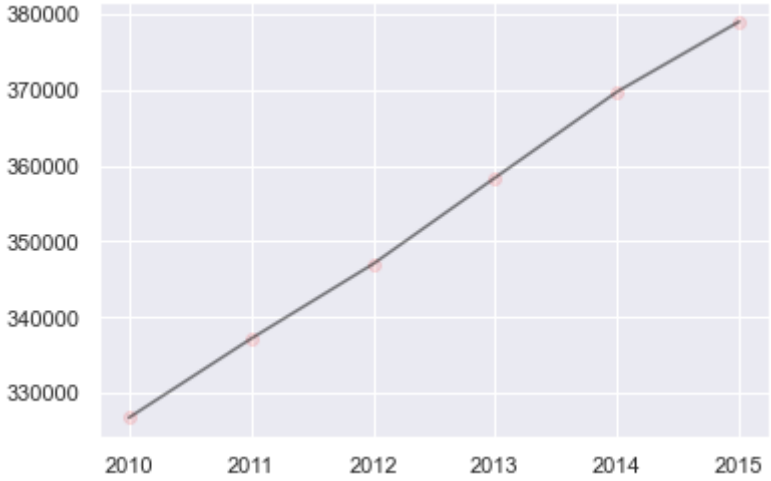

69000
68000
67000
66000
65000
64000
63000
62000
2010
2011
2012
2013
2014
2015

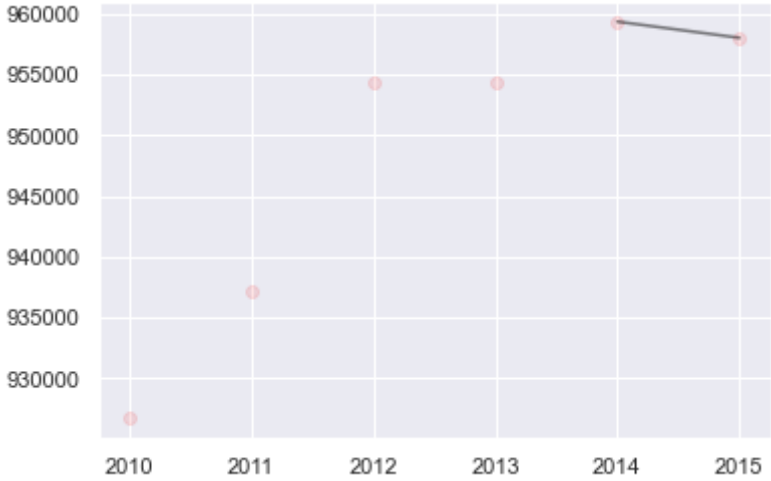

680000
670000
660000
650000
640000
630000
620000
610000
2010
2011
2012
2013
2014
2015

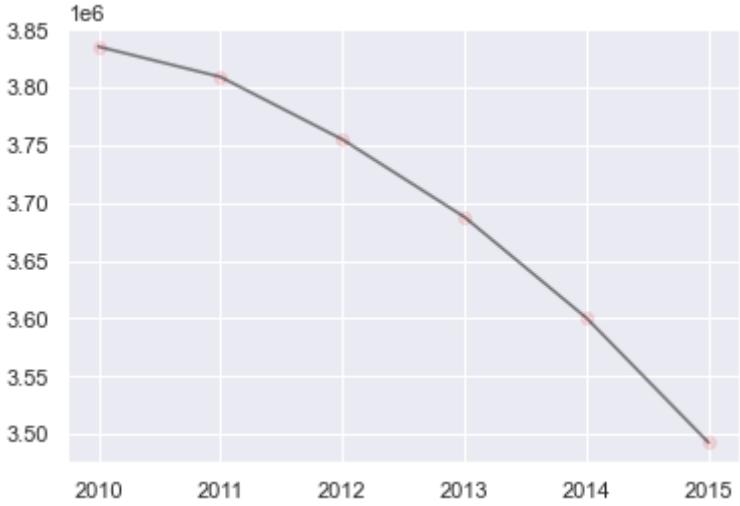

1e6
4.8
4.7
4.6
4.5
4.4
2010
2011
2012
2013
2014
2015

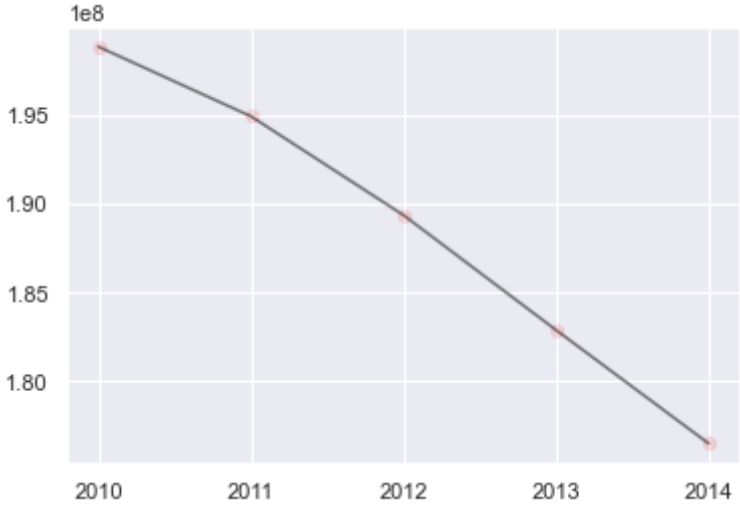

94500
94000
93500
93000
92500
92000
2010
2011
2012
2013
2014

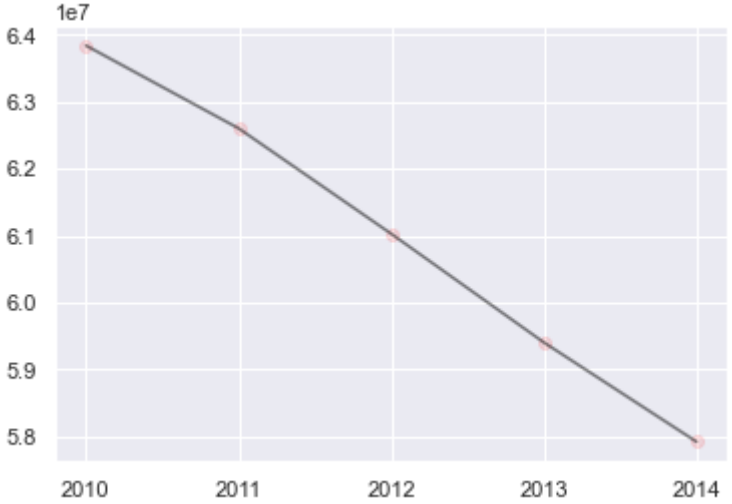

7100
7050
7000
6950
6900
6850
6800
2010
2011
2012
2013
2014
2015

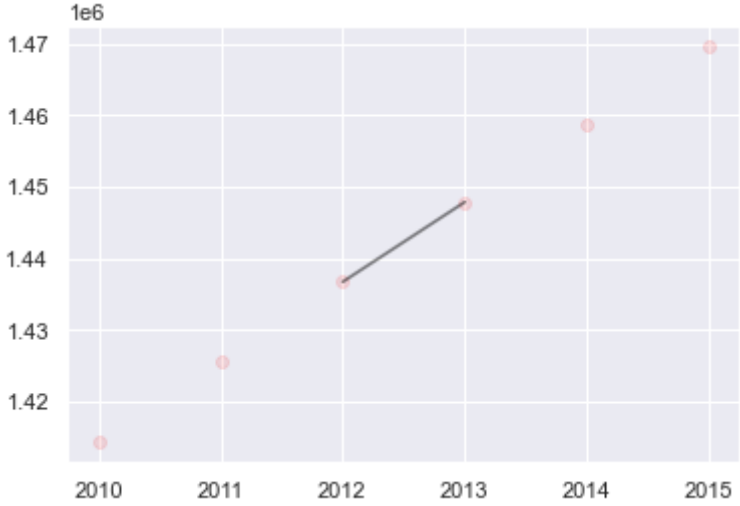

380000
370000
360000
350000
340000
330000
2010
2011
2012
2013
2014
2015

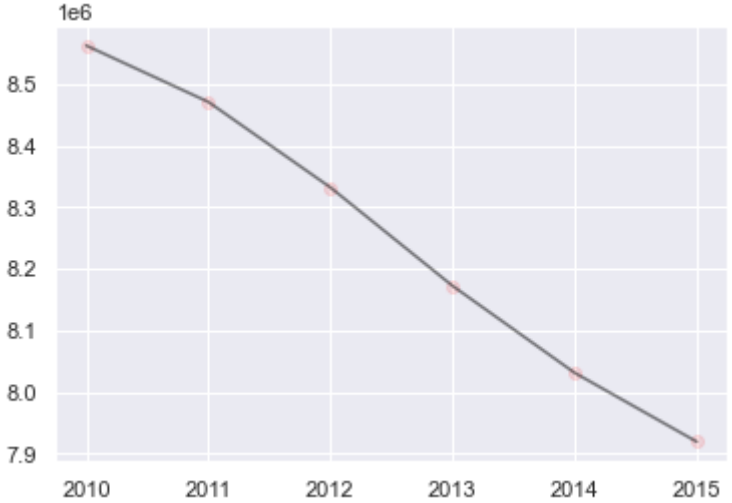

960000
955000
950000
945000
940000
935000
930000
2010
2011
2012
2013
2014
2015

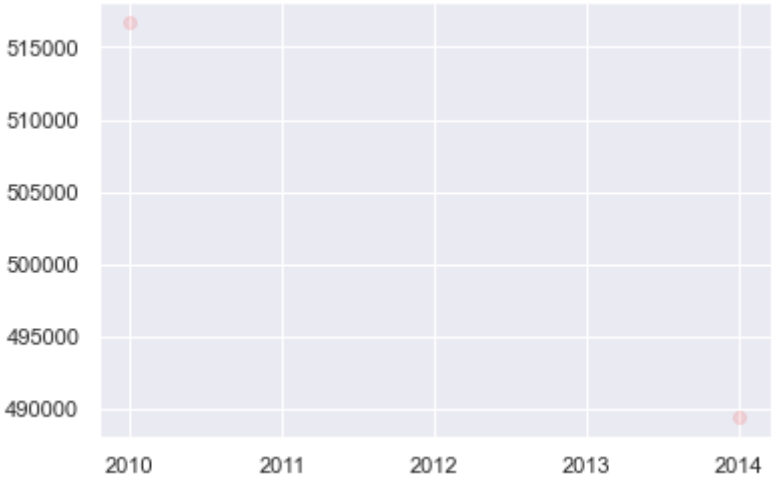

1e6
3.85
3.80
3.75
3.70
3.65
3.60
3.55
3.50
2010
2011
2012
2013
2014
2015

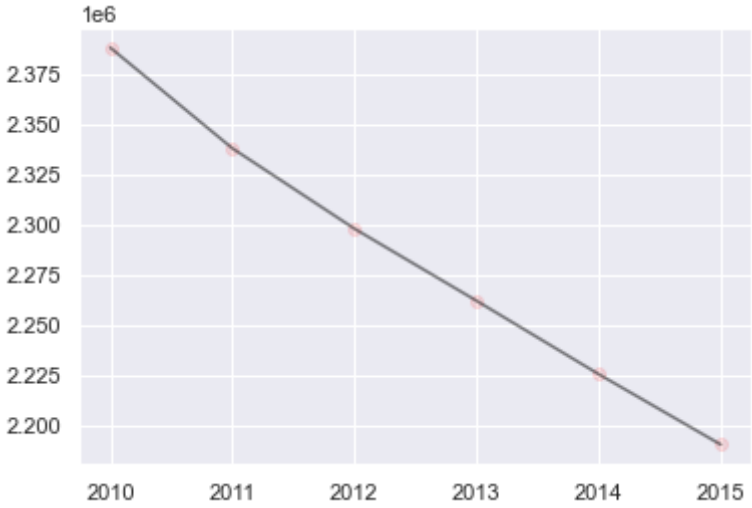

1e8
1.95
1.90
1.85
1.80
2010
2011
2012
2013
2014


1e7
6.4
6.3
6.2
6.1
6.0
5.9
5.8
2010
2011
2012
2013
2014

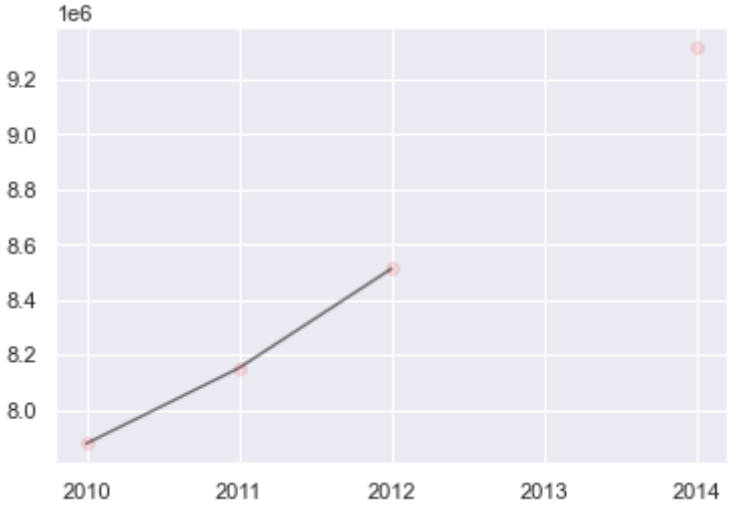

1e6
1.47
1.46
1.45
1.44
1.43
1.42
2010
2011
2012
2013
2014
2015

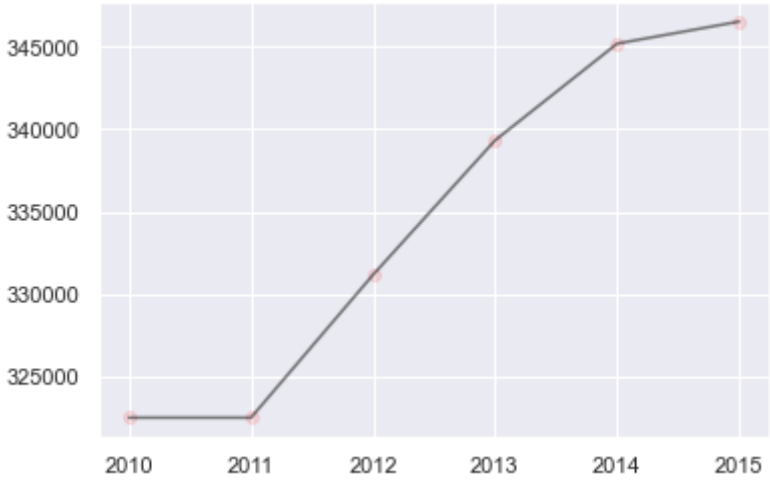

1e6
8.5
8.4
8.3
8.2
8.1
8.0
7.9
2010
2011
2012
2013
2014
2015

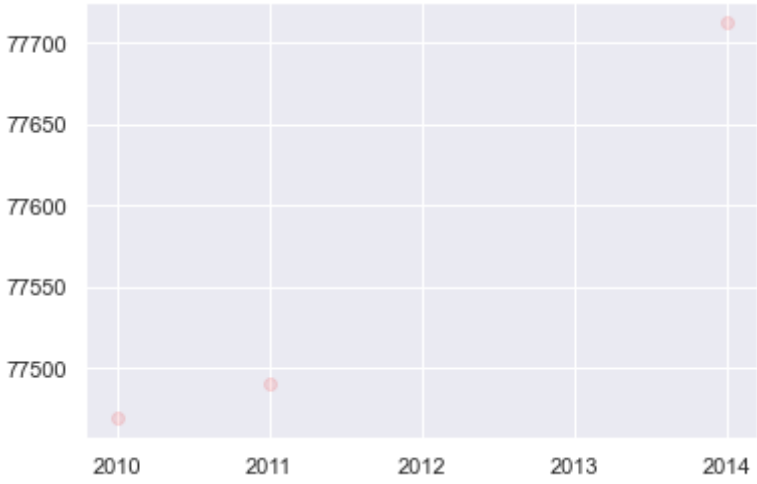

515000
510000
505000
500000
495000
490000
2010
2011
2012
2013
2014

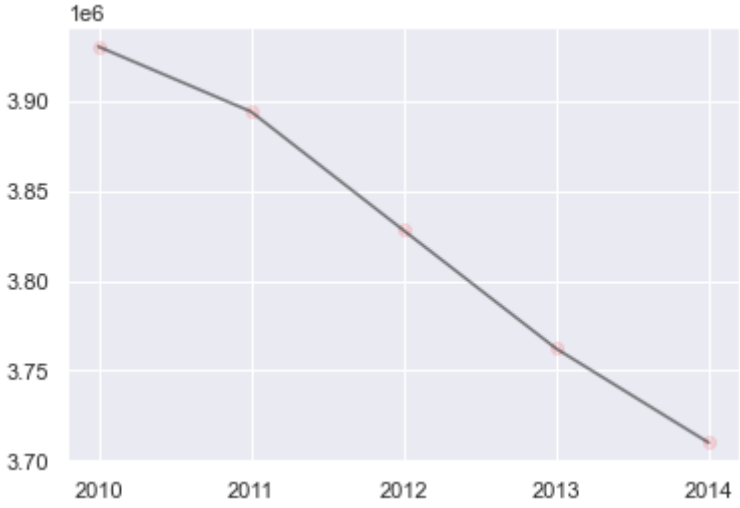

1e6
2.375
2.350
2.325
2.300
2.275
2.250
2.225
2.200
2010
2011
2012
2013
2014
2015

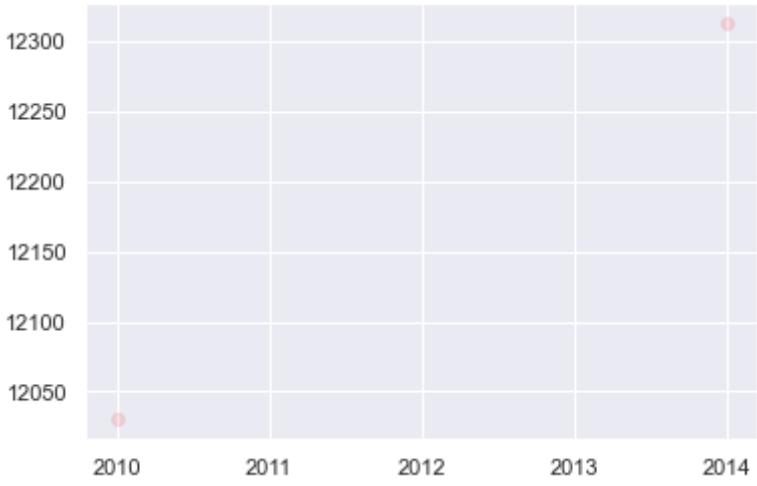
100000
95000
90000
85000
80000
2010
2011
2012
2013
2014
2015

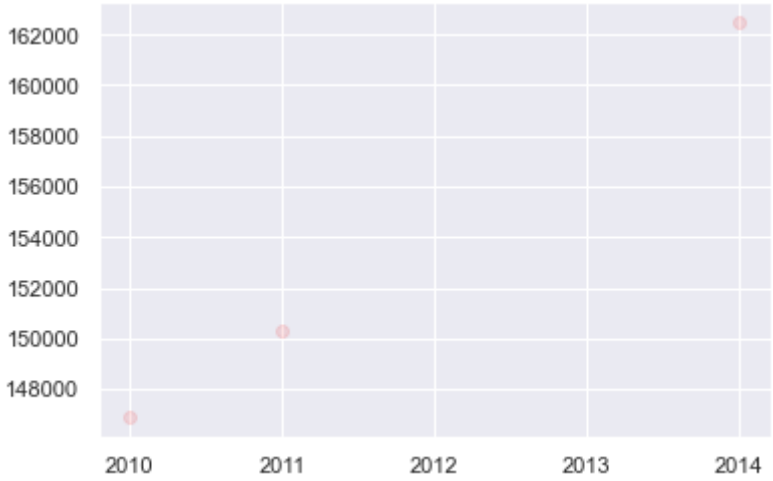

1e6
9.2
9.0
8.8
8.6
8.4
8.2
8.0
2010
2011
2012
2013
2014

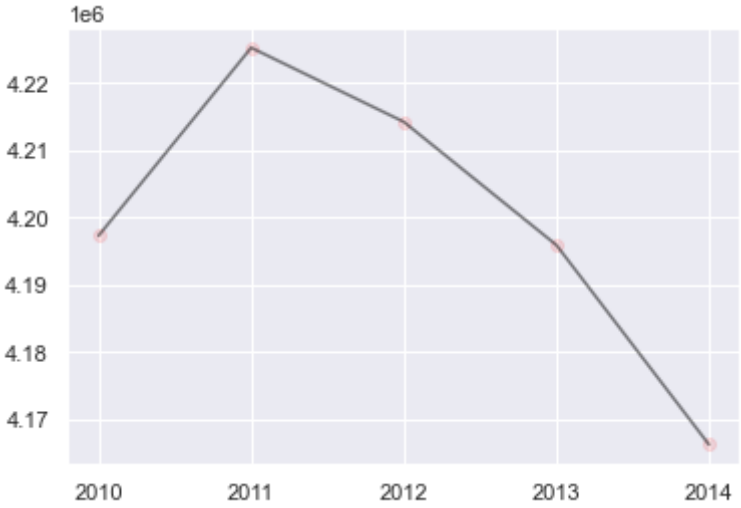

345000
340000
335000
330000
325000
2010
2011
2012
2013
2014
2015

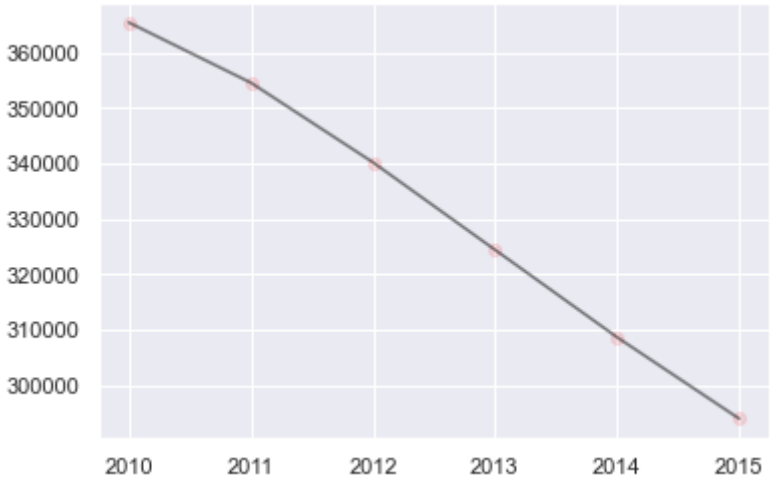

77700
77650
77600
77550
77500
2010
2011
2012
2013
2014

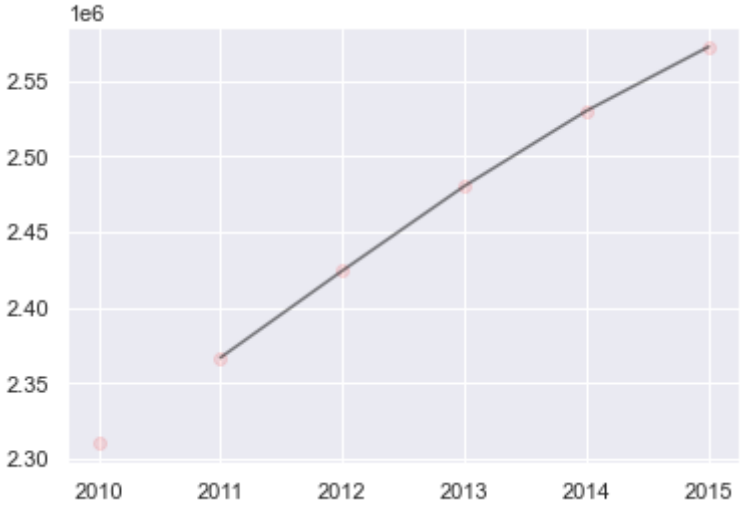

1e6
3.90
3.85
3.80
3.75
3.70
2010
2011
2012
2013
2014

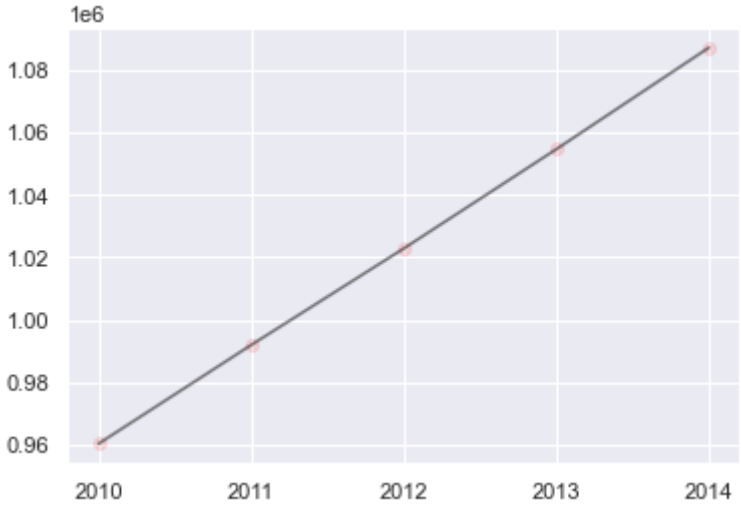

12300
12250
12200
12150
12100
12050
2010
2011
2012
2013
2014

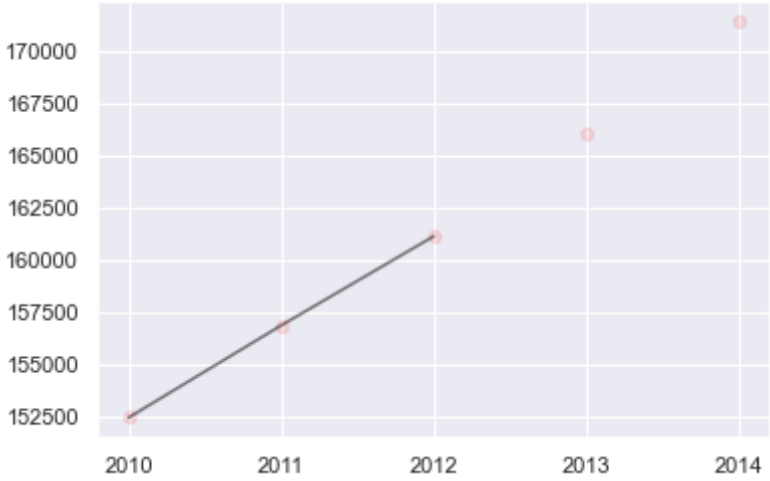

162000
160000
158000
156000
154000
152000
150000
148000
2010
2011
2012
2013
2014

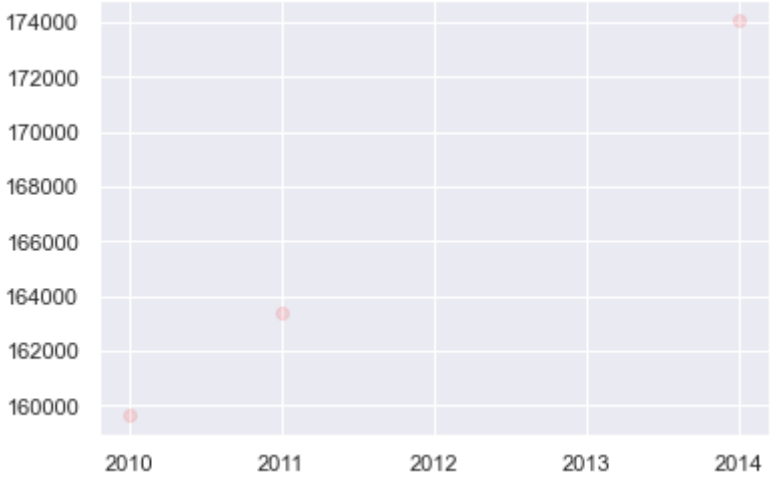

1e6
4.22
4.21
4.20
4.19
4.18
4.17
2010
2011
2012
2013
2014


360000
350000
340000
330000
320000
310000
300000
2010
2011
2012
2013
2014
2015


1e6
2.55
2.50
2.45
2.40
2.35
2.30
2010
2011
2012
2013
2014
2015

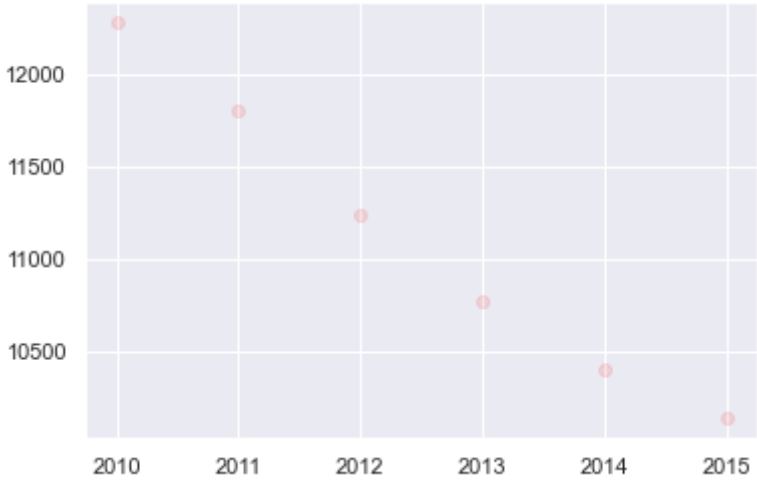

1e6
1.08
1.06
1.04
1.02
1.00
0.98
0.96
2010
2011
2012
2013
2014

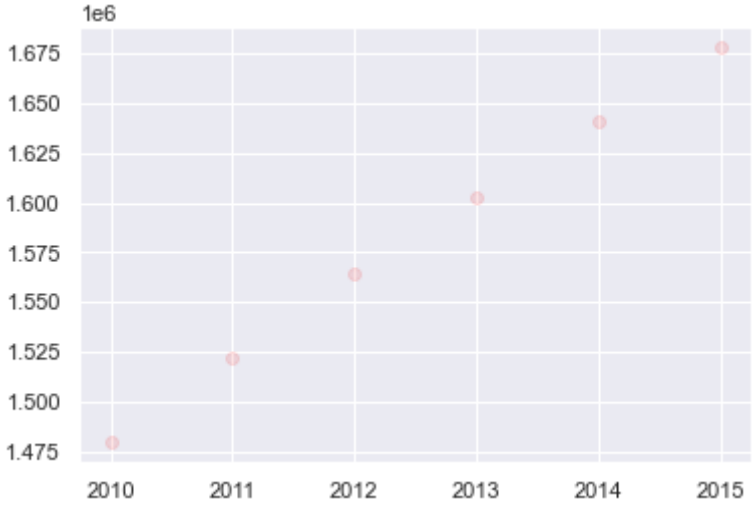

170000
167500
165000
162500
160000
157500
155000
152500
2010
2011
2012
2013
2014

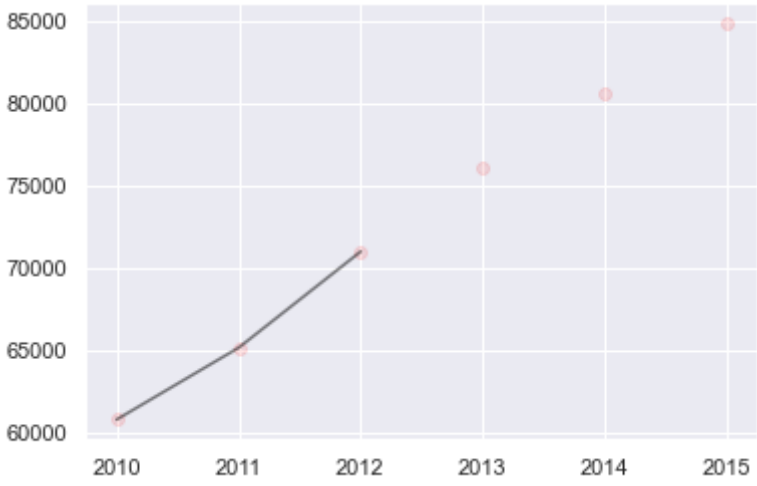

174000
172000
170000
168000
166000
164000
162000
160000
2010
2011
2012
2013
2014

74000
72000
70000
68000
66000
2010
2011
2012
2013
2014

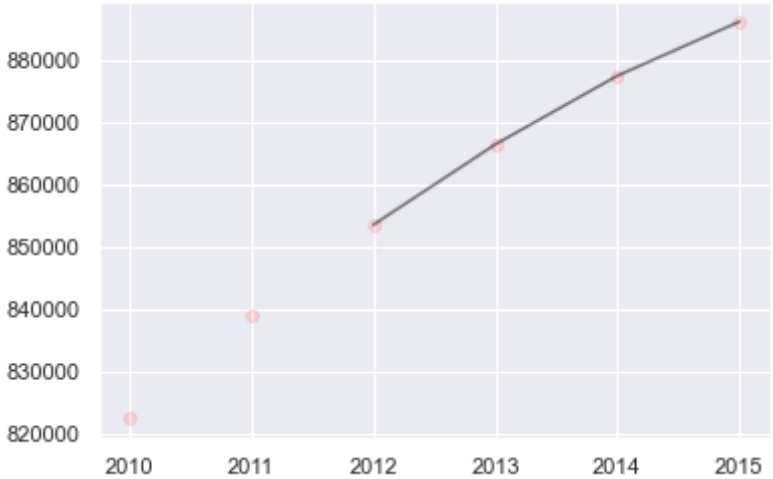
625000
620000
615000
610000
605000
600000
595000
2010
2011
2012
2013
2014


12000
11500
11000
10500
2010
2011
2012
2013
2014
2015

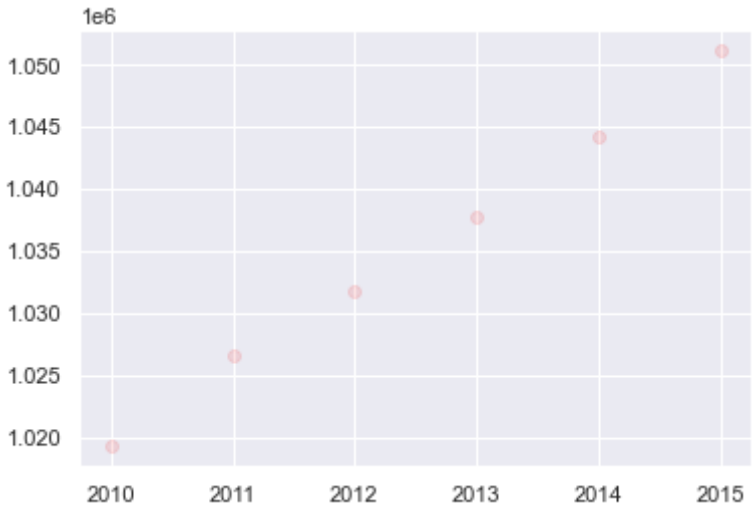

1e6
1.675
1.650
1.625
1.600
1.575
1.550
1.525
1.500
1.475
2010
2011
2012
2013
2014
2015

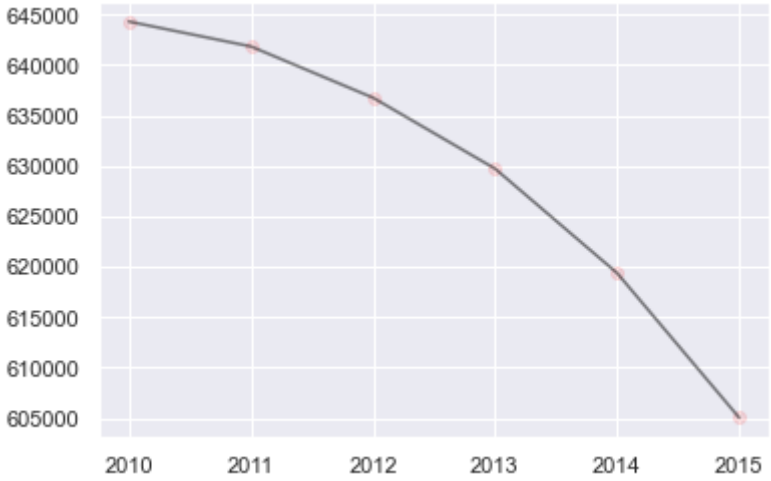

85000
80000
75000
70000
65000
60000
2010
2011
2012
2013
2014
2015

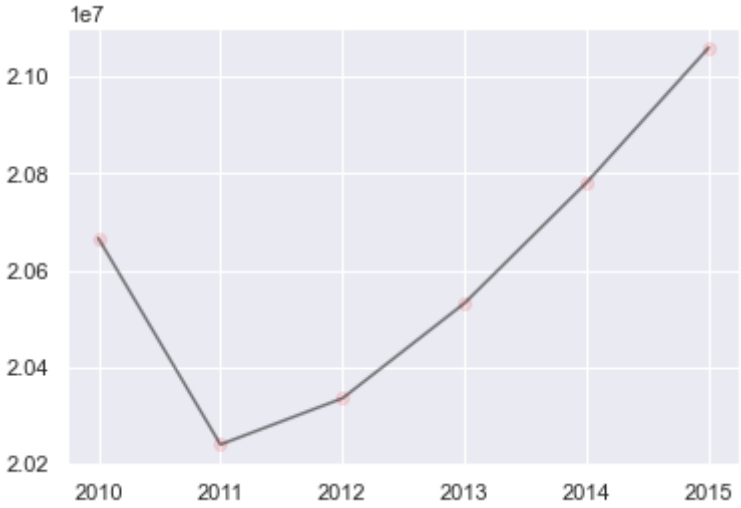
452500
450000
447500
445000
442500
440000
437500
2010
2011
2012
2013
2014
2015

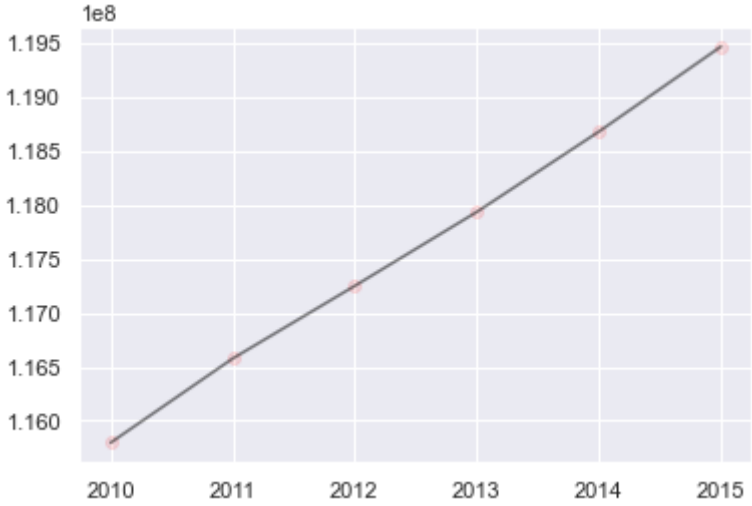

880000
870000
860000
850000
840000
830000
820000
2010
2011
2012
2013
2014
2015

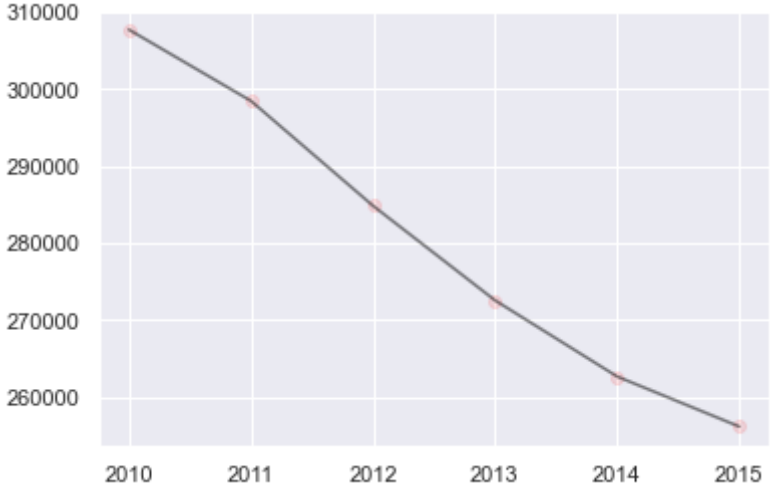
275000
270000
265000
260000
255000
250000
245000
240000
235000
2010
2011
2012
2013
2014
2015

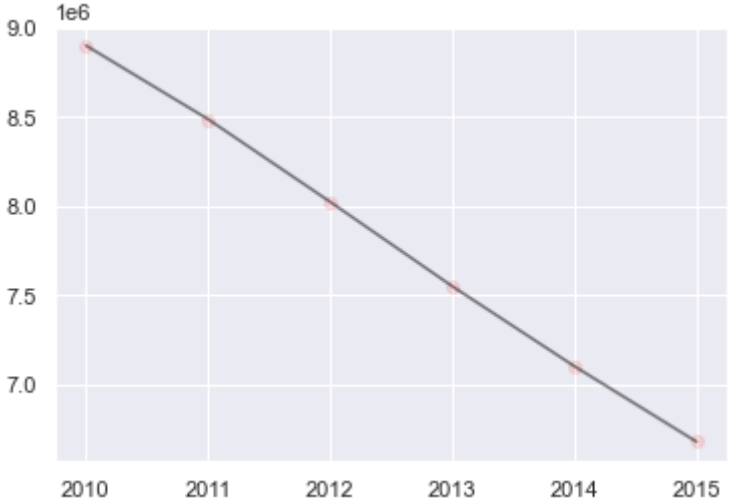

1e6
1.050
1.045
1.040
1.035
1.030
1.025
1.020
2010
2011
2012
2013
2014
2015

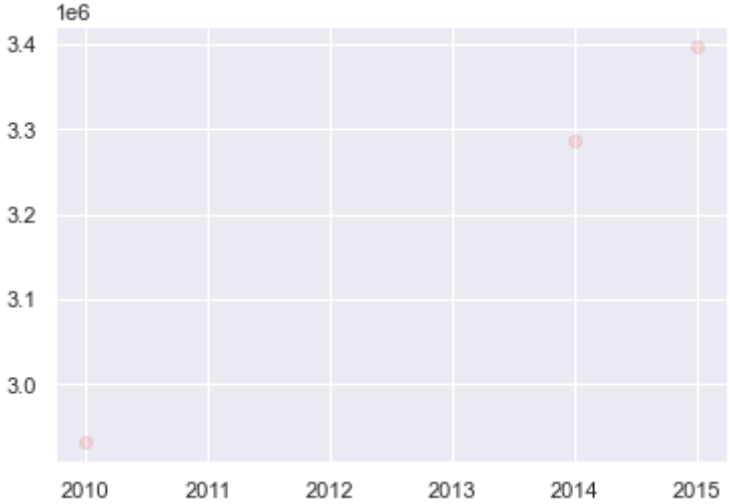

645000
640000
635000
630000
625000
620000
615000
610000
605000
2010
2011
2012
2013
2014
2015

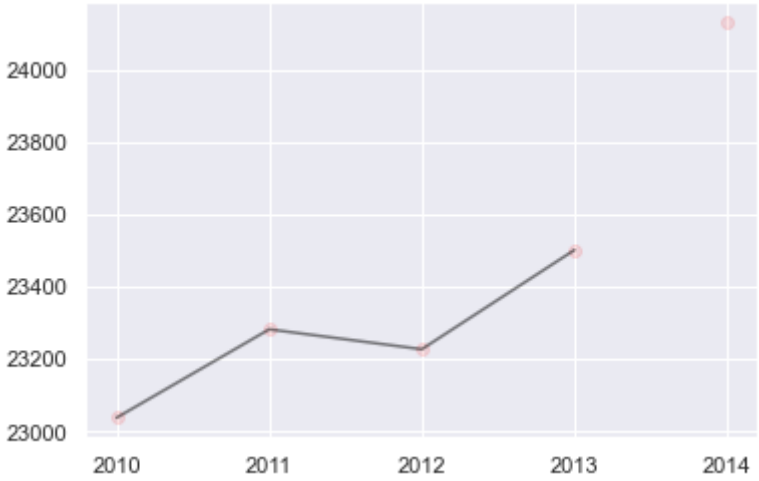

1e7
2.10
2.08
2.06
2.04
2.02
2010
2011
2012
2013
2014
2015

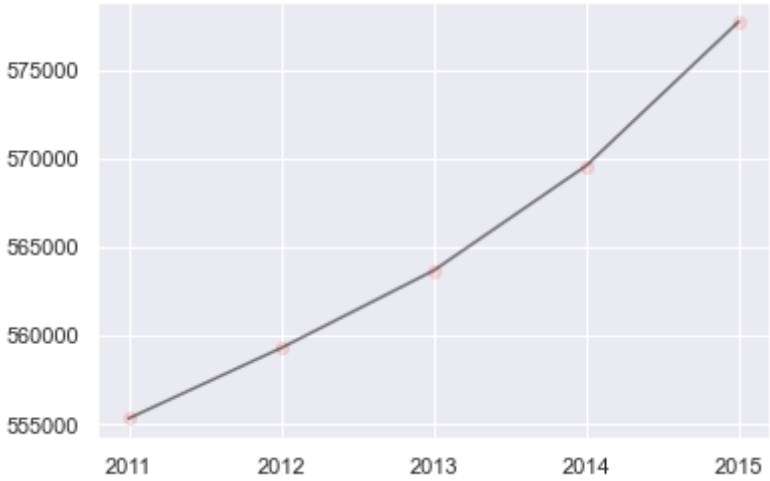

1e8
1.195
1.190
1.185
1.180
1.175
1.170
1.165
1.160
2010
2011
2012
2013
2014
2015

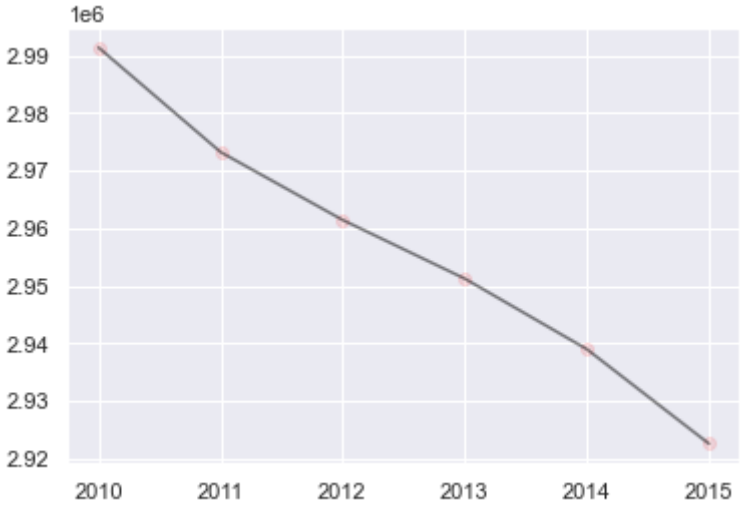

310000
300000
290000
280000
270000
260000
2010
2011
2012
2013
2014
2015


1e6
9.0
8.5
8.0
7.5
7.0
2010
2011
2012
2013
2014
2015

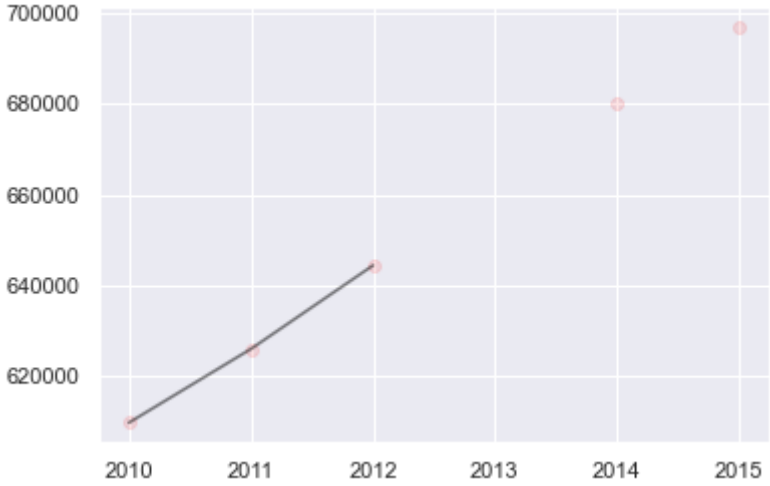

1e6
3.4
3.3
3.2
3.1
3.0
2010
2011
2012
2013
2014
2015

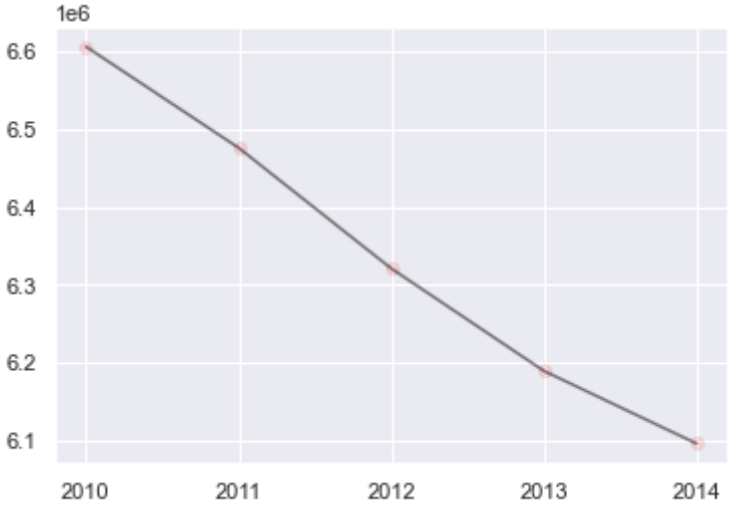

24000
23800
23600
23400
23200
23000
2010
2011
2012
2013
2014

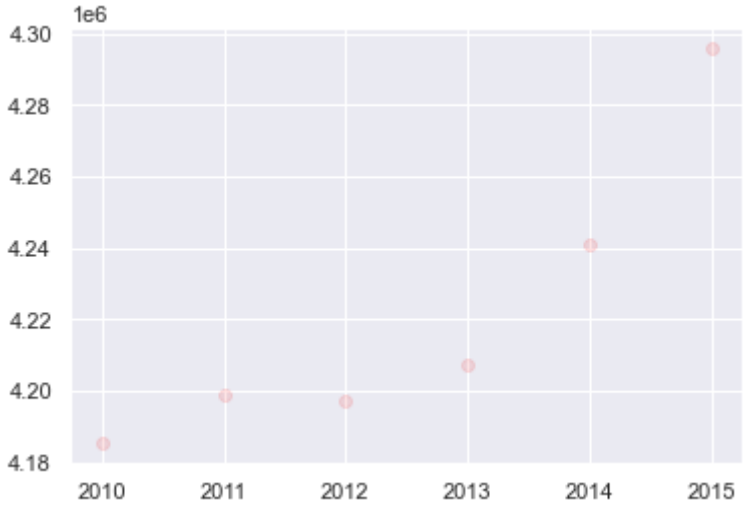

575000
570000
565000
560000
555000
2011
2012
2013
2014
2015

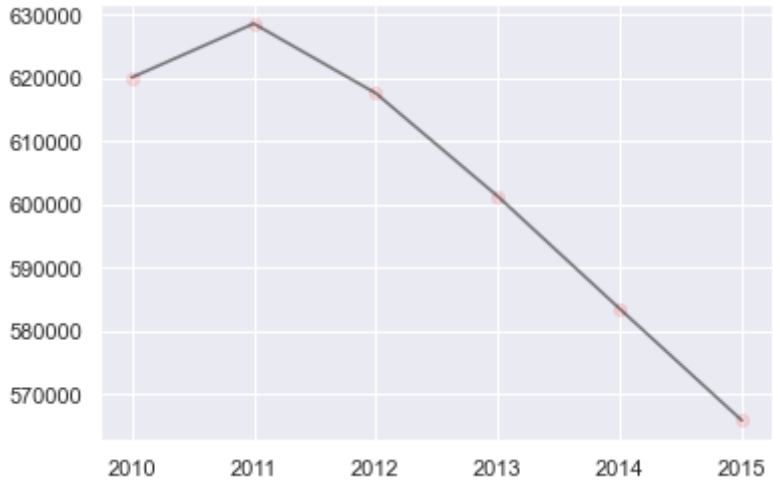

1e6
2.99
2.98
2.97
2.96
2.95
2.94
2.93
2.92
2010
2011
2012
2013
2014
2015

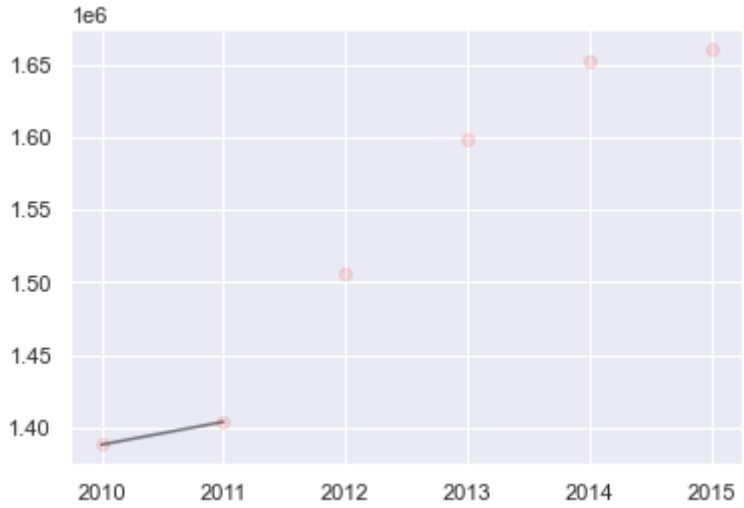
274000
272000
270000
268000
266000
264000
2010
2011
2012
2013
2014
2015

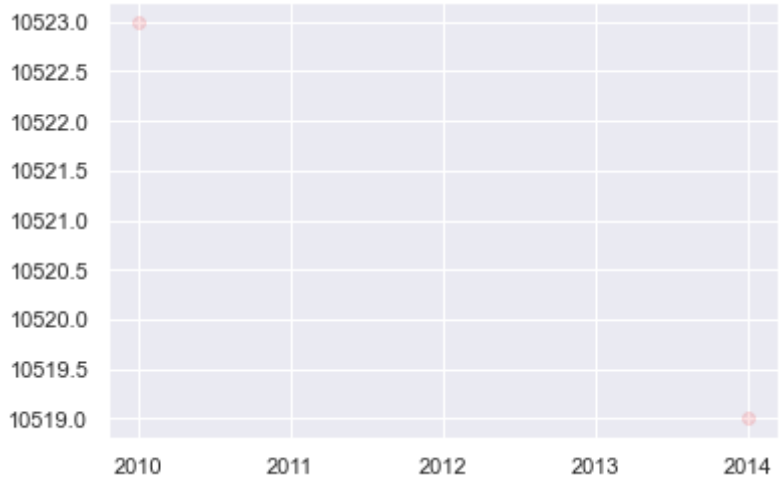

700000
680000
660000
640000
620000
2010
2011
2012
2013
2014
2015

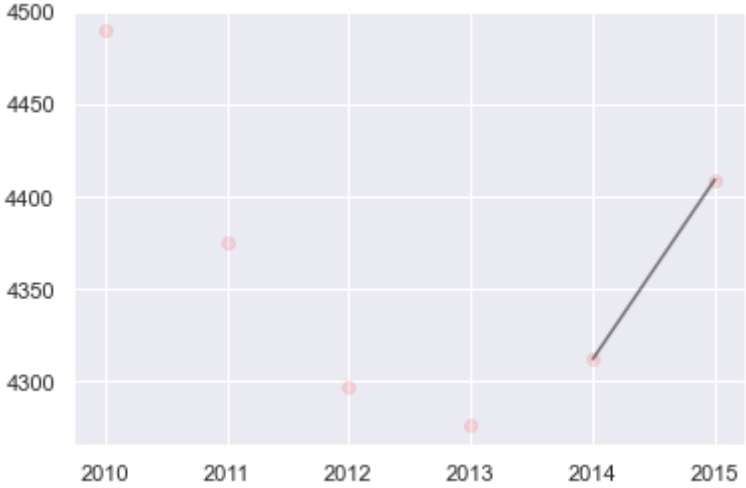

1e6
6.6
6.5
6.4
6.3
6.2
6.1
2010
2011
2012
2013
2014

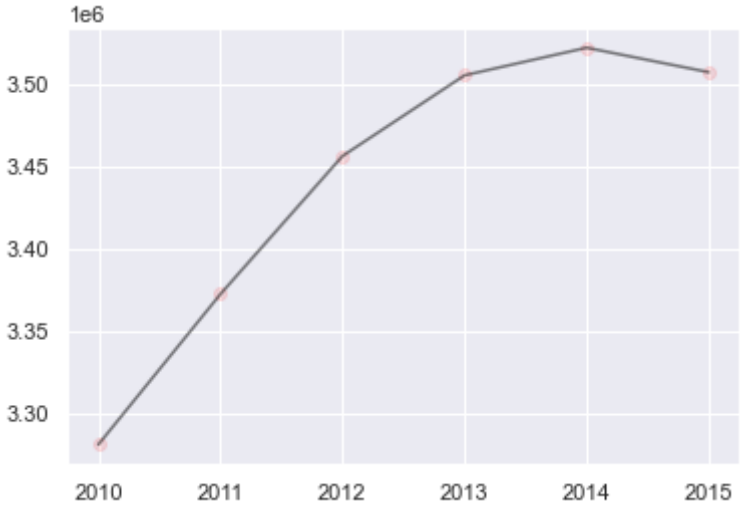

1e6
4.30
4.28
4.26
4.24
4.22
4.20
4.18
2010
2011
2012
2013
2014
2015

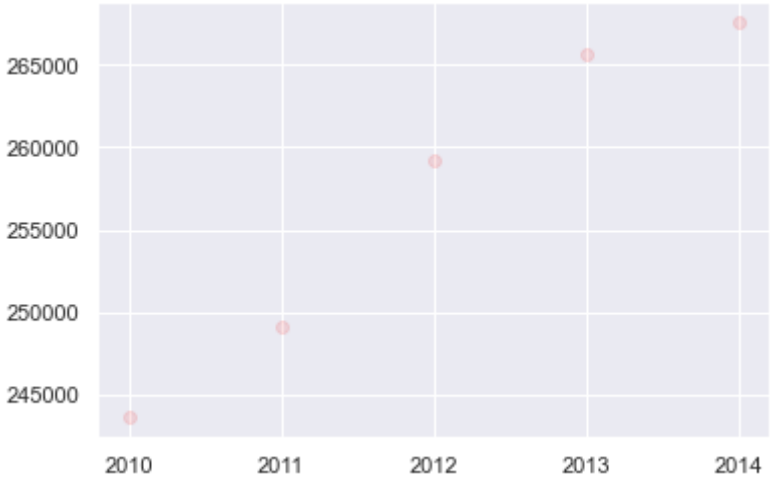

630000
620000
610000
600000
590000
580000
570000
2010
2011
2012
2013
2014
2015

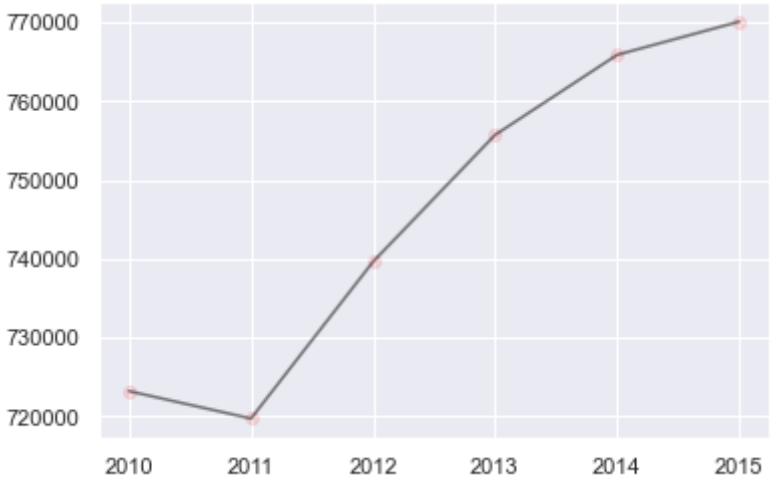

1e6
1.65
1.60
1.55
1.50
1.45
1.40
2010
2011
2012
2013
2014
2015

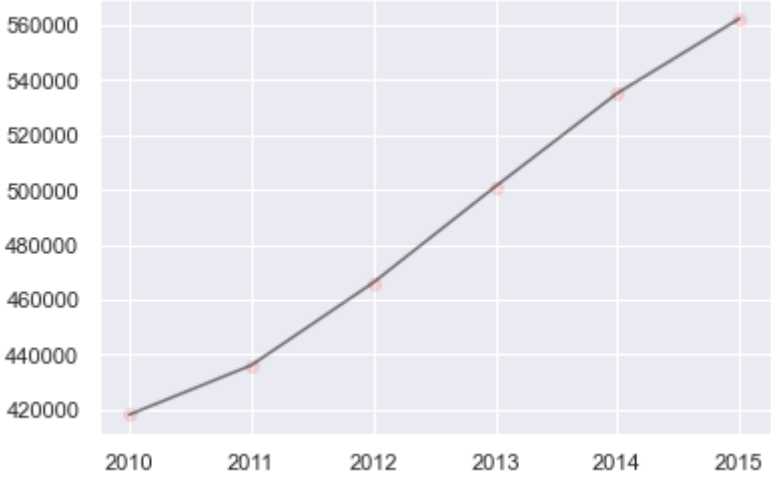

10523.0
10522.5
10522.0
10521.5
10521.0
10520.5
10520.0
10519.5
10519.0
2010
2011
2012
2013
2014

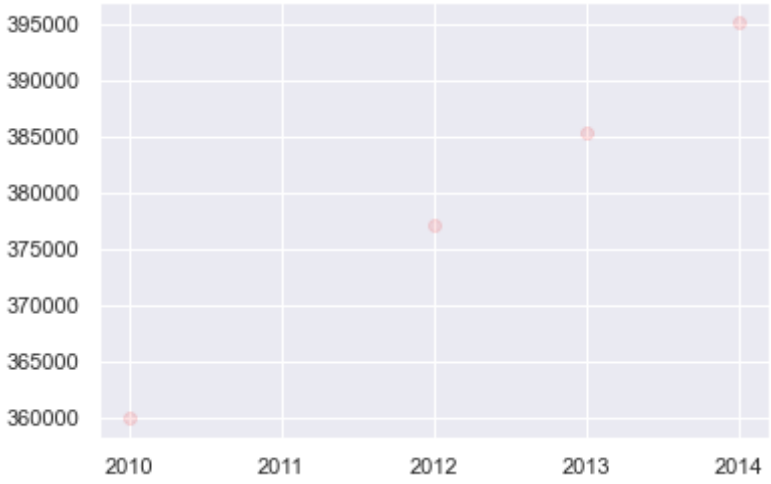

4500
4450
4400
4350
4300
2010
2011
2012
2013
2014
2015


1e6
3.50
3.45
3.40
3.35
3.30
2010
2011
2012
2013
2014
2015

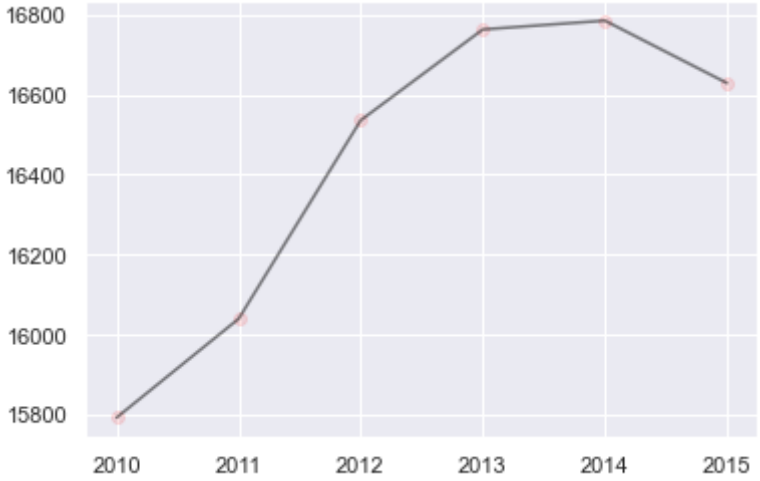

265000
260000
255000
250000
245000
2010
2011
2012
2013
2014

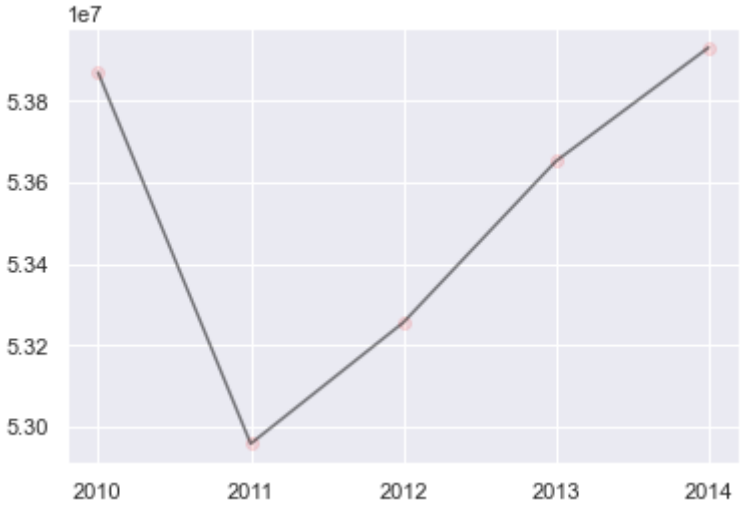

770000
760000
750000
740000
730000
720000
2010
2011
2012
2013
2014
2015

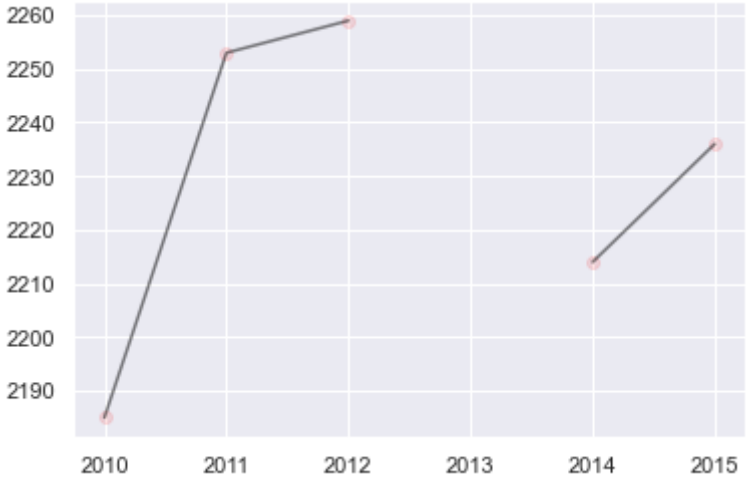

560000
540000
520000
500000
480000
460000
440000
420000
2010
2011
2012
2013
2014
2015

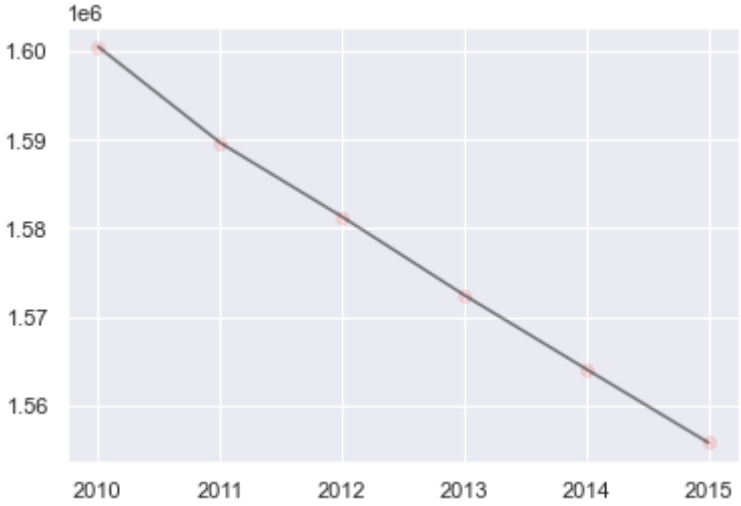

395000
390000
385000
380000
375000
370000
365000
360000
2010
2011
2012
2013
2014

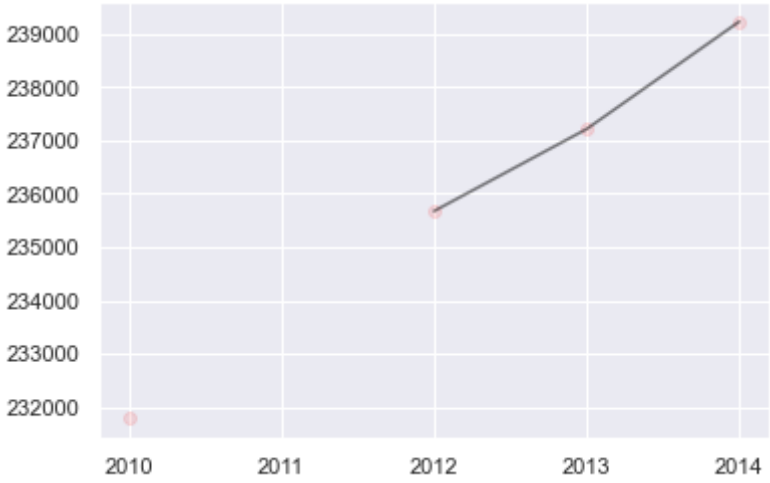
620000
600000
580000
560000
540000
520000
2010
2011
2012
2013
2014

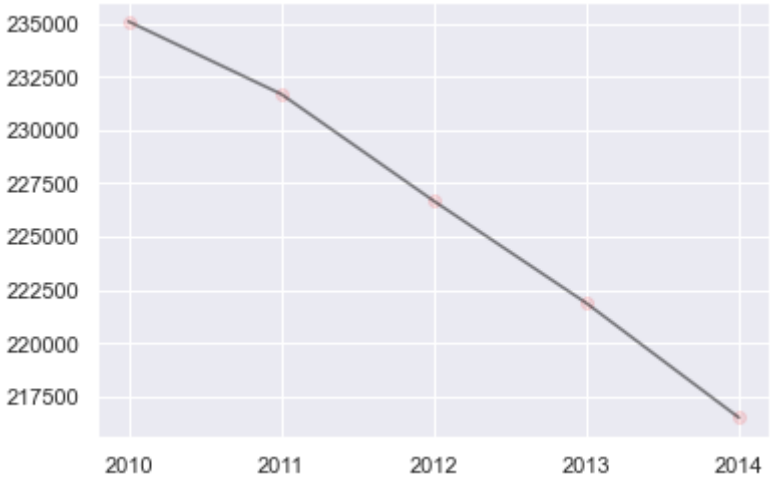

16800
16600
16400
16200
16000
15800
2010
2011
2012
2013
2014
2015

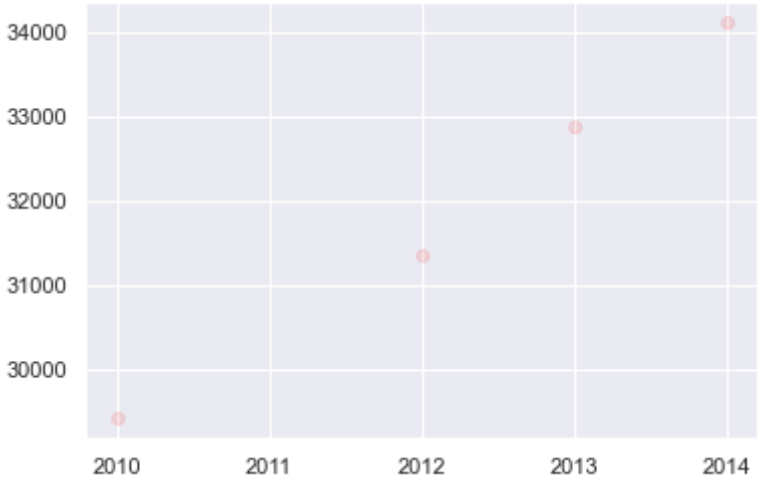

1e7
5.38
5.36
5.34
5.32
5.30
2010
2011
2012
2013
2014

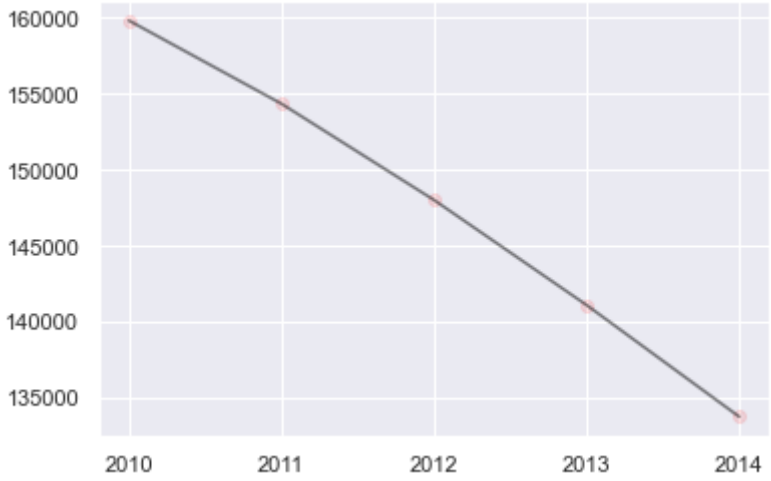

2260
2250
2240
2230
2220
2210
2200
2190
2010
2011
2012
2013
2014
2015

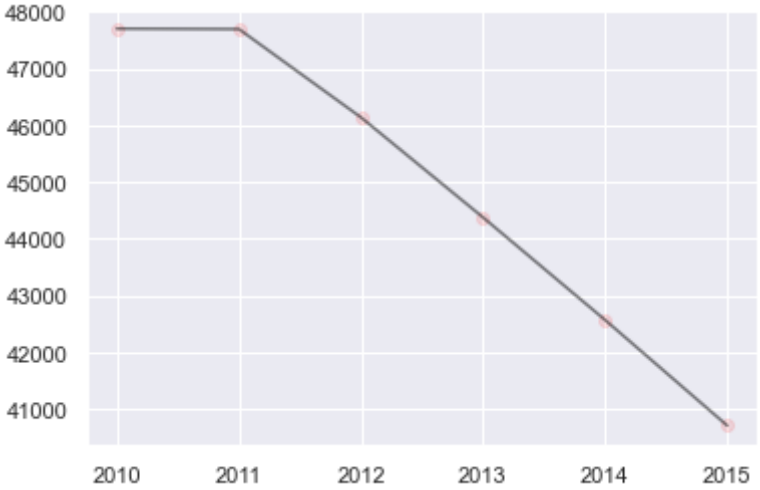

1e6
1.60
1.59
1.58
1.57
1.56
2010
2011
2012
2013
2014
2015

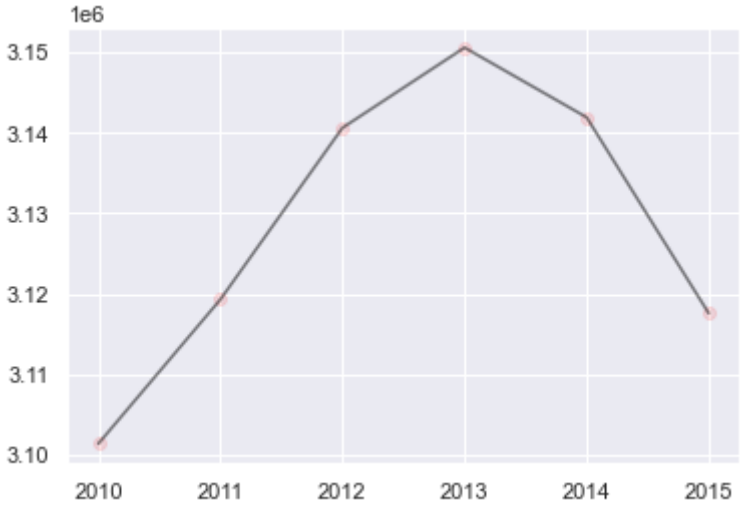

239000
238000
237000
236000
235000
234000
233000
232000
2010
2011
2012
2013
2014


235000
232500
230000
227500
225000
222500
220000
217500
2010
2011
2012
2013
2014

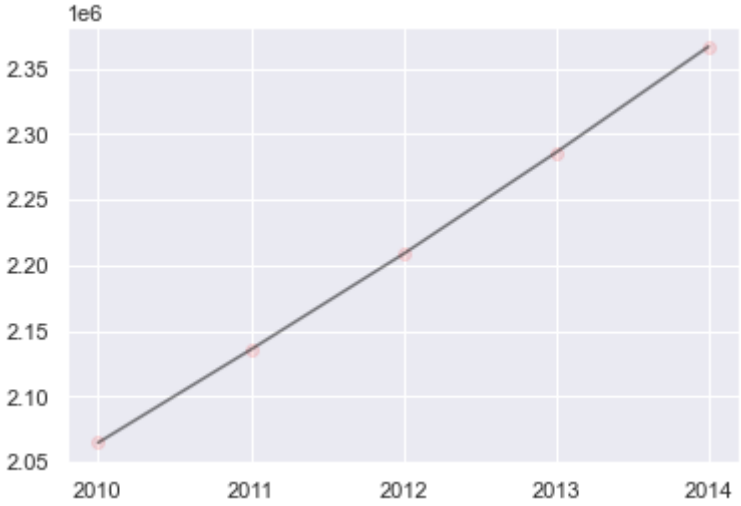

34000
33000
32000
31000
30000
2010
2011
2012
2013
2014

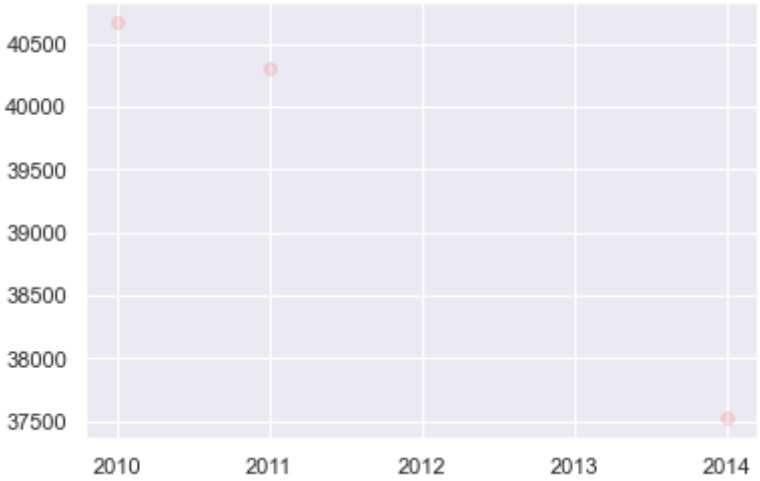

160000
155000
150000
145000
140000
135000
2010
2011
2012
2013
2014

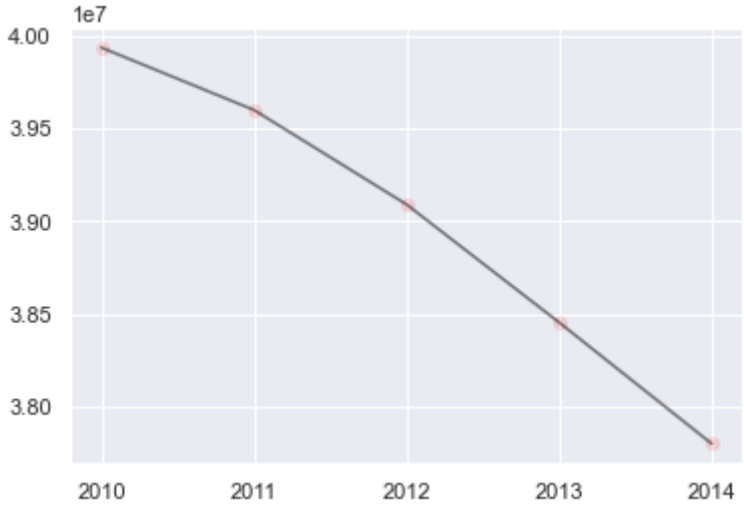

48000
47000
46000
45000
44000
43000
42000
41000
2010
2011
2012
2013
2014
2015

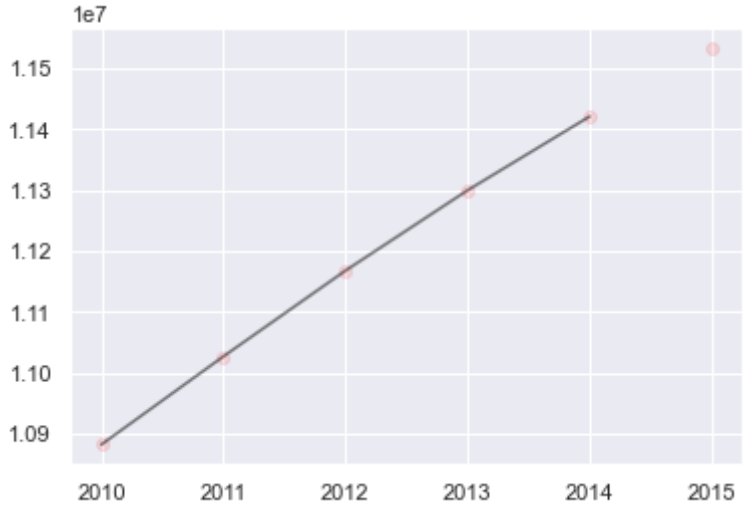

1e6
3.15
3.14
3.13
3.12
3.11
3.10
2010
2011
2012
2013
2014
2015

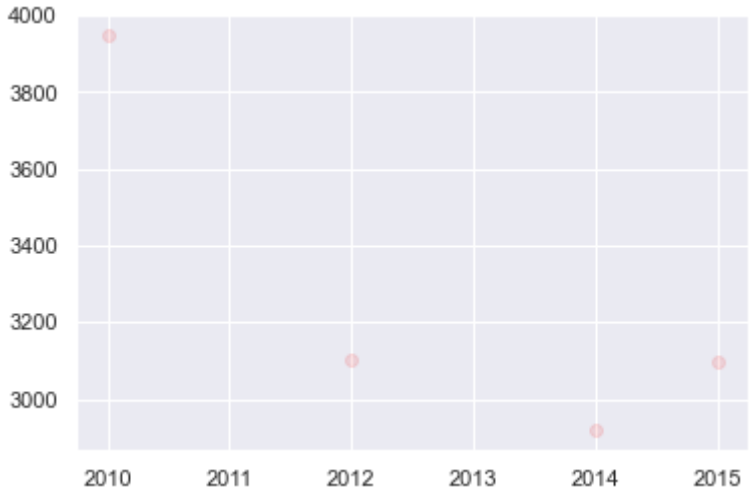
340000
330000
320000
310000
300000
290000
280000
270000
2010
2011
2012
2013
2014
2015

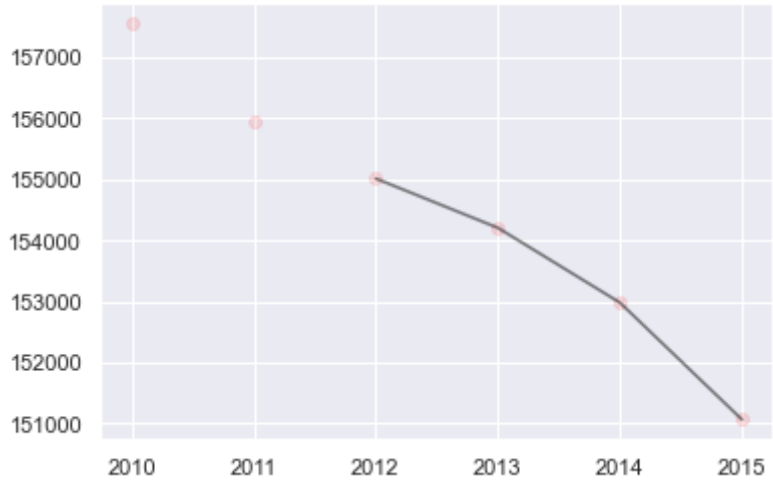

1e6
2.35
2.30
2.25
2.20
2.15
2.10
2.05
2010
2011
2012
2013
2014

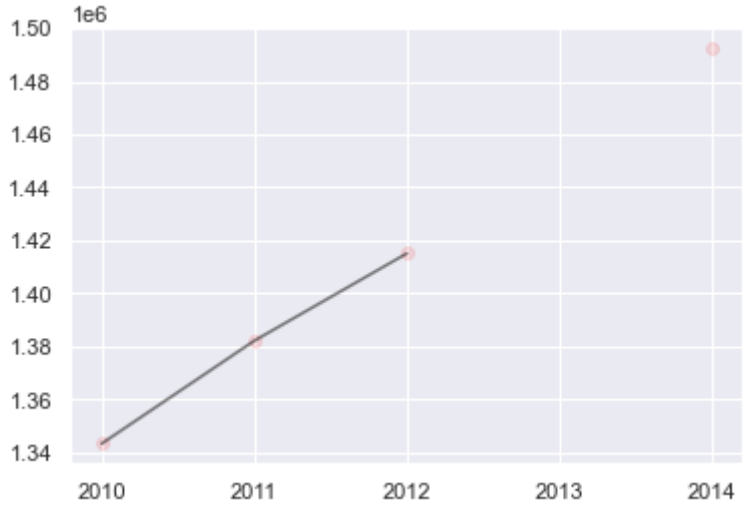

40500
40000
39500
39000
38500
38000
37500
2010
2011
2012
2013
2014


1e7
4.00
3.95
3.90
3.85
3.80
2010
2011
2012
2013
2014

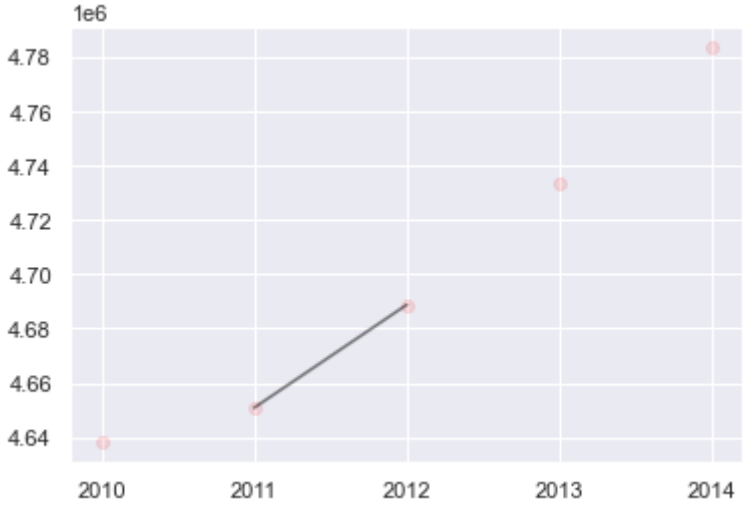

1e7
1.15
1.14
1.13
1.12
1.11
1.10
1.09
2010
2011
2012
2013
2014
2015

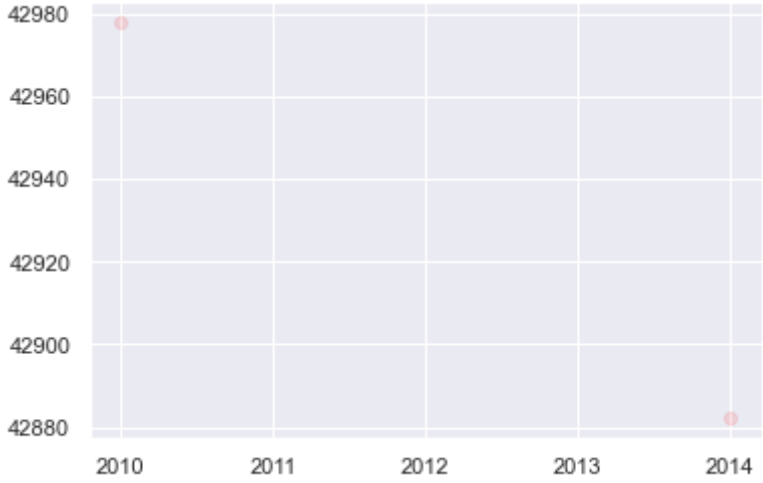

4000
3800
3600
3400
3200
3000
2010
2011
2012
2013
2014
2015

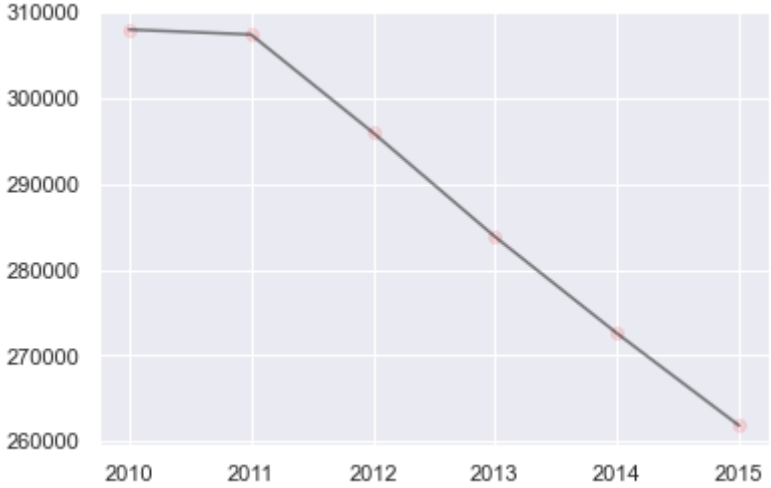

157000
156000
155000
154000
153000
152000
151000
2010
2011
2012
2013
2014
2015

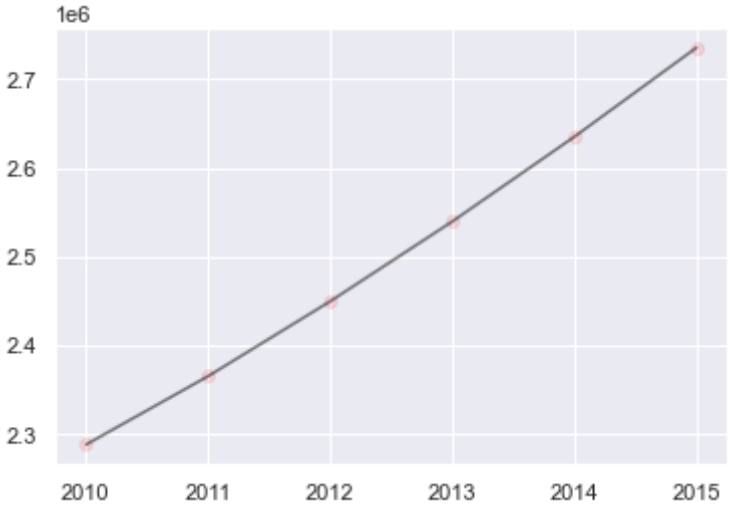

1e6
1.50
1.48
1.46
1.44
1.42
1.40
1.38
1.36
1.34
2010
2011
2012
2013
2014

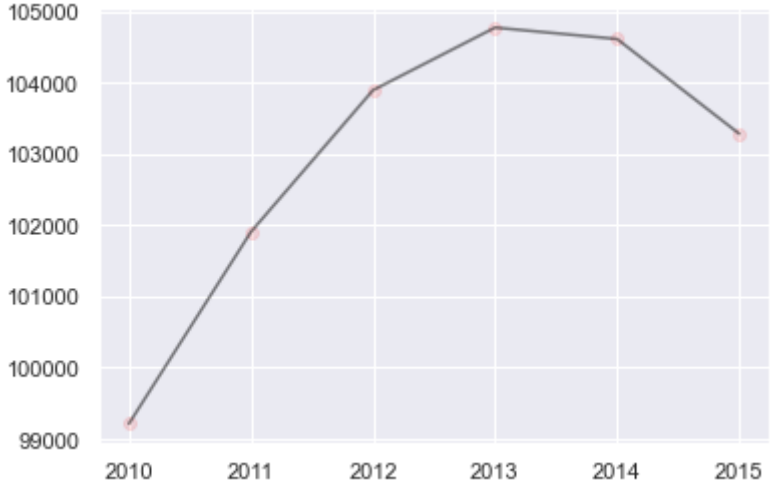
29200
29000
28800
28600
28400
28200
28000
27800
2010
2011
2012
2013
2014
2015

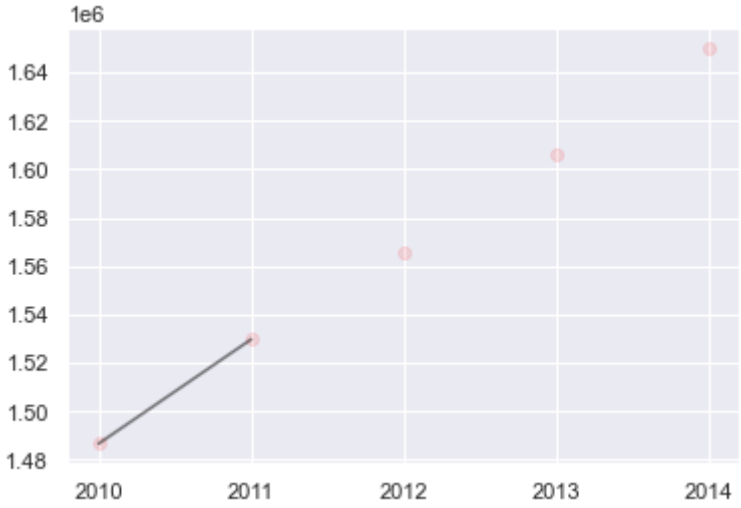

1e6
4.78
4.76
4.74
4.72
4.70
4.68
4.66
4.64
2010
2011
2012
2013
2014

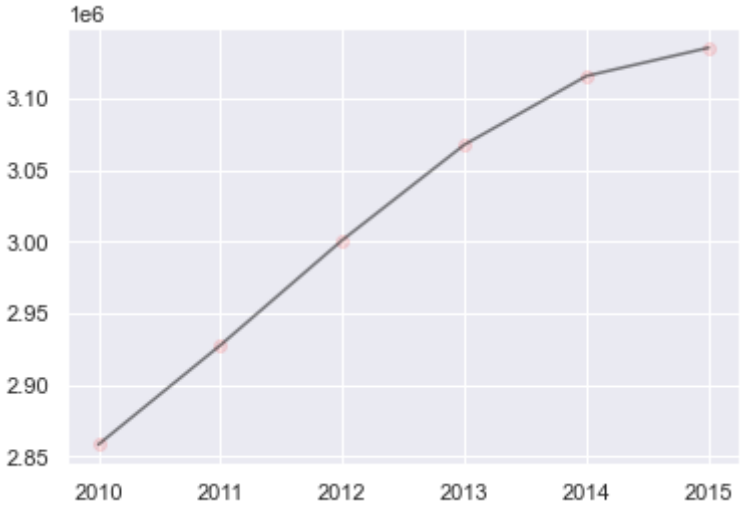

42980
42960
42940
42920
42900
42880
2010
2011
2012
2013
2014

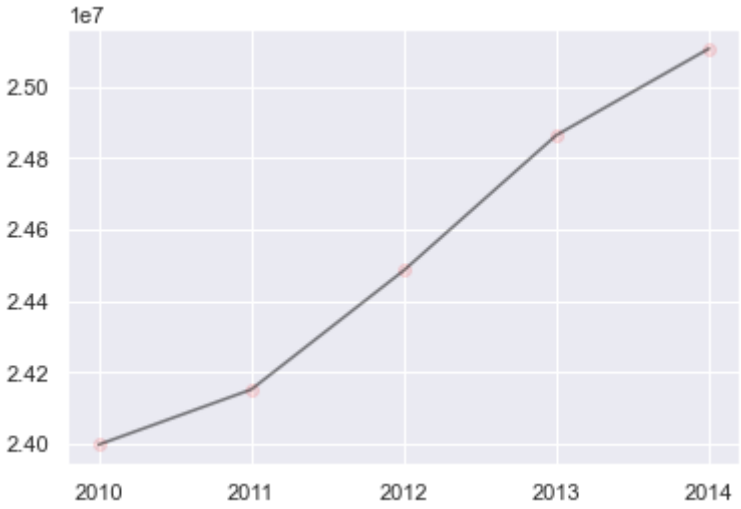

310000
300000
290000
280000
270000
260000
2010
2011
2012
2013
2014
2015

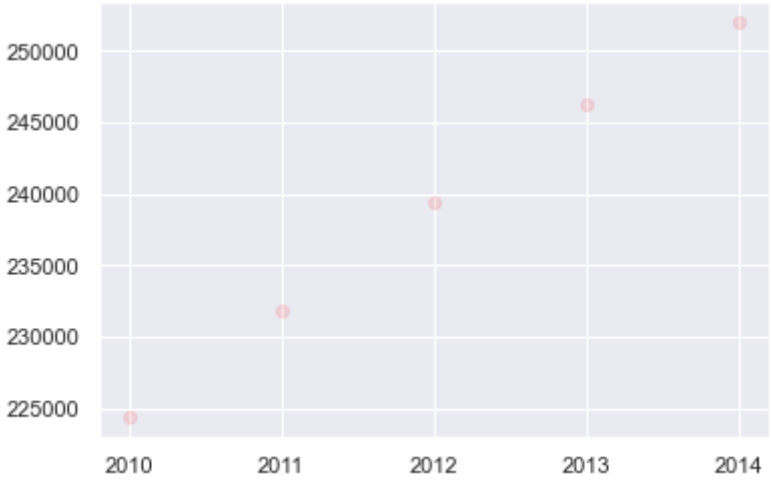

1e6
2.7
2.6
2.5
2.4
2.3
2010
2011
2012
2013
2014
2015

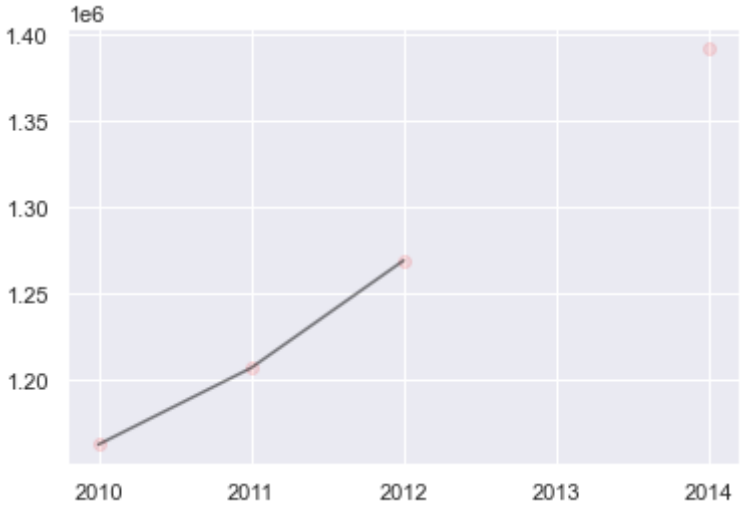

105000
104000
103000
102000
101000
100000
99000
2010
2011
2012
2013
2014
2015

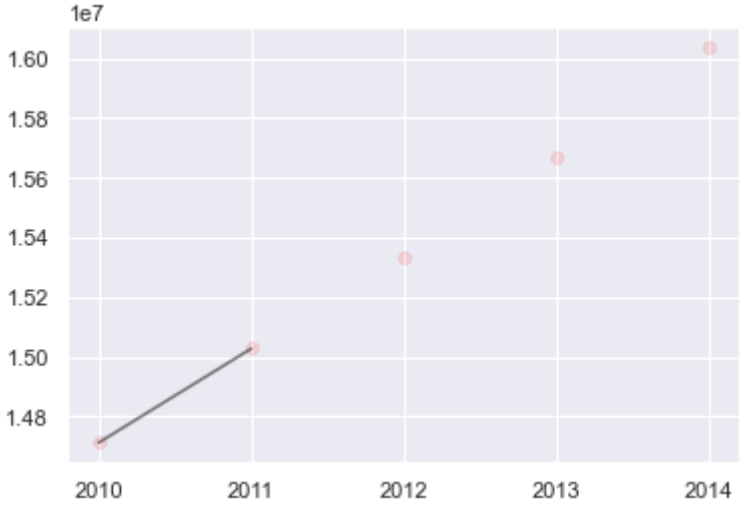

1e6
1.64
1.62
1.60
1.58
1.56
1.54
1.52
1.50
1.48
2010
2011
2012
2013
2014

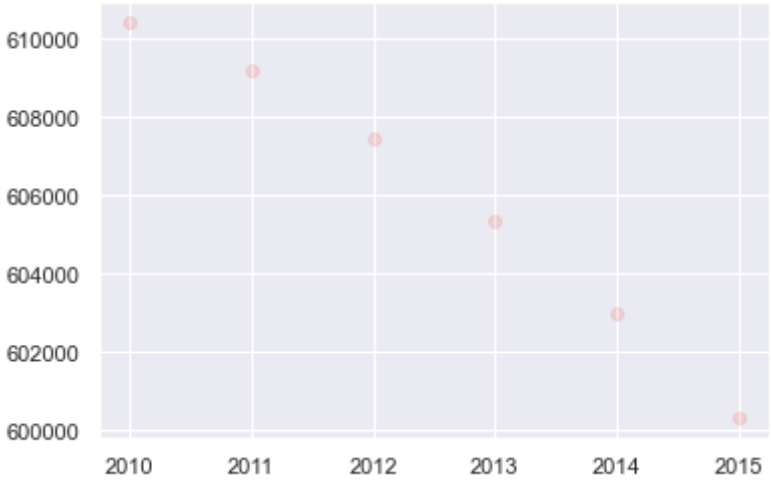

1e6
3.10
3.05
3.00
2.95
2.90
2.85
2010
2011
2012
2013
2014
2015

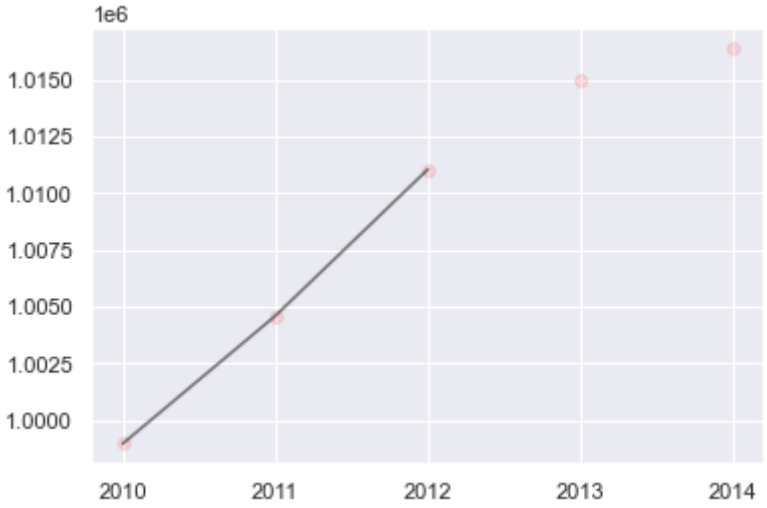

1e7
2.50
2.48
2.46
2.44
2.42
2.40
2010
2011
2012
2013
2014

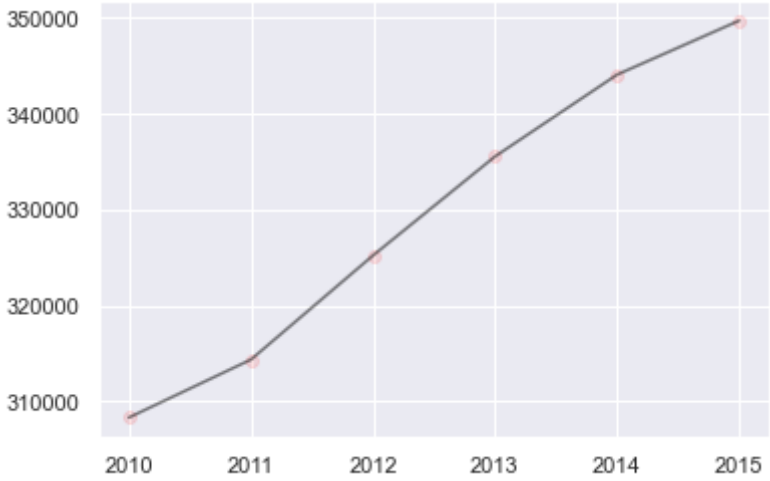

250000
245000
240000
235000
230000
225000
2010
2011
2012
2013
2014

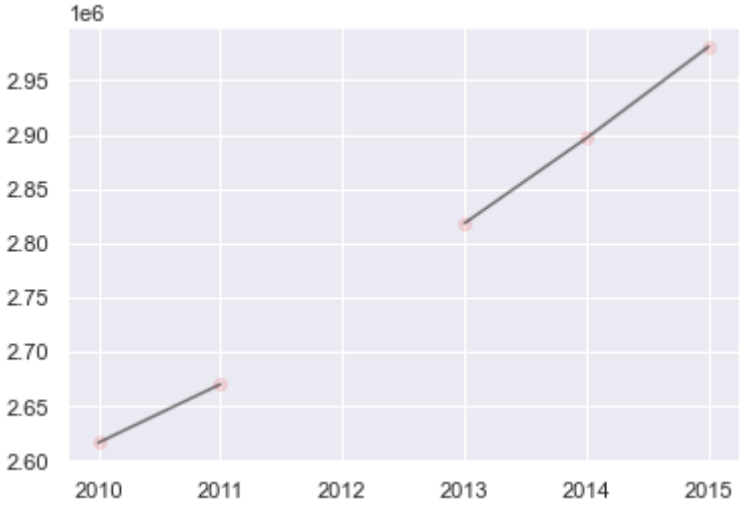

1e6
1.40
1.35
1.30
1.25
1.20
2010
2011
2012
2013
2014

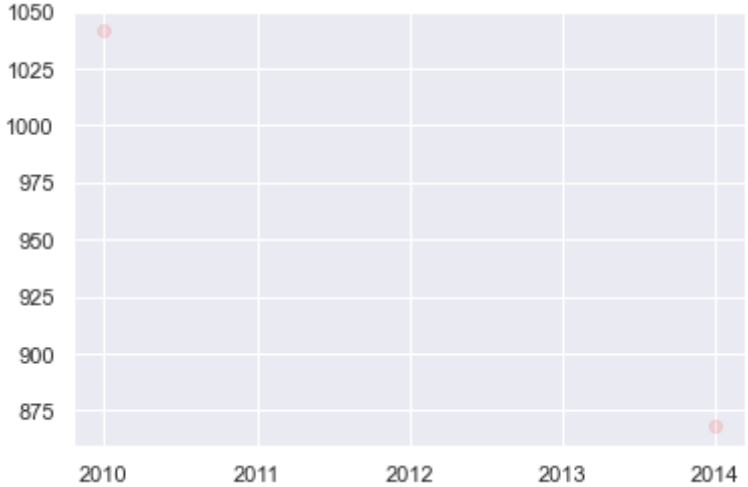

1e7
1.60
1.58
1.56
1.54
1.52
1.50
1.48
2010
2011
2012
2013
2014

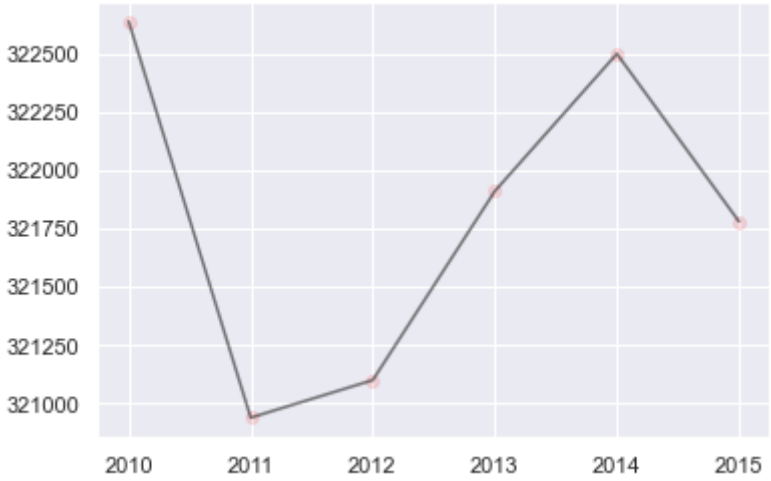

610000
608000
606000
604000
602000
600000
2010
2011
2012
2013
2014
2015

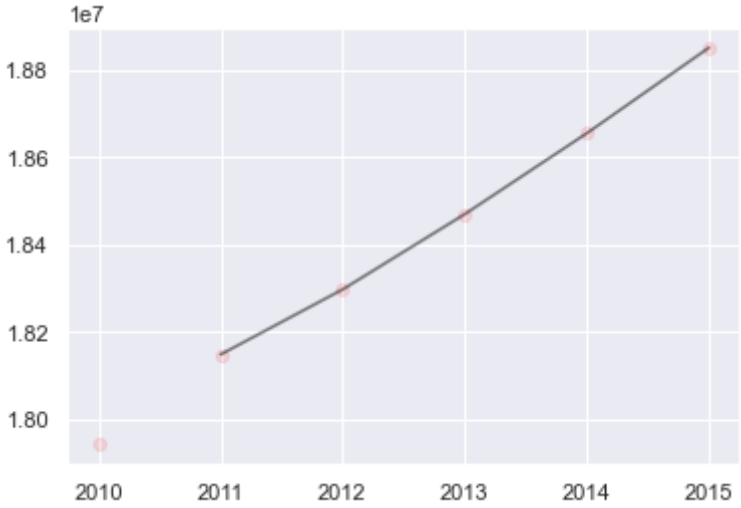

1e6
1.0150
1.0125
1.0100
1.0075
1.0050
1.0025
1.0000
2010
2011
2012
2013
2014

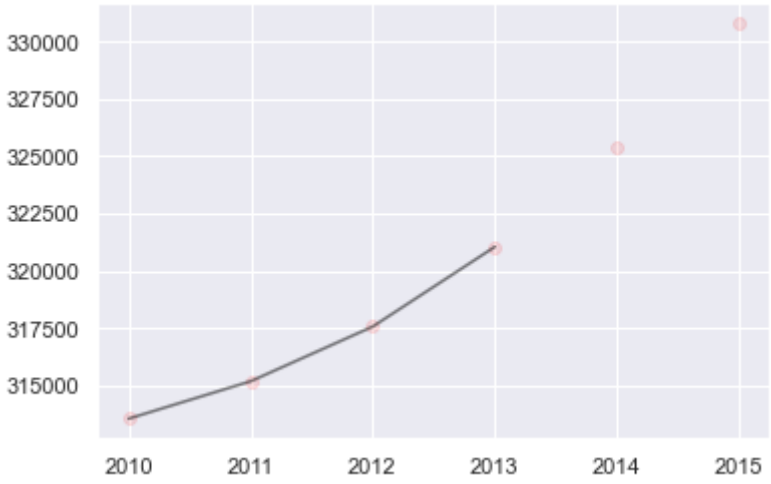

350000
340000
330000
320000
310000
2010
2011
2012
2013
2014
2015

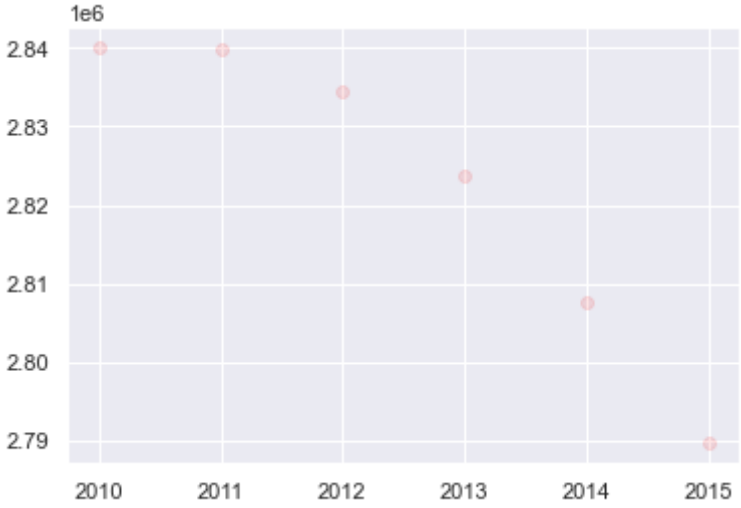

1e6
2.95
2.90
2.85
2.80
2.75
2.70
2.65
2.60
2010
2011
2012
2013
2014
2015

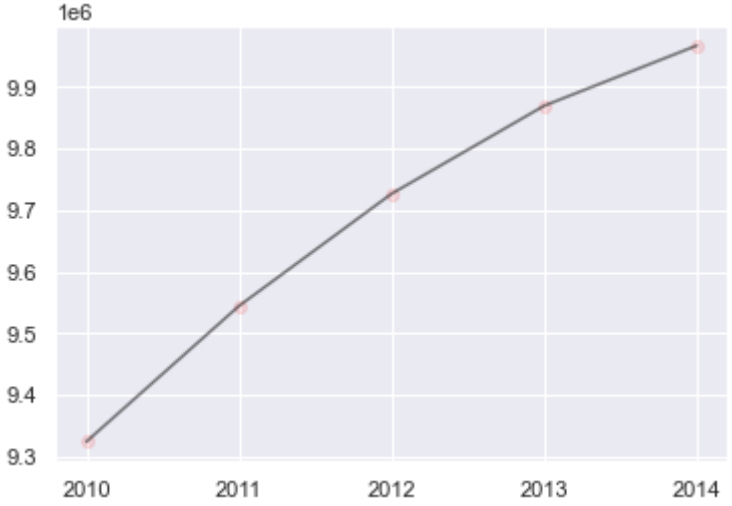

1050
1025
1000
975
950
925
900
875
2010
2011
2012
2013
2014

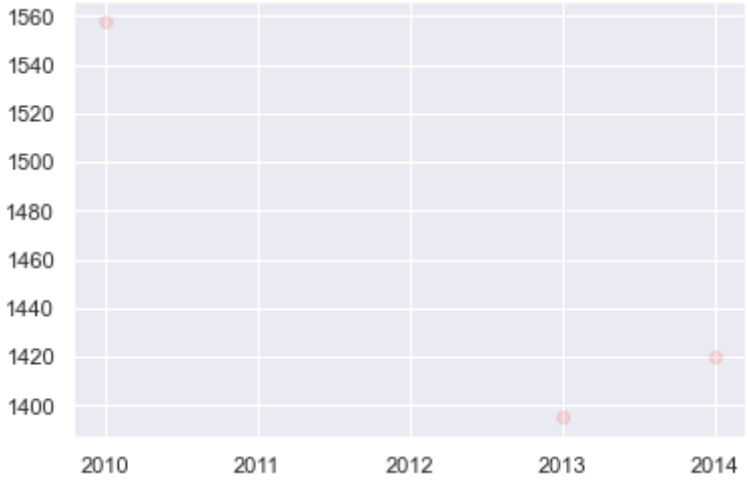

322500
322250
322000
321750
321500
321250
321000
2010
2011
2012
2013
2014
2015

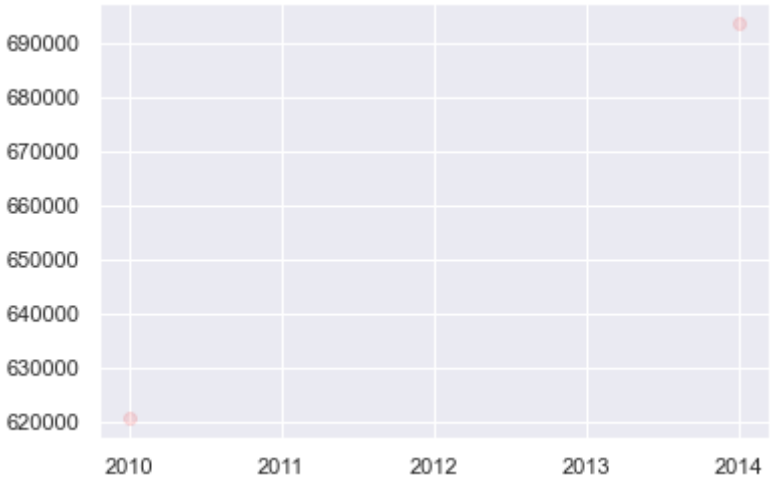

1e7
1.88
1.86
1.84
1.82
1.80
2010
2011
2012
2013
2014
2015

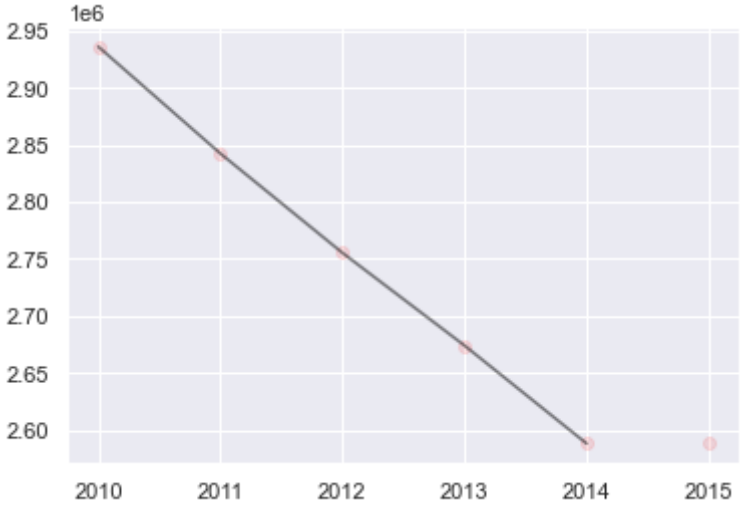

330000
327500
325000
322500
320000
317500
315000
2010
2011
2012
2013
2014
2015


1e6
2.84
2.83
2.82
2.81
2.80
2.79
2010
2011
2012
2013
2014
2015

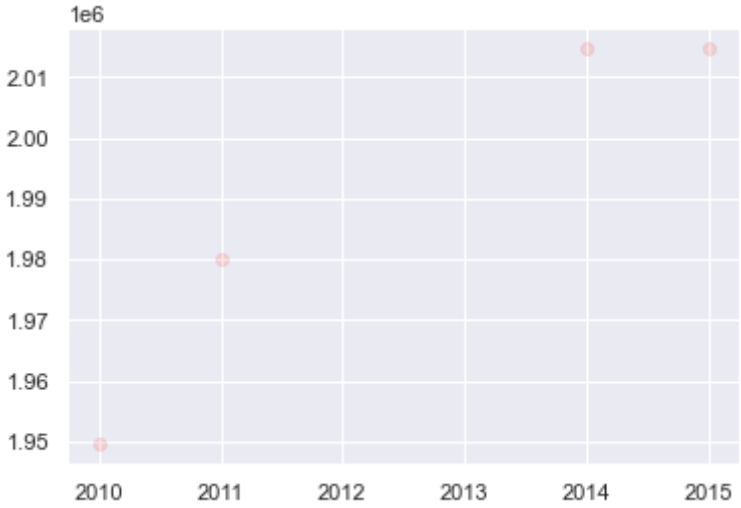

1e6
9.9
9.8
9.7
9.6
9.5
9.4
9.3
2010
2011
2012
2013
2014

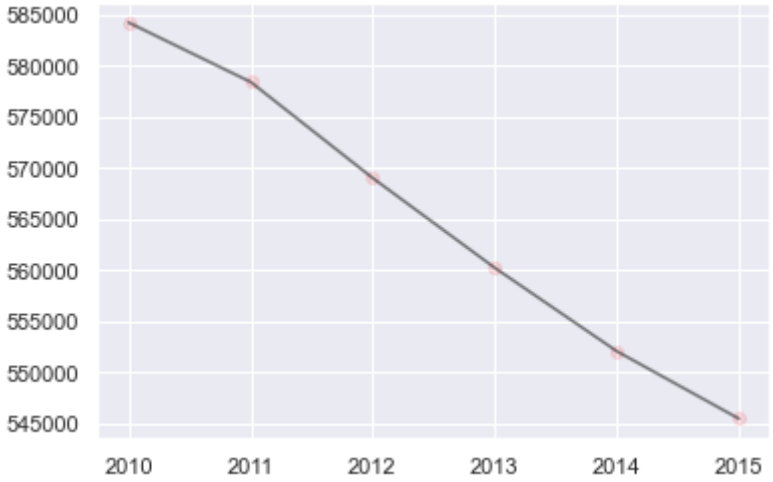

1560
1540
1520
1500
1480
1460
1440
1420
1400
2010
2011
2012
2013
2014

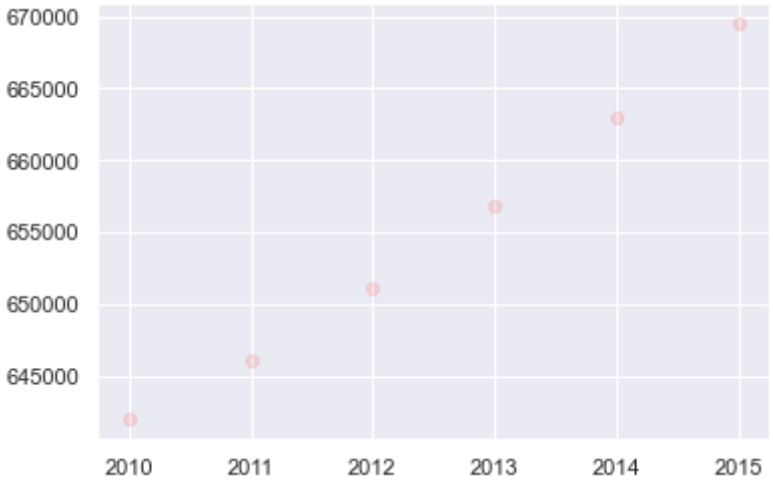

690000
680000
670000
660000
650000
640000
630000
620000
2010
2011
2012
2013
2014

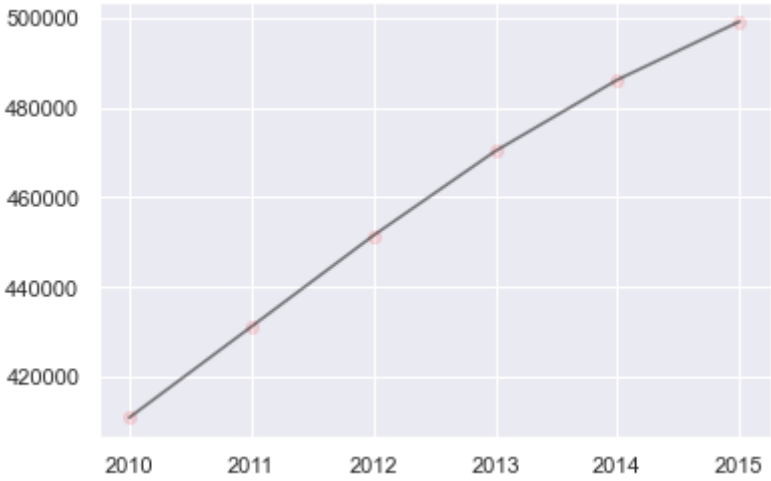

1e6
2.95
2.90
2.85
2.80
2.75
2.70
2.65
2.60
2010
2011
2012
2013
2014
2015

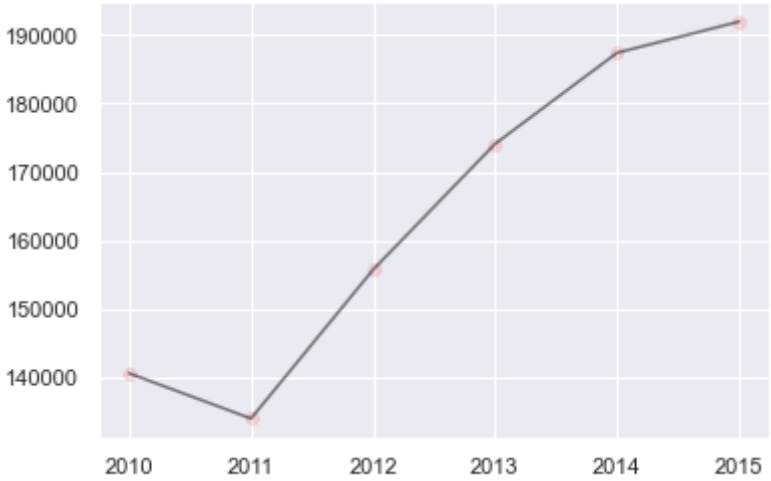
289000
288000
287000
286000
285000
284000
2010
2011
2012
2013
2014
2015

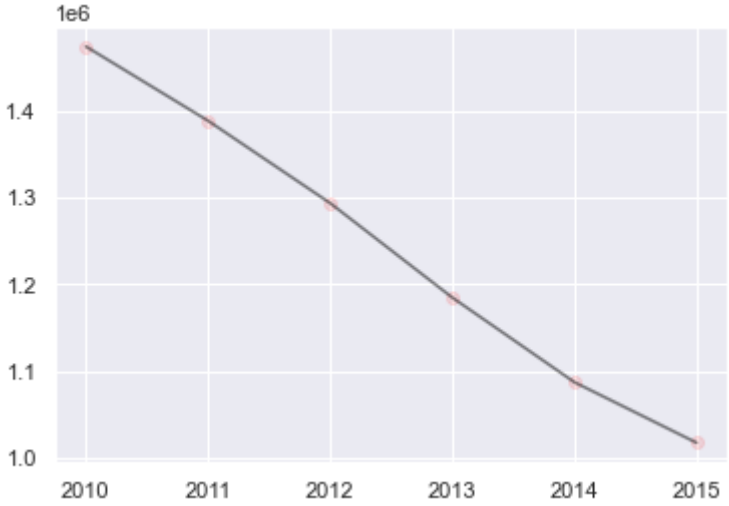

1e6
2.01
2.00
1.99
1.98
1.97
1.96
1.95
2010
2011
2012
2013
2014
2015

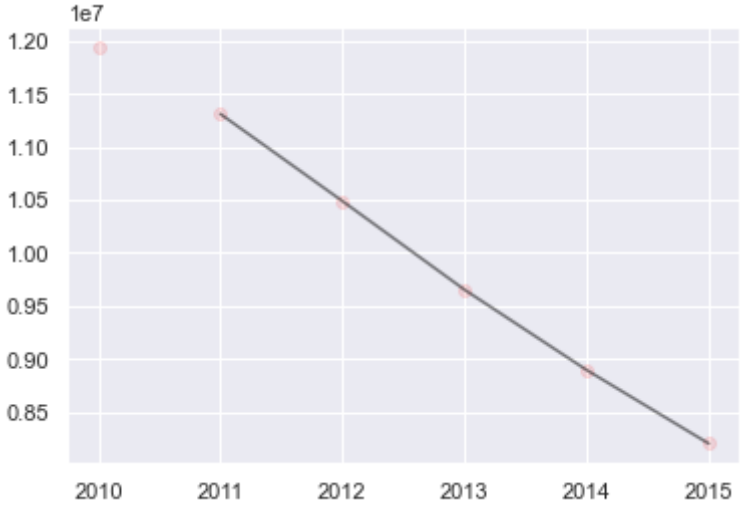

585000
580000
575000
570000
565000
560000
555000
550000
545000
2010
2011
2012
2013
2014
2015

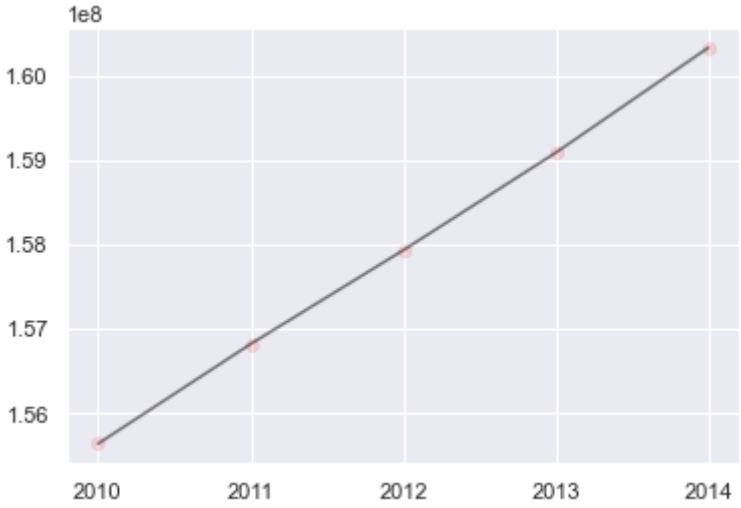

670000
665000
660000
655000
650000
645000
2010
2011
2012
2013
2014
2015

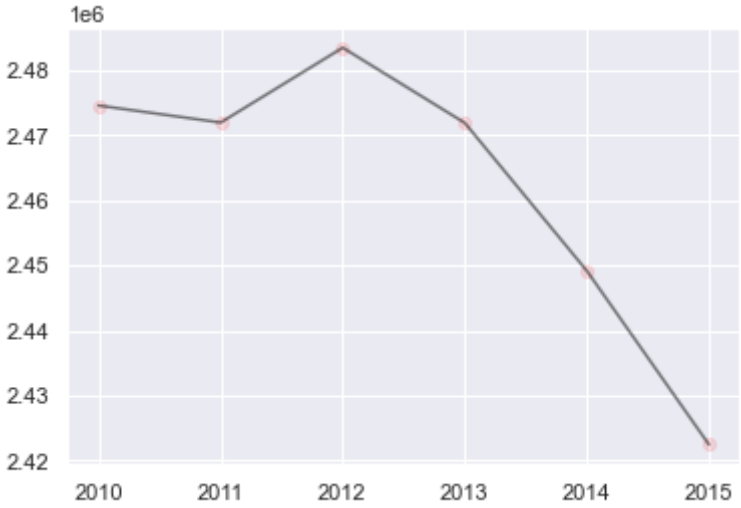

500000
480000
460000
440000
420000
2010
2011
2012
2013
2014
2015

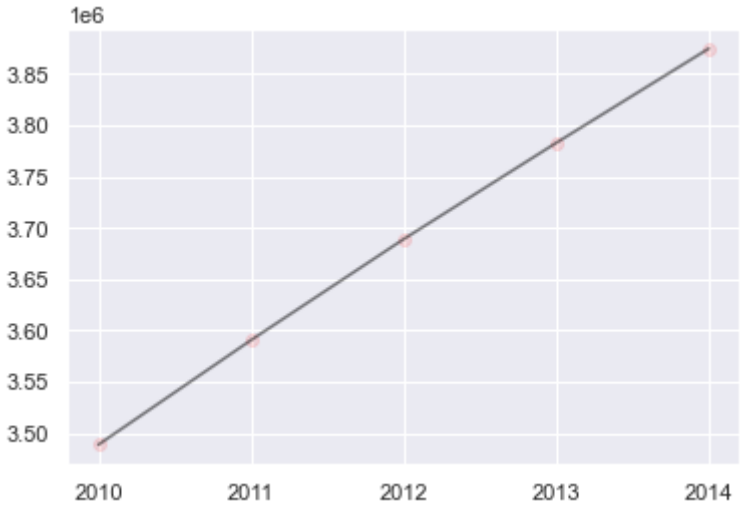

190000
180000
170000
160000
150000
140000
2010
2011
2012
2013
2014
2015

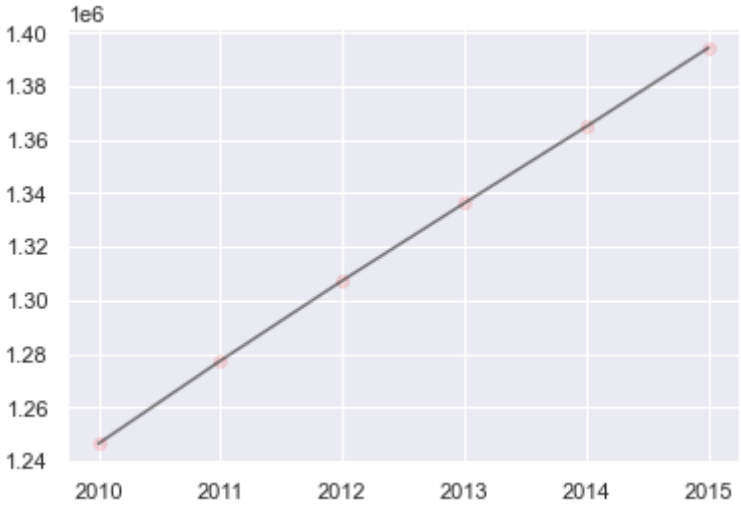

1e6
1.4
1.3
1.2
1.1
1.0
2010
2011
2012
2013
2014
2015

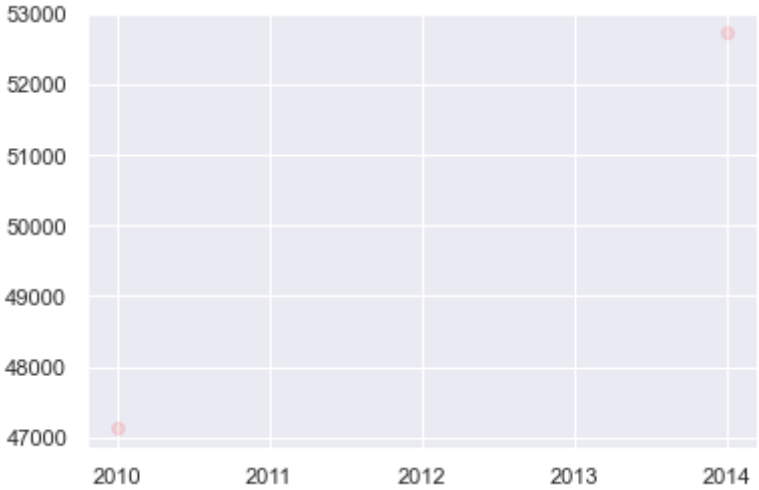

1e7
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
2010
2011
2012
2013
2014
2015

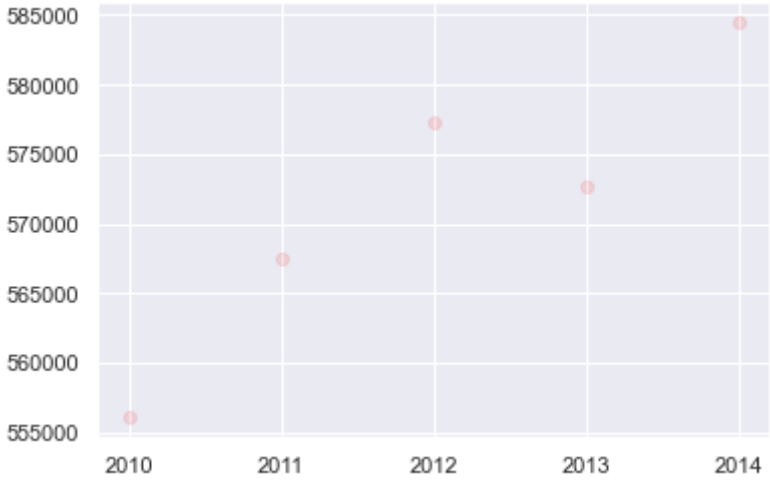

1e8
1.60
1.59
1.58
1.57
1.56
2010
2011
2012
2013
2014

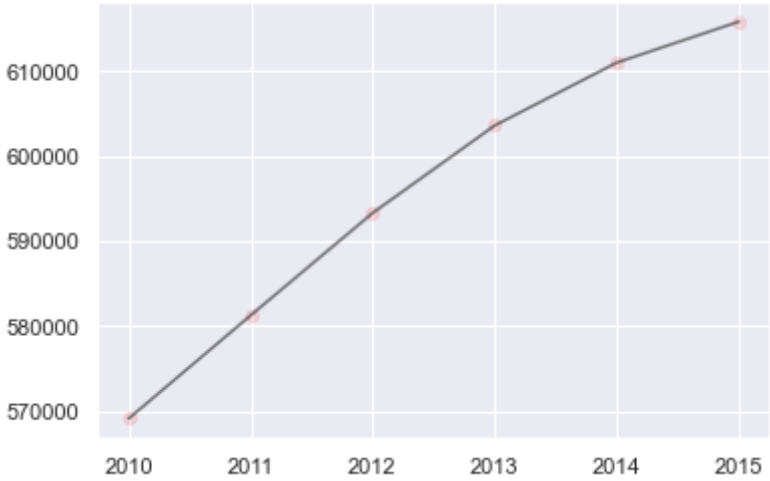

1e6
2.48
2.47
2.46
2.45
2.44
2.43
2.42
2010
2011
2012
2013
2014
2015

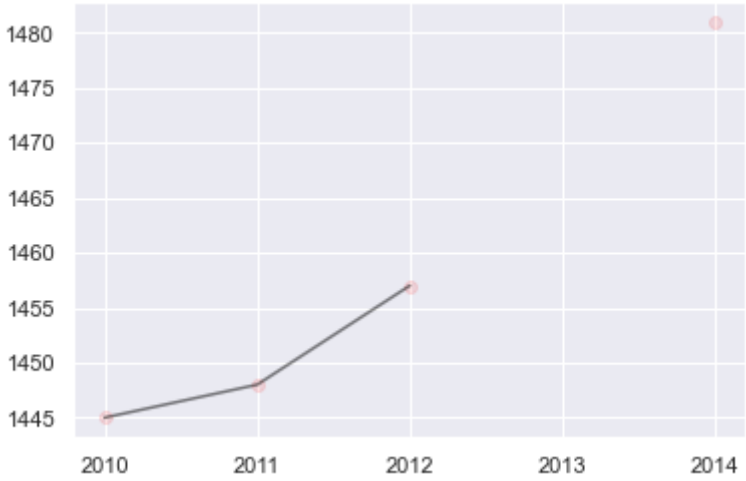

1e6
3.85
3.80
3.75
3.70
3.65
3.60
3.55
3.50
2010
2011
2012
2013
2014

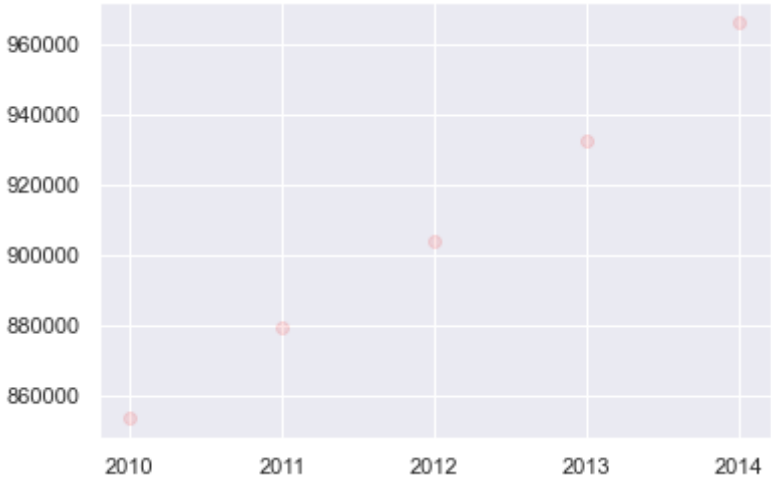

1e6
1.40
1.38
1.36
1.34
1.32
1.30
1.28
1.26
1.24
2010
2011
2012
2013
2014
2015


53000
52000
51000
50000
49000
48000
47000
2010
2011
2012
2013
2014

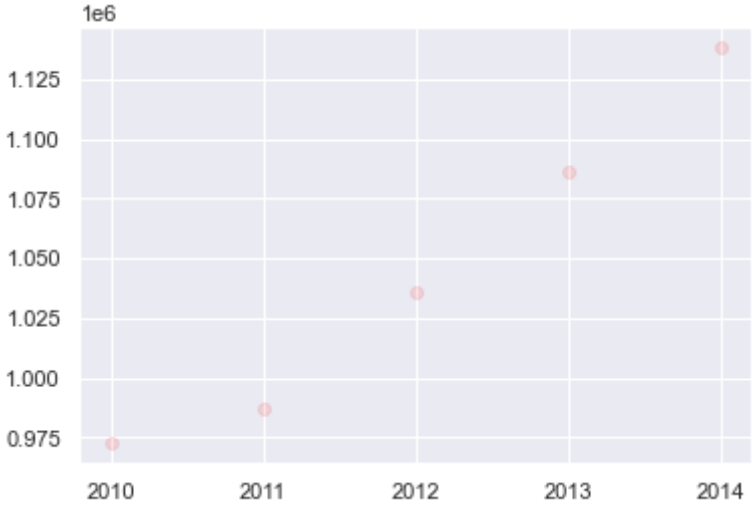

585000
580000
575000
570000
565000
560000
555000
2010
2011
2012
2013
2014

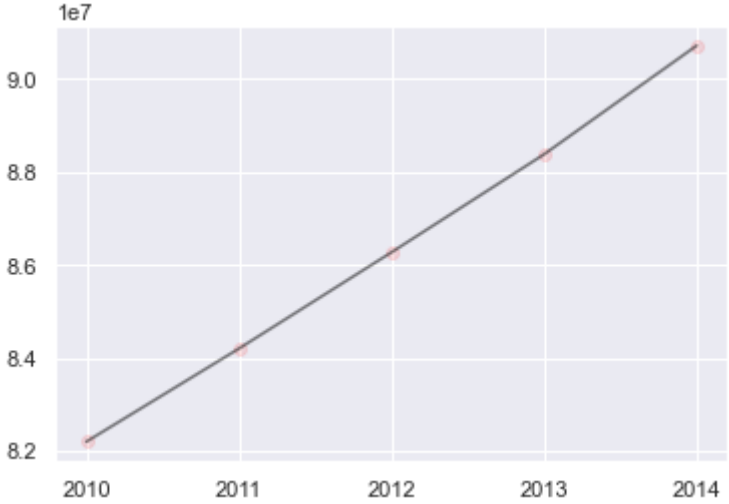

610000
600000
590000
580000
570000
2010
2011
2012
2013
2014
2015

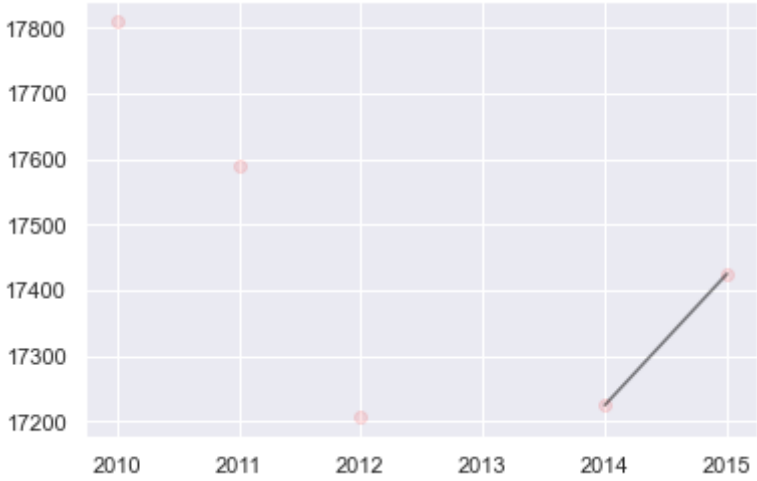

1480
1475
1470
1465
1460
1455
1450
1445
2010
2011
2012
2013
2014

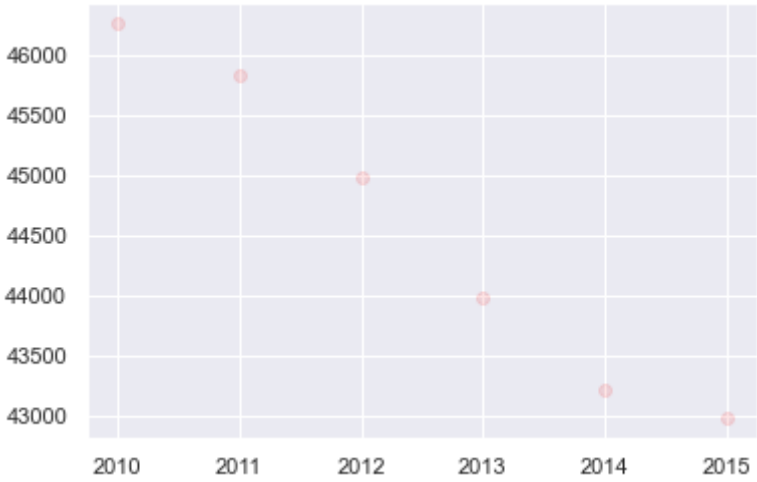

960000
940000
920000
900000
880000
860000
2010
2011
2012
2013
2014

460000
450000
440000
430000
420000
2010
2011
2012
2013
2014
2015

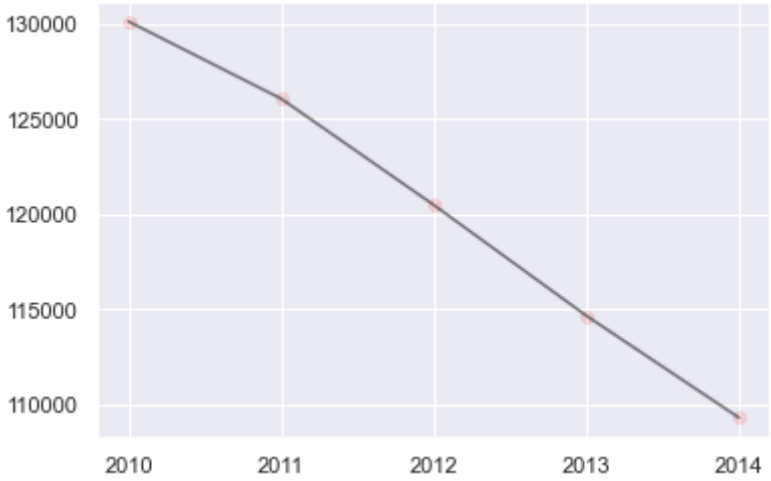

1e6
1.125
1.100
1.075
1.050
1.025
1.000
0.975
2010
2011
2012
2013
2014

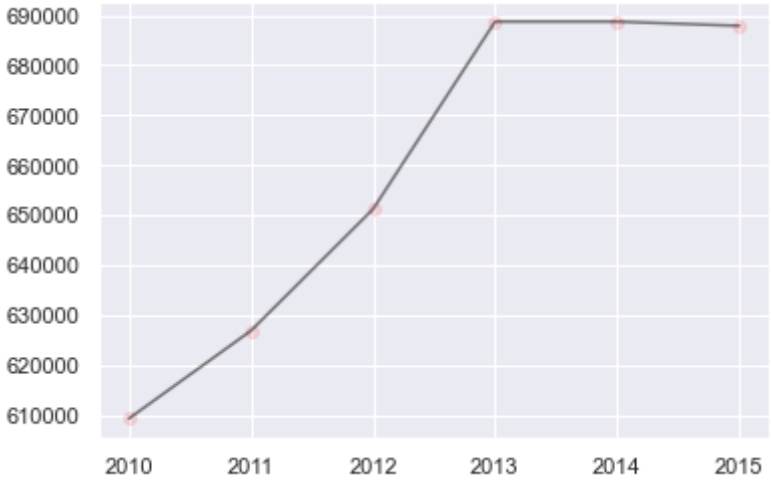

1e7
9.0
8.8
8.6
8.4
8.2
2010
2011
2012
2013
2014

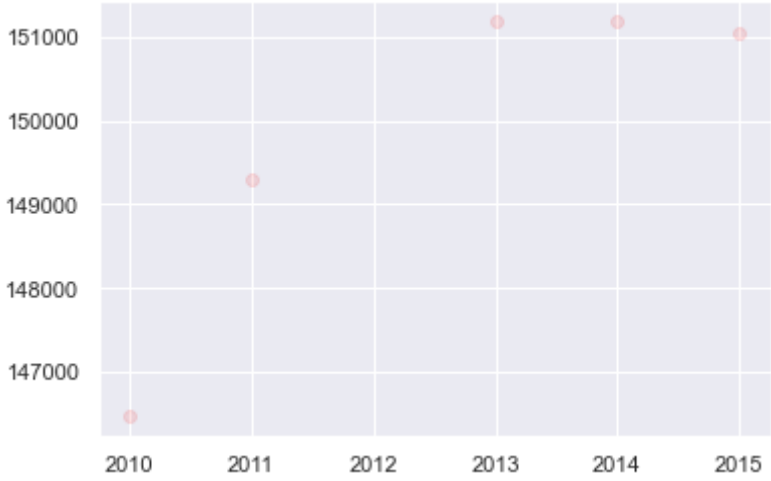

17800
17700
17600
17500
17400
17300
17200
2010
2011
2012
2013
2014
2015

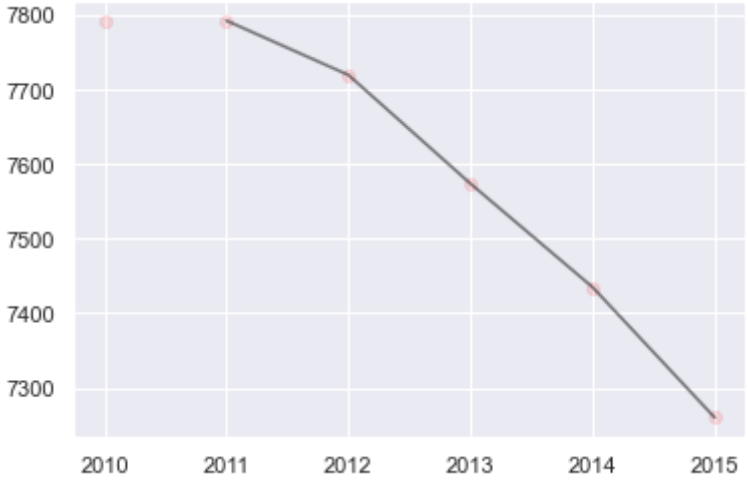

46000
45500
45000
44500
44000
43500
43000
2010
2011
2012
2013
2014
2015

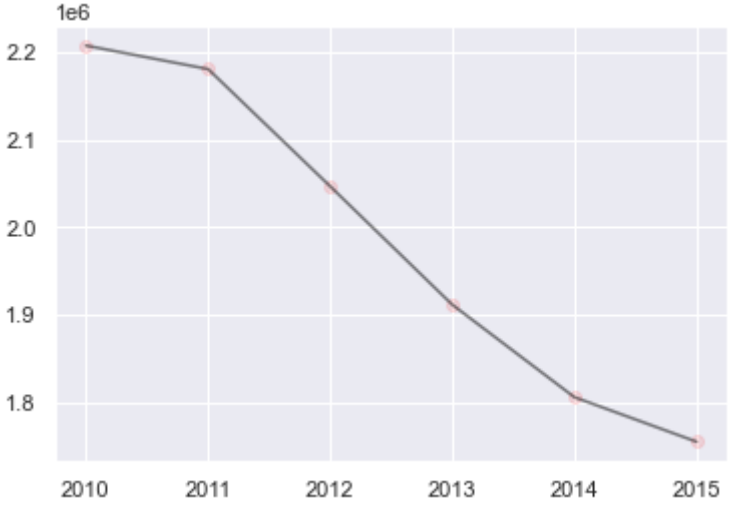
410000
400000
390000
380000
370000
360000
2010
2011
2012
2013
2014
2015

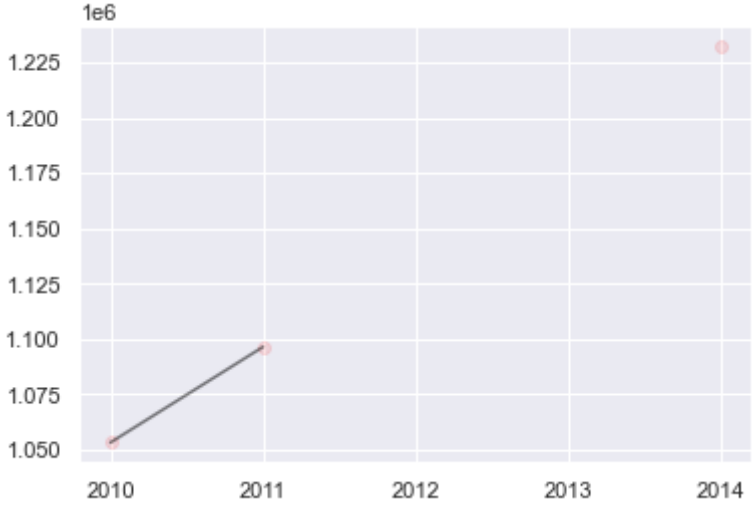

130000
125000
120000
115000
110000
2010
2011
2012
2013
2014

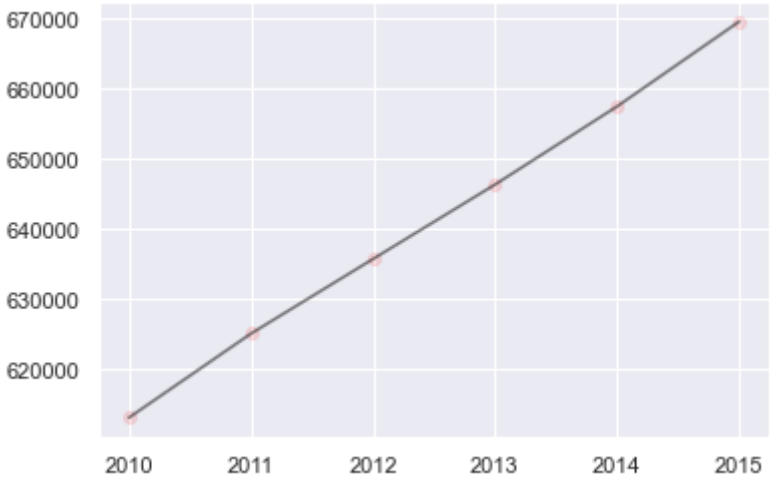

690000
680000
670000
660000
650000
640000
630000
620000
610000
2010
2011
2012
2013
2014
2015

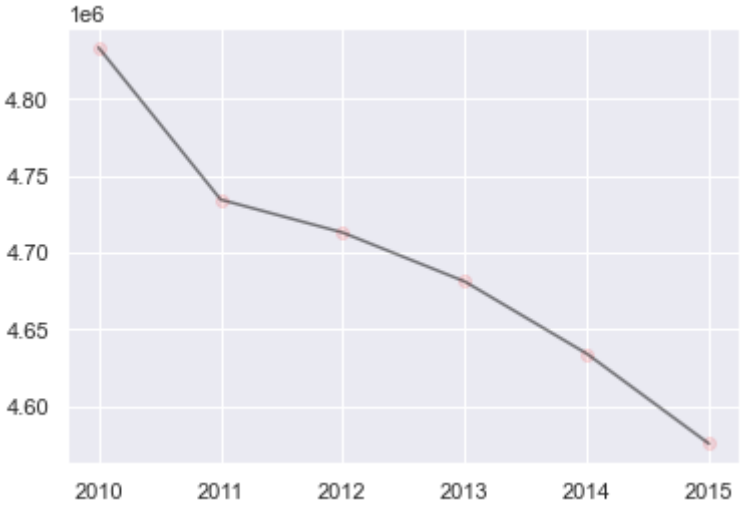

151000
150000
149000
148000
147000
2010
2011
2012
2013
2014
2015

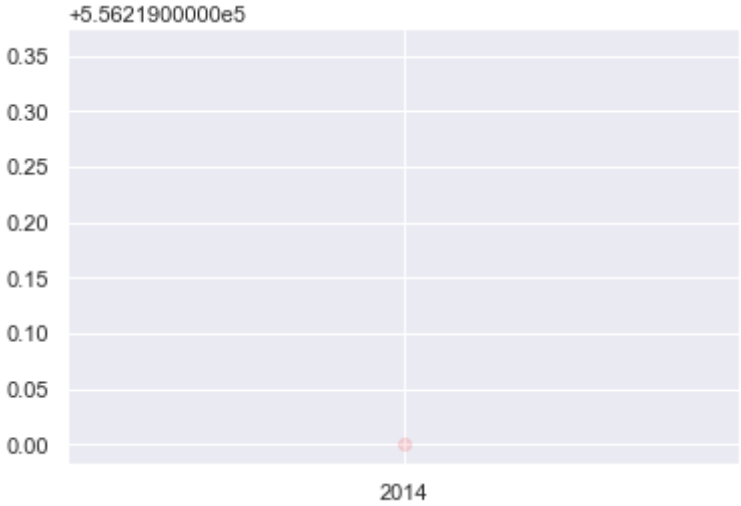

7800
7700
7600
7500
7400
7300
2010
2011
2012
2013
2014
2015

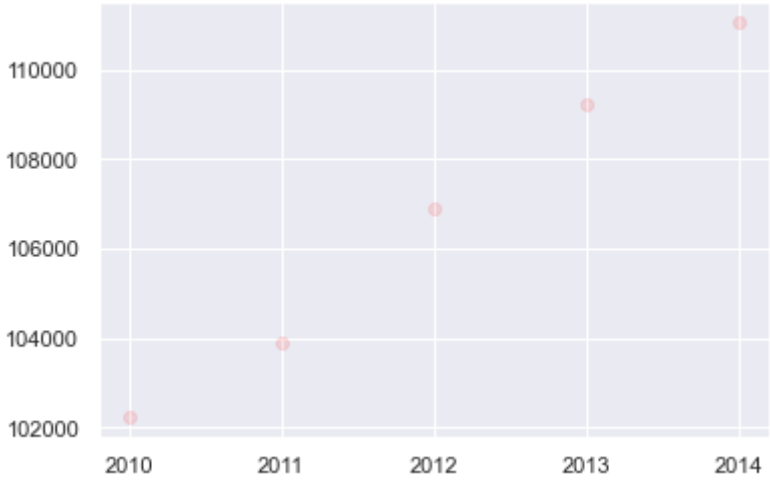

1e6
2.2
2.1
2.0
1.9
1.8
2010
2011
2012
2013
2014
2015

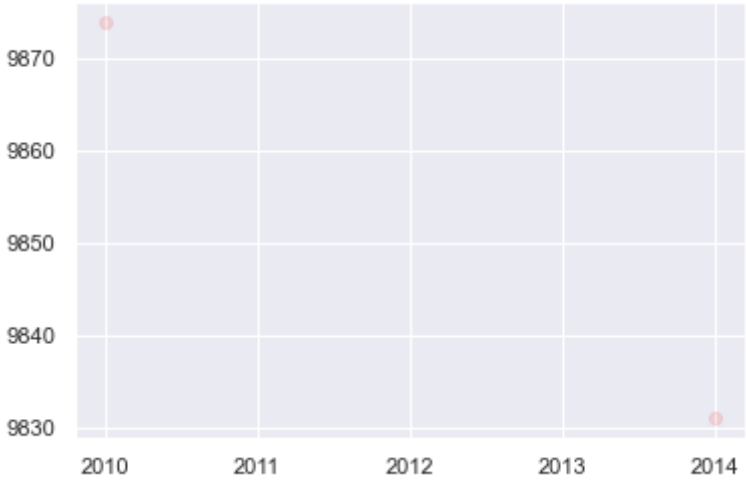

1e6
1.225
1.200
1.175
1.150
1.125
1.100
1.075
1.050
2010
2011
2012
2013
2014

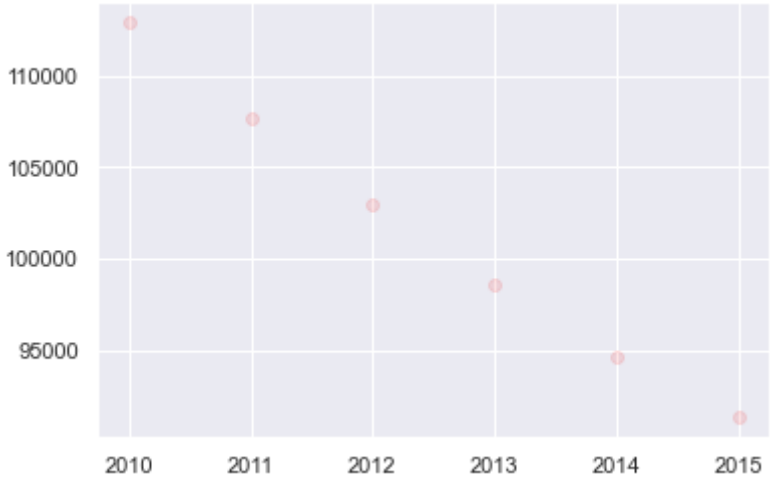

670000
660000
650000
640000
630000
620000
2010
2011
2012
2013
2014
2015

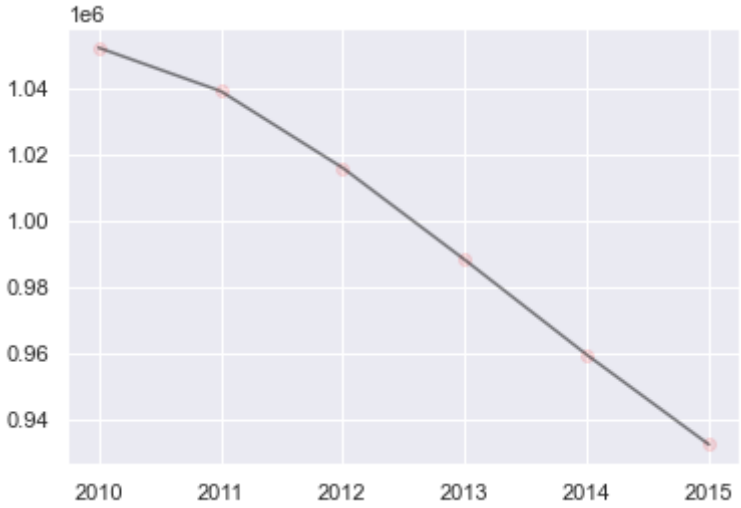

1e6
4.80
4.75
4.70
4.65
4.60
2010
2011
2012
2013
2014
2015

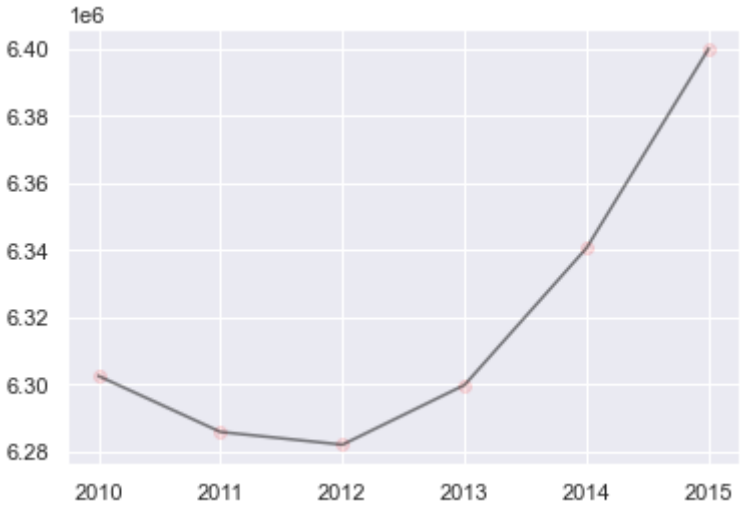

+5.5621900000e5
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
2014

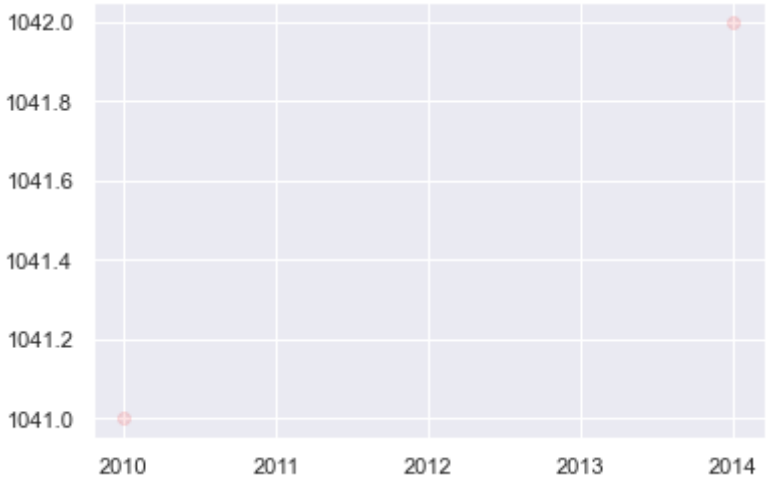

110000
108000
106000
104000
102000
2010
2011
2012
2013
2014

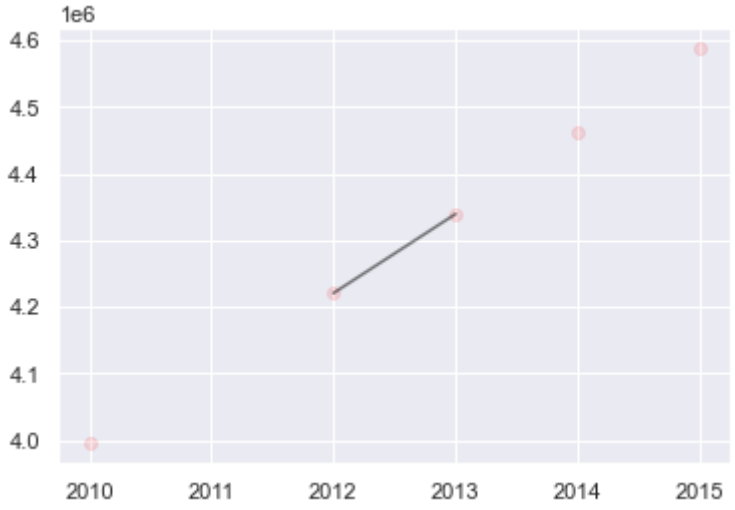

9870
9860
9850
9840
9830
2010
2011
2012
2013
2014

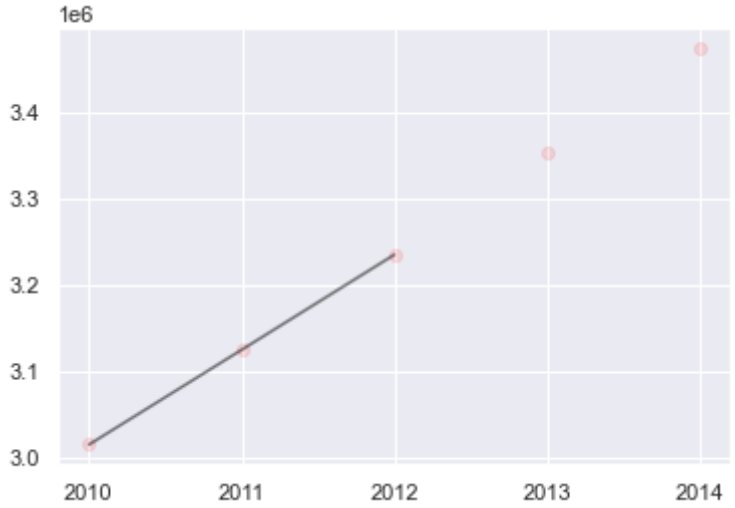

110000
105000
100000
95000
2010
2011
2012
2013
2014
2015

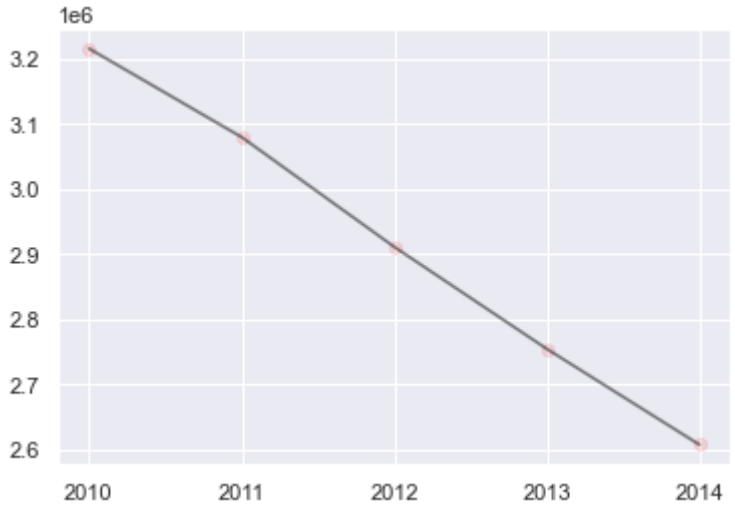

1e6
1.04
1.02
1.00
0.98
0.96
0.94
2010
2011
2012
2013
2014
2015


1e6
6.40
6.38
6.36
6.34
6.32
6.30
6.28
2010
2011
2012
2013
2014
2015

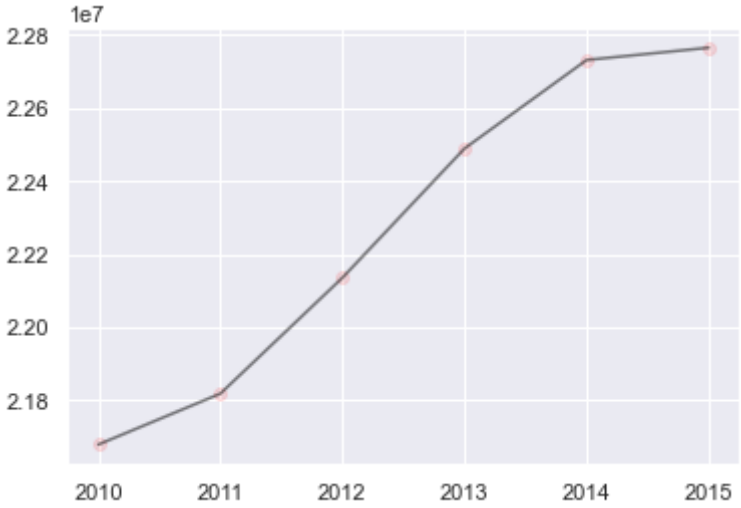

1042.0
1041.8
1041.6
1041.4
1041.2
1041.0
2010
2011
2012
2013
2014

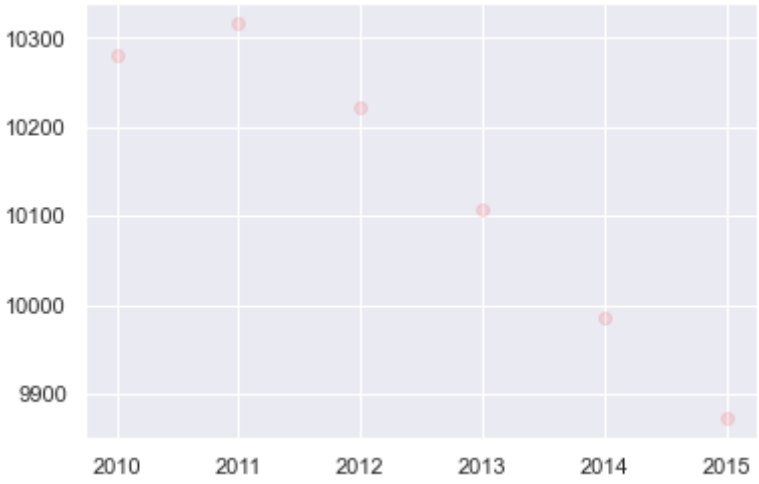

1e6
4.6
4.5
4.4
4.3
4.2
4.1
4.0
2010
2011
2012
2013
2014
2015

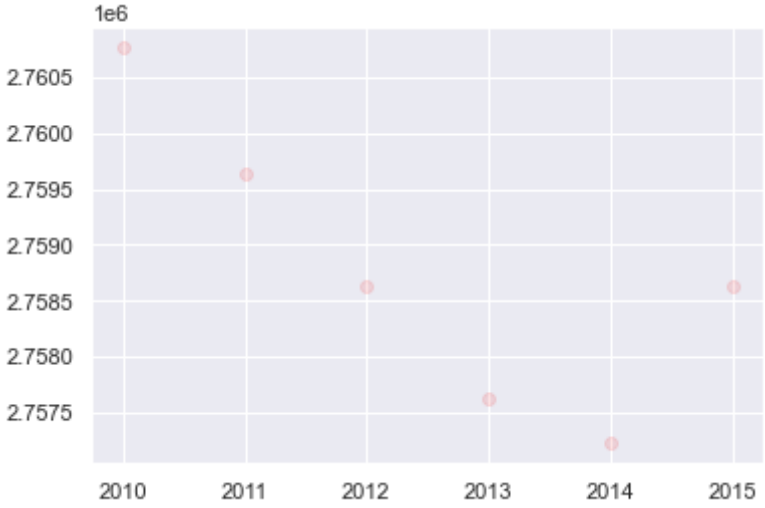

1e6
3.4
3.3
3.2
3.1
3.0
2010
2011
2012
2013
2014

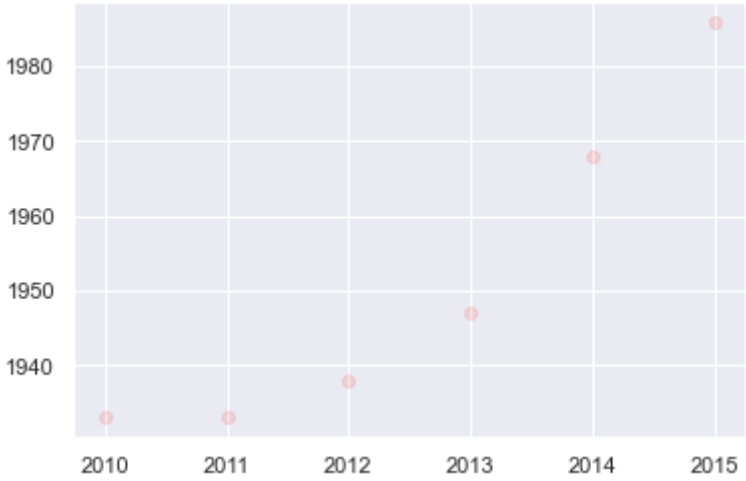

1e6
3.2
3.1
3.0
2.9
2.8
2.7
2.6
2010
2011
2012
2013
2014

263000
262000
261000
260000
259000
2010
2011
2012
2013
2014
2015

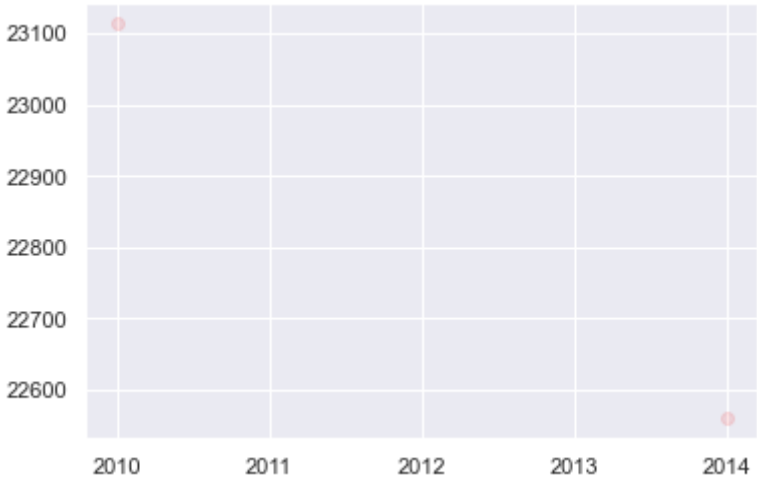

1e7
2.28
2.26
2.24
2.22
2.20
2.18
2010
2011
2012
2013
2014
2015

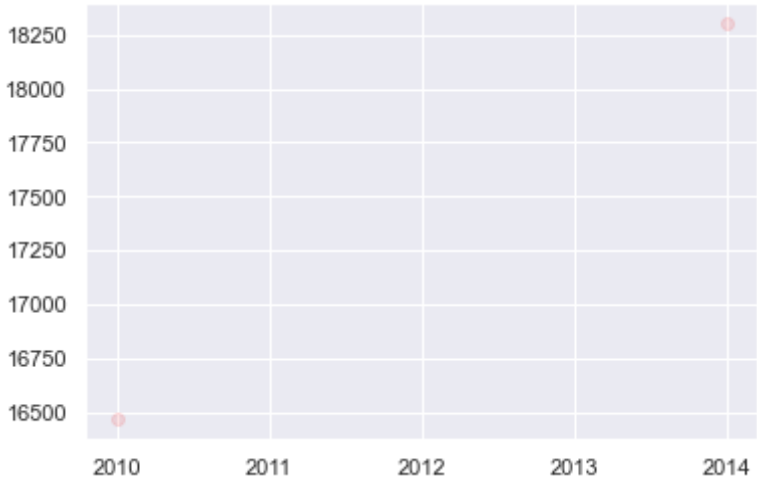

10300
10200
10100
10000
9900
2010
2011
2012
2013
2014
2015

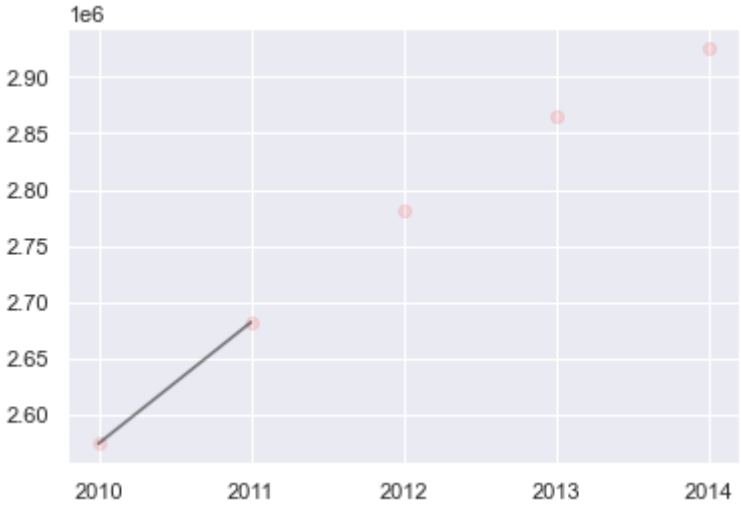

1e6
2.7605
2.7600
2.7595
2.7590
2.7585
2.7580
2.7575
2010
2011
2012
2013
2014
2015

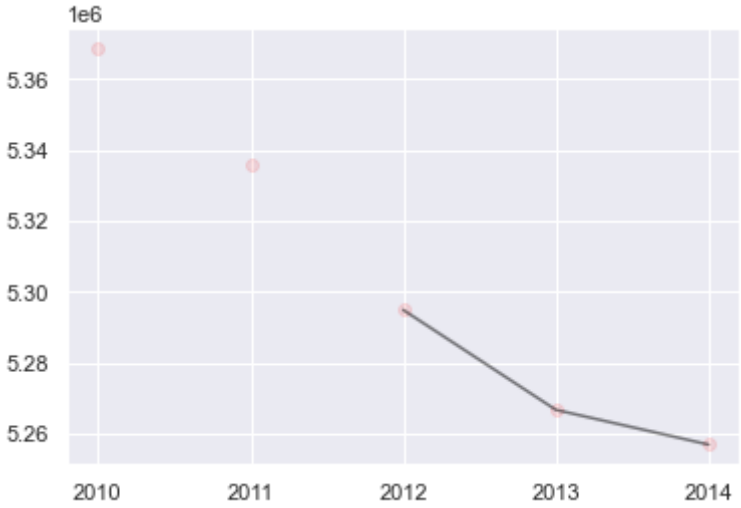

1980
1970
1960
1950
1940
2010
2011
2012
2013
2014
2015

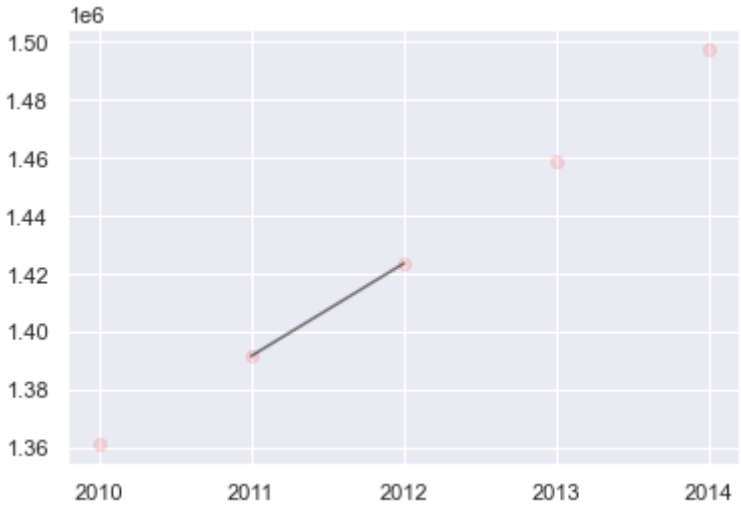
1e6
9.0
8.9
8.8
8.7
8.6
2010
2011
2012
2013
2014
2015

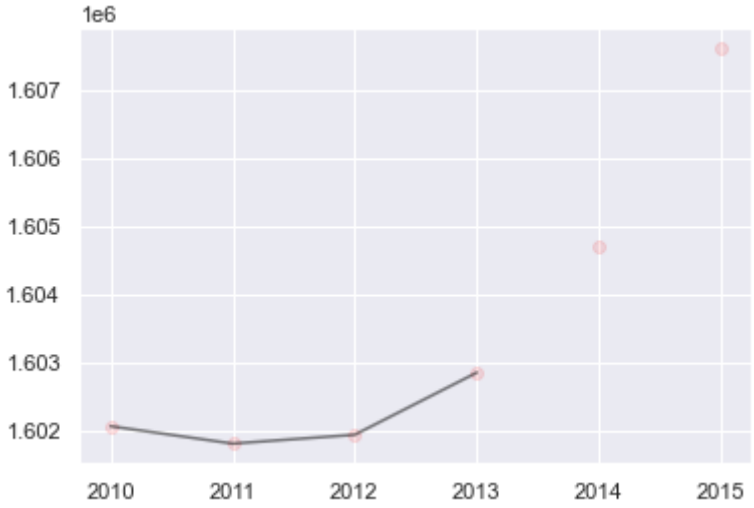

23100
23000
22900
22800
22700
22600
2010
2011
2012
2013
2014


18250
18000
17750
17500
17250
17000
16750
16500
2010
2011
2012
2013
2014


1e6
2.90
2.85
2.80
2.75
2.70
2.65
2.60
2010
2011
2012
2013
2014


1e6
5.36
5.34
5.32
5.30
5.28
5.26
2010
2011
2012
2013
2014


1e6
1.50
1.48
1.46
1.44
1.42
1.40
1.38
1.36
2010
2011
2012
2013
2014



In [59]:

```
print(data.isna().sum().sum())
print(data.fillna(value = data4).isna().sum().sum())
```

```
349
349
```

Résumé : SP.TER.TOTL.IN, SE.TER.ENRL, SE.TER.ENRR

- SE.TER.ENRR = SE.TER.ENRL / SP.TER.TOTL.IN

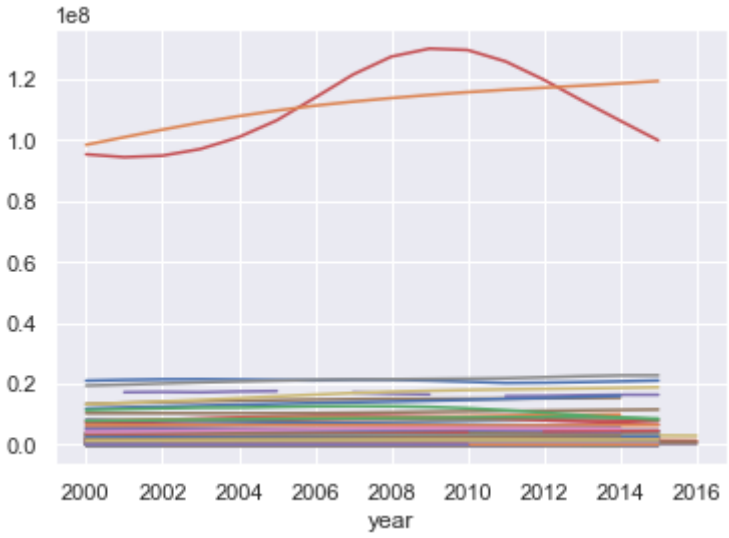
SP.TER.TOTL.IN

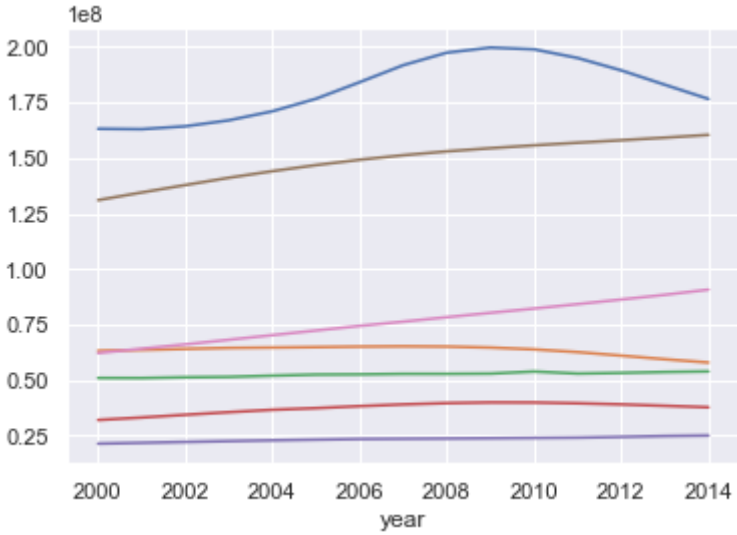
Population avec l'âge pour 'teriary education'.

In [60]:

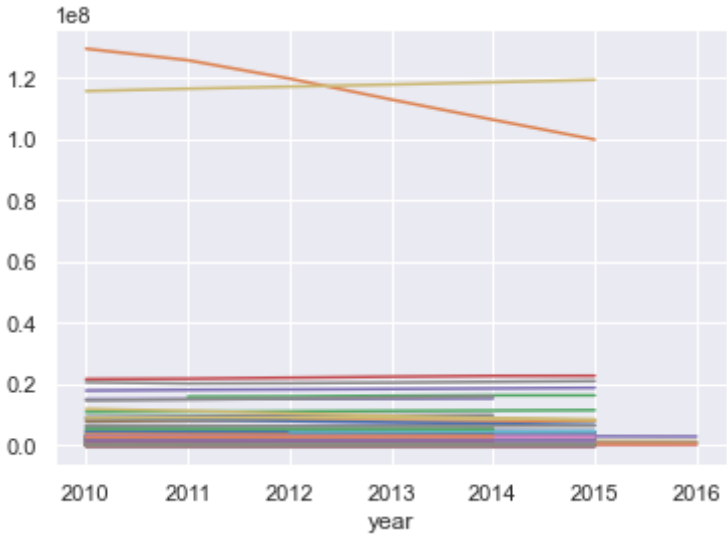
```
ind = 'SP.TER.TOTL.IN'
initial_investigation(ind)[0].T.info()
```

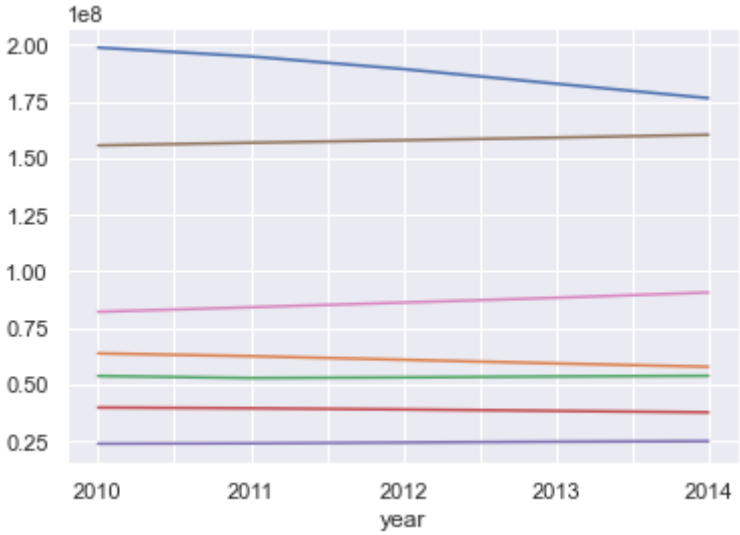
```
**************************************************
SP.TER.TOTL.IN
```






```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 206 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI'
 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BMU' 'BOL'
 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR'
 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI'
 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH'
 'FIN' 'FJI' 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIB' 'GIN' 'GMB' 'GNB'
 'GNQ' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND'
 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ'
 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE'
 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX'
 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS'
 'NAC' 'NAM' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'OMN' 'PAK'
 'PAN' 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT'
 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM'
 'SRB' 'SSD' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD'
 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA'
 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VGB' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF'
 'ZMB' 'ZWE']

These countries have absolutely no data : {'GRL', 'MCO', 'CHI', 'XKX', 'SXM', 'FRO',
'MNP', 'CYM', 'ASM', 'VIR', 'GUM', 'MAF', 'AND', 'PYF', 'IMN', 'NCL', 'TCA', 'SGP'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SP.TER.TOTL.IN
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 202 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL'
 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA' 'BRB'
 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG'
 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK'
 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI'
 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD'
 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ'
 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA'
 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU'
 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI'
 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NER'
 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'PLW'
 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS'
 'SAU' 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSD' 'SSF' 'STP'
 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK' 'TKM'
 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB'
 'VCT' 'VEN' 'VGB' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'CHI', 'MNP', 'CYM', 'PYF', 'SGP', 'ARE',
'GRL', 'MCO', 'FRO', 'VIR', 'ASM', 'OMN', 'NCL', 'TCA', 'IMN', 'XKX', 'SXM', 'GUM',
'MAF', 'AND', 'BIH', 'GIB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.7531824611032532

year           2010          2011          2012          2013          2014  \
count  1.990000e+02  1.820000e+02  1.760000e+02  1.700000e+02  2.020000e+02
mean   6.113856e+06  6.695563e+06  6.857382e+06  6.996587e+06  5.955561e+06
std    2.306221e+07  2.384630e+07  2.389222e+07  2.389969e+07  2.172879e+07
min    1.041000e+03  1.448000e+03  1.457000e+03  1.395000e+03  8.680000e+02
25%    1.435380e+05  2.317242e+05  2.581600e+05  2.657060e+05  1.516418e+05
50%    6.242340e+05  7.627780e+05  7.601635e+05  8.630915e+05  6.267365e+05
75%    2.482447e+06  2.740180e+06  2.853672e+06  2.871583e+06  2.602514e+06
max    1.988264e+08  1.949116e+08  1.893518e+08  1.828877e+08  1.765053e+08

year           2015          2016
count  1.290000e+02  7.000000e+00
mean   3.672495e+06  1.208216e+06
std    1.393025e+07  9.147148e+05
min    1.986000e+03  2.200870e+05
25%    2.429840e+05  6.173370e+05
50%    6.695660e+05  1.030103e+06
75%    2.422484e+06  1.525426e+06
max    1.194700e+08  2.921796e+06
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 202 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    199 non-null    float64
 1   2011    182 non-null    float64
 2   2012    176 non-null    float64
 3   2013    170 non-null    float64
 4   2014    202 non-null    float64
 5   2015    129 non-null    float64
 6   2016    7 non-null      float64
dtypes: float64(7)
memory usage: 12.6+ KB
```

Résumé: SP.TER.TOTL.IN

- Pour 195 pays, il y a de la data pour les années 2010-2016.
  - 22 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 75% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Pour les 202 pays et régions, il y a de data pour l'année 2014.**
- Garder que tous ces 205 pays.
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [61]:

```
ind = 'SP.TER.TOTL.IN'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data = drop_countries(data,3)
```

```
The following countries were dropped :
['AND', 'ARE', 'ASM', 'BIH', 'CHI', 'CYM', 'FRO', 'GIB', 'GRL', 'GUM', 'IMN', 'MAF',
'MCO', 'MNP', 'NCL', 'OMN', 'PYF', 'SGP', 'SXM', 'TCA', 'VIR', 'XKX']
```

In [62]:

```
len(data.columns.values)
```

Out[62]: 202

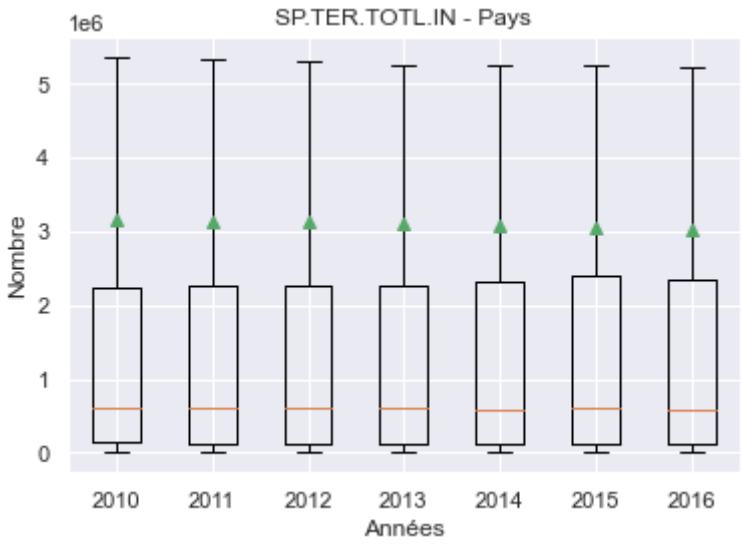
In [63]:

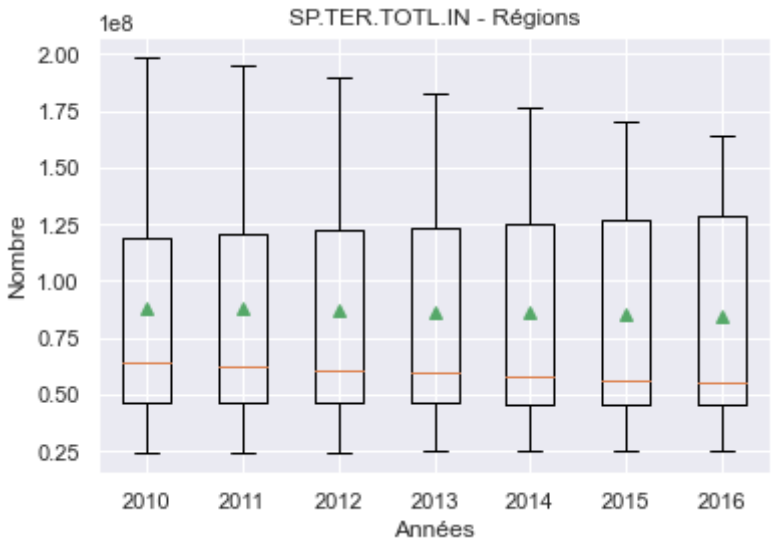
```
data_clean = fill_indicator_linear(data)
data_clean.T[years].describe()
```

Out[63]:

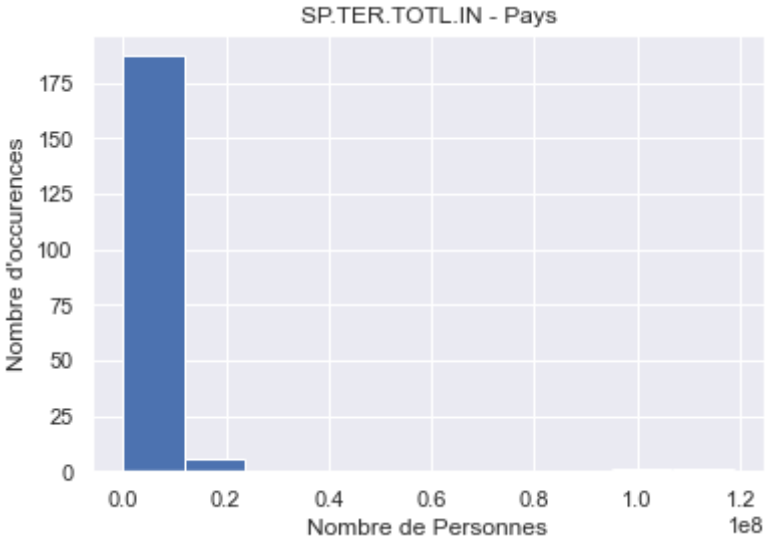
	2010	2011	2012	2013	2014	2015	
count	2.020000e+02	2.020000e+02	2.020000e+02	2.020000e+02	2.020000e+02	2.020000e+02	2.020(
mean	6.108873e+06	6.083296e+06	6.045661e+06	5.999616e+06	5.955561e+06	5.910514e+06	5.865
std	2.290728e+07	2.270690e+07	2.239605e+07	2.204517e+07	2.172879e+07	2.143240e+07	2.115
min	1.041000e+03	9.985000e+02	9.550000e+02	9.115000e+02	8.680000e+02	8.245000e+02	7.810(
25%	1.465625e+05	1.495585e+05	1.512834e+05	1.519482e+05	1.516418e+05	1.510522e+05	1.495
50%	6.223820e+05	6.277225e+05	6.405560e+05	6.380365e+05	6.267365e+05	6.123205e+05	6.143
75%	2.486400e+06	2.564956e+06	2.664337e+06	2.640306e+06	2.602514e+06	2.584005e+06	2.608
max	1.988264e+08	1.949116e+08	1.893518e+08	1.828877e+08	1.765053e+08	1.701229e+08	1.637

In [64]:
```
plot_boxplot(ind, data_clean, unit = "Nombre", years = years, showmeans = True, show
print_info_year('2014', data_clean, unit = "Nombre de Personnes", ind = ind)
print_info_year('2016', data_clean, unit = "Nombre de Personnes", ind = ind)
```



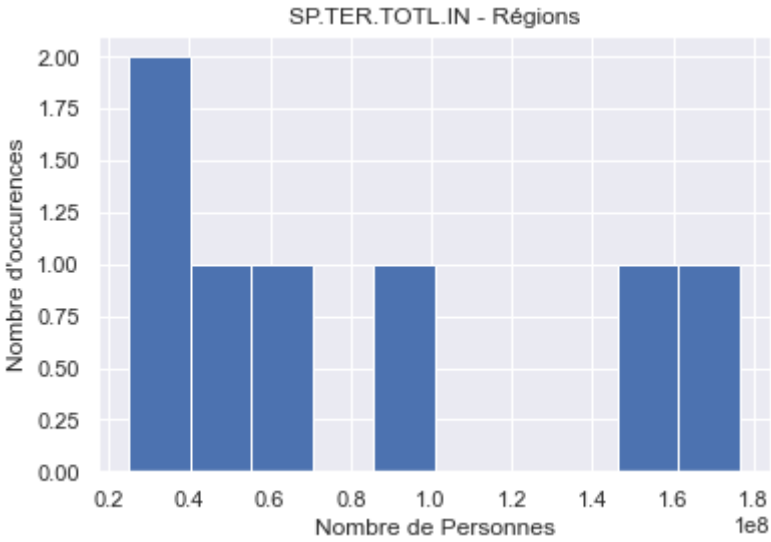



```
2014
Pays
Median:  594905.0
Mean:  3080556.553846154
Std:  11753540.096847855
```

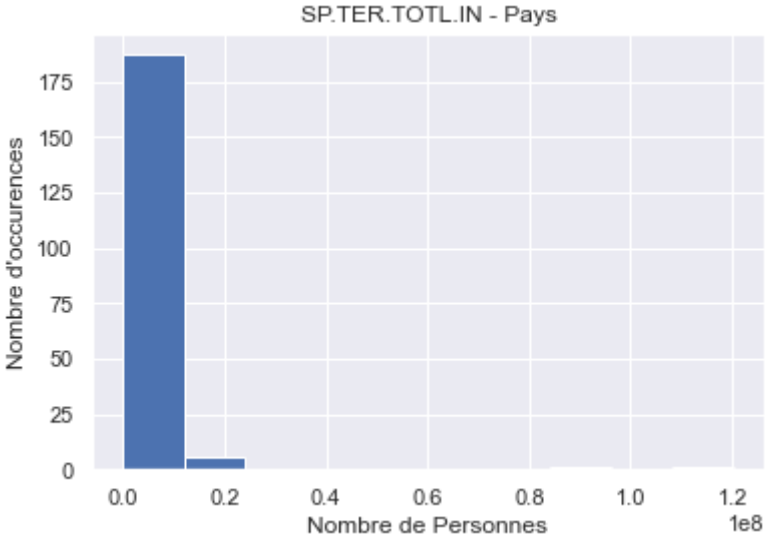



```
Régions
Median:  57917924.0
```

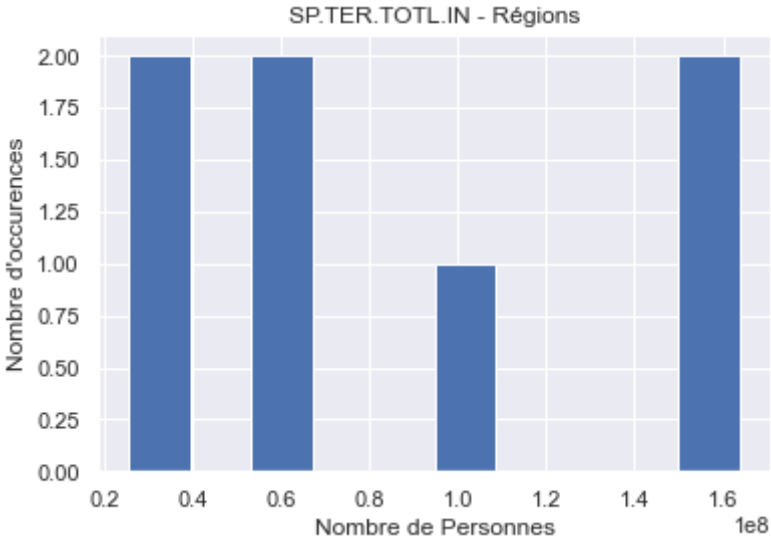
```
Mean:  86044976.0
Std:  59983195.82877391
```




```
2016
Pays
Median:  589675.0
Mean:  3032523.487008547
Std:  11284319.487551749
```




```
Régions
Median:  54949364.0
Mean:  84783474.28571428
Std:  57852846.47421718
```



Résumé: SP.TER.TOTL.IN

- **Garde 195 pays et 7 régions**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 590 000, 3 030 000 et 11 200 000 personnes pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 54 900 000, 84 800 000 et 57 900 000 personnes pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne information sur la taille du marché - nombre absolute des potentielle clients.

In [65]:

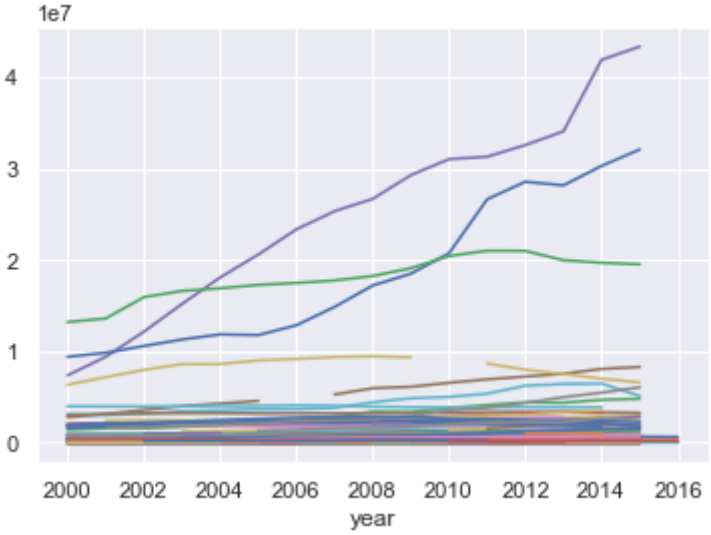
```
#save the cleaned data
data_clean['indicator'] = ind
#DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().renam
DATA_CLEAN = pd.concat([DATA_CLEAN, deconstruct_df(data_clean.copy().reset_index().r
```

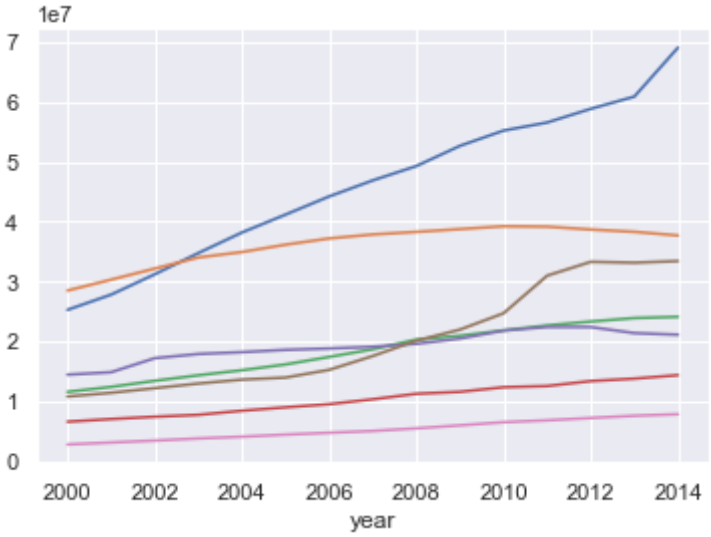
SE.TERR.ENRL

In [66]:

```
ind = 'SE.TER.ENRL'
initial_investigation(ind)[0].T.info()
```

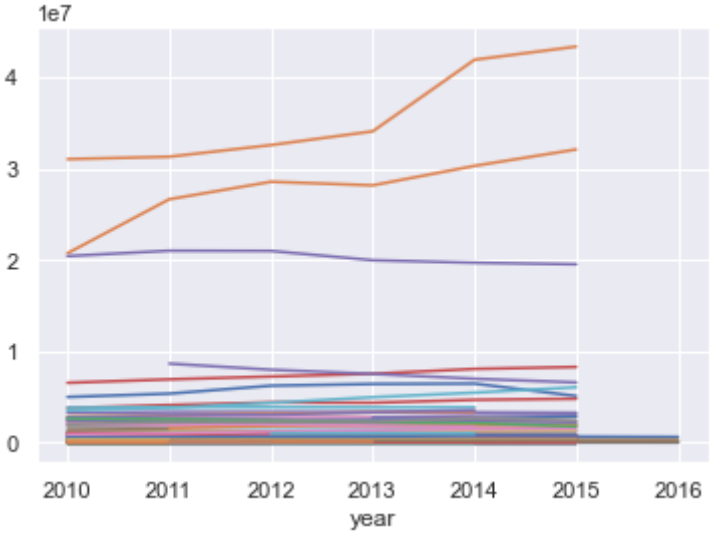
```
**************************************************
SE.TER.ENRL
```

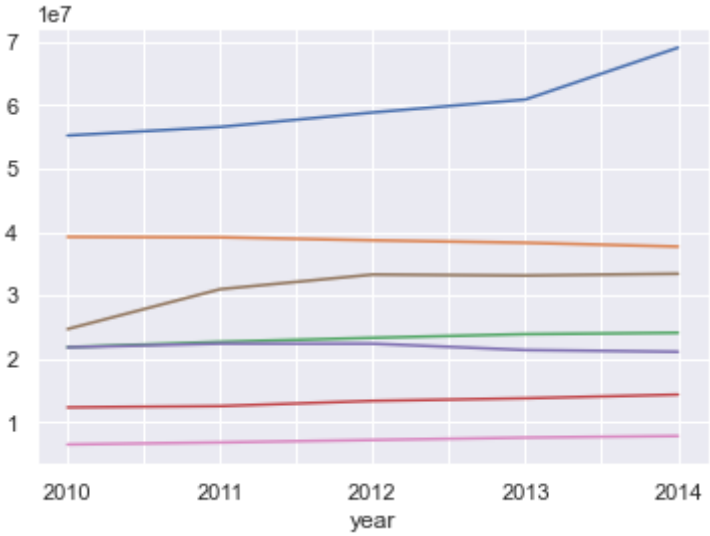





```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']
```

```
We have data for the 201 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATG' 'AUS' 'AUT'
 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BIH' 'BLR' 'BLZ' 'BMU'
 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV'
 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYM' 'CYP' 'CZE'
 'DEU' 'DJI' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST'
 'ETH' 'FIN' 'FJI' 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB'
 'GNQ' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN' 'IND' 'IRL'
 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM'
 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO'
 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD'
 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL'
 'PLW' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS'
 'SAU' 'SDN' 'SEN' 'SGP' 'SLE' 'SLV' 'SMR' 'SRB' 'SSF' 'STP' 'SUR' 'SVK'
 'SVN' 'SWE' 'SWZ' 'SXM' 'SYC' 'SYR' 'TCA' 'TCD' 'TGO' 'THA' 'TJK' 'TKM'
 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VEN'
 'VGB' 'VIR' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'DMA', 'CHI', 'VCT', 'MNP', 'PYF', 'NRU',
'SLB', 'SSD', 'TUV', 'BHS', 'GRL', 'HTI', 'MCO', 'SOM', 'FRO', 'NCL', 'KIR', 'IMN',
'XKX', 'PNG', 'GUM', 'MAF', 'GIB'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SE.TER.ENRL
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']
```

```
We have data for the 179 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE'
 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BIH' 'BLR' 'BLZ' 'BMU' 'BRA'
 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG'
 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DNK' 'DOM'
 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FRA' 'GBR'
 'GEO' 'GHA' 'GIN' 'GMB' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN'
 'IDN' 'IND' 'IRL' 'IRN' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KGZ'
 'KHM' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO'
 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD'
 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NER'
 'NGA' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'POL'
 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN'
 'SEN' 'SGP' 'SLV' 'SMR' 'SRB' 'SSF' 'STP' 'SVK' 'SVN' 'SWE' 'SWZ' 'SXM'
 'SYC' 'SYR' 'TCA' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TUN' 'TUR' 'TZA'
 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VGB' 'VNM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'DMA', 'IRQ', 'CHI', 'VCT', 'LBY', 'GNQ',
'SUR', 'MNP', 'CYM', 'VUT', 'PYF', 'NRU', 'VEN', 'SLE', 'GNB', 'NIC', 'TON', 'SLB',
'TTO', 'WSM', 'SSD', 'FJI', 'TUV', 'BHS', 'GRL', 'HTI', 'MCO', 'SOM', 'FRO', 'VIR',
'FSM', 'ASM', 'KEN', 'NCL', 'KIR', 'IMN', 'XKX', 'BOL', 'PNG', 'GUM', 'MAF', 'CAN',
'NAM', 'GIB', 'GAB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.6624102154828412

year           2010          2011          2012          2013          2014  \
count  1.470000e+02  1.490000e+02  1.480000e+02  1.380000e+02  1.310000e+02
mean   2.298180e+06  2.480601e+06  2.569238e+06  2.773283e+06  3.085502e+06
std    7.208971e+06  7.549402e+06  7.826913e+06  8.182494e+06  9.102269e+06
min    5.000000e+00  1.890000e+02  1.000000e+02  2.380000e+02  1.780000e+02
25%    5.886150e+04  7.068300e+04  6.710950e+04  9.721850e+04  1.130575e+05
50%    2.023450e+05  2.368910e+05  2.408850e+05  2.560121e+05  2.832940e+05
75%    8.522200e+05  1.021288e+06  1.023460e+06  1.232937e+06  1.629698e+06
max    5.523363e+07  5.658473e+07  5.887421e+07  6.091618e+07  6.909780e+07

year           2015           2016
count  1.100000e+02       7.000000
mean   1.573105e+06  197838.428571
std    5.494281e+06  210245.804130
min    1.940000e+02   20280.000000
25%    8.172150e+04   66778.000000
50%    2.329090e+05   94728.000000
75%    5.324260e+05  256385.500000
max    4.336739e+07  623534.000000
<class 'pandas.core.frame.DataFrame'>
Index: 179 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    147 non-null    float64
 1   2011    149 non-null    float64
 2   2012    148 non-null    float64
 3   2013    138 non-null    float64
 4   2014    131 non-null    float64
 5   2015    110 non-null    float64
 6   2016    7 non-null      float64
dtypes: float64(7)
memory usage: 11.2+ KB
```

Résumé: SE.TER.ENRL

- Pour 172 pays, il y a de la data pour les années 2010-2016.
  - 45 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.

- La data pour cette variable est complète à 66% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Le plus complète année entre 2010-2016 est 2011.**
- **Presque pas de data pour 2016**
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [67]:

```
ind = 'SE.TER.ENRL'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data = drop_countries(data,7)
len(data.columns.values)
```

```
The following countries were dropped :
['ASM', 'BHS', 'BOL', 'CAN', 'CHI', 'CYM', 'DMA', 'FJI', 'FRO', 'FSM', 'GAB', 'GIB',
'GNB', 'GNQ', 'GRL', 'GUM', 'HTI', 'IMN', 'IRQ', 'KEN', 'KIR', 'LBY', 'MAF', 'MCO',
'MNP', 'NAM', 'NCL', 'NIC', 'NRU', 'PNG', 'PYF', 'SLB', 'SLE', 'SOM', 'SSD', 'SUR',
'TON', 'TTO', 'TUV', 'VCT', 'VEN', 'VIR', 'VUT', 'WSM', 'XKX']
```

Out[67]: 179

In [68]:

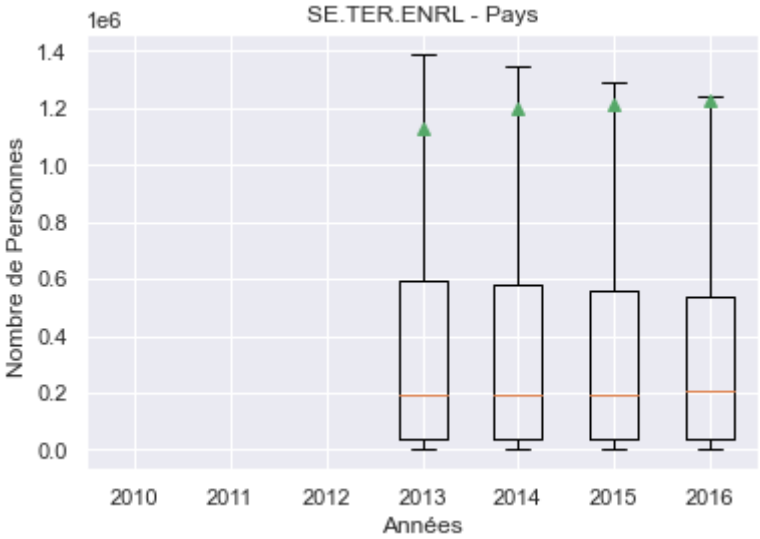
```
data_clean = fill_indicator_linear(data)
data_clean.T[years].describe()
```

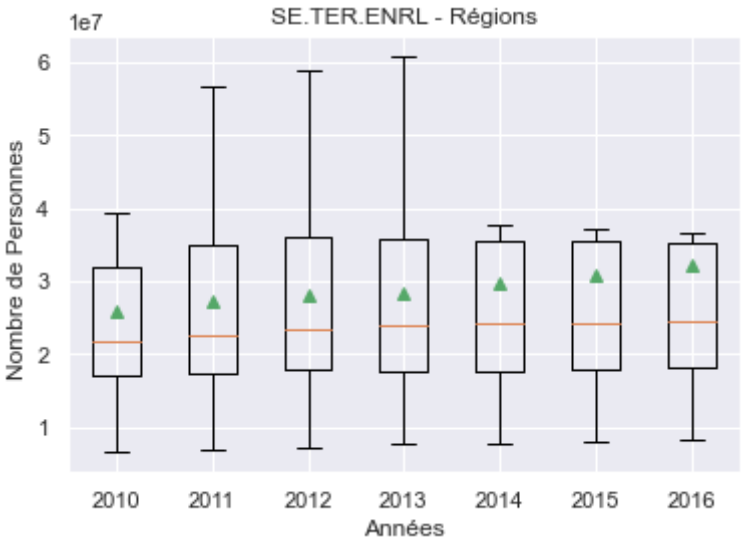
Out[68]:

	2010	2011	2012	2013	2014	2015	
count	1.740000e+02	1.760000e+02	1.780000e+02	1.790000e+02	1.790000e+02	1.790000e+02	1.790(
mean	2.033117e+06	2.119672e+06	2.169032e+06	2.194427e+06	2.309677e+06	2.371853e+06	2.434
std	6.685459e+06	6.995208e+06	7.189889e+06	7.260554e+06	7.886376e+06	8.356682e+06	8.849
min	5.000000e+00	6.120000e+01	1.000000e+02	1.736000e+02	1.780000e+02	1.940000e+02	2.100(
25%	3.343775e+04	3.541150e+04	3.951850e+04	3.935750e+04	4.146000e+04	4.165600e+04	4.253
50%	1.954315e+05	1.930655e+05	1.933920e+05	2.052470e+05	2.091700e+05	2.164680e+05	2.158
75%	6.367699e+05	6.598178e+05	6.608245e+05	6.838275e+05	7.422880e+05	7.342255e+05	7.338
max	5.523363e+07	5.658473e+07	5.887421e+07	6.091618e+07	6.909780e+07	7.727942e+07	8.546

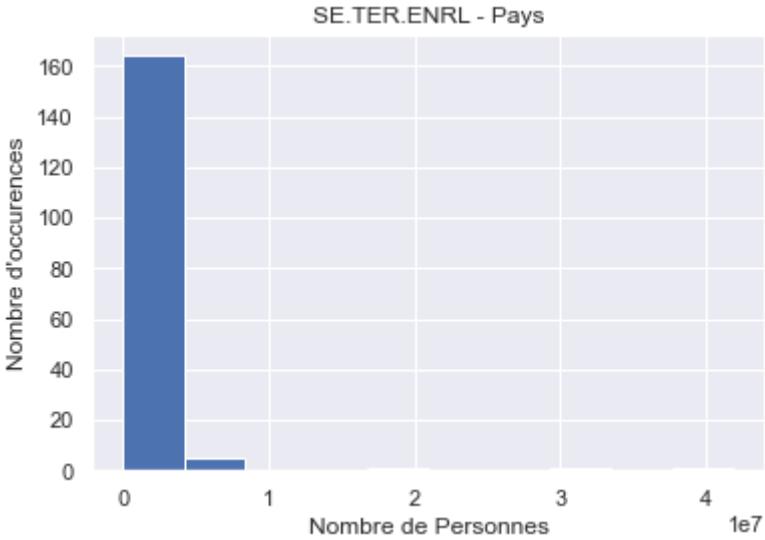
In [69]:

```
plot_boxplot(ind, data_clean, unit = "Nombre de Personnes ", years = years, showmean
print_info_year('2014', data_clean, unit = "Nombre de Personnes", ind = ind)
print_info_year('2016', data_clean, unit = "Nombre de Personnes", ind = ind)
```

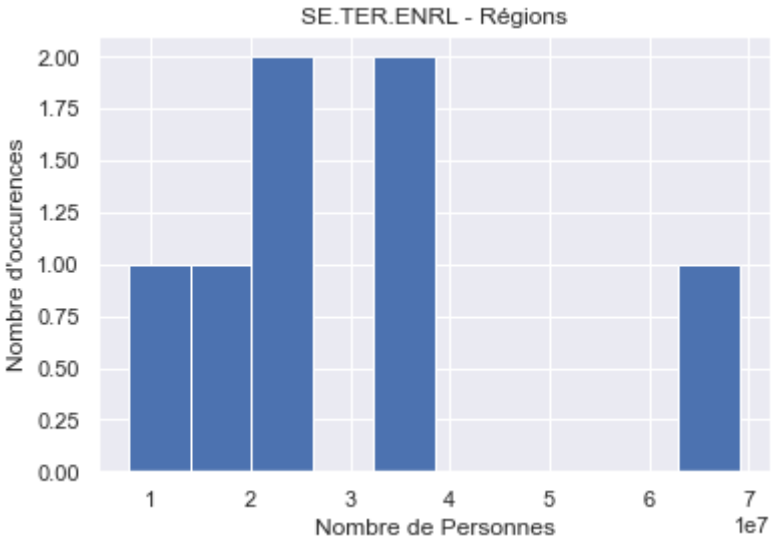




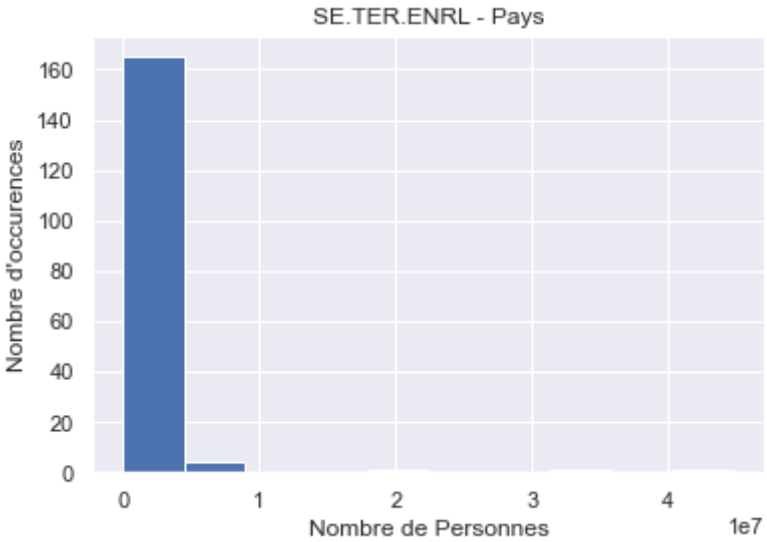

```
2014
Pays
Median:  190638.5
Mean:  1197189.7753050125
Std:  4309610.820330442
```



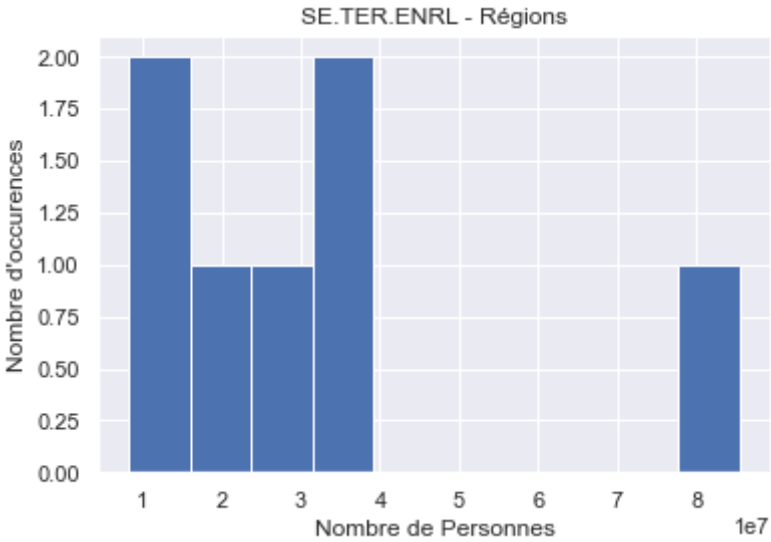

```
Régions
Median:  24087544.0
Mean:  29645084.214285713
Std:  20212674.8408585
```




```
2016
Pays
Median:  207552.5
Mean:  1226464.124146626
Std:  4602886.837419653
```




```
Régions
Median:  24503504.0
Mean:  32109215.92857143
Std:  25505352.558783434
```



Résumé: SE.TER.ENRL

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 207 000, 1 230 000 et 4 600 000 personnes pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 24 500 000, 32 100 000 et 25 500 000 personnes pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne information sur la taille du marché - nombre absolute des potentielle clients.

In [70]:

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
print(DATA.isna().sum().sum())
print(DATA_CLEANED.isna().sum().sum())
```

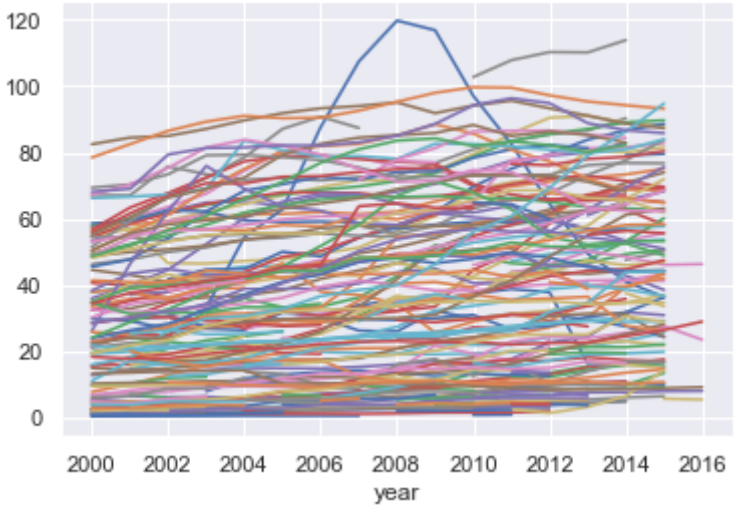
```
164156
163108
```

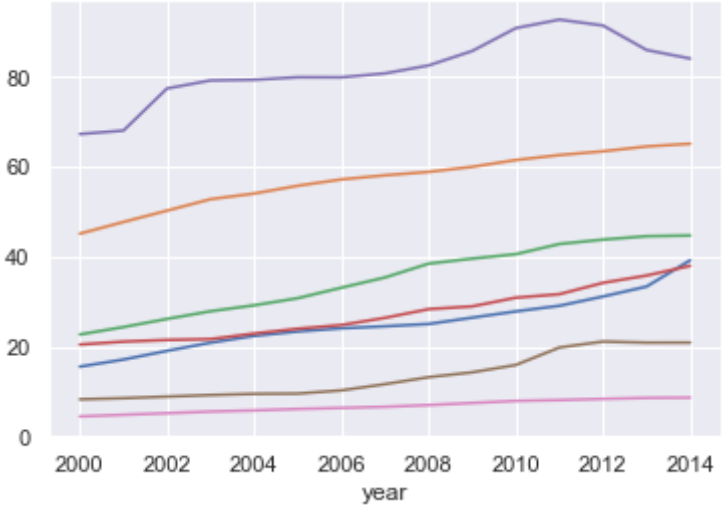
SE.TER.ENRR

In [71]:

```
ind = 'SE.TER.ENRR'
initial_investigation(ind)[0].T.info()
```

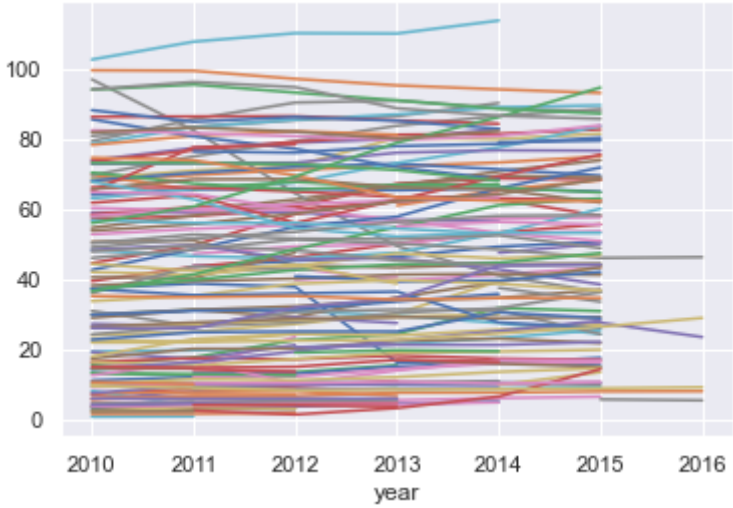
```
**************************************************
SE.TER.ENRR
```

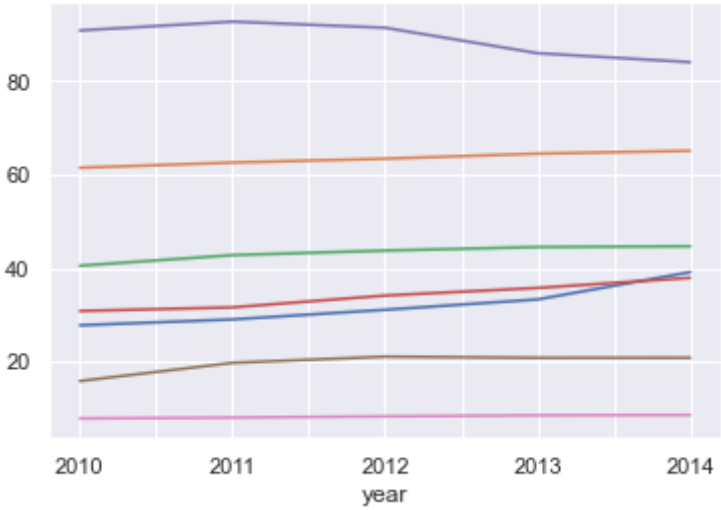





```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
```

```
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 192 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL'
 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BIH' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA' 'BRB'
 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG'
 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DNK' 'DOM'
 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA'
 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GTM'
 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KNA' 'KOR' 'KWT' 'LAO'
 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC'
 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE'
 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NER' 'NGA' 'NIC' 'NLD'
 'NOR' 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'POL' 'PRI' 'PRK' 'PRT'
 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SLE' 'SLV'
 'SMR' 'SRB' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD'
 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'UZB' 'VEN' 'VGB' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'DMA', 'CHI', 'VCT', 'MNP', 'CYM', 'PYF',
'NRU', 'SLB', 'SSD', 'TUV', 'BHS', 'SGP', 'ARE', 'GRL', 'HTI', 'MCO', 'SOM', 'FRO',
'VIR', 'ASM', 'OMN', 'NCL', 'KIR', 'TCA', 'IMN', 'XKX', 'SXM', 'PNG', 'GUM', 'MAF',
'AND', 'GIB'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SE.TER.ENRR
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']
```

```
We have data for the 172 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL'
 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BLR' 'BLZ' 'BMU' 'BRA' 'BRB' 'BRN' 'BTN'
 'BWA' 'CAF' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV'
 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS'
 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FRA' 'GBR' 'GEO' 'GHA' 'GIN'
 'GMB' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN' 'IND' 'IRL'
 'IRN' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KGZ' 'KHM' 'KNA' 'KOR'
 'KWT' 'LAO' 'LBN' 'LBR' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA'
 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR'
 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NER' 'NGA' 'NLD' 'NOR'
 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'POL' 'PRI' 'PRK' 'PRT' 'PRY'
 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SLV' 'SMR' 'SRB'
 'SSF' 'STP' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK'
 'TKM' 'TLS' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VGB' 'VNM'
 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'DMA', 'IRQ', 'CHI', 'VCT', 'LBY', 'GNQ',
'SUR', 'MNP', 'CYM', 'VUT', 'PYF', 'NRU', 'VEN', 'SLE', 'GNB', 'NIC', 'TON', 'SLB',
'TTO', 'WSM', 'SSD', 'FJI', 'TUV', 'BHS', 'SGP', 'ARE', 'GRL', 'HTI', 'MCO', 'SOM',
'FRO', 'VIR', 'FSM', 'ASM', 'OMN', 'KEN', 'NCL', 'KIR', 'TCA', 'IMN', 'XKX', 'SXM',
'BOL', 'PNG', 'GUM', 'MAF', 'AND', 'BIH', 'CAN', 'NAM', 'GIB', 'GAB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.6586378737541528

year         2010        2011        2012        2013        2014        2015  \
count  140.000000  145.000000  143.000000  132.000000  126.000000  100.000000
mean    38.491967   38.647410   40.398885   42.749837   44.428420   44.863470
std     27.826331   28.307418   27.415387   26.960216   27.209240   25.503212
min      0.692570    0.797730    1.295340    3.142330    2.565380    5.621730
25%     13.860443   12.078200   13.474695   18.784398   19.309497   24.013793
50%     35.591730   34.827381   39.863510   40.103275   43.312885   43.405649
75%     61.536581   62.561241   61.562691   63.710171   66.002331   64.801764
max    102.730873  107.828552  110.263123  110.162682  113.871788   94.732300

year        2016
count   7.000000
mean   18.078432
std    15.503049
min     5.343410
25%     6.760450
50%     9.093000
75%    26.163945
max    46.263821
<class 'pandas.core.frame.DataFrame'>
Index: 172 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    140 non-null    float64
 1   2011    145 non-null    float64
 2   2012    143 non-null    float64
 3   2013    132 non-null    float64
 4   2014    126 non-null    float64
 5   2015    100 non-null    float64
 6   2016    7 non-null      float64
dtypes: float64(7)
memory usage: 10.8+ KB
```

Résumé: SE.TER.ENRR

- Pour 165 pays, il y a de la data pour les années 2010-2016.
  - 52 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.

- La data pour cette variable est complète à 66% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Le plus complète année entre 2010-2016 est 2011.**
- **Presque pas de data pour 2016**
- Garder que tous ces 175 pays (pour l'instant).
- Voir si on peut remplir cette variable avec SE.TER.ENRL and SP.TER.TOTL.IN
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [72]:

```
ind = 'SE.TER.ENRR'
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind

ind2 = 'SE.TER.ENRL'
ind3 = 'SP.TER.TOTL.IN'
data2 = make_2D_map(DATA_CLEANED, ind2, ['indicator', 'year'], 'country').drop(colum
data3 = make_2D_map(DATA_CLEANED, ind3, ['indicator', 'year'], 'country').drop(colum
data4 = data2.div(data3).multiply(100)
```

In [73]:

```
print(data.T.isna().sum())
print(data4.T.isna().sum())
```

```
year
2000     98
2001     96
2002     86
2003     87
2004     86
2005     92
2006     92
2007     94
2008     91
2009     84
2010     84
2011     79
2012     81
2013     92
2014     98
2015    124
2016    217
dtype: int64
year
2000    96
2001    86
2002    76
2003    70
2004    66
2005    66
2006    65
2007    64
2008    61
2009    56
2010    56
2011    54
2012    53
2013    52
2014    52
2015    52
2016    52
dtype: int64
```

In [74]:

```
#fill holes will cleand data from the previous two indicators
years = [str(y) for y in range(2010, 2017)]
```

```
data_clean = data.fillna(value = data4)
data_clean.T[years].describe()
```

Out[74]:

year	2010	2011	2012	2013	2014	2015	2016
count	168.000000	170.000000	171.000000	172.000000	172.000000	172.000000	172.000000
mean	36.741256	37.415041	38.009371	38.661046	39.413400	39.830921	40.250979
std	27.010125	27.308132	27.144390	27.104799	27.402937	27.666776	28.214153
min	0.692570	0.797730	0.901064	0.997396	1.086190	1.170342	1.250207
25%	12.346864	12.149930	12.331515	13.277122	13.468025	14.452610	15.418338
50%	32.604329	32.383079	32.231331	34.332636	37.702419	36.794924	36.352352
75%	58.178133	59.595551	59.823154	62.172310	63.241416	62.792811	62.524028
max	102.730873	107.828552	110.263123	110.162682	113.871788	117.625553	121.424768

In [75]:

```
#drop countries for which there is no data in the 2010-2016
data_clean = drop_countries(data_clean.loc[years,:],7)
data_clean.T.describe()
```

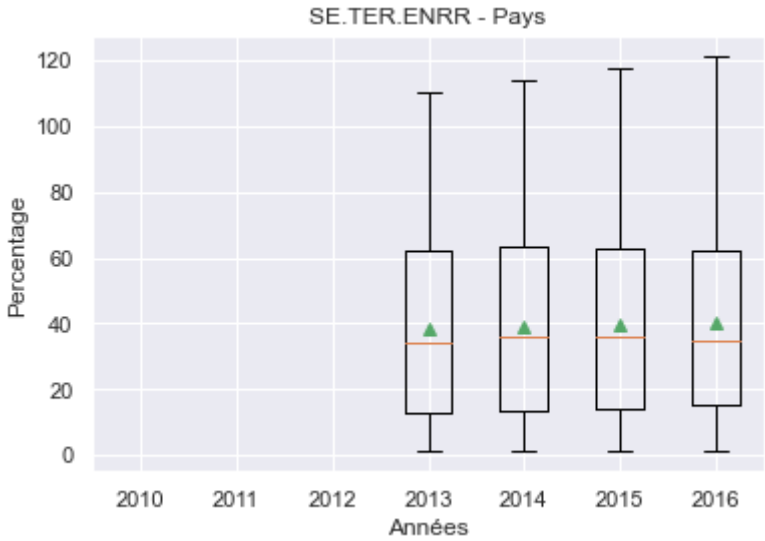
```
The following countries were dropped :
['AND', 'ARE', 'ASM', 'BHS', 'BIH', 'BOL', 'CAN', 'CHI', 'CYM', 'DMA', 'FJI', 'FRO',
'FSM', 'GAB', 'GIB', 'GNB', 'GNQ', 'GRL', 'GUM', 'HTI', 'IMN', 'IRQ', 'KEN', 'KIR',
'LBY', 'MAF', 'MCO', 'MNP', 'NAM', 'NCL', 'NIC', 'NRU', 'OMN', 'PNG', 'PYF', 'SGP',
'SLB', 'SLE', 'SOM', 'SSD', 'SUR', 'SXM', 'TCA', 'TON', 'TTO', 'TUV', 'VCT', 'VEN',
'VIR', 'VUT', 'WSM', 'XKX']
```

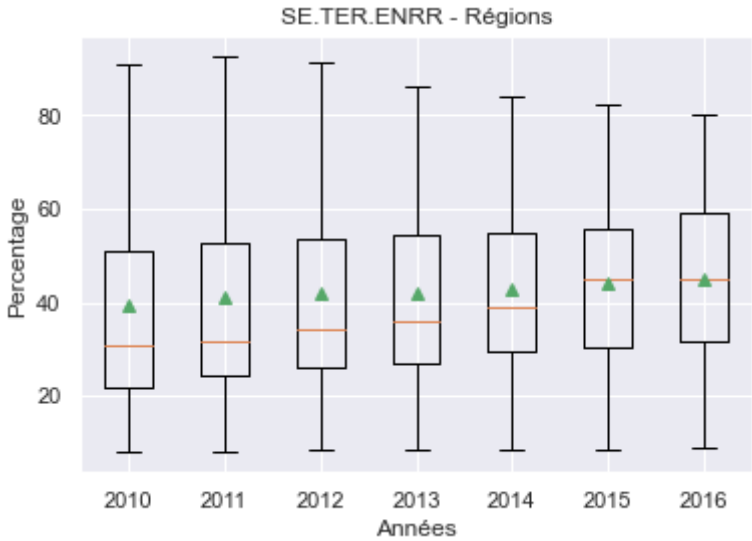
Out[75]:

year	2010	2011	2012	2013	2014	2015	2016
count	168.000000	170.000000	171.000000	172.000000	172.000000	172.000000	172.000000
mean	36.741256	37.415041	38.009371	38.661046	39.413400	39.830921	40.250979
std	27.010125	27.308132	27.144390	27.104799	27.402937	27.666776	28.214153
min	0.692570	0.797730	0.901064	0.997396	1.086190	1.170342	1.250207
25%	12.346864	12.149930	12.331515	13.277122	13.468025	14.452610	15.418338
50%	32.604329	32.383079	32.231331	34.332636	37.702419	36.794924	36.352352
75%	58.178133	59.595551	59.823154	62.172310	63.241416	62.792811	62.524028
max	102.730873	107.828552	110.263123	110.162682	113.871788	117.625553	121.424768

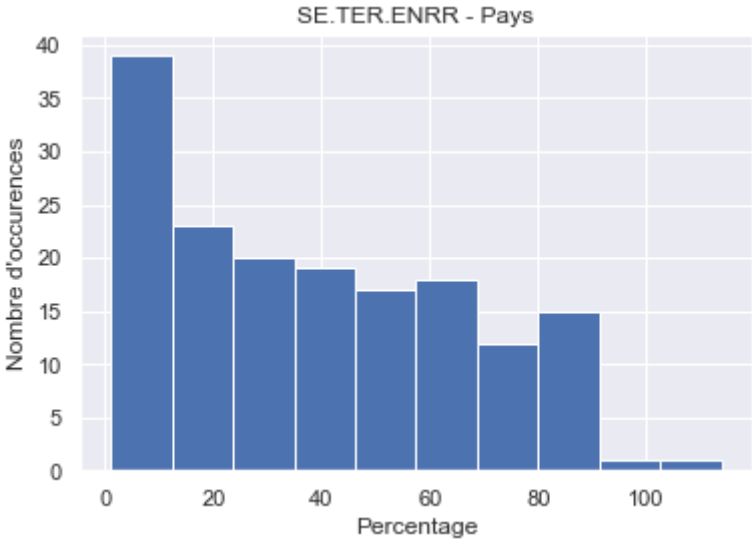
In [76]:

```
plot_boxplot(ind, data_clean, unit = "Percentage", years = years, showmeans = True)
print_info_year('2014', data_clean, unit = "Percentage", ind = ind)
print_info_year('2016', data_clean, unit = "Percentage", ind = ind)
```

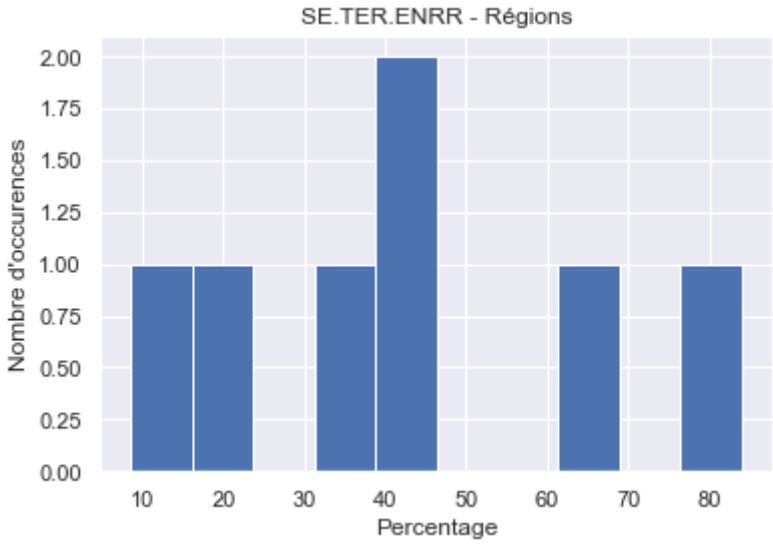




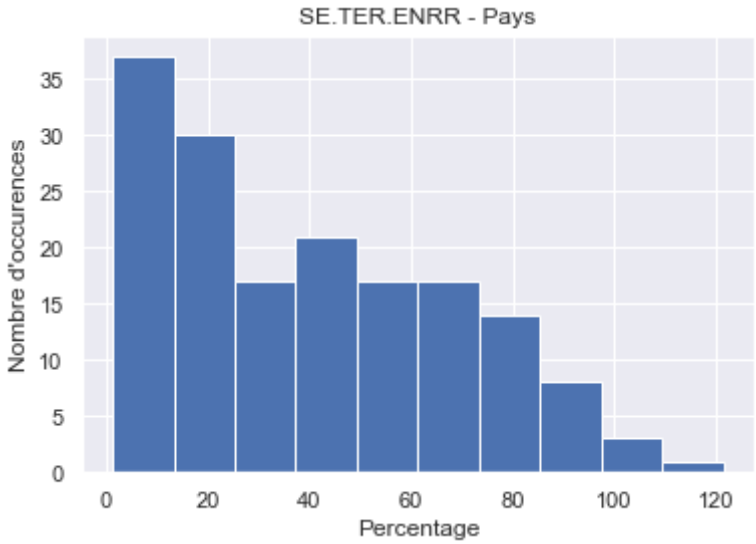

```
2014
Pays
Median:  35.7533302307129
Mean:  39.26564681410641
Std:  27.544802548059355
```




```
Régions
Median:  39.1477203369141
Mean:  42.89615508488248
Std:  25.460831132711878
```




```
2016
Pays
Median:  34.608745120348395
Mean:  40.04410691173803
Std:  28.40114885669824
```




```
Régions
Median:  44.970908533516415
Mean:  45.12724820549618
Std:  24.706674832090915
```

Résumé: SE.TER.ENRR

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 34.6%, 40.0% et 28.4% pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 45.0%, 45.1% et 24.7% pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne une indication sur le niveau d'education du pays / de la région.

In [77]:

```
#See which countries have a rate above 100%
for c in data_clean.loc[:, data_clean.loc['2016',:] >= 100].columns.values:
    print(c, RAW_DATA[RAW_DATA['Country Code'] == c]['Country Name'].iloc[0])
```

```
GRC Greece
TUR Turkey
```

In [78]:

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

Education secondaire

In [79]:

```
my_indicators = ['SE.SEC.ENRL', 'SE.SEC.ENRL.GC', 'SE.SEC.ENRR.UP',
                 'UIS.E.3','UIS.E.3.GPV','UIS.NERA.3',
                 'SP.SEC.UTOT.IN']
```

In [80]:

```
num_c = []
for ind in my_indicators :
    print(50*'*')
    print(ind)
    data, dy, dc = initial_investigation(ind, output=False)
    num_c.append(len(dc))
    print()
    print(f"There are {len(dc)} countries for the indicator {ind}.")

print()
print(f"{my_indicators[num_c.index(max(num_c))]} has the most countries.")
```

```
**************************************************
SE.SEC.ENRL
Completeness for the years 2010-2016: 0.6768885564697082

year           2010          2011          2012          2013          2014  \
count  1.580000e+02  1.630000e+02  1.570000e+02  1.480000e+02  1.480000e+02
mean   6.585123e+06  6.555549e+06  6.847195e+06  7.388847e+06  7.183491e+06
std    2.199931e+07  2.205076e+07  2.274832e+07  2.361695e+07  2.367130e+07
min    2.029000e+03  1.923000e+03  9.450000e+02  1.672000e+03  1.028000e+03
25%    1.247535e+05  1.288475e+05  1.259070e+05  2.159632e+05  1.285220e+05
50%    6.302390e+05  6.631150e+05  6.595580e+05  7.190543e+05  6.368005e+05
75%    2.659874e+06  2.605746e+06  2.611168e+06  2.757749e+06  2.648677e+06
max    1.626082e+08  1.616448e+08  1.597344e+08  1.596127e+08  1.542121e+08

year           2015          2016
count  1.230000e+02  8.000000e+00
mean   3.442501e+06  1.678644e+06
std    1.424738e+07  1.533353e+06
min    1.289000e+03  1.351000e+03
25%    1.233845e+05  5.177100e+04
50%    5.534480e+05  1.822722e+06
75%    1.554642e+06  2.703206e+06
max    1.295421e+08  3.908744e+06
```

```
There are 191 countries for the indicator SE.SEC.ENRL.
**************************************************
SE.SEC.ENRL.GC
Completeness for the years 2010-2016: 0.6843205574912892

year           2010          2011          2012          2013          2014  \
count  1.640000e+02  1.700000e+02  1.700000e+02  1.630000e+02  1.630000e+02
mean   4.985700e+06  4.991486e+06  5.039983e+06  6.026051e+06  5.921578e+06
std    1.769099e+07  1.762286e+07  1.782101e+07  2.029692e+07  2.051024e+07
min    1.817000e+03  1.835000e+03  9.450000e+02  1.672000e+03  1.028000e+03
25%    1.053198e+05  1.152208e+05  1.170765e+05  1.719895e+05  9.578200e+04
50%    4.780330e+05  4.874450e+05  5.178450e+05  5.793050e+05  5.525470e+05
75%    2.057766e+06  2.144308e+06  2.431328e+06  2.500209e+06  2.511417e+06
max    1.352937e+08  1.407599e+08  1.474491e+08  1.493493e+08  1.506857e+08

year           2015          2016
count  1.400000e+02  1.200000e+01
mean   2.930014e+06  1.064239e+06
std    1.245703e+07  1.187859e+06
min    1.289000e+03  1.037000e+03
25%    1.094588e+05  4.721600e+04
50%    4.585145e+05  6.045655e+05
75%    1.542604e+06  1.842795e+06
max    1.277872e+08  3.252523e+06

There are 205 countries for the indicator SE.SEC.ENRL.GC.
**************************************************
SE.SEC.ENRR.UP
Completeness for the years 2010-2016: 0.6762422360248447

year         2010        2011        2012        2013        2014        2015  \
count  152.000000  157.000000  153.000000  144.000000  142.000000  115.000000
mean    73.684680   72.513417   74.378254   79.556849   81.937742   80.488933
std     31.045156   32.364155   32.325351   33.665308   34.069408   35.725014
min      4.539990    4.614590    6.595380    8.227430   10.214850    5.020860
25%     50.799011   48.371780   52.597542   56.605623   57.734553   57.270521
50%     78.434319   78.178322   78.891022   83.800797   85.145084   84.507851
75%     94.822105   94.961700   95.746696   99.155041  100.225294   99.781548
max    168.520767  175.029724  181.990967  181.605743  186.427887  194.101990

year         2016
count    8.000000
mean    74.943194
std     29.076538
min     40.489799
25%     49.768929
50%     76.880419
75%    100.143446
max    107.495377

There are 184 countries for the indicator SE.SEC.ENRR.UP.
**************************************************
UIS.E.3
Completeness for the years 2010-2015: 0.6693262411347518

year           2010          2011          2012          2013          2014  \
count  1.560000e+02  1.610000e+02  1.590000e+02  1.470000e+02  1.250000e+02
mean   2.845537e+06  2.837781e+06  2.936010e+06  3.277591e+06  3.047275e+06
std    9.581047e+06  9.741906e+06  1.015763e+07  1.073568e+07  1.043027e+07
min    1.221000e+03  1.259000e+03  2.540000e+02  1.150000e+03  3.530000e+02
25%    6.487025e+04  6.895500e+04  6.193450e+04  9.068100e+04  5.081500e+04
50%    2.492330e+05  2.443700e+05  2.415230e+05  2.835860e+05  2.312500e+05
75%    9.911418e+05  9.799870e+05  9.244300e+05  1.258817e+06  8.043760e+05
max    6.869628e+07  6.940019e+07  6.968986e+07  7.171130e+07  6.985116e+07

year           2015
count  7.000000e+00
mean   3.883340e+05
```

```
std    5.298411e+05
min    8.000000e+02
25%    1.449950e+04
50%    2.936400e+04
75%    6.590840e+05
max    1.341007e+06

There are 188 countries for the indicator UIS.E.3.
**************************************************
UIS.E.3.GPV
Completeness for the years 2010-2015: 0.6708333333333333

year           2010          2011          2012          2013          2014  \
count  1.640000e+02  1.680000e+02  1.670000e+02  1.600000e+02  1.380000e+02
mean   1.776256e+06  1.786979e+06  1.861807e+06  2.316515e+06  2.098210e+06
std    6.447122e+06  6.610929e+06  6.929807e+06  8.234839e+06  7.653458e+06
min    2.960000e+02  3.270000e+02  2.540000e+02  8.420000e+02  3.530000e+02
25%    3.547600e+04  3.691050e+04  4.014450e+04  5.255350e+04  4.012150e+04
50%    1.248595e+05  1.385600e+05  1.341640e+05  1.519625e+05  1.360345e+05
75%    6.414100e+05  6.439515e+05  6.935440e+05  9.058162e+05  7.138005e+05
max    5.829052e+07  6.199414e+07  6.681197e+07  6.671974e+07  6.734470e+07

year           2015
count  8.000000e+00
mean   3.336176e+05
std    4.820599e+05
min    8.000000e+02
25%    8.679500e+03
50%    1.252175e+05
75%    4.110245e+05
max    1.317580e+06

There are 200 countries for the indicator UIS.E.3.GPV.
**************************************************
UIS.NERA.3
Completeness for the years 2010-2015: 0.42857142857142855

year        2010       2011       2012        2013       2014       2015
count  57.000000  57.000000  53.000000  100.000000  90.000000   3.000000
mean   44.446078  48.107520  47.454504   58.217231  58.335861  51.858532
std    26.405450  26.605186  25.969731   26.553220  27.348290  35.966802
min     1.963860   2.116620   2.430520    3.171050   3.523080  24.985229
25%    23.125990  24.829790  26.958281   34.884692  37.780292  31.429079
50%    49.621479  52.960419  52.095139   63.232365  63.511044  37.872929
75%    66.383163  68.674171  65.475487   80.650431  81.868164  65.295183
max    93.581871  94.768677  94.926010   97.071823  94.302933  92.717438

There are 140 countries for the indicator UIS.NERA.3.
**************************************************
SP.SEC.UTOT.IN
Completeness for the years 2010-2016: 0.957983193277311

year           2010          2011          2012          2013          2014  \
count  2.020000e+02  2.030000e+02  2.030000e+02  2.030000e+02  2.040000e+02
mean   3.809156e+06  3.822181e+06  3.795956e+06  3.780282e+06  3.735734e+06
std    1.489671e+07  1.475462e+07  1.465482e+07  1.456892e+07  1.444357e+07
min    4.320000e+02  3.570000e+02  3.630000e+02  3.750000e+02  3.930000e+02
25%    8.389075e+04  8.490000e+04  8.405200e+04  8.104250e+04  7.675700e+04
50%    3.525685e+05  3.687200e+05  3.835140e+05  3.717900e+05  3.669760e+05
75%    1.387539e+06  1.463500e+06  1.461264e+06  1.474812e+06  1.467564e+06
max    1.295778e+08  1.305978e+08  1.315686e+08  1.325174e+08  1.333664e+08

year           2015          2016
count  1.950000e+02  1.580000e+02
mean   1.943047e+06  1.255435e+06
std    8.080003e+06  4.045436e+06
min    4.080000e+02  4.160000e+02
25%    7.299050e+04  5.009550e+04
50%    3.301490e+05  2.953205e+05
```

```
75%    1.296611e+06  1.073032e+06
max    9.919816e+07  4.514491e+07

There are 204 countries for the indicator SP.SEC.UTOT.IN.

SE.SEC.ENRL.GC has the most countries.
```

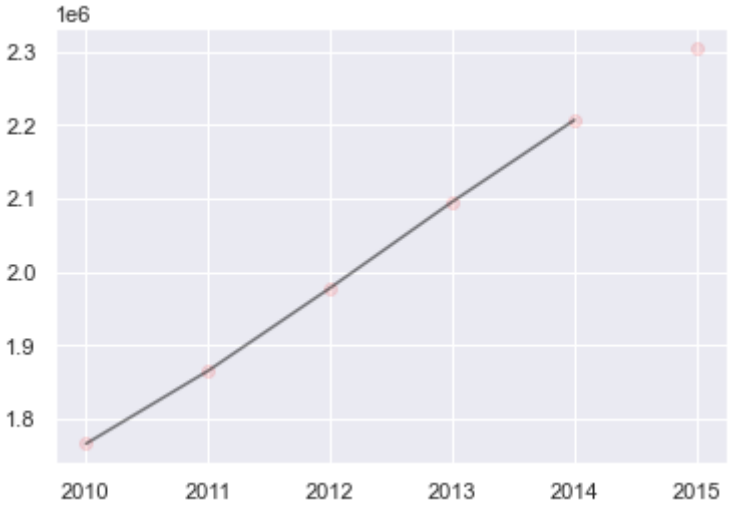
In [81]:

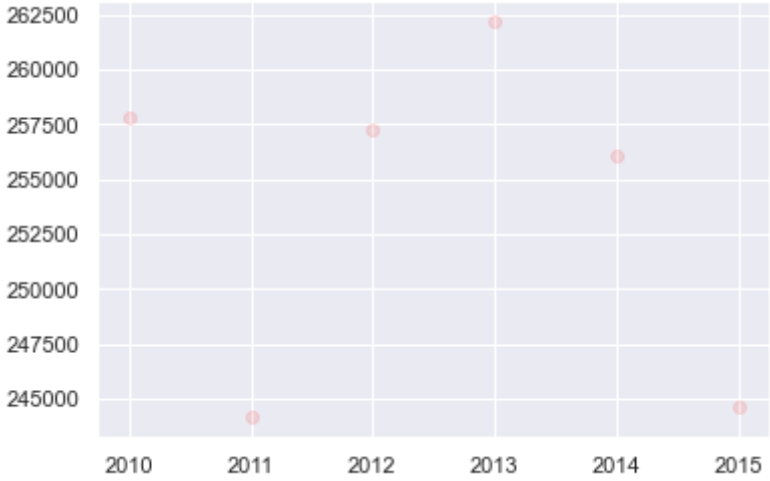
```
ind = 'SP.SEC.UTOT.IN'
ind2 = 'UIS.E.3'
ind3 = 'SE.SEC.ENRR.UP'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
data = drop_countries(data,len(years))

data2 = make_2D_map(DATA, ind2, ['indicator', 'year'], 'country').drop(columns = ['i
data3 = make_2D_map(DATA, ind3, ['indicator', 'year'], 'country').drop(columns = ['i
data4 = data2.div(data3).multiply(100)
```

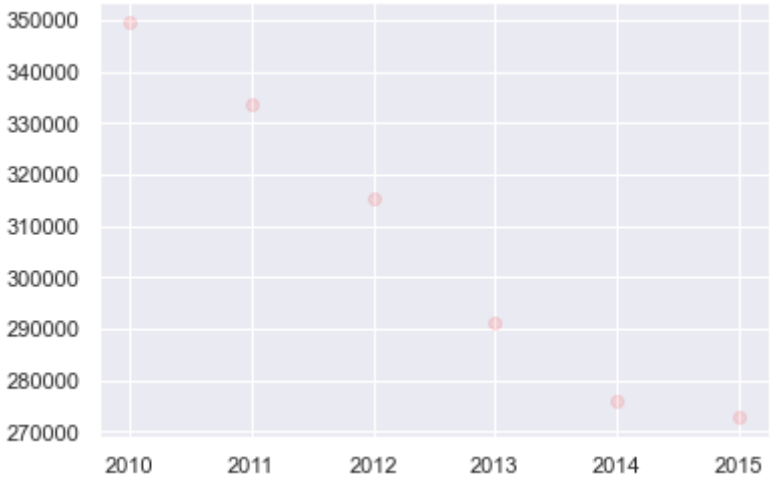
```
The following countries were dropped :
['AND', 'ASM', 'BIH', 'CHI', 'CYM', 'FRO', 'GIB', 'GRL', 'GUM', 'IMN', 'MAF', 'MCO',
'MNP', 'NCL', 'PYF', 'SGP', 'SXM', 'TCA', 'VIR', 'XKX']
```

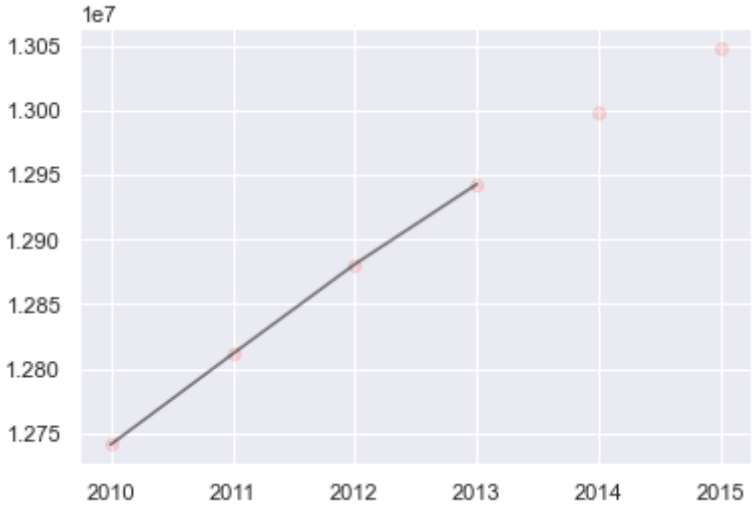
In [82]:

```
data_holes = data.loc[: ,data.isna().sum() != 0]
for c in data_holes.columns.values :
    plt.plot(data_holes[c], 'o', alpha=0.1, color = 'red')
    plt.plot(data4[c],'-', alpha=0.5, color = 'black')
    plt.show()
```

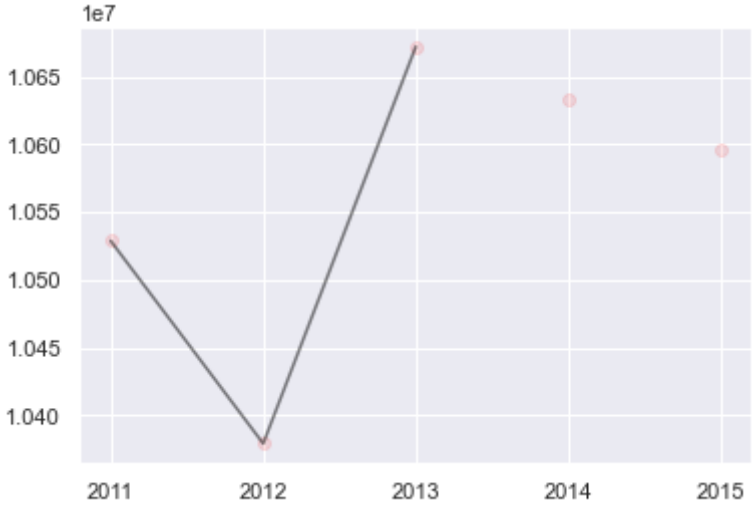


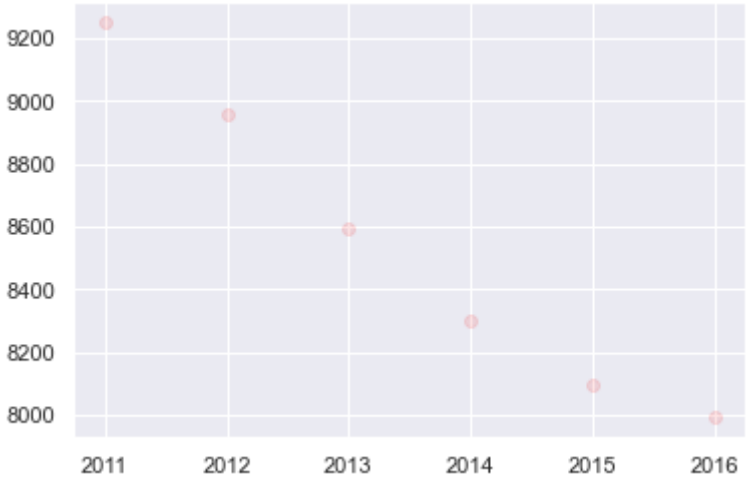


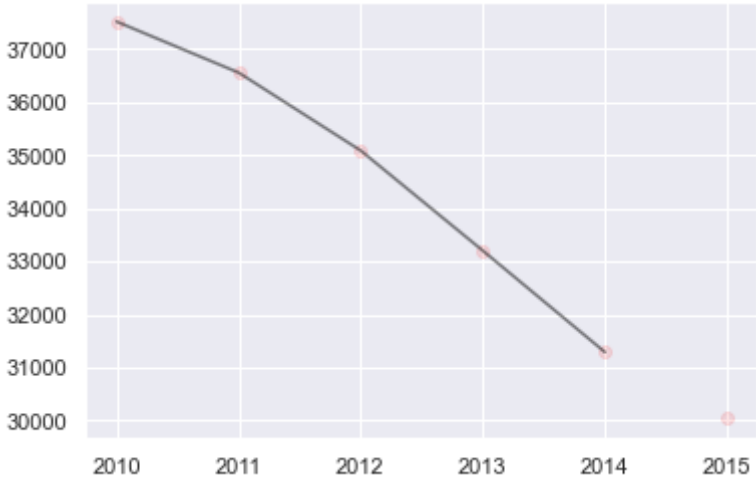

350000
340000
330000
320000
310000
300000
290000
280000
270000
2010
2011
2012
2013
2014
2015

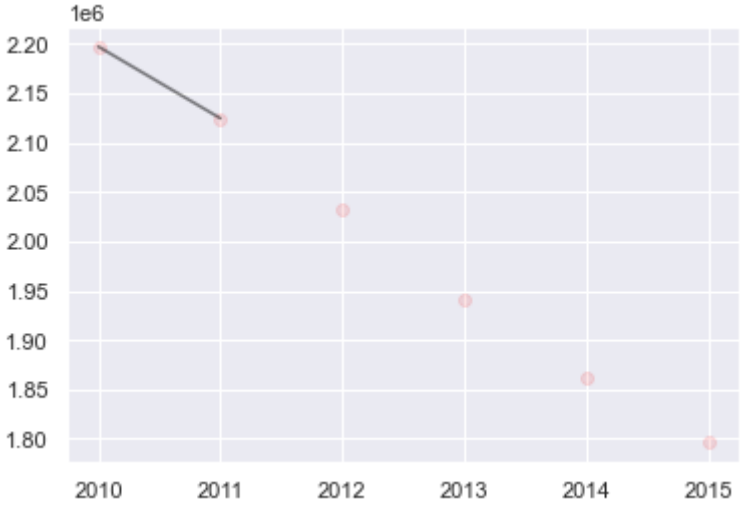

1e7
1.305
1.300
1.295
1.290
1.285
1.280
1.275
2010
2011
2012
2013
2014
2015

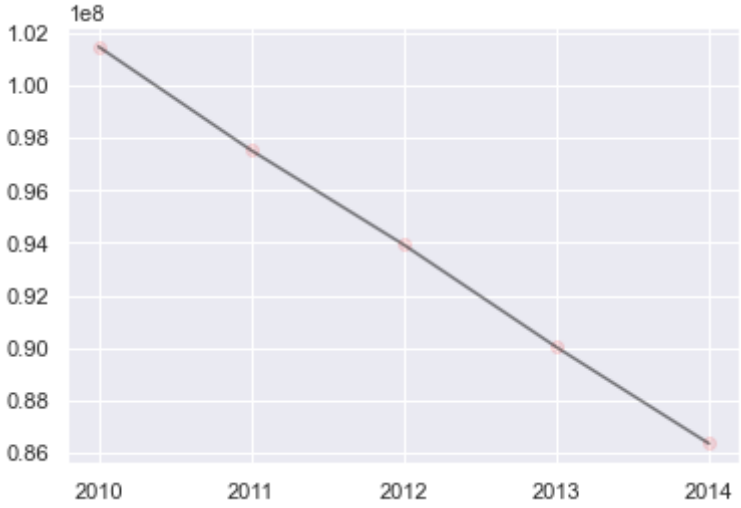
45000
44000
43000
42000
41000
2010
2011
2012
2013
2014
2015

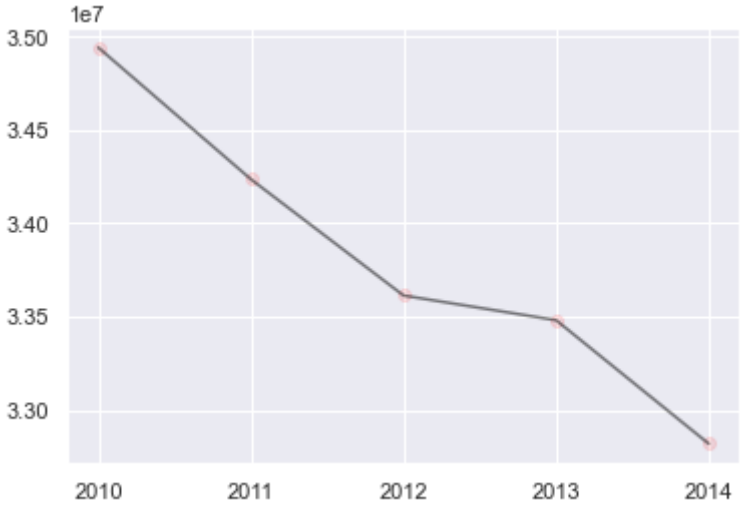

1e7
1.065
1.060
1.055
1.050
1.045
1.040
2011
2012
2013
2014
2015

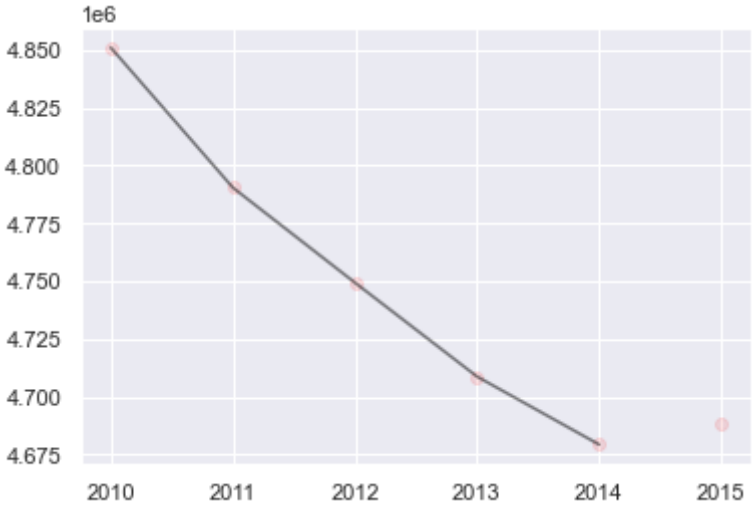

9200
9000
8800
8600
8400
8200
8000
2011
2012
2013
2014
2015
2016

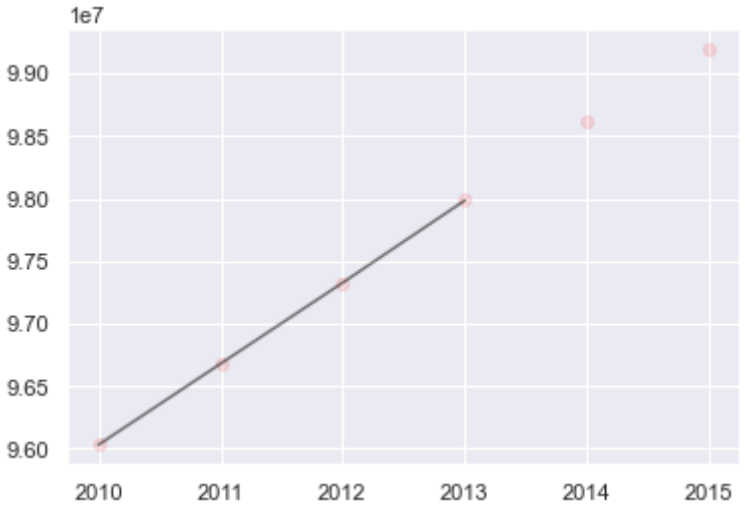

37000
36000
35000
34000
33000
32000
31000
30000
2010
2011
2012
2013
2014
2015

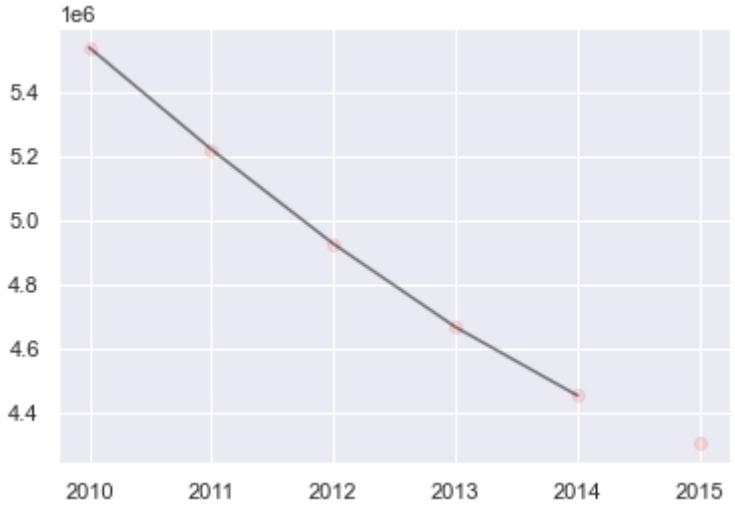

1e6
2.20
2.15
2.10
2.05
2.00
1.95
1.90
1.85
1.80
2010
2011
2012
2013
2014
2015

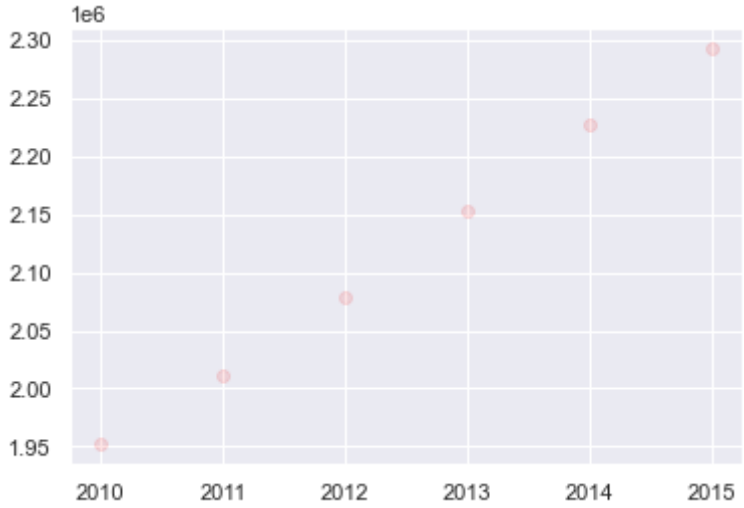

1e8
1.02
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
2010
2011
2012
2013
2014

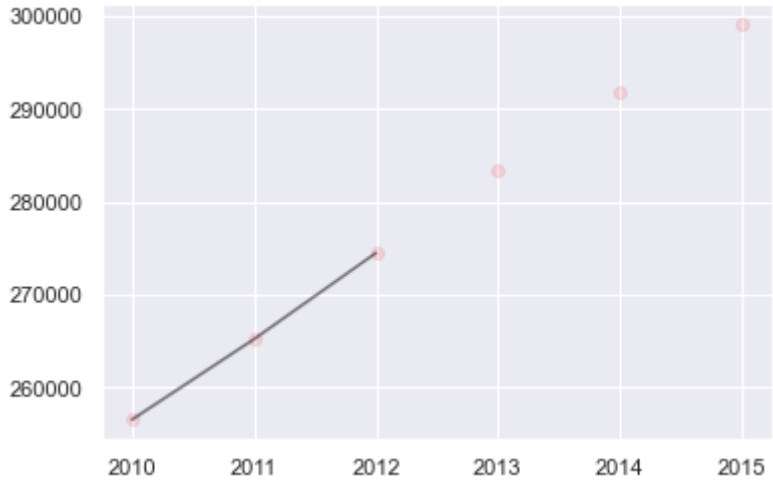

1e7
3.50
3.45
3.40
3.35
3.30
2010
2011
2012
2013
2014

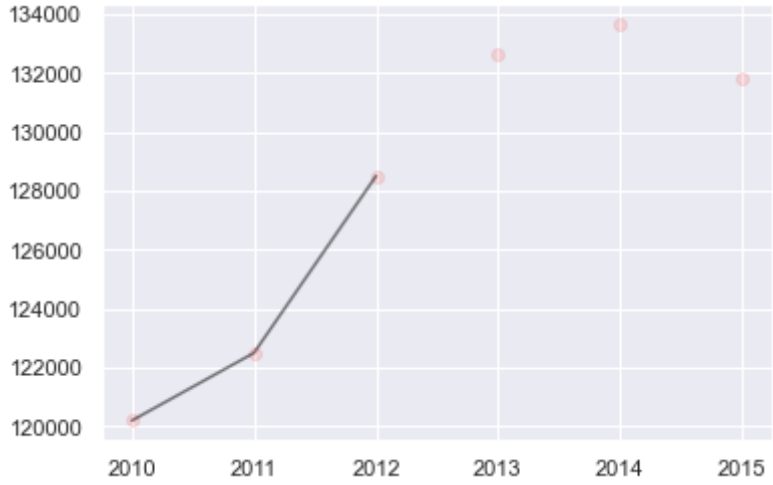

1e6
4.850
4.825
4.800
4.775
4.750
4.725
4.700
4.675
2010
2011
2012
2013
2014
2015

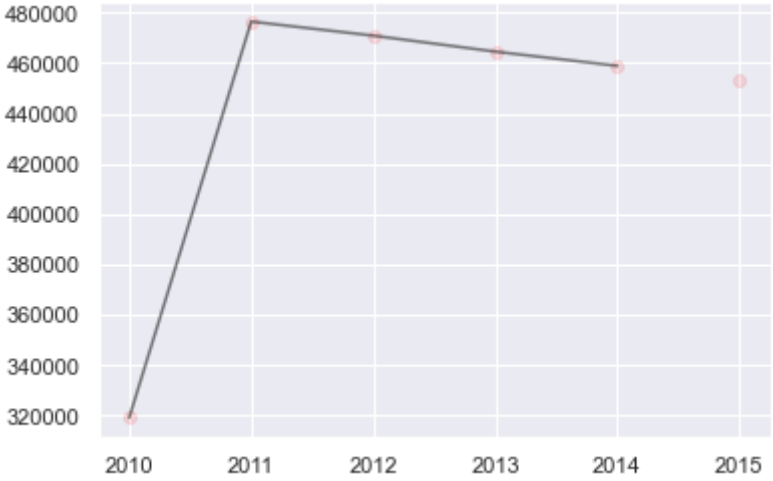

1e7
9.90
9.85
9.80
9.75
9.70
9.65
9.60
2010
2011
2012
2013
2014
2015

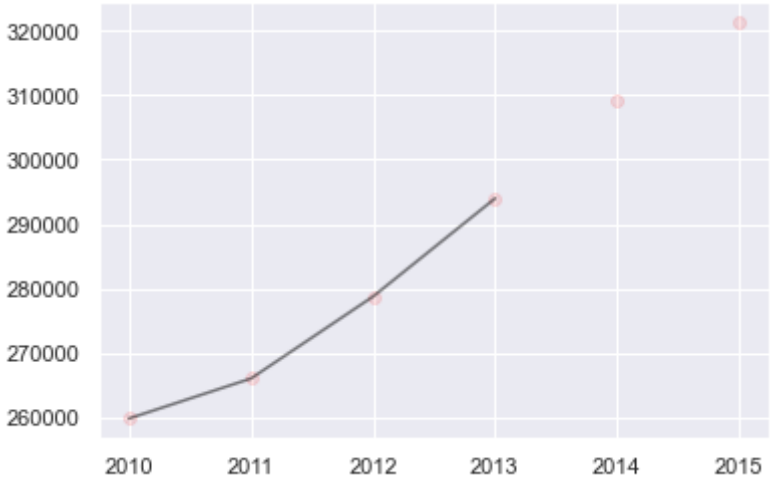

1e6
5.4
5.2
5.0
4.8
4.6
4.4
2010
2011
2012
2013
2014
2015

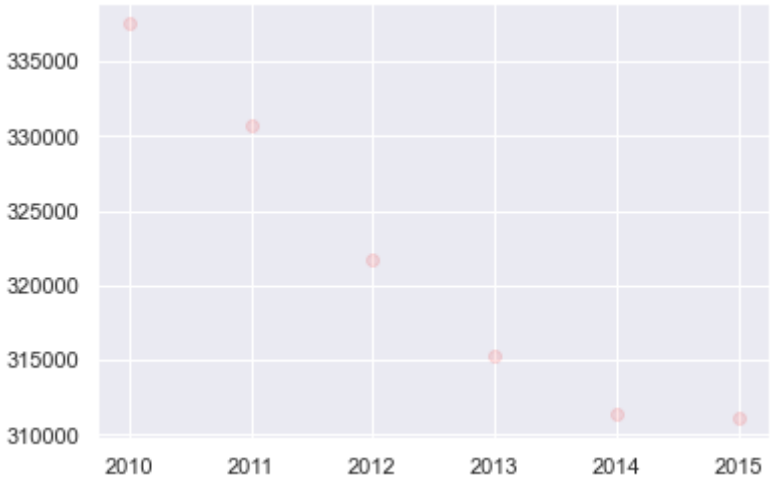

1e6
2.30
2.25
2.20
2.15
2.10
2.05
2.00
1.95
2010
2011
2012
2013
2014
2015

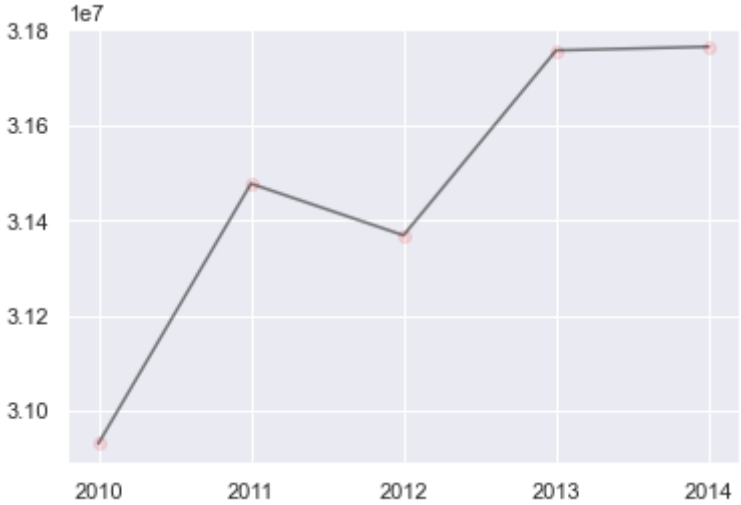

300000
290000
280000
270000
260000
2010
2011
2012
2013
2014
2015

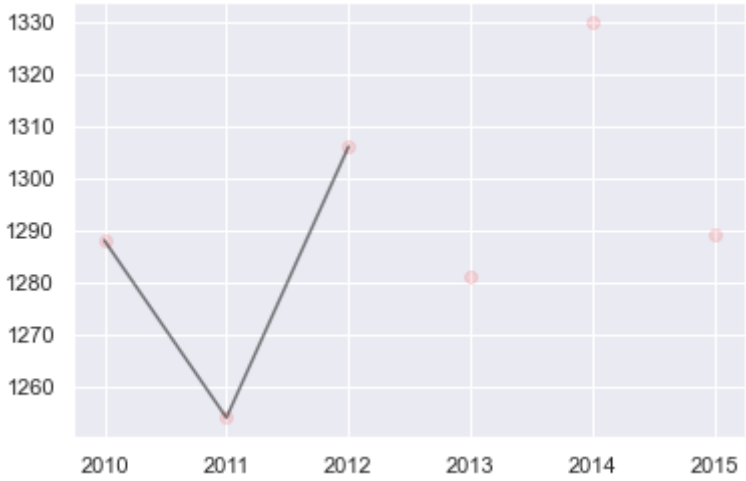

134000
132000
130000
128000
126000
124000
122000
120000
2010
2011
2012
2013
2014
2015

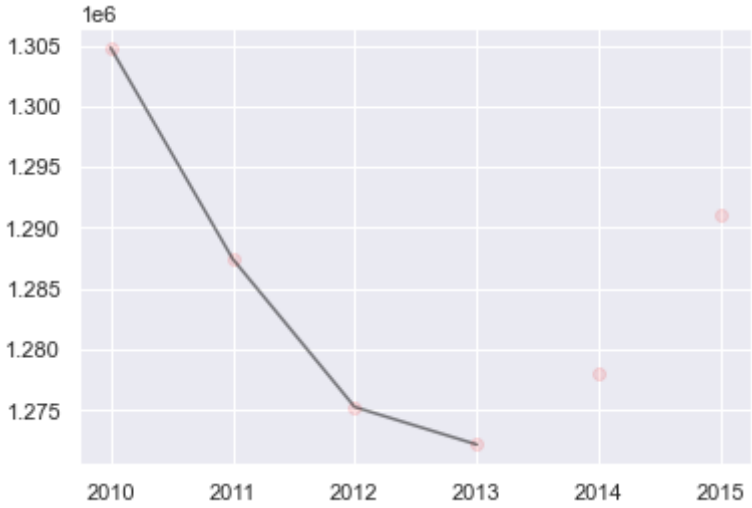

480000
460000
440000
420000
400000
380000
360000
340000
320000
2010
2011
2012
2013
2014
2015

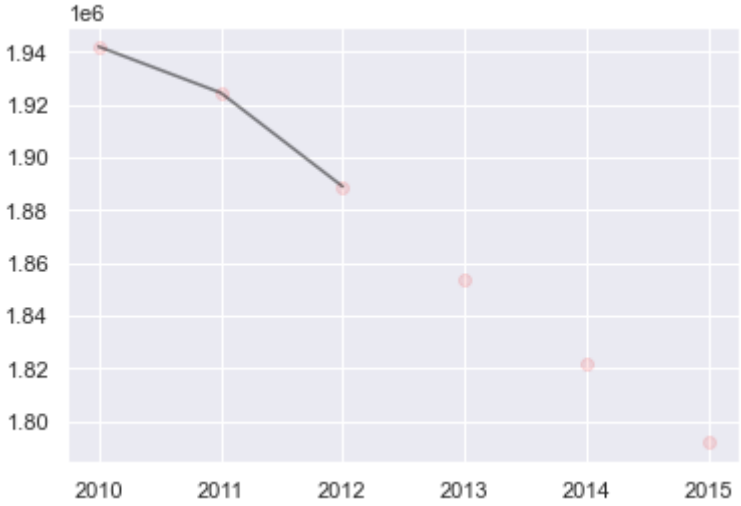

320000
310000
300000
290000
280000
270000
260000
2010
2011
2012
2013
2014
2015

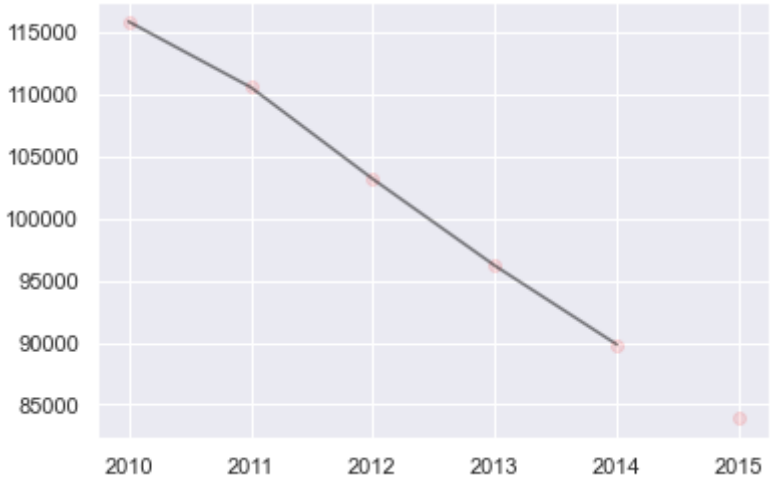

335000
330000
325000
320000
315000
310000
2010
2011
2012
2013
2014
2015

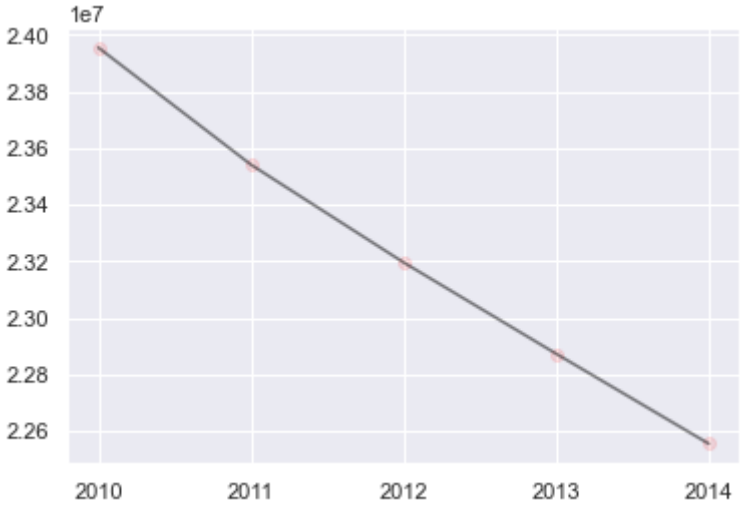

1e7
3.18
3.16
3.14
3.12
3.10
2010
2011
2012
2013
2014

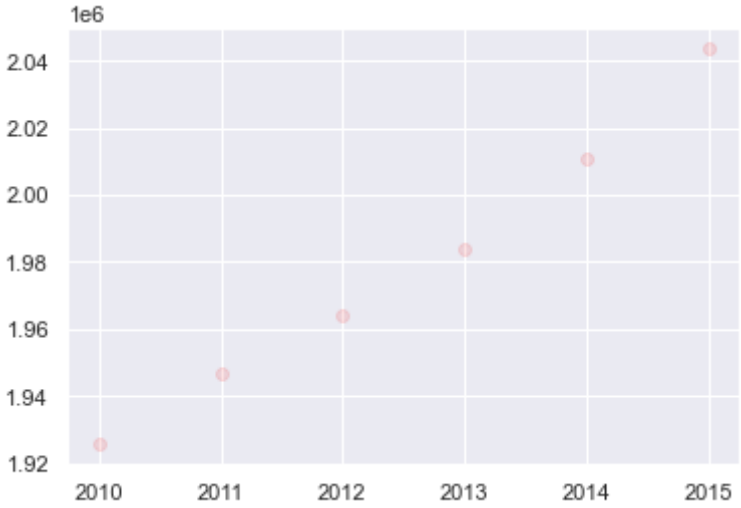

1330
1320
1310
1300
1290
1280
1270
1260
2010
2011
2012
2013
2014
2015

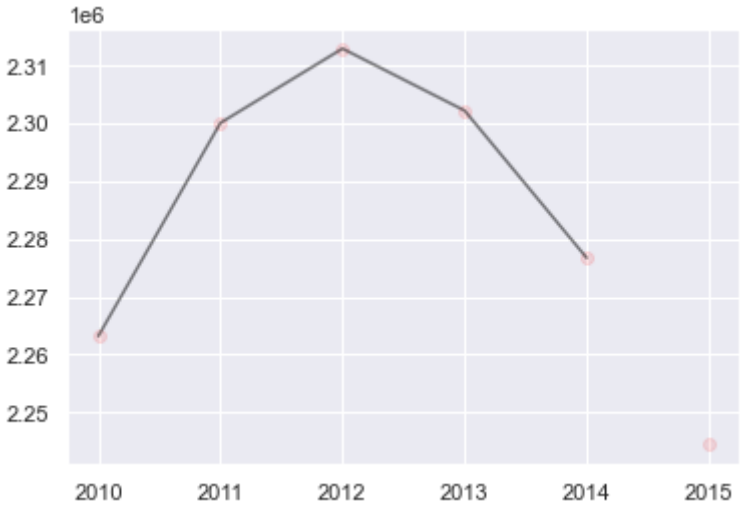

1e6
1.305
1.300
1.295
1.290
1.285
1.280
1.275
2010
2011
2012
2013
2014
2015

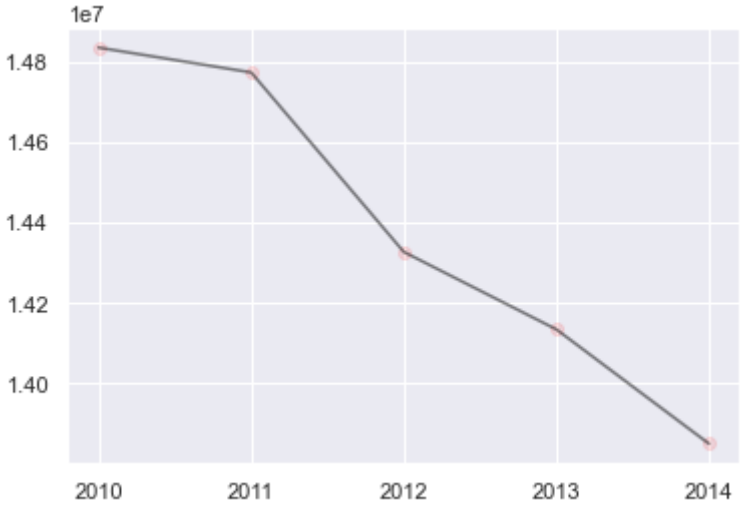

1e6
1.94
1.92
1.90
1.88
1.86
1.84
1.82
1.80
2010
2011
2012
2013
2014
2015

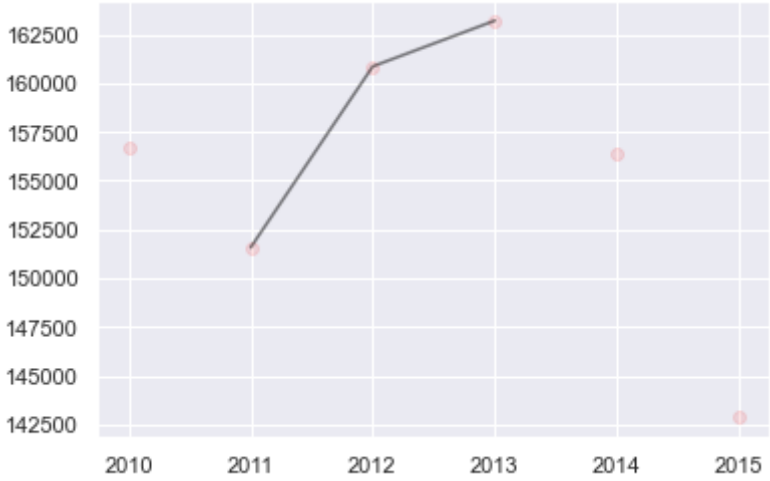

115000
110000
105000
100000
95000
90000
85000
2010
2011
2012
2013
2014
2015

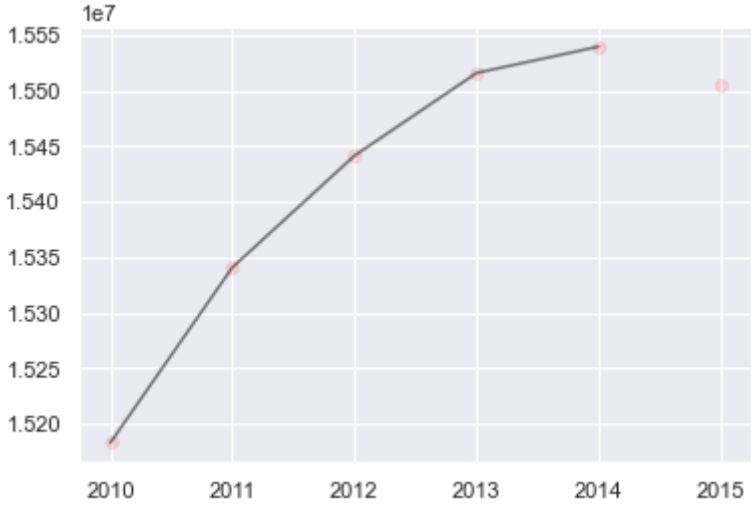

1e7
2.40
2.38
2.36
2.34
2.32
2.30
2.28
2.26
2010
2011
2012
2013
2014

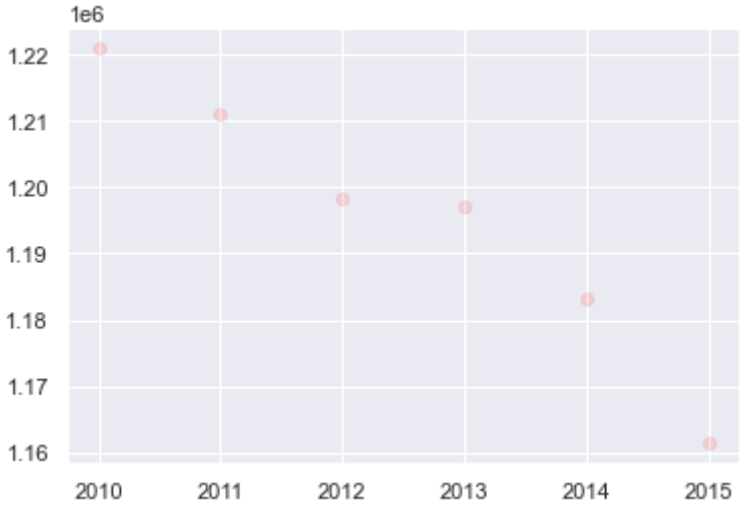

1e6
2.04
2.02
2.00
1.98
1.96
1.94
1.92
2010
2011
2012
2013
2014
2015

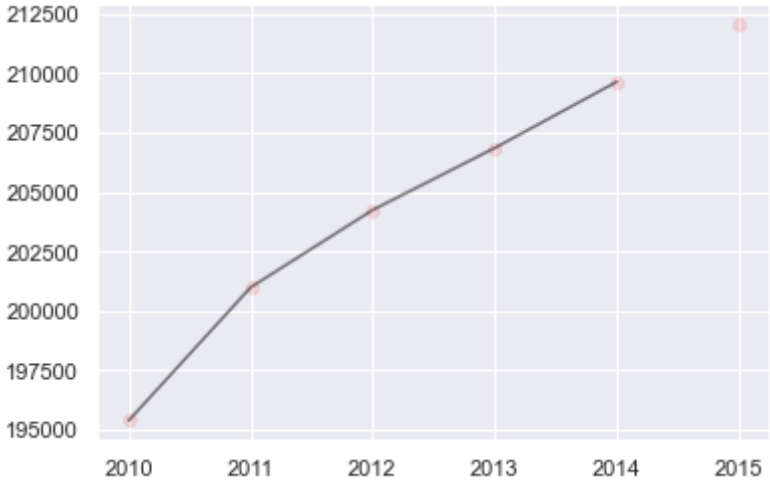

1e6
2.31
2.30
2.29
2.28
2.27
2.26
2.25
2010
2011
2012
2013
2014
2015

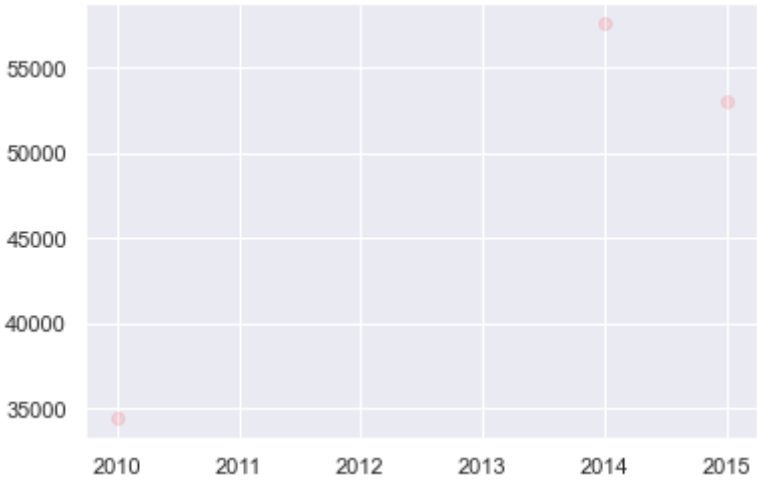

1e7
1.48
1.46
1.44
1.42
1.40
2010
2011
2012
2013
2014

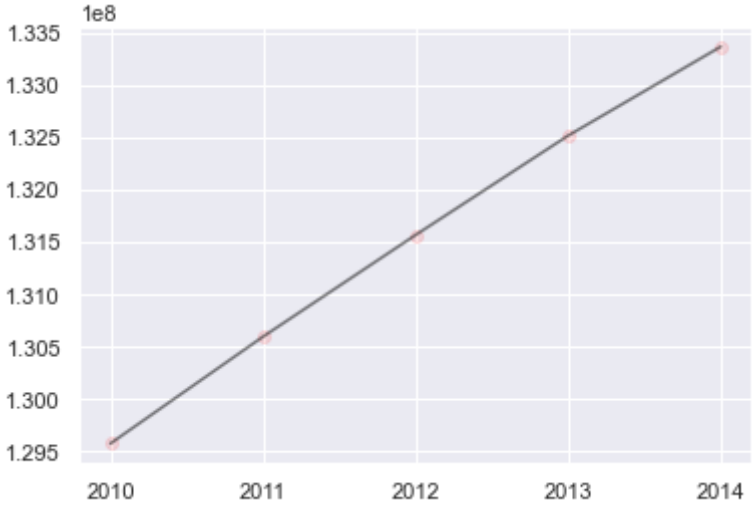

162500
160000
157500
155000
152500
150000
147500
145000
142500
2010
2011
2012
2013
2014
2015

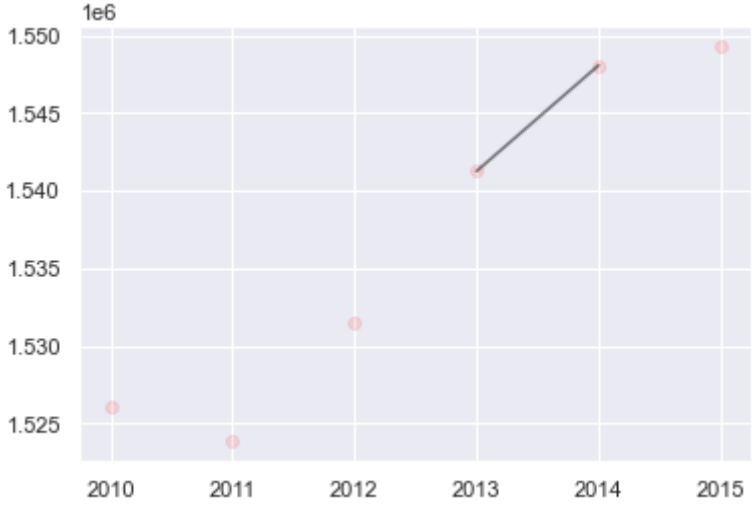

1e7
1.555
1.550
1.545
1.540
1.535
1.530
1.525
1.520
2010
2011
2012
2013
2014
2015

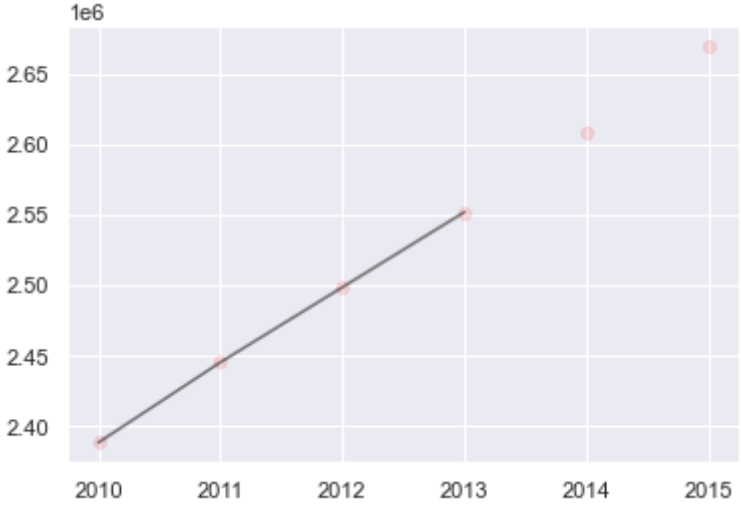

1e6
1.22
1.21
1.20
1.19
1.18
1.17
1.16
2010
2011
2012
2013
2014
2015

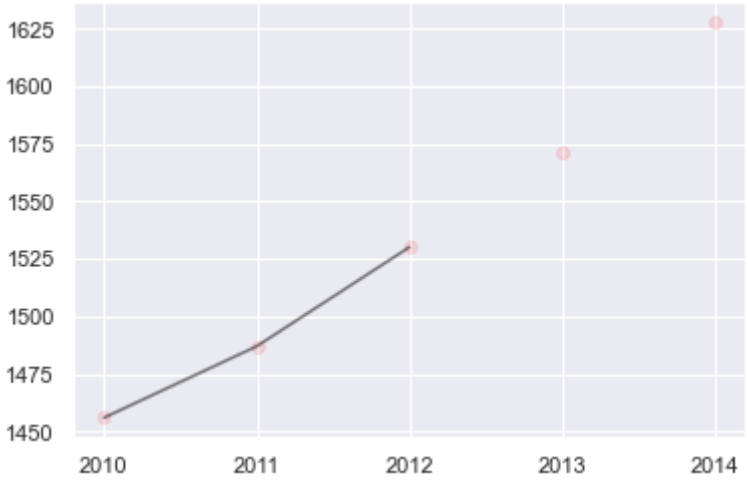

212500
210000
207500
205000
202500
200000
197500
195000
2010
2011
2012
2013
2014
2015


55000
50000
45000
40000
35000
2010
2011
2012
2013
2014
2015

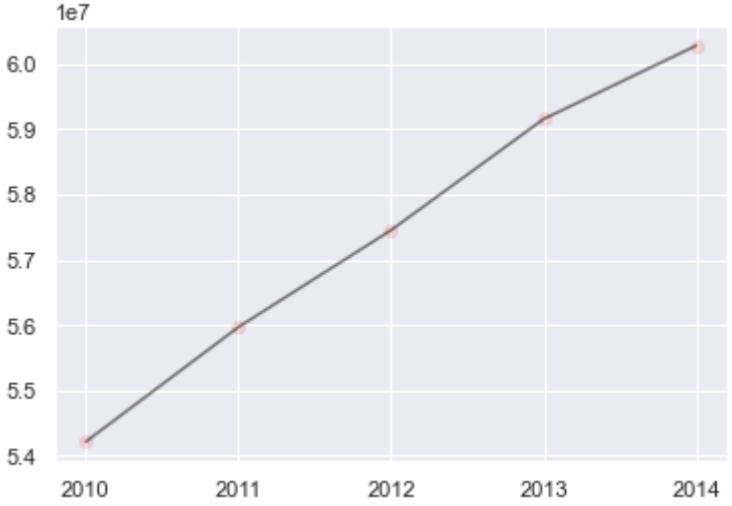

1e8
1.335
1.330
1.325
1.320
1.315
1.310
1.305
1.300
1.295
2010
2011
2012
2013
2014

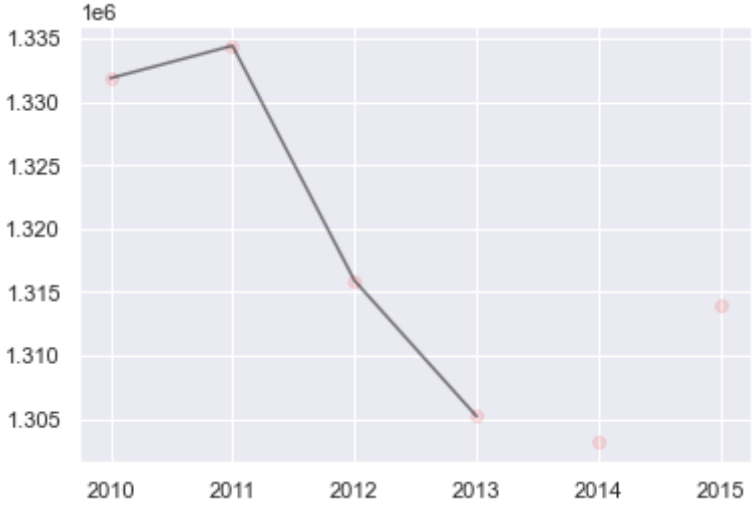

1e6
1.550
1.545
1.540
1.535
1.530
1.525
2010
2011
2012
2013
2014
2015

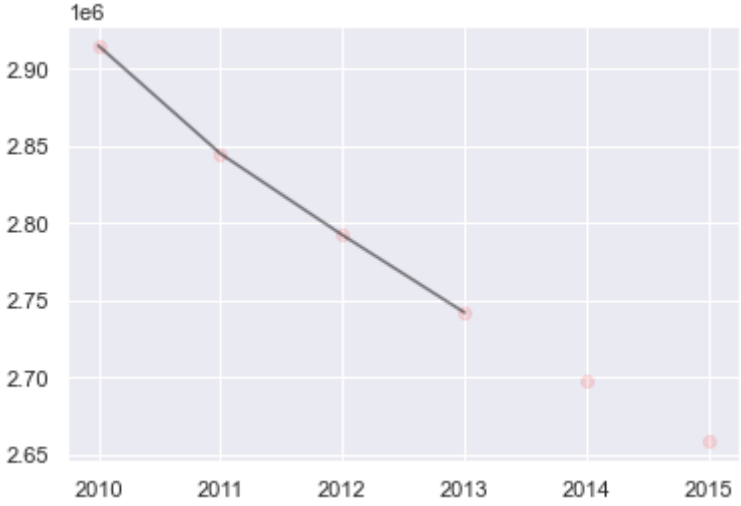

1e6
2.65
2.60
2.55
2.50
2.45
2.40
2010
2011
2012
2013
2014
2015


1625
1600
1575
1550
1525
1500
1475
1450
2010
2011
2012
2013
2014

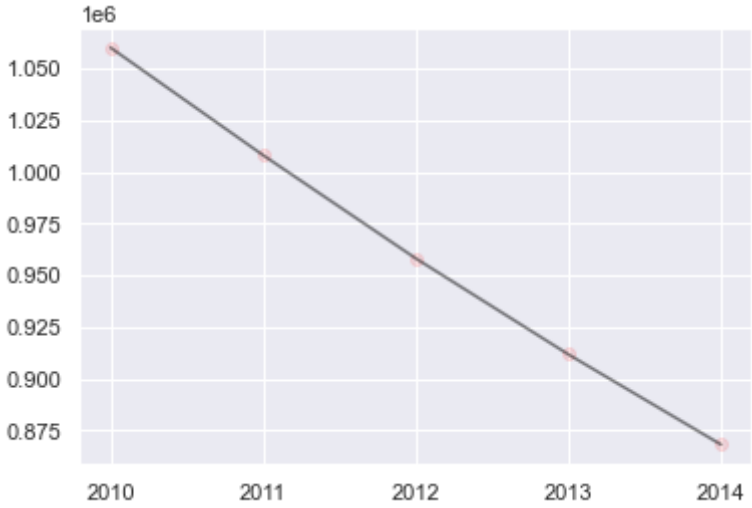
340000
330000
320000
310000
300000
290000
2010
2011
2012
2013
2014
2015

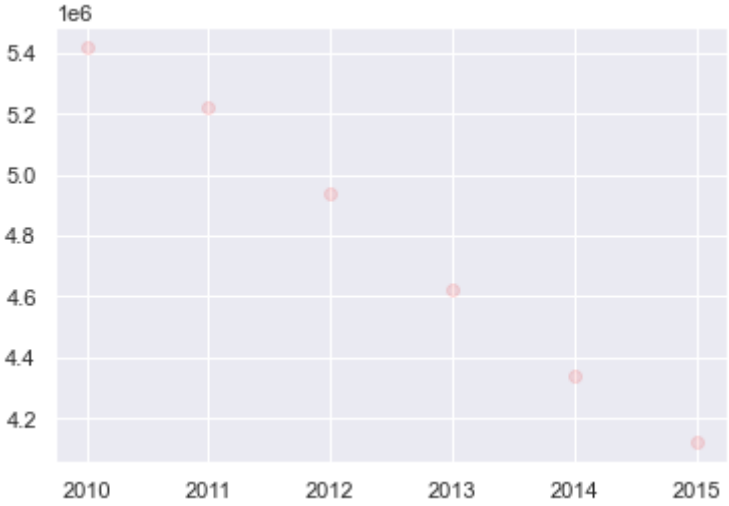

1e7
6.0
5.9
5.8
5.7
5.6
5.5
5.4
2010
2011
2012
2013
2014

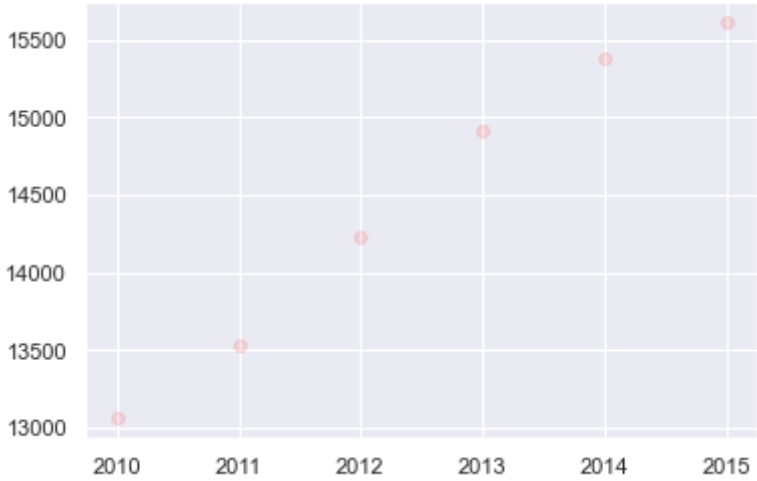

1e6
1.335
1.330
1.325
1.320
1.315
1.310
1.305
2010
2011
2012
2013
2014
2015

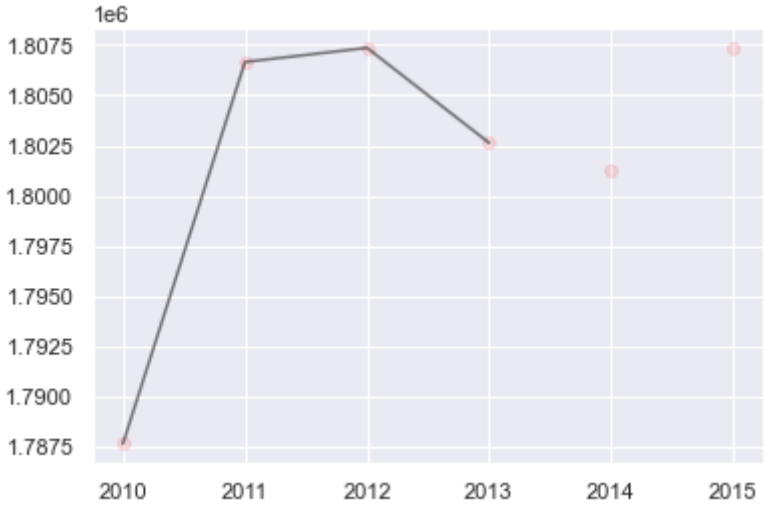

1e6
2.90
2.85
2.80
2.75
2.70
2.65
2010
2011
2012
2013
2014
2015

760000
740000
720000
700000
680000
2010
2011
2012
2013
2014
2015


1e6
1.050
1.025
1.000
0.975
0.950
0.925
0.900
0.875
2010
2011
2012
2013
2014


1e6
5.4
5.2
5.0
4.8
4.6
4.4
4.2
2010
2011
2012
2013
2014
2015


15500
15000
14500
14000
13500
13000
2010
2011
2012
2013
2014
2015



Résumé : SP.SEC.UTOT.IN, UIS.E.3, SE.SEC.ENRR.UP

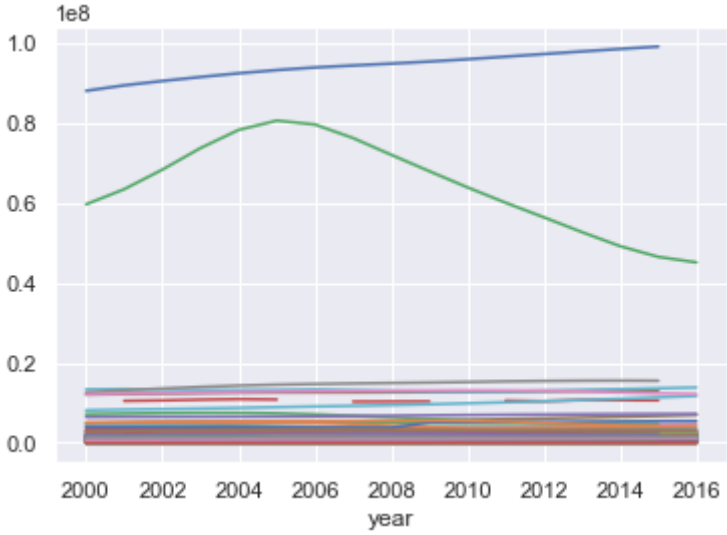
- SE.SEC.ENRR.UP = UIS.E.3 / SP.SEC.UTOT.IN

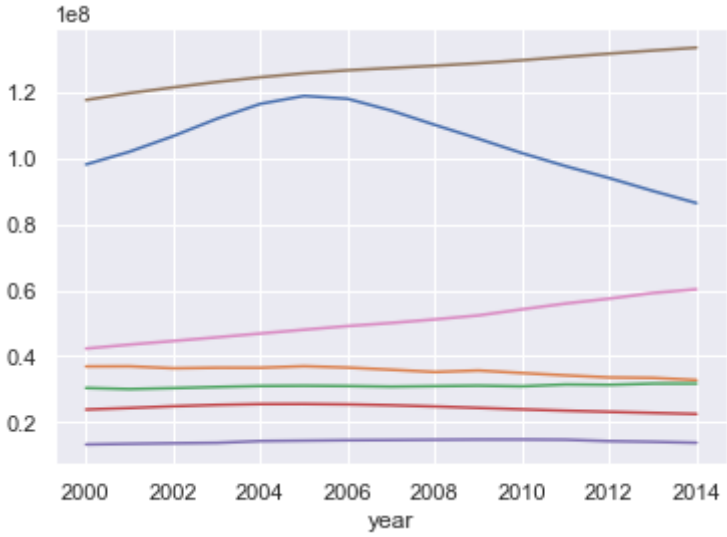
SP.SEC.UTOT.IN

In [83]:

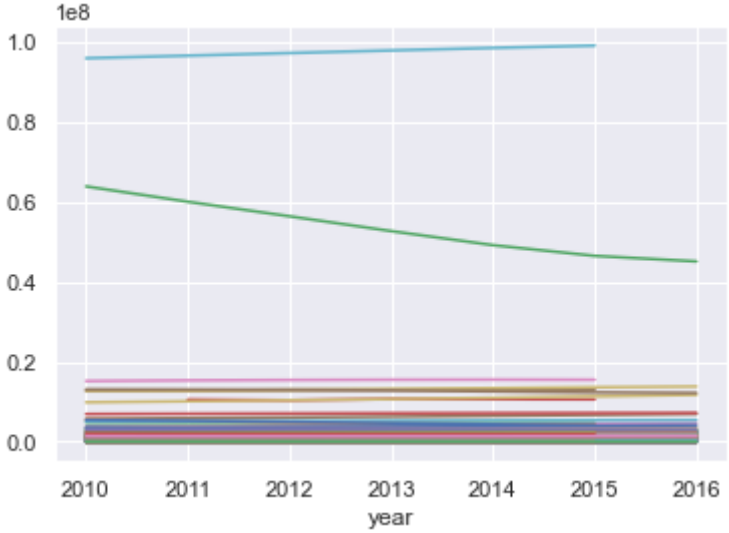
```
ind = 'SP.SEC.UTOT.IN'
initial_investigation(ind)[0].T.info()
```

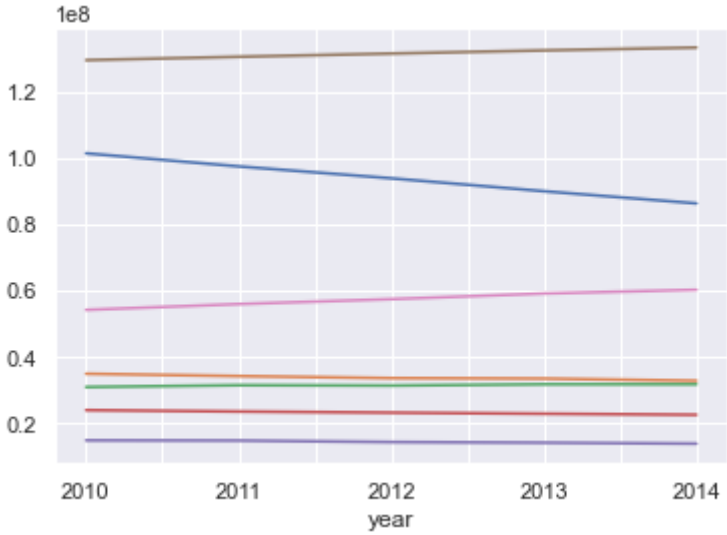
```
**************************************************
SP.SEC.UTOT.IN
```





```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 205 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI'
 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA'
 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD'
 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA'
 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN'
 'FJI' 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIB' 'GIN' 'GMB' 'GNB' 'GNQ'
 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL'
 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM'
 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA'
 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC'
 'NAM' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'OMN' 'PAK' 'PAN'
 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU'
 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB'
 'SSD' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO'
 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VGB' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB'
 'ZWE']

These countries have absolutely no data : {'CHI', 'MNP', 'CYM', 'PYF', 'SGP', 'GRL',
'MCO', 'FRO', 'VIR', 'ASM', 'NCL', 'IMN', 'TCA', 'XKX', 'SXM', 'GUM', 'MAF', 'AND',
'BIH'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SP.SEC.UTOT.IN
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 204 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE' 'BDI'
 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA'
 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD'
 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA'
 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN'
 'FJI' 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC'
 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN'
 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR'
 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO'
 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD'
 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'OMN' 'PAK' 'PAN' 'PER'
 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS'
 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSD'
 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA'
 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY'
 'USA' 'UZB' 'VCT' 'VEN' 'VGB' 'VNM' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'CHI', 'MNP', 'CYM', 'PYF', 'SGP', 'GRL',
'MCO', 'FRO', 'VIR', 'ASM', 'NCL', 'IMN', 'TCA', 'XKX', 'SXM', 'GUM', 'MAF', 'AND',
'BIH', 'GIB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.957983193277311

year           2010          2011          2012          2013          2014  \
count  2.020000e+02  2.030000e+02  2.030000e+02  2.030000e+02  2.040000e+02
mean   3.809156e+06  3.822181e+06  3.795956e+06  3.780282e+06  3.735734e+06
std    1.489671e+07  1.475462e+07  1.465482e+07  1.456892e+07  1.444357e+07
min    4.320000e+02  3.570000e+02  3.630000e+02  3.750000e+02  3.930000e+02
25%    8.389075e+04  8.490000e+04  8.405200e+04  8.104250e+04  7.675700e+04
50%    3.525685e+05  3.687200e+05  3.835140e+05  3.717900e+05  3.669760e+05
75%    1.387539e+06  1.463500e+06  1.461264e+06  1.474812e+06  1.467564e+06
max    1.295778e+08  1.305978e+08  1.315686e+08  1.325174e+08  1.333664e+08

year           2015          2016
count  1.950000e+02  1.580000e+02
mean   1.943047e+06  1.255435e+06
std    8.080003e+06  4.045436e+06
min    4.080000e+02  4.160000e+02
25%    7.299050e+04  5.009550e+04
50%    3.301490e+05  2.953205e+05
75%    1.296611e+06  1.073032e+06
max    9.919816e+07  4.514491e+07
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 204 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    202 non-null    float64
 1   2011    203 non-null    float64
 2   2012    203 non-null    float64
 3   2013    203 non-null    float64
 4   2014    204 non-null    float64
 5   2015    195 non-null    float64
 6   2016    158 non-null    float64
dtypes: float64(7)
memory usage: 12.8+ KB
```

Résumé: SP.SEC.UTOT.IN

- Pour 197 pays, il y a de la data pour les années 2010-2016.
  - 20 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 95% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Pour les 207 pays et régions, il y a de data pour l'année 2014.**
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [84]:

```
ind = 'SP.SEC.UTOT.IN'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data = drop_countries(data,7)
data.T[years].describe()
```

```
The following countries were dropped :
['AND', 'ASM', 'BIH', 'CHI', 'CYM', 'FRO', 'GIB', 'GRL', 'GUM', 'IMN', 'MAF', 'MCO',
'MNP', 'NCL', 'PYF', 'SGP', 'SXM', 'TCA', 'VIR', 'XKX']
```

Out[84]:

year	2010	2011	2012	2013	2014	2015	
count	2.020000e+02	2.030000e+02	2.030000e+02	2.030000e+02	2.040000e+02	1.950000e+02	1.580(
mean	3.809156e+06	3.822181e+06	3.795956e+06	3.780282e+06	3.735734e+06	1.943047e+06	1.255
std	1.489671e+07	1.475462e+07	1.465482e+07	1.456892e+07	1.444357e+07	8.080003e+06	4.045
min	4.320000e+02	3.570000e+02	3.630000e+02	3.750000e+02	3.930000e+02	4.080000e+02	4.160(
25%	8.389075e+04	8.490000e+04	8.405200e+04	8.104250e+04	7.675700e+04	7.299050e+04	5.009
50%	3.525685e+05	3.687200e+05	3.835140e+05	3.717900e+05	3.669760e+05	3.301490e+05	2.953
75%	1.387539e+06	1.463500e+06	1.461264e+06	1.474812e+06	1.467564e+06	1.296611e+06	1.073(
max	1.295778e+08	1.305978e+08	1.315686e+08	1.325174e+08	1.333664e+08	9.919816e+07	4.514

In [85]:

```
data_clean = fill_indicator_linear(data)
data_clean.T[years].describe()
```

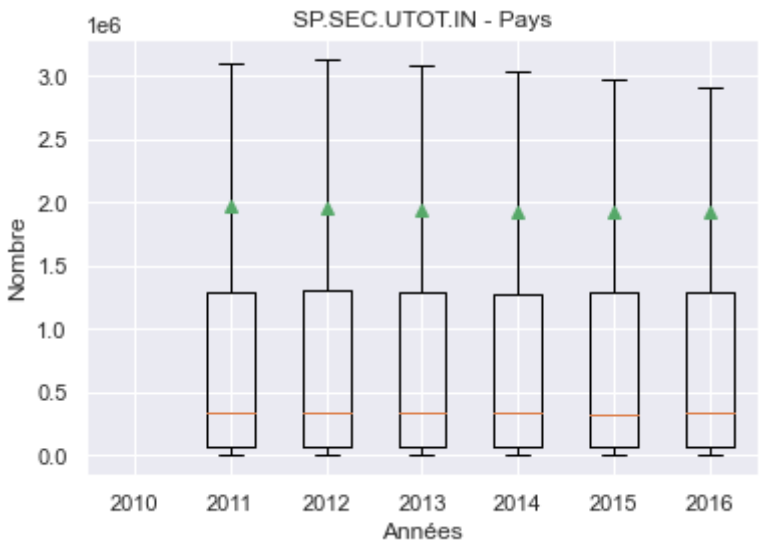
Out[85]:

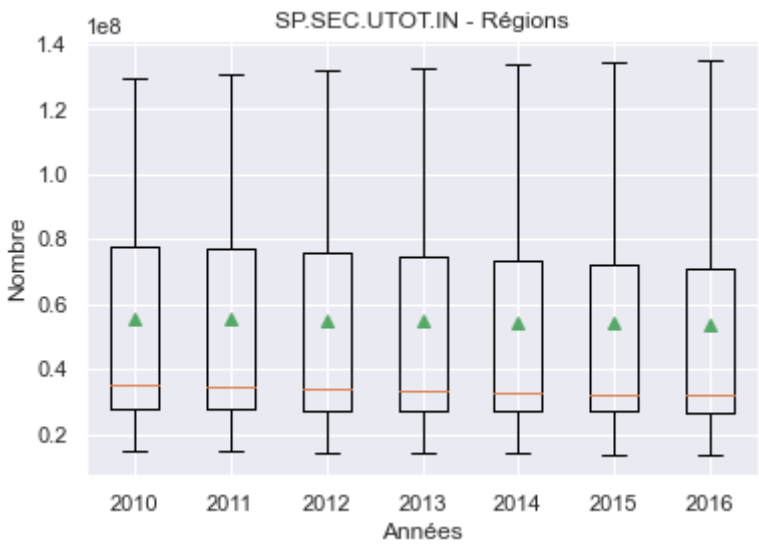
	2010	2011	2012	2013	2014	2015	
count	2.030000e+02	2.040000e+02	2.040000e+02	2.040000e+02	2.040000e+02	2.040000e+02	2.040(
mean	3.841700e+06	3.803642e+06	3.777574e+06	3.762005e+06	3.735734e+06	3.714450e+06	3.698

	2010	2011	2012	2013	2014	2015	
std	1.486703e+07	1.472061e+07	1.462104e+07	1.453534e+07	1.444357e+07	1.437181e+07	1.432
min	4.320000e+02	3.570000e+02	3.630000e+02	3.750000e+02	3.930000e+02	4.080000e+02	4.160
25%	8.398350e+04	8.193450e+04	7.998800e+04	7.849925e+04	7.675700e+04	7.696375e+04	7.671
50%	3.543780e+05	3.632660e+05	3.779025e+05	3.684380e+05	3.669760e+05	3.616355e+05	3.524
75%	1.414082e+06	1.451153e+06	1.450089e+06	1.444523e+06	1.467564e+06	1.489494e+06	1.520
max	1.295778e+08	1.305978e+08	1.315686e+08	1.325174e+08	1.333664e+08	1.342154e+08	1.350

In [86]:

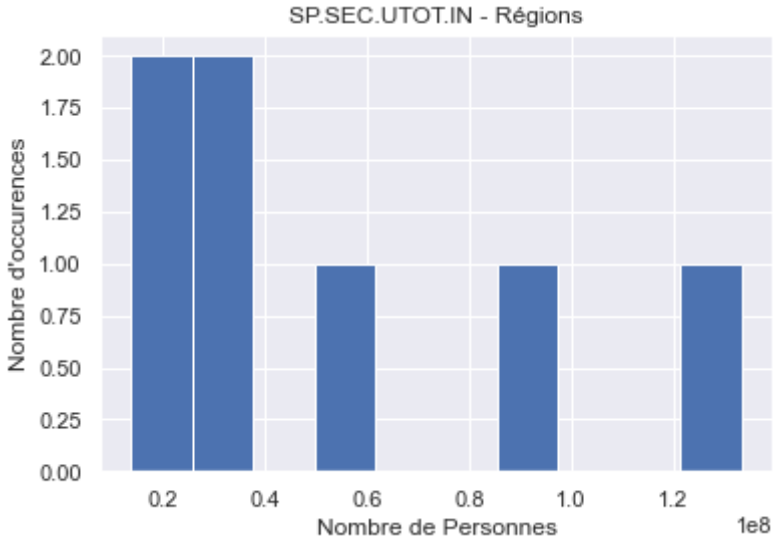
```
plot_boxplot(ind, data_clean, unit = "Nombre", years = years, showmeans = True, show
print_info_year('2014', data_clean, unit = "Nombre de Personnes", ind = ind)
print_info_year('2016', data_clean, unit = "Nombre de Personnes", ind = ind)
```



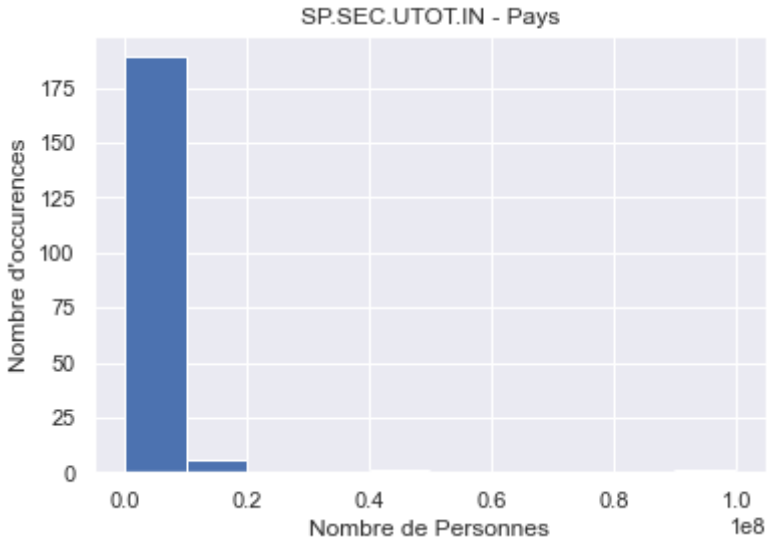



```
2014
Pays
Median:  337891.0
Mean:  1934550.3147208123
Std:  8079897.4907415295
```

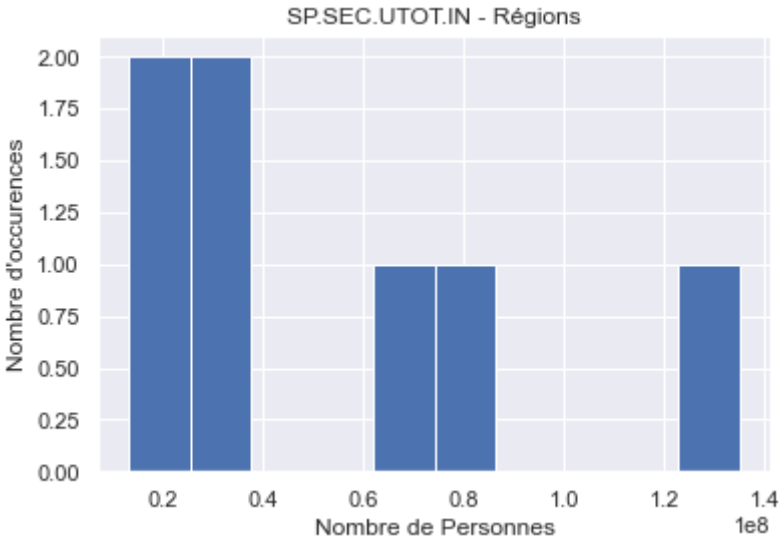
```
Régions
Median:  32818504.0
Mean:  54426201.428571425
Std:  42676279.99977718
```




```
2016
Pays
Median:  333591.0
Mean:  1925609.2335025382
Std:  8041586.734902801
```




```
Régions
Median:  31780436.0
Mean:  53581634.0
Std:  42730535.42920336
```



Résumé: SP.SEC.UTOT.IN

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 334 000, 1 930 000 et 8 040 000 personnes pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 31 800 000, 53 600 000 et 42 700 000 personnes pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne information sur la taille du marché - nombre absolute des potentielle clients.

In [87]:

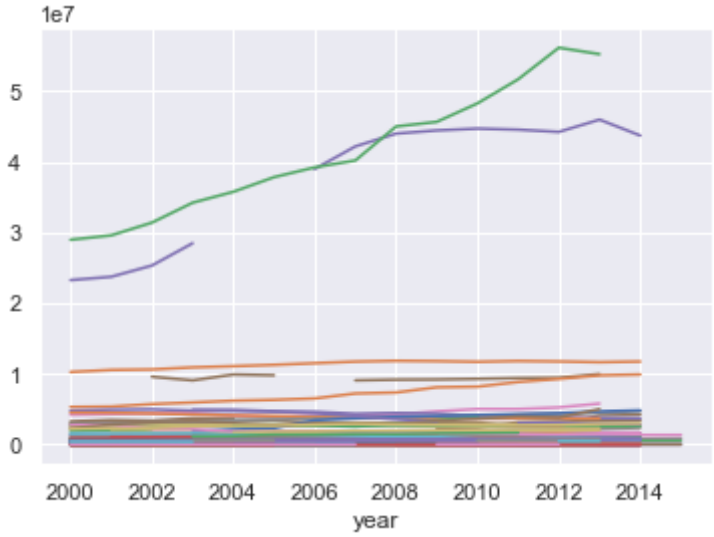
```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

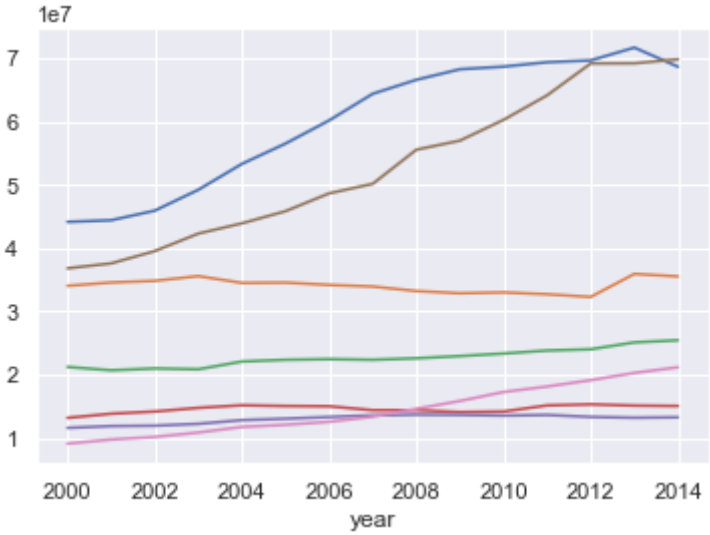
UIS.E.3

In [88]:

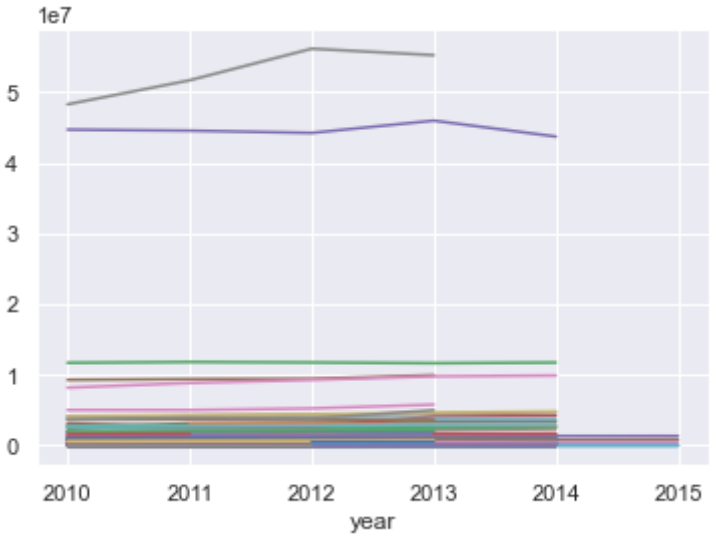
```
ind = 'UIS.E.3'
initial_investigation(ind)[0].T.info()
```

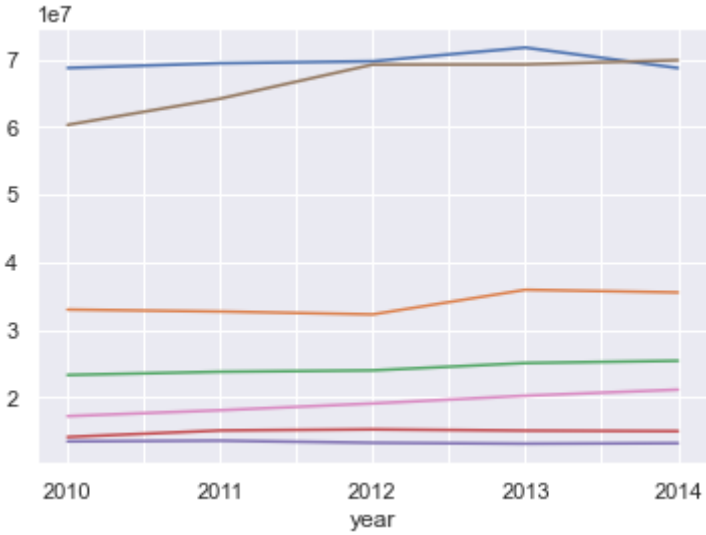
```
**************************************************
UIS.E.3
```






```
We have data between 2000 and 2015.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015']

We have data for the 207 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE'
 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BMU'
 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV'
 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYM' 'CYP' 'CZE'
 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP'
 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIB' 'GIN'
 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN'
 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN'
 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN'
 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO' 'MDA' 'MDG' 'MDV'
 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS'
 'MWI' 'MYS' 'NAC' 'NAM' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL'
 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY'
 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SGP' 'SLB' 'SLE'
 'SLV' 'SMR' 'SOM' 'SRB' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SXM'
 'SYC' 'SYR' 'TCA' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN'
 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VUT' 'WSM'
 'YEM' 'ZAF' 'ZWE']

These countries have absolutely no data : {'GRL', 'HTI', 'VGB', 'VNM', 'CHI', 'XKX',
'FRO', 'MNP', 'VIR', 'ASM', 'GUM', 'MAF', 'ZMB', 'PYF', 'SSD', 'NCL', 'IMN'}
These regions have absolutely no data : set()
**************************************************
**************************************************
UIS.E.3
```






```
We have data between 2010 and 2015.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015']

We have data for the 188 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'AZE'
 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BMU'
 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV'
 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYM' 'CYP' 'CZE'
 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP'
 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GRC' 'GRD'
 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN'
 'LBR' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO'
 'MDA' 'MDG' 'MEA' 'MEX' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT'
 'MUS' 'MWI' 'MYS' 'NAC' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NRU' 'NZL'
 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRT' 'PRY' 'PSE'
 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SLB' 'SLE' 'SLV' 'SMR' 'SRB'
 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SXM' 'SYC' 'SYR' 'TCA' 'TCD'
 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TUN' 'TUR' 'TUV' 'TZA' 'UKR' 'URY' 'USA'
 'UZB' 'VCT' 'VEN' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZWE']

These countries have absolutely no data : {'IRQ', 'CHI', 'LBY', 'MHL', 'GNQ', 'MNP',
'KHM', 'PYF', 'GNB', 'TON', 'PRK', 'TTO', 'SSD', 'SGP', 'GRL', 'HTI', 'SOM', 'FRO',
'VIR', 'FSM', 'ASM', 'NCL', 'KIR', 'IMN', 'VGB', 'VNM', 'XKX', 'UGA', 'MDV', 'GUM',
'MAF', 'ZMB', 'SEN', 'NAM', 'GIB', 'GAB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2015: 0.6693262411347518

year           2010          2011          2012          2013          2014  \
count  1.560000e+02  1.610000e+02  1.590000e+02  1.470000e+02  1.250000e+02
mean   2.845537e+06  2.837781e+06  2.936010e+06  3.277591e+06  3.047275e+06
std    9.581047e+06  9.741906e+06  1.015763e+07  1.073568e+07  1.043027e+07
min    1.221000e+03  1.259000e+03  2.540000e+02  1.150000e+03  3.530000e+02
25%    6.487025e+04  6.895500e+04  6.193450e+04  9.068100e+04  5.081500e+04
50%    2.492330e+05  2.443700e+05  2.415230e+05  2.835860e+05  2.312500e+05
75%    9.911418e+05  9.799870e+05  9.244300e+05  1.258817e+06  8.043760e+05
max    6.869628e+07  6.940019e+07  6.968986e+07  7.171130e+07  6.985116e+07

year           2015
count  7.000000e+00
mean   3.883340e+05
std    5.298411e+05
min    8.000000e+02
25%    1.449950e+04
50%    2.936400e+04
75%    6.590840e+05
max    1.341007e+06
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 188 entries, ABW to ZWE
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    156 non-null    float64
 1   2011    161 non-null    float64
 2   2012    159 non-null    float64
 3   2013    147 non-null    float64
 4   2014    125 non-null    float64
 5   2015    7 non-null      float64
dtypes: float64(6)
memory usage: 10.3+ KB
```

Résumé: UIS.E.3

- Pour 181 pays, il y a de la data pour les années 2010-2016.
  - 36 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 67% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Il n'y a pas de data pour l'année 2016.**
- **Il n'y a presque pas de data pour l'année 2015.**
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [89]:

```
ind = 'UIS.E.3'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data2 = drop_countries(data.loc[years,:],7)
print(len(data2.columns.values))
data.T[years].describe()
```

```
The following countries were dropped :
['ASM', 'CHI', 'FRO', 'FSM', 'GAB', 'GIB', 'GNB', 'GNQ', 'GRL', 'GUM', 'HTI', 'IMN',
'IRQ', 'KHM', 'KIR', 'LBY', 'MAF', 'MDV', 'MHL', 'MNP', 'NAM', 'NCL', 'PRK', 'PYF',
'SEN', 'SGP', 'SOM', 'SSD', 'TON', 'TTO', 'UGA', 'VGB', 'VIR', 'VNM', 'XKX', 'ZMB']
188
```

Out[89]:

year	2010	2011	2012	2013	2014	2015	2016
count	1.560000e+02	1.610000e+02	1.590000e+02	1.470000e+02	1.250000e+02	7.000000e+00	0.0
mean	2.845537e+06	2.837781e+06	2.936010e+06	3.277591e+06	3.047275e+06	3.883340e+05	NaN
std	9.581047e+06	9.741906e+06	1.015763e+07	1.073568e+07	1.043027e+07	5.298411e+05	NaN
min	1.221000e+03	1.259000e+03	2.540000e+02	1.150000e+03	3.530000e+02	8.000000e+02	NaN
25%	6.487025e+04	6.895500e+04	6.193450e+04	9.068100e+04	5.081500e+04	1.449950e+04	NaN
50%	2.492330e+05	2.443700e+05	2.415230e+05	2.835860e+05	2.312500e+05	2.936400e+04	NaN
75%	9.911418e+05	9.799870e+05	9.244300e+05	1.258817e+06	8.043760e+05	6.590840e+05	NaN
max	6.869628e+07	6.940019e+07	6.968986e+07	7.171130e+07	6.985116e+07	1.341007e+06	NaN

In [90]:

```
data_clean = fill_indicator_linear(data[data2.columns.values])
data_clean.T.describe()
```

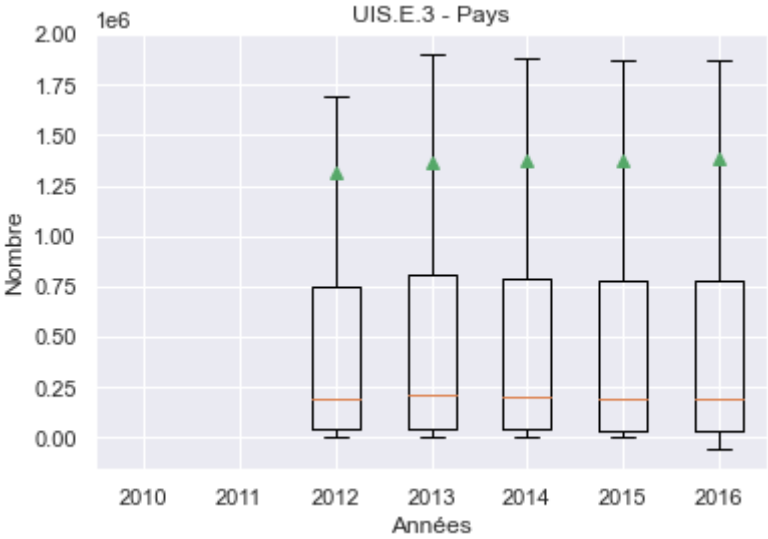
Out[90]:

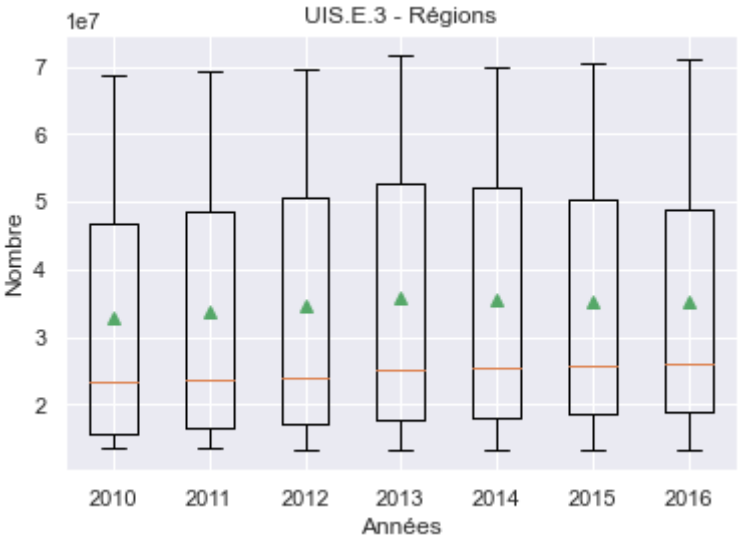
	2000	2001	2002	2003	2004	2005

	2000	2001	2002	2003	2004	2005	
count	1.480000e+02	1.640000e+02	1.710000e+02	1.760000e+02	1.760000e+02	1.780000e+02	1.800(
mean	2.096641e+06	1.957804e+06	1.992274e+06	2.066772e+06	2.147531e+06	2.199765e+06	2.250(
std	6.557442e+06	6.339898e+06	6.474601e+06	6.789779e+06	7.143830e+06	7.449501e+06	7.779(
min	2.820000e+02	3.800000e+01	3.500000e+01	6.900000e+01	1.030000e+02	1.370000e+02	1.710(
25%	3.922900e+04	3.126900e+04	2.939800e+04	2.916412e+04	3.099250e+04	3.219425e+04	3.361(
50%	1.632185e+05	1.471025e+05	1.441510e+05	1.505450e+05	1.602905e+05	1.704101e+05	1.803(
75%	7.590758e+05	6.527372e+05	6.296465e+05	7.039428e+05	6.561965e+05	6.749932e+05	6.810(
max	4.415272e+07	4.440643e+07	4.592420e+07	4.922797e+07	5.332024e+07	5.653658e+07	6.019(

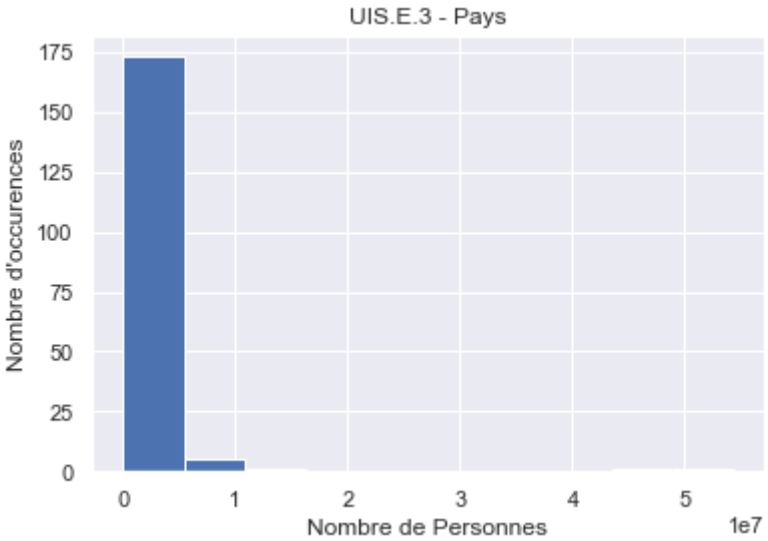
In [91]:

```
plot_boxplot(ind, data_clean, unit = "Nombre", years = years, showmeans = True, show
print_info_year('2014', data_clean, unit = "Nombre de Personnes", ind = ind)
print_info_year('2016', data_clean, unit = "Nombre de Personnes", ind = ind)
```

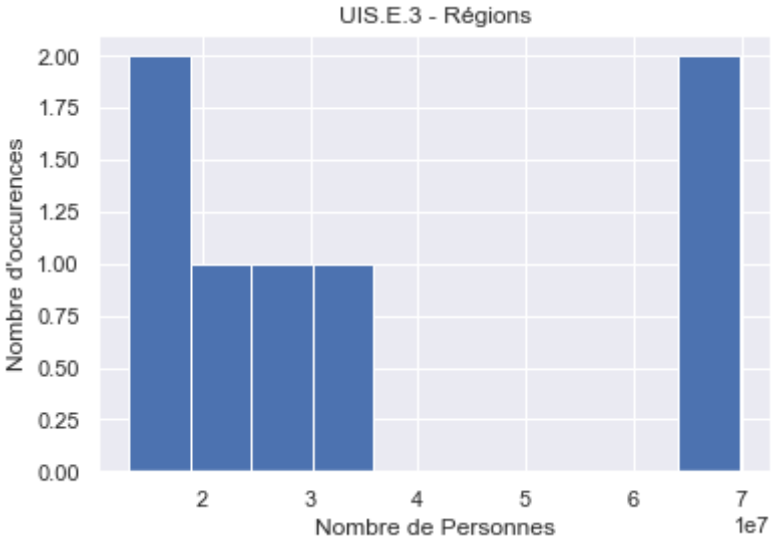




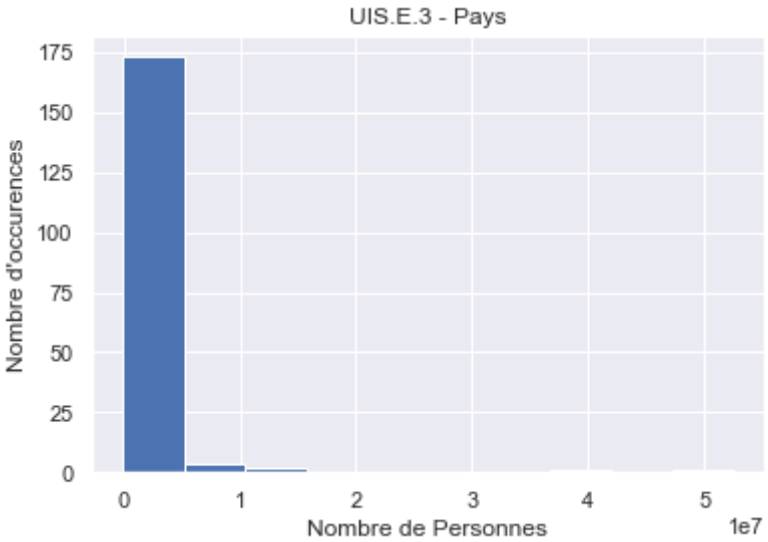

```
2014
Pays
Median:  200900.0
Mean:  1370822.920733579
Std:  5371576.907691944
```



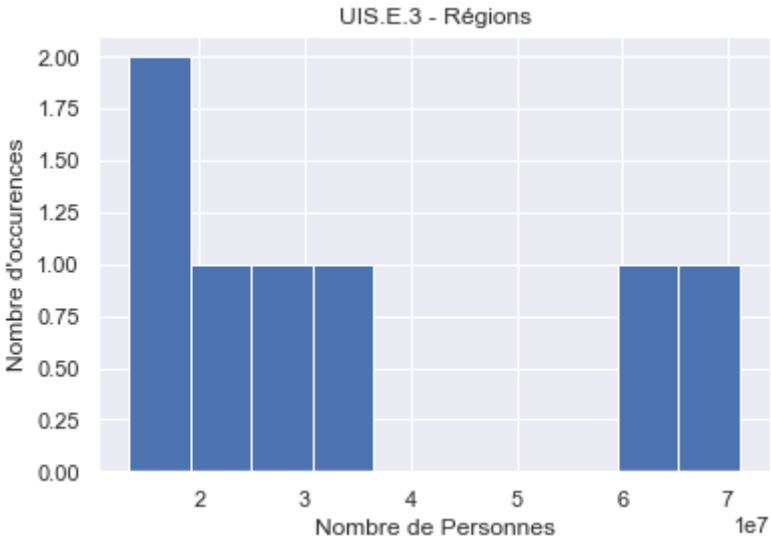

```
Régions
Median:  25425662.0
Mean:  35561217.14285714
Std:  24162802.240447517
```




```
2016
Pays
Median:  193102.25
Mean:  1383772.9305540798
Std:  5128756.465520181
```




```
Régions
Median:  26097022.0
Mean:  35115825.71428572
Std:  22964096.589051504
```



Résumé: UIS.E.3

- **Pour l'année 2014 la médian, le moyenne et l'écart-type sont 201 000, 1 370 000 et 5 370 000 personnes pour les pays**
- **Pour l'année 2014 la médian, le moyenne et l'écart-type sont 25 400 000, 35 600 000 et 24 200 000 personnes pour les régions**
- **Pour l'année 2016 la médian, le moyenne et l'écart-type sont 193 000, 1 380 000 et 5 130 000 personnes pour les pays**
- **Pour l'année 2016 la médian, le moyenne et l'écart-type sont 26 100 000, 35 100 000 et 23 000 000 personnes pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne information sur la taille du marché - nombre absolute des potentielle clients.

In [92]:

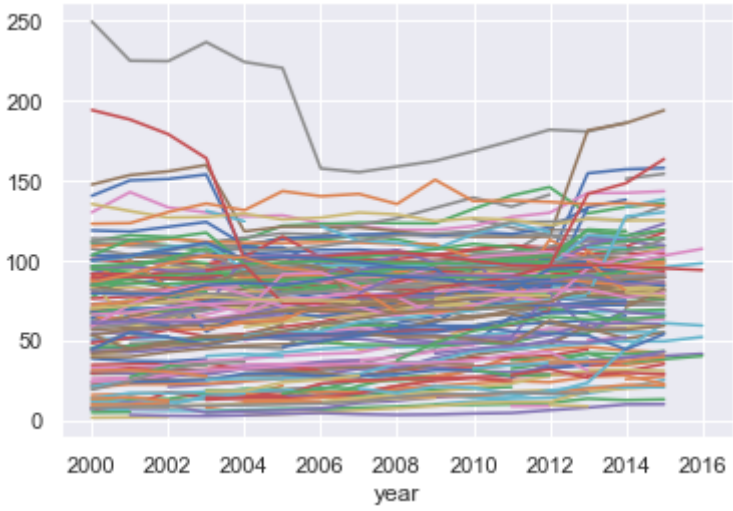
```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

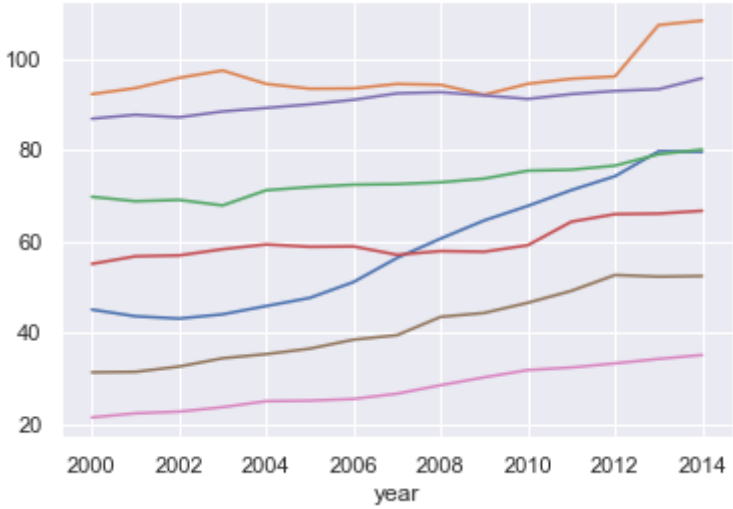
SE.SEC.ENRR.UP

In [93]:

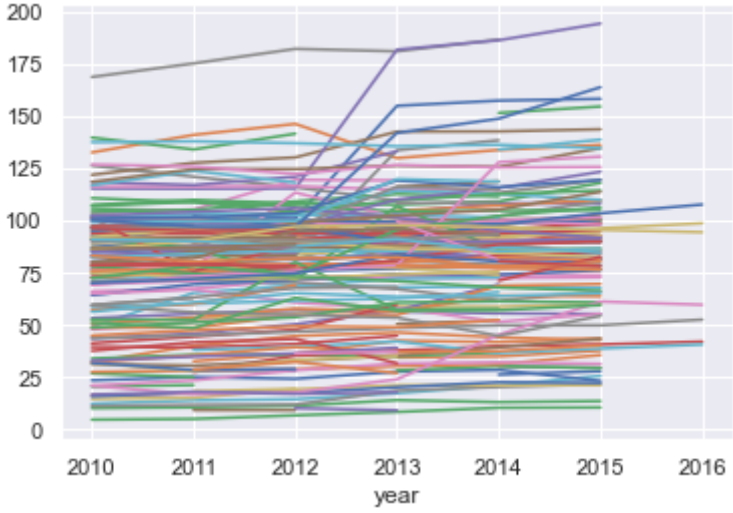
```
ind = 'SE.SEC.ENRR.UP'
initial_investigation(ind)[0].T.info()
```

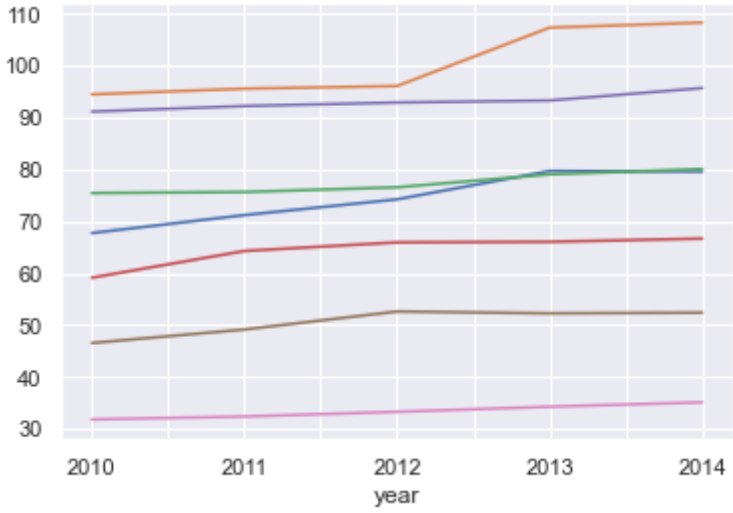
```
**************************************************
SE.SEC.ENRR.UP
```






```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 199 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'BDI' 'BEL' 'BEN'
 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA' 'BRB' 'BRN'
 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL'
 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM'
 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA'
 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GTM'
 'GUY' 'HKG' 'HND' 'HRV' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT'
 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA'
 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR'
 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NER' 'NGA' 'NIC'
 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG'
 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU'
 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSD' 'SSF' 'STP' 'SUR'
 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS'
 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT'
 'VEN' 'VGB' 'VUT' 'WSM' 'YEM' 'ZAF' 'ZWE']

These countries have absolutely no data : {'CHI', 'MNP', 'CYM', 'PYF', 'SGP', 'ARE',
'GRL', 'HTI', 'MCO', 'FRO', 'VIR', 'ASM', 'AZE', 'NCL', 'TCA', 'IMN', 'VNM', 'XKX',
'SXM', 'GUM', 'MAF', 'AND', 'ZMB', 'BIH', 'GIB'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SE.SEC.ENRR.UP
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 184 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'ARG' 'ARM' 'ATG' 'AUS' 'AUT' 'BDI' 'BEL' 'BEN'
 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BLR' 'BLZ' 'BMU' 'BOL' 'BRA' 'BRB' 'BRN'
 'BTN' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM'
 'CPV' 'CRI' 'CUB' 'CUW' 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA'
 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'GBR'
 'GEO' 'GHA' 'GIN' 'GMB' 'GRC' 'GRD' 'GTM' 'GUY' 'HKG' 'HND' 'HRV' 'HUN'
 'IDN' 'IND' 'IRL' 'IRN' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KGZ'
 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU'
 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT'
 'MMR' 'MNE' 'MNG' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NER' 'NGA' 'NIC'
 'NLD' 'NOR' 'NPL' 'NRU' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG'
 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU'
 'SDN' 'SEN' 'SLB' 'SLE' 'SLV' 'SMR' 'SRB' 'SSD' 'SSF' 'STP' 'SUR' 'SVK'
 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON'
 'TUN' 'TUR' 'TUV' 'TZA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VGB' 'VUT'
 'WSM' 'YEM' 'ZAF' 'ZWE']

These countries have absolutely no data : {'BWA', 'IRQ', 'CHI', 'LBY', 'GNQ', 'MNP',
'CYM', 'KHM', 'PYF', 'GNB', 'TTO', 'SGP', 'ARE', 'GRL', 'HTI', 'MCO', 'SOM', 'FRO',
'VIR', 'FSM', 'ASM', 'AZE', 'KEN', 'NCL', 'KIR', 'TCA', 'IMN', 'VNM', 'XKX', 'UGA',
'SXM', 'MDV', 'GUM', 'MAF', 'AND', 'ZMB', 'BIH', 'NAM', 'GIB', 'GAB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.6762422360248447

year          2010        2011        2012        2013        2014        2015  \
count  152.000000  157.000000  153.000000  144.000000  142.000000  115.000000
```

```
mean    73.684680   72.513417   74.378254   79.556849   81.937742   80.488933
std     31.045156   32.364155   32.325351   33.665308   34.069408   35.725014
min      4.539990    4.614590    6.595380    8.227430   10.214850    5.020860
25%     50.799011   48.371780   52.597542   56.605623   57.734553   57.270521
50%     78.434319   78.178322   78.891022   83.800797   85.145084   84.507851
75%     94.822105   94.961700   95.746696   99.155041  100.225294   99.781548
max    168.520767  175.029724  181.990967  181.605743  186.427887  194.101990

year         2016
count    8.000000
mean    74.943194
std     29.076538
min     40.489799
25%     49.768929
50%     76.880419
75%    100.143446
max    107.495377
<class 'pandas.core.frame.DataFrame'>
Index: 184 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    152 non-null    float64
 1   2011    157 non-null    float64
 2   2012    153 non-null    float64
 3   2013    144 non-null    float64
 4   2014    142 non-null    float64
 5   2015    115 non-null    float64
 6   2016    8 non-null      float64
dtypes: float64(7)
memory usage: 11.5+ KB
```

Résumé: SE.SEC.ENRR.UP

- Pour 177 pays, il y a de la data pour les années 2010-2016.
  - 40 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 68% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Le plus complète année entre 2010-2016 est 2011.**
- **Presque pas de data pour 2016**
- Voir si on peut remplir cette variable avec UIS.E.3 and SP.SEC.UTOT.IN
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [94]:

```
ind = 'SE.SEC.ENRR.UP'
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind

ind2 = 'UIS.E.3'
ind3 = 'SP.SEC.UTOT.IN'
data2 = make_2D_map(DATA_CLEANED, ind2, ['indicator', 'year'], 'country').drop(colum
data3 = make_2D_map(DATA_CLEANED, ind3, ['indicator', 'year'], 'country').drop(colum
data4 = data2.div(data3).multiply(100)
```

In [95]:

```
print(data.T.isna().sum())
print(data4.T.isna().sum())
```

```
year
2000     77
2001     73
2002     71
2003     75
2004     65
```

```
2005     64
2006     72
2007     67
2008     73
2009     72
2010     72
2011     67
2012     71
2013     80
2014     82
2015    109
2016    216
dtype: int64
year
2000    73
2001    56
2002    50
2003    47
2004    45
2005    44
2006    43
2007    44
2008    44
2009    45
2010    44
2011    43
2012    42
2013    42
2014    42
2015    42
2016    42
dtype: int64
```

In [96]:

```
#fill holes will cleand data from the previous two indicators
years = [str(y) for y in range(2010, 2017)]
data_clean = data.fillna(value = data4)
data_clean.T[years].describe()
```

Out[96]:

year	2010	2011	2012	2013	2014	2015	2016
count	181.000000	183.000000	183.000000	183.000000	184.000000	187.000000	182.000000
mean	70.697760	71.639824	73.320119	76.803752	77.657950	79.160216	80.961761
std	31.406552	31.598845	31.606309	34.502389	35.160904	37.215797	41.760949
min	4.539990	4.614590	6.595380	8.227430	5.893076	5.020860	-9.463003
25%	47.085247	47.263718	49.352152	52.310698	51.859070	51.714381	50.283073
50%	75.415329	76.223846	77.882736	80.714943	80.976017	82.024597	82.696126
75%	93.307961	92.954090	94.812717	98.519215	98.118362	99.983200	104.742004
max	168.520767	175.029724	181.990967	181.605743	186.427887	194.101990	207.824575

In [97]:

```
data2 = drop_countries(data_clean.loc[years,:],7)
data_clean[data2.columns.values].T[years].describe()
```

```
The following countries were dropped :
['AND', 'ASM', 'BIH', 'CHI', 'CYM', 'FRO', 'FSM', 'GAB', 'GIB', 'GNB', 'GNQ', 'GRL',
'GUM', 'HTI', 'IMN', 'IRQ', 'KHM', 'KIR', 'LBY', 'MAF', 'MCO', 'MDV', 'MNP', 'NAM',
'NCL', 'PYF', 'SGP', 'SOM', 'SXM', 'TCA', 'TTO', 'UGA', 'VIR', 'VNM', 'XKX', 'ZMB']
```

Out[97]:

year	2010	2011	2012	2013	2014	2015	2016
count	181.000000	183.000000	183.000000	183.000000	184.000000	187.000000	182.000000

year	2010	2011	2012	2013	2014	2015	2016
mean	70.697760	71.639824	73.320119	76.803752	77.657950	79.160216	80.961761
std	31.406552	31.598845	31.606309	34.502389	35.160904	37.215797	41.760949
min	4.539990	4.614590	6.595380	8.227430	5.893076	5.020860	-9.463003
25%	47.085247	47.263718	49.352152	52.310698	51.859070	51.714381	50.283073
50%	75.415329	76.223846	77.882736	80.714943	80.976017	82.024597	82.696126
75%	93.307961	92.954090	94.812717	98.519215	98.118362	99.983200	104.742004
max	168.520767	175.029724	181.990967	181.605743	186.427887	194.101990	207.824575

In [98]:

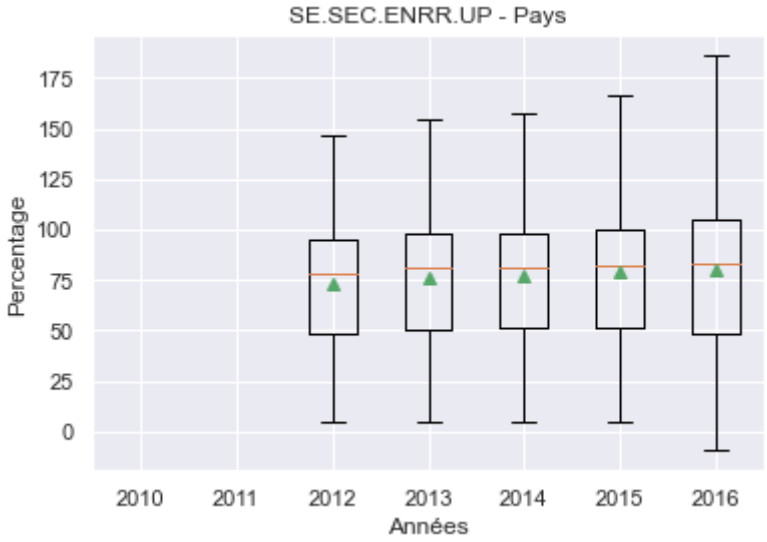
```
data_clean = fill_indicator_linear(data_clean[data2.columns.values])
data_clean.T[years].describe()
```

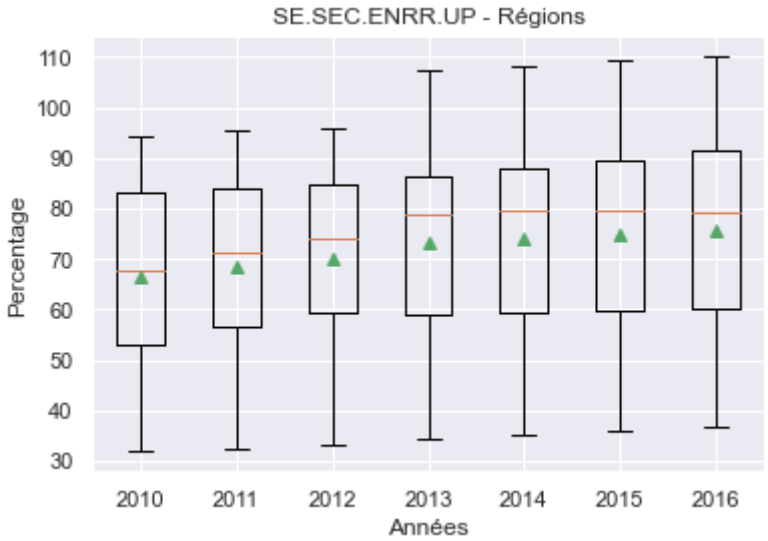
Out[98]:

	2010	2011	2012	2013	2014	2015	2016
count	185.000000	187.000000	188.000000	188.000000	188.000000	188.000000	188.000000
mean	70.720909	71.602066	72.841603	76.192835	77.101359	79.012615	80.130448
std	31.386441	31.531633	31.819996	34.657459	35.360100	37.171290	41.670793
min	4.539990	4.614590	4.716158	4.817725	4.919293	5.020860	-9.463003
25%	47.085247	47.263718	49.157682	51.771240	51.436852	51.288172	48.849576
50%	75.415329	76.223846	77.773572	80.192394	80.626278	81.995463	82.273093
75%	93.307961	92.954090	94.737085	97.814043	97.752191	99.963299	103.494818
max	168.520767	175.029724	181.990967	181.605743	186.427887	194.101990	207.824575

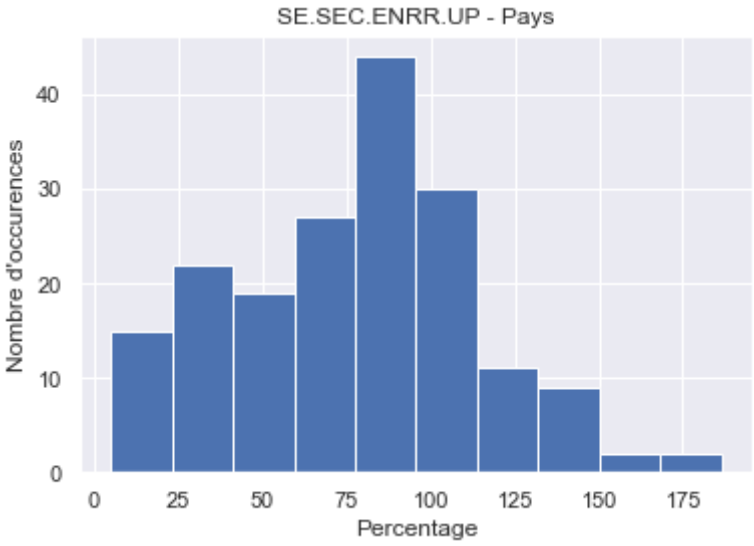
In [99]:

```
plot_boxplot(ind, data_clean, unit = "Percentage", years = years, showmeans = True,
print_info_year('2014', data_clean, unit = "Percentage", ind = ind)
print_info_year('2016', data_clean, unit = "Percentage", ind = ind)
```

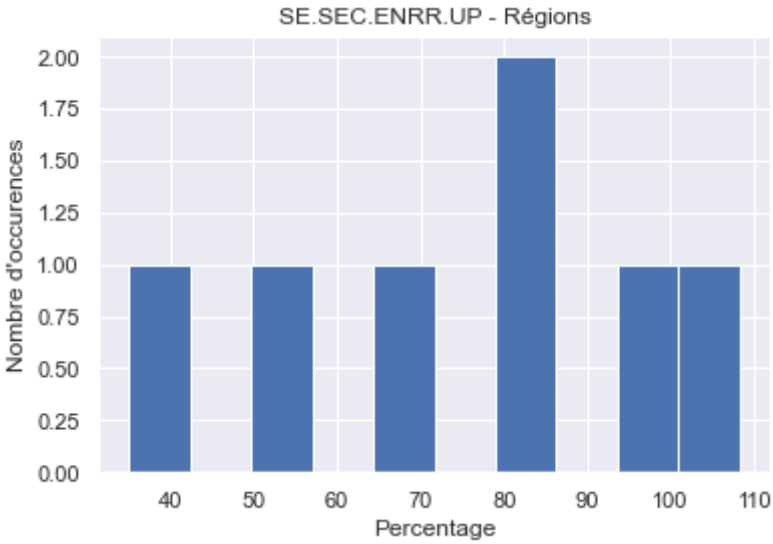





```
2014
Pays
Median:  81.0454864501953
Mean:  77.22351946980481
Std:  35.74577663696868
```

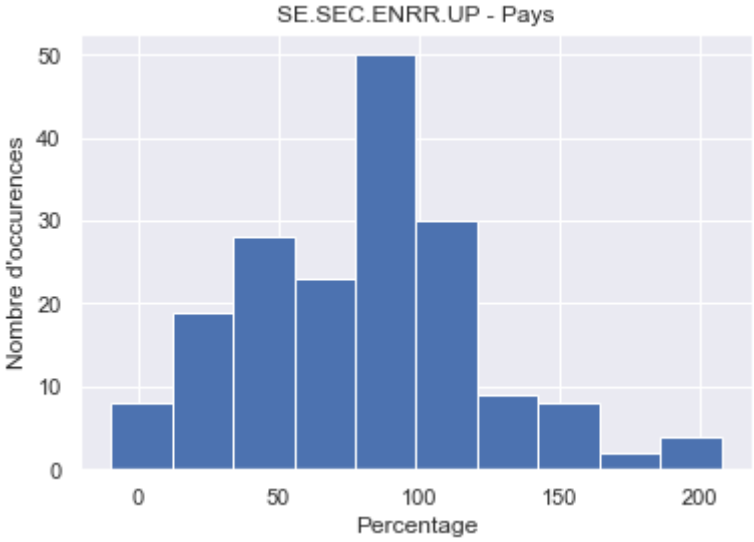



```
Régions
Median:  79.5493469238281
Mean:  73.94263512747628
Std:  24.97835649005094
```

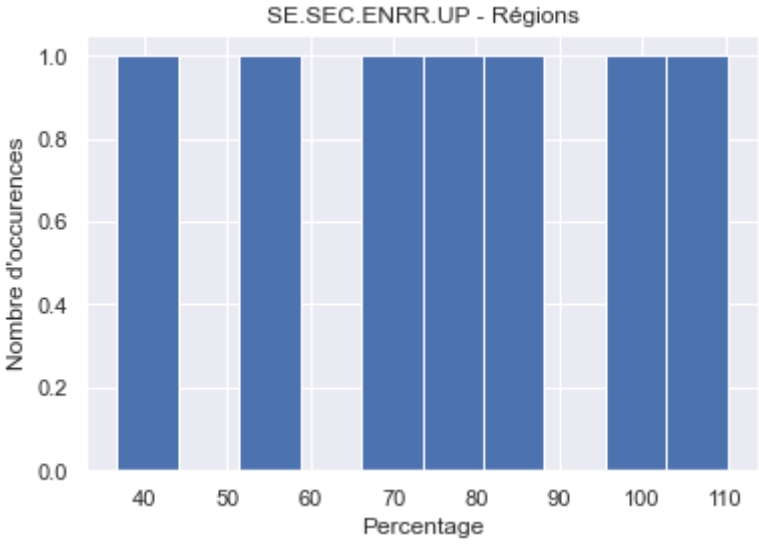



```
2016
Pays
Median:  82.7377853393554
```

```
Mean:  80.30267777674578
Std:  42.2020694524826
```




```
Régions
Median:  79.3032215416714
Mean:  75.67708570678174
Std:  25.788378318004487
```



Résumé: SE.SEC.ENRR.UP

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 82.7%, 80.3% et 42.2% pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 79.3%, 75.7% et 25.8% pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne une indication sur le niveau d'ecducation du pays / la régions.

In [100…

```
#See which countries have a rate above 100%
print(data_clean.loc[:, data_clean.loc['2016',:] >= 100])
for c in data_clean.loc[:, data_clean.loc['2016',:] >= 100].columns.values:
    print(c, RAW_DATA[RAW_DATA['Country Code'] == c]['Country Name'].iloc[0])
```

```
country         ABW         AUS         AZE         BEL         BGR  \
2000      82.165443  249.607498         NaN  140.671967  101.134552
2001      85.361458  225.125687         NaN  150.290283  102.303978
2002      88.880478  224.854965         NaN  151.160828  104.499741
2003      89.847588  236.741791         NaN  153.979172   84.427452
2004      86.599899  224.360306         NaN  104.403603   89.033051
```

2005	84.785622	220.559540	NaN	105.426781	90.763153
2006	86.693459	157.797073	107.761619	106.723442	92.481628
2007	98.751671	155.434784	108.805957	107.022377	93.818207
2008	89.368263	158.950897	111.546248	105.188988	93.479012
2009	86.303452	162.460724	113.421871	103.545998	94.448219
2010	84.642014	168.520767	113.364468	102.225456	96.561630
2011	86.417068	175.029724	115.213142	101.621918	101.196510
2012	106.325500	181.990967	118.697327	100.904678	98.992943
2013	126.120556	180.813995	126.986498	154.740646	103.600769
2014	145.964523	186.427887	135.031979	157.230682	108.145897
2015	165.939448	192.612589	137.699778	158.077530	107.801353
2016	186.450168	196.460693	140.432680	160.608750	115.948435

country	BHR	BHS	BLR	BRA	BRB	... \
2000	87.634850	NaN	NaN	NaN	110.742279	...
2001	88.187851	NaN	NaN	NaN	111.477638	...
2002	82.997551	94.300056	NaN	90.600227	109.628609	...
2003	84.318573	88.729272	NaN	84.298866	106.211182	...
2004	86.701492	82.517014	NaN	90.831230	108.137154	...
2005	89.166672	80.423752	NaN	90.629448	114.835243	...
2006	91.561272	82.243347	NaN	89.539260	112.986679	...
2007	85.802635	86.082947	NaN	88.391533	115.747726	...
2008	84.187218	86.532951	NaN	89.121162	113.290733	...
2009	87.006508	87.633408	NaN	89.306290	108.710709	...
2010	90.681213	87.462700	126.082542	89.135489	105.503761	...
2011	95.825638	88.000833	120.778152	89.238762	104.526428	...
2012	90.148972	90.193392	115.749290	90.972198	107.653602	...
2013	95.656937	92.704598	109.357552	93.230927	110.605858	...
2014	97.559898	96.095635	113.157600	93.478493	112.885910	...
2015	102.816643	100.841866	119.316940	91.420326	114.427195	...
2016	105.608421	106.884425	110.000669	108.630677	115.573880	...

country	POL	PRT	PRY	QAT	RUS \
2000	100.063850	96.723709	48.533901	84.787613	NaN
2001	102.547791	98.264183	51.952831	79.748894	NaN
2002	108.720543	100.672119	57.004070	79.804718	NaN
2003	110.581444	98.502228	52.716091	85.887848	96.674271
2004	94.748741	83.548012	54.025761	92.844131	96.390167
2005	98.001549	87.041260	53.490040	98.698822	92.727310
2006	97.134453	82.181297	53.548672	96.123222	90.205849
2007	97.566612	86.575279	54.416019	72.806862	87.640869
2008	97.637657	89.624542	55.950310	71.161171	83.977928
2009	97.070259	97.627083	56.495579	75.501228	82.427650
2010	96.575058	100.929611	56.912849	93.307961	86.824443
2011	96.625069	101.251312	59.918468	84.852704	90.287827
2012	96.649490	103.300919	68.884468	79.070727	98.772331
2013	116.053131	119.009392	77.816119	70.546030	102.253548
2014	114.558609	115.826851	87.142226	71.093338	105.064331
2015	152.602434	119.224350	97.335522	82.431358	113.567223
2016	184.522396	109.647525	108.384690	102.923191	104.774669

country	SAU	SVN	SWE	TKM	TUR
2000	NaN	103.415916	194.291534	119.703447	59.035358
2001	NaN	113.185783	188.375183	115.465231	72.604080
2002	NaN	114.415611	179.227234	111.247835	81.337677
2003	NaN	117.613380	164.170822	107.908741	86.030586
2004	NaN	97.823280	100.788353	105.333490	79.048729
2005	84.107338	98.941811	101.846260	103.584799	71.356522
2006	85.438277	99.957161	102.462990	102.559221	77.087372
2007	87.255318	99.522072	104.637352	102.374196	80.205437
2008	88.561450	98.447639	105.235451	101.722983	77.345352
2009	90.535179	98.118523	103.432419	101.614663	66.364021
2010	95.536556	99.124382	100.079674	103.647713	72.301422
2011	100.263714	99.392036	97.093910	108.070354	78.178322
2012	104.327011	99.604507	96.739517	113.568864	74.078979
2013	108.197021	119.680840	141.753723	120.022999	95.422012
2014	108.431297	118.469719	148.578979	120.321960	102.081360
2015	109.053750	117.195865	163.634064	126.444851	106.123123
2016	109.675183	115.830869	160.633382	128.893789	158.304357

```
[17 rows x 53 columns]
ABW Aruba
AUS Australia
AZE Azerbaijan
BEL Belgium
BGR Bulgaria
BHR Bahrain
BHS Bahamas, The
BLR Belarus
BRA Brazil
BRB Barbados
CAN Canada
CHE Switzerland
CRI Costa Rica
CUB Cuba
CZE Czech Republic
DEU Germany
DNK Denmark
ECS Europe & Central Asia
ESP Spain
EST Estonia
FIN Finland
FRA France
GBR United Kingdom
GRC Greece
HUN Hungary
IRL Ireland
ISL Iceland
ISR Israel
JPN Japan
KAZ Kazakhstan
KGZ Kyrgyz Republic
LIE Liechtenstein
LTU Lithuania
LUX Luxembourg
LVA Latvia
MLT Malta
NAC North America
NLD Netherlands
NOR Norway
NRU Nauru
NZL New Zealand
PER Peru
PLW Palau
POL Poland
PRT Portugal
PRY Paraguay
QAT Qatar
RUS Russian Federation
SAU Saudi Arabia
SVN Slovenia
SWE Sweden
TKM Turkmenistan
TUR Turkey
```

In [101…

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

4.2.3 Marché / pouvoir d'achat

- PIB et PIB per habitant
- choisi PIB / habitant - 'NY.GDP.PCAP.CD'

- la population et le nombres des personnes inscrites dans l'education (secondaire et tertiare) donne aussi information sur la taille du marché.

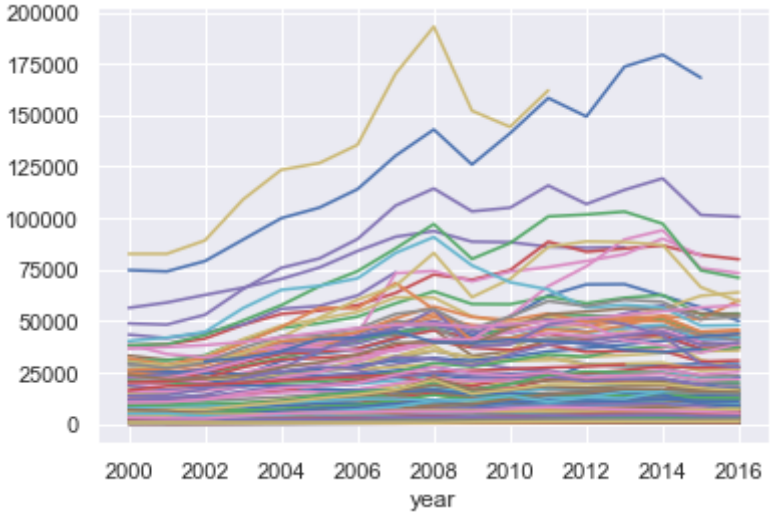
NY.GDP.PCAP.CD

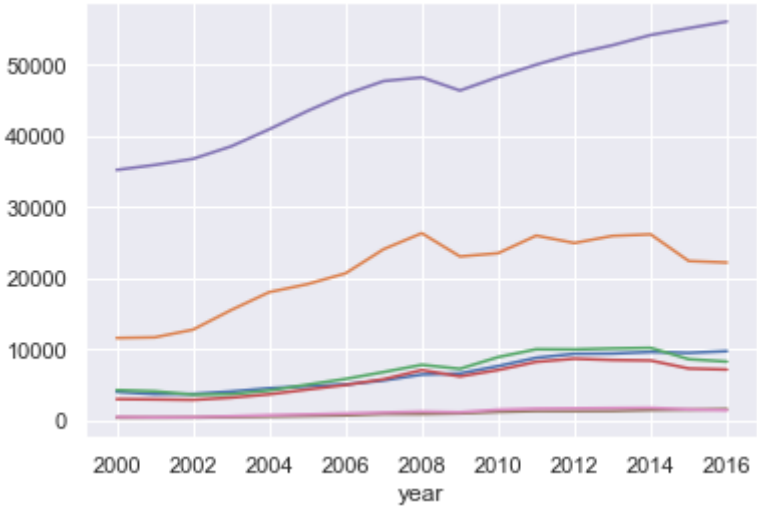
- PIB (Produit Intérieur Brut) per habitant
- donne information sur la richesse du pays /régions - sur le marché en générale

In [102…

```
ind = 'NY.GDP.PCAP.CD'
initial_investigation(ind)[0].T.info()
```

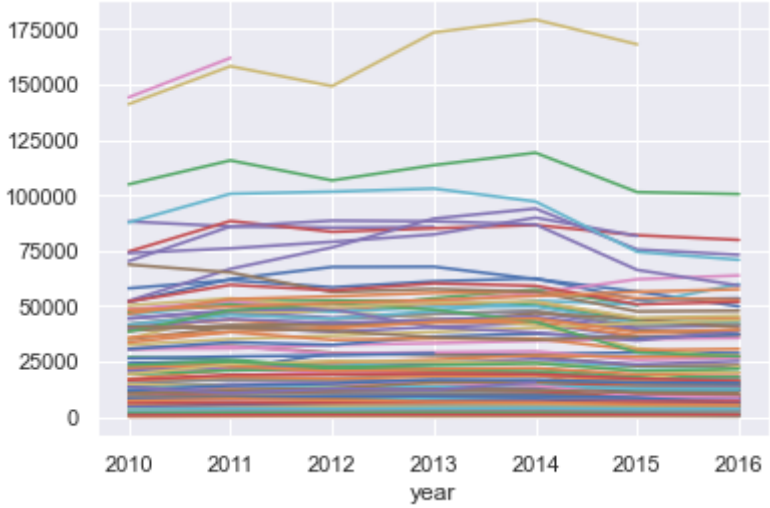
```
**************************************************
NY.GDP.PCAP.CD
```

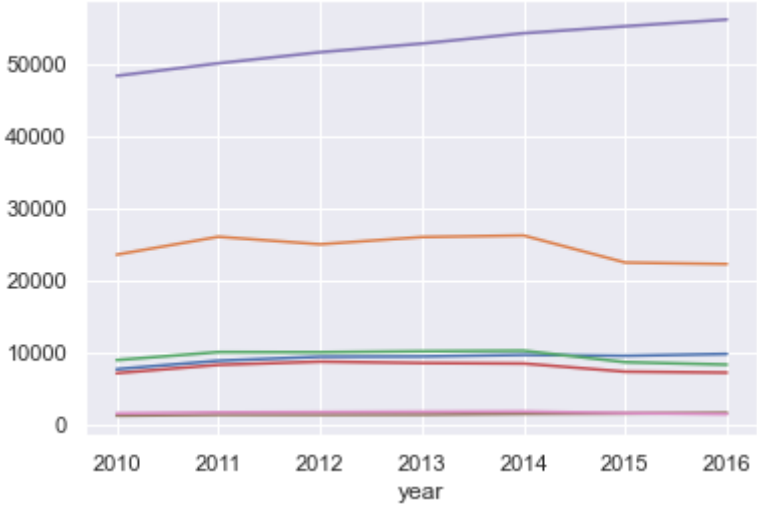





```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 216 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATG' 'AUS' 'AUT'
 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ'
 'BMU' 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHI' 'CHL'
 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CYM' 'CYP'
 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI'
 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA'
 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY' 'HKG' 'HND'
 'HRV' 'HTI' 'HUN' 'IDN' 'IMN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA'
 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO'
 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC'
```

```
 'MAR' 'MCO' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR'
 'MNE' 'MNG' 'MNP' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NCL' 'NER'
 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW'
 'PNG' 'POL' 'PRI' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS'
 'SAU' 'SDN' 'SEN' 'SGP' 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSD' 'SSF'
 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYC' 'SYR' 'TCD' 'TGO' 'THA' 'TJK'
 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA'
 'UZB' 'VCT' 'VEN' 'VIR' 'VNM' 'VUT' 'WSM' 'XKX' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'VGB', 'SXM', 'MAF', 'PRK', 'NRU', 'TCA',
'GIB', 'CUW'}
These regions have absolutely no data : set()
**************************************************
**************************************************
NY.GDP.PCAP.CD
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 211 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATG' 'AUS' 'AUT'
 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ'
 'BMU' 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHL' 'CHN'
 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CYP' 'CZE' 'DEU'
 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY' 'ERI' 'ESP' 'EST'
 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB'
 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY' 'HKG' 'HND' 'HRV' 'HTI'
 'HUN' 'IDN' 'IMN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR' 'ITA' 'JAM' 'JOR'
 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR' 'KWT' 'LAO' 'LBN' 'LBR'
 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MCO'
 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG'
 'MNP' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC' 'NAM' 'NER' 'NGA' 'NIC' 'NLD'
```

```
 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN' 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI'
 'PRT' 'PRY' 'PSE' 'QAT' 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SGP'
 'SLB' 'SLE' 'SLV' 'SMR' 'SOM' 'SRB' 'SSD' 'SSF' 'STP' 'SUR' 'SVK' 'SVN'
 'SWE' 'SWZ' 'SYC' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN'
 'TUR' 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VIR' 'VNM'
 'VUT' 'WSM' 'XKX' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'VGB', 'CHI', 'SXM', 'CYM', 'MAF', 'PYF',
'PRK', 'SYR', 'NRU', 'NCL', 'TCA', 'GIB', 'CUW'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.972241029113067

year            2010           2011           2012           2013  \
count     210.000000     210.000000     206.000000     207.000000
mean    15362.394127   16882.923062   16135.532080   16644.524102
std     22948.091782   25382.070112   22914.352265   24208.321211
min       231.194326     260.479907     265.285605     282.755525
25%      1501.494867    1685.371763    1770.871134    1865.654630
50%      5563.604181    5965.269402    6406.863200    6597.211381
75%     19275.457330   21505.951836   20365.519606   20081.506235
max    144246.368775  162009.877819  149295.646044  173528.150454

year            2014           2015           2016
count     204.000000     203.000000     196.000000
mean    16568.321231   14842.064439   13377.976868
std     24307.888662   21701.939919   18039.795507
min       312.748979     300.676557     285.727442
25%      2015.603992    1892.104331    1853.821155
50%      6487.818716    5814.863134    5235.308547
75%     19430.097692   17088.226055   15928.850971
max    179308.075616  168146.015281  100573.139978
<class 'pandas.core.frame.DataFrame'>
Index: 211 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    210 non-null    float64
 1   2011    210 non-null    float64
 2   2012    206 non-null    float64
 3   2013    207 non-null    float64
 4   2014    204 non-null    float64
 5   2015    203 non-null    float64
 6   2016    196 non-null    float64
dtypes: float64(7)
memory usage: 13.2+ KB
```

Résumé: NY.GDP.PCAP.CD

- Pour 204 pays, il y a de la data pour les années 2010-2016.
  - 13 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 97% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Le plus complète année entre 2010-2016 est 2011.**.
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [103…

```
ind = 'NY.GDP.PCAP.CD'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data2 = drop_countries(data.loc[years,:],7)
print(len(data2.columns.values))
data.T[years].describe()
```

```
The following countries were dropped :
['CHI', 'CUW', 'CYM', 'GIB', 'MAF', 'NCL', 'NRU', 'PRK', 'PYF', 'SXM', 'SYR', 'TCA',
'VGB']
211
```

Out[103…

year	2010	2011	2012	2013	2014	2015
count	210.000000	210.000000	206.000000	207.000000	204.000000	203.000000
mean	15362.394127	16882.923062	16135.532080	16644.524102	16568.321231	14842.064439
std	22948.091782	25382.070112	22914.352265	24208.321211	24307.888662	21701.939919
min	231.194326	260.479907	265.285605	282.755525	312.748979	300.676557
25%	1501.494867	1685.371763	1770.871134	1865.654630	2015.603992	1892.104331
50%	5563.604181	5965.269402	6406.863200	6597.211381	6487.818716	5814.863134
75%	19275.457330	21505.951836	20365.519606	20081.506235	19430.097692	17088.226055
max	144246.368775	162009.877819	149295.646044	173528.150454	179308.075616	168146.015281

In [104…

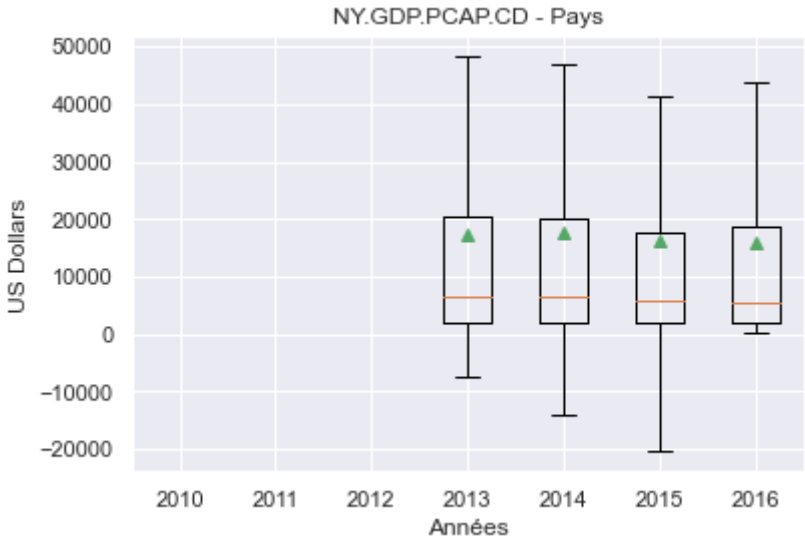
```
data_clean = fill_indicator_linear(data[data2.columns.values])
data_clean.T[years].describe()
```

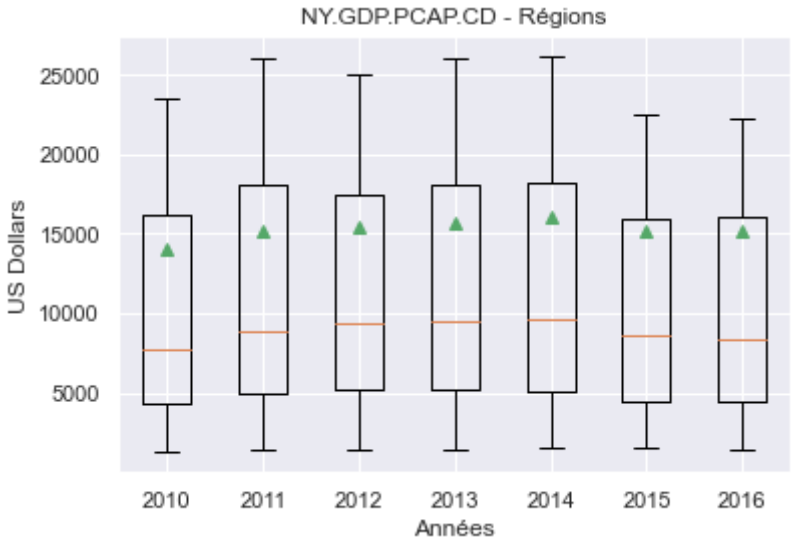
Out[104…

	2010	2011	2012	2013	2014	2015
count	210.000000	210.000000	210.000000	211.000000	211.000000	211.000000
mean	15362.394127	16882.923062	16808.754449	17363.669278	17665.502458	16081.702009
std	22948.091782	25382.070112	25411.015383	27105.208195	28093.474745	26694.670269
min	231.194326	260.479907	-915.463301	-7433.476036	-13951.488771	-20469.501506
25%	1501.494867	1685.371763	1756.864855	1848.995704	1992.558665	1847.333163
50%	5563.604181	5965.269402	6406.863200	6597.211381	6491.052450	5949.106307
75%	19275.457330	21505.951836	21312.590259	20757.874728	20204.016823	17893.196816
max	144246.368775	162009.877819	179773.386864	197536.895908	215300.404952	233063.913997

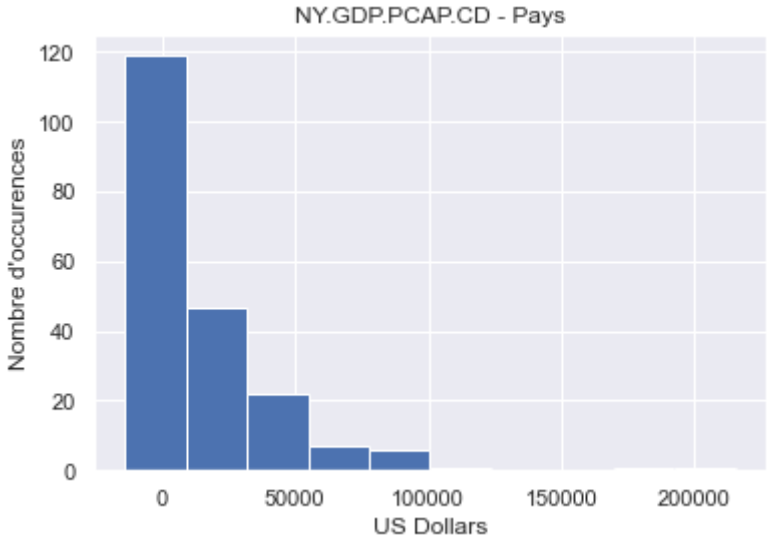
In [105…

```
plot_boxplot(ind, data_clean, unit = "US Dollars", years = years, showmeans = True,
print_info_year('2014', data_clean, unit = "US Dollars", ind = ind)
print_info_year('2016', data_clean, unit = "US Dollars", ind = ind)
```

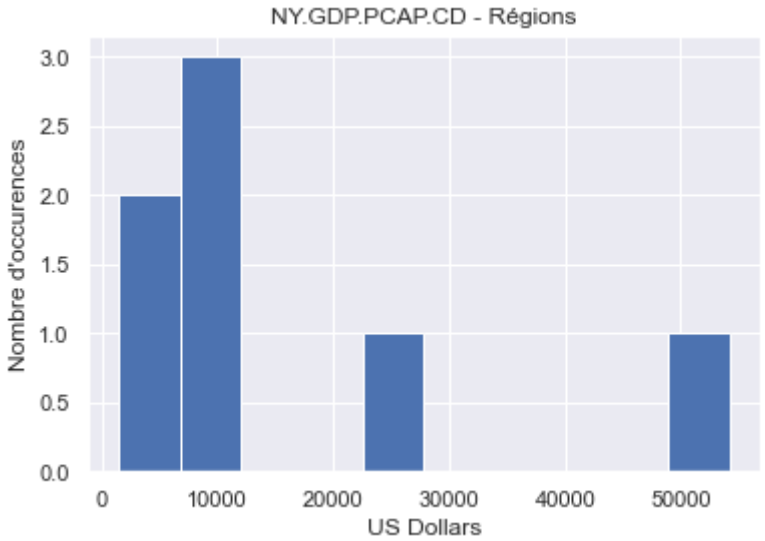




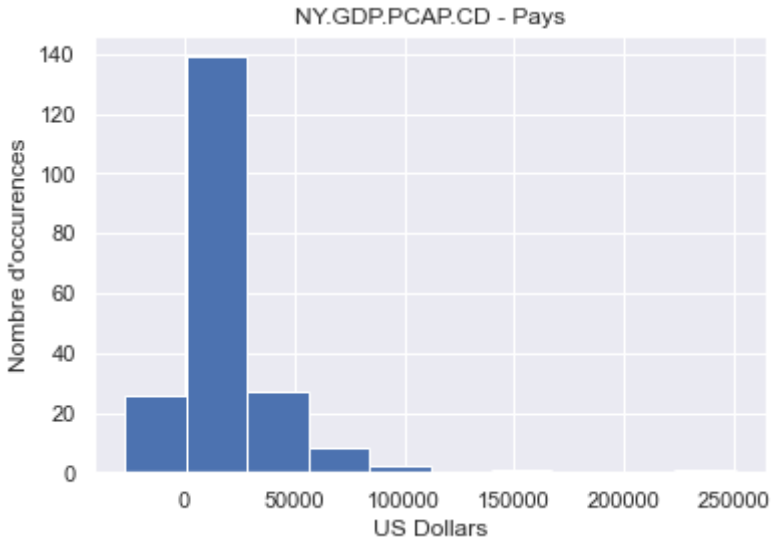

```
2014
Pays
Median:  6440.607733002995
Mean:  17722.342311783974
Std:  28390.09495041479
```



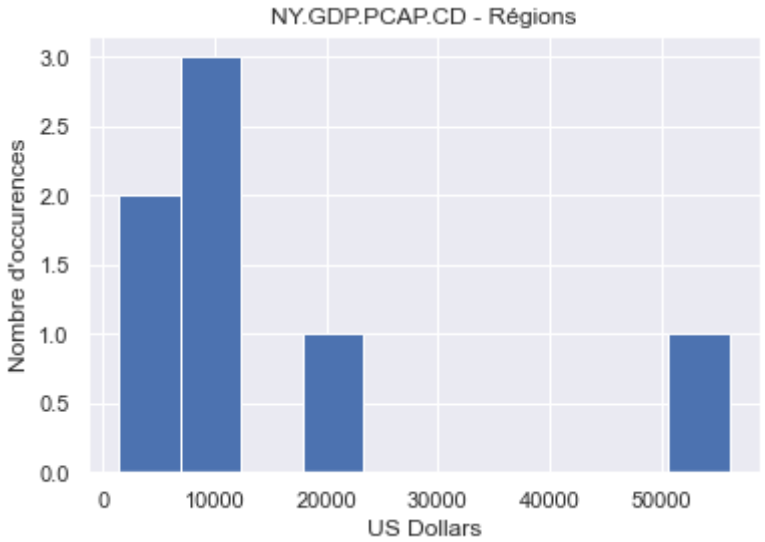

```
Régions
Median:  9665.05607927331
Mean:  16009.02671265367
Std:  18724.94187978487
```




```
2016
Pays
Median:  5355.7507752448
Mean:  16022.8985280936
Std:  27225.472466026924
```




```
Régions
Median:  8311.37017802892
Mean:  15244.838504984276
Std:  19296.324162122706
```



Résumé: NP.GDP.PCAP.CD

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 5 360, 16 000 et 27 200 US-Dollar pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 8 310, 15 200 et 19 300 US-Dollar pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne une indication sur la taille du marché du pays / régions (par personne).

In [106…

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEAN = DATA_CLEAN.drop_duplicates()
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

4.2.4 Population

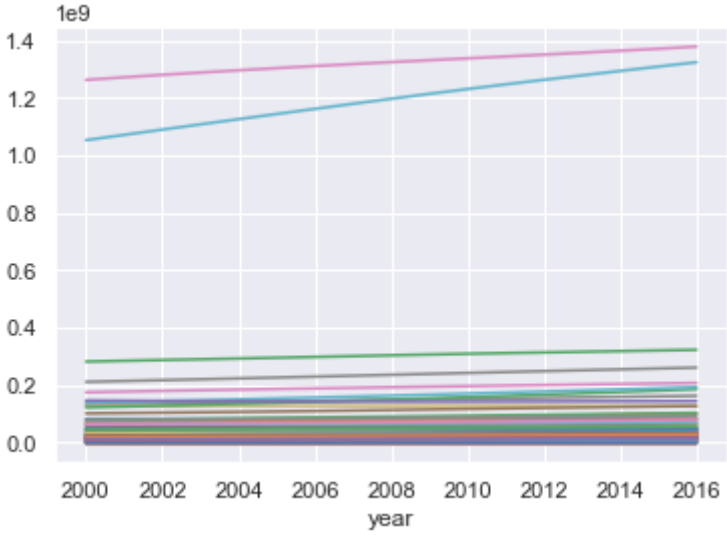
- Population = donne information sur le nombre de potentiels clients = taille de marché
- C'est aussi une unité avec laquelle ont peut normaliser des autre variables
- Choisi SP.POP.TOTL car c'est la variable le plus complète.

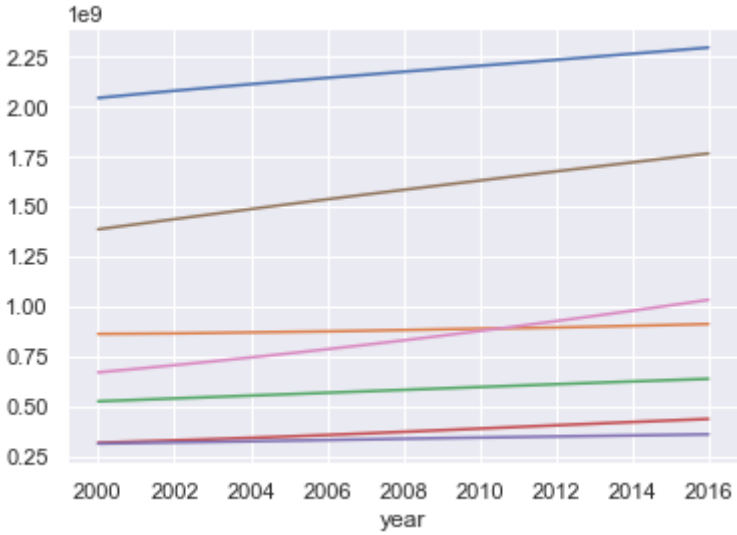
SP.POP.TOTL

In [107…

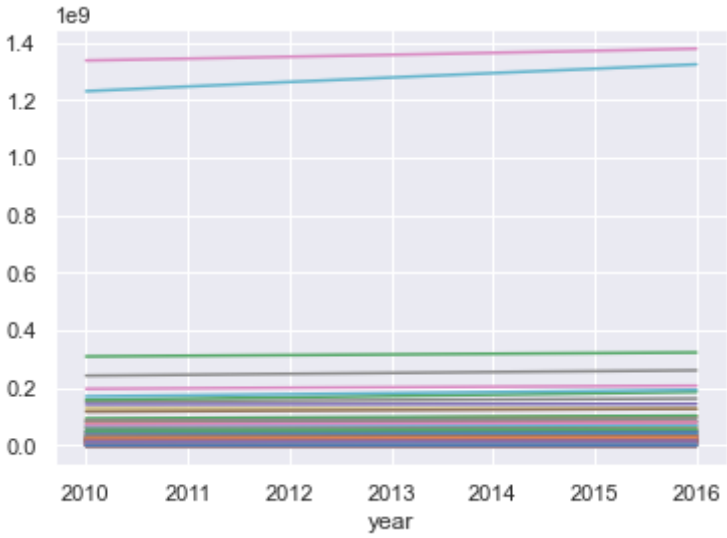
```
ind = 'SP.POP.TOTL'
initial_investigation(ind)[0].T.info()
```

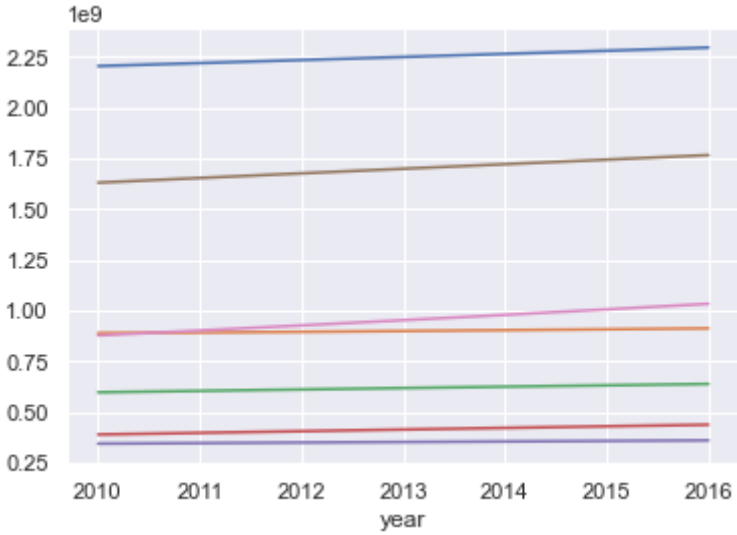
```
**************************************************
SP.POP.TOTL
```






```
We have data between 2000 and 2016.
The exact years are :
['2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009'
 '2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 222 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATG' 'AUS' 'AUT'
 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ'
 'BMU' 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHI' 'CHL'
 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYM'
 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY'
 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO'
 'GHA' 'GIB' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IMN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL'
 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR'
 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX'
 'LVA' 'MAC' 'MAF' 'MAR' 'MCO' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD'
 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MNP' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC'
 'NAM' 'NCL' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN'
 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT'
 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SGP' 'SLB' 'SLE' 'SLV' 'SMR'
 'SOM' 'SRB' 'SSD' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SXM' 'SYC'
 'SYR' 'TCA' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR'
 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VIR' 'VNM' 'VUT'
 'WSM' 'XKX' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'NRU', 'VGB'}
These regions have absolutely no data : set()
**************************************************
**************************************************
SP.POP.TOTL
```






```
We have data between 2010 and 2016.
The exact years are :
['2010' '2011' '2012' '2013' '2014' '2015' '2016']

We have data for the 222 countries.
The countries are :
['ABW' 'AFG' 'AGO' 'ALB' 'AND' 'ARE' 'ARG' 'ARM' 'ASM' 'ATG' 'AUS' 'AUT'
 'AZE' 'BDI' 'BEL' 'BEN' 'BFA' 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ'
 'BMU' 'BOL' 'BRA' 'BRB' 'BRN' 'BTN' 'BWA' 'CAF' 'CAN' 'CHE' 'CHI' 'CHL'
 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB' 'CUW' 'CYM'
 'CYP' 'CZE' 'DEU' 'DJI' 'DMA' 'DNK' 'DOM' 'DZA' 'EAS' 'ECS' 'ECU' 'EGY'
 'ERI' 'ESP' 'EST' 'ETH' 'FIN' 'FJI' 'FRA' 'FRO' 'FSM' 'GAB' 'GBR' 'GEO'
 'GHA' 'GIB' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GRD' 'GRL' 'GTM' 'GUM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IMN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL'
 'ISR' 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KIR' 'KNA' 'KOR'
 'KWT' 'LAO' 'LBN' 'LBR' 'LBY' 'LCA' 'LCN' 'LIE' 'LKA' 'LSO' 'LTU' 'LUX'
 'LVA' 'MAC' 'MAF' 'MAR' 'MCO' 'MDA' 'MDG' 'MDV' 'MEA' 'MEX' 'MHL' 'MKD'
 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MNP' 'MOZ' 'MRT' 'MUS' 'MWI' 'MYS' 'NAC'
 'NAM' 'NCL' 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'OMN' 'PAK' 'PAN'
 'PER' 'PHL' 'PLW' 'PNG' 'POL' 'PRI' 'PRK' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT'
 'ROU' 'RUS' 'RWA' 'SAS' 'SAU' 'SDN' 'SEN' 'SGP' 'SLB' 'SLE' 'SLV' 'SMR'
 'SOM' 'SRB' 'SSD' 'SSF' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SXM' 'SYC'
 'SYR' 'TCA' 'TCD' 'TGO' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR'
 'TUV' 'TZA' 'UGA' 'UKR' 'URY' 'USA' 'UZB' 'VCT' 'VEN' 'VIR' 'VNM' 'VUT'
 'WSM' 'XKX' 'YEM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'NRU', 'VGB'}
These regions have absolutely no data : set()
**************************************************
Completeness for the years 2010-2016: 0.989060489060489

year           2010          2011          2012          2013          2014  \
count  2.220000e+02  2.220000e+02  2.210000e+02  2.210000e+02  2.170000e+02
mean   6.233340e+07  6.307390e+07  6.410391e+07  6.487695e+07  6.681123e+07
std    2.386456e+08  2.410239e+08  2.439683e+08  2.464174e+08  2.510360e+08
min    1.053100e+04  1.062800e+04  1.072500e+04  1.081900e+04  1.090800e+04
25%    8.827385e+05  8.988870e+05  8.811850e+05  8.966880e+05  1.152309e+06
50%    6.352497e+06  6.472590e+06  6.766103e+06  6.922079e+06  7.130576e+06
75%    2.406775e+07  2.460478e+07  2.509615e+07  2.599834e+07  2.696256e+07
max    2.205615e+09  2.220358e+09  2.235430e+09  2.250552e+09  2.265898e+09

year           2015          2016
count  2.170000e+02  2.170000e+02
mean   6.760359e+07  6.840231e+07
std    2.535391e+08  2.560779e+08
min    1.100100e+04  1.109700e+04
25%    1.175389e+06  1.221490e+06
50%    7.177991e+06  7.127822e+06
75%    2.785930e+07  2.881346e+07
max    2.281203e+09  2.296786e+09
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 222 entries, ABW to ZWE
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    222 non-null    float64
 1   2011    222 non-null    float64
 2   2012    221 non-null    float64
 3   2013    221 non-null    float64
 4   2014    217 non-null    float64
 5   2015    217 non-null    float64
 6   2016    217 non-null    float64
dtypes: float64(7)
memory usage: 13.9+ KB
```

Résumé: SP.POP.TOTL

- Pour 215 pays, il y a de la data pour les années 2010-2016.
  - que 2 des 217 pays n'ont pas de data pour cette variable.
- Il y a de la data pour les 7 régions.
- La data pour cette variable est complète à 98.9% pour les années 2010-2016 (après avoir écarter les pays, pour lequels il n'y a pas de data)
- **Les années 2010 et 2011 sont complètes, mais même les autres années sont est quasi complète**
- Utiliser interpolation / extrapolation lineaire pour remplir les trous dans les données.

In [108…

```
ind = 'SP.POP.TOTL'
years = [str(y) for y in range(2010, 2017)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
#drop data, where half of the data points are missing and there are no data points f
data2 = drop_countries(data.loc[years,:],7)
print(len(data2.columns.values))
data.T[years].describe()
```

```
The following countries were dropped :
['NRU', 'VGB']
222
```

Out[108…

year	2010	2011	2012	2013	2014	2015	
count	2.220000e+02	2.220000e+02	2.210000e+02	2.210000e+02	2.170000e+02	2.170000e+02	2.170(
mean	6.233340e+07	6.307390e+07	6.410391e+07	6.487695e+07	6.681123e+07	6.760359e+07	6.8402
std	2.386456e+08	2.410239e+08	2.439683e+08	2.464174e+08	2.510360e+08	2.535391e+08	2.5607
min	1.053100e+04	1.062800e+04	1.072500e+04	1.081900e+04	1.090800e+04	1.100100e+04	1.1097
25%	8.827385e+05	8.988870e+05	8.811850e+05	8.966880e+05	1.152309e+06	1.175389e+06	1.2214
50%	6.352497e+06	6.472590e+06	6.766103e+06	6.922079e+06	7.130576e+06	7.177991e+06	7.1278
75%	2.406775e+07	2.460478e+07	2.509615e+07	2.599834e+07	2.696256e+07	2.785930e+07	2.8813
max	2.205615e+09	2.220358e+09	2.235430e+09	2.250552e+09	2.265898e+09	2.281203e+09	2.2967

In [109…

```
data_clean = fill_indicator_linear(data[data2.columns.values])
data_clean.T[years].describe()
```

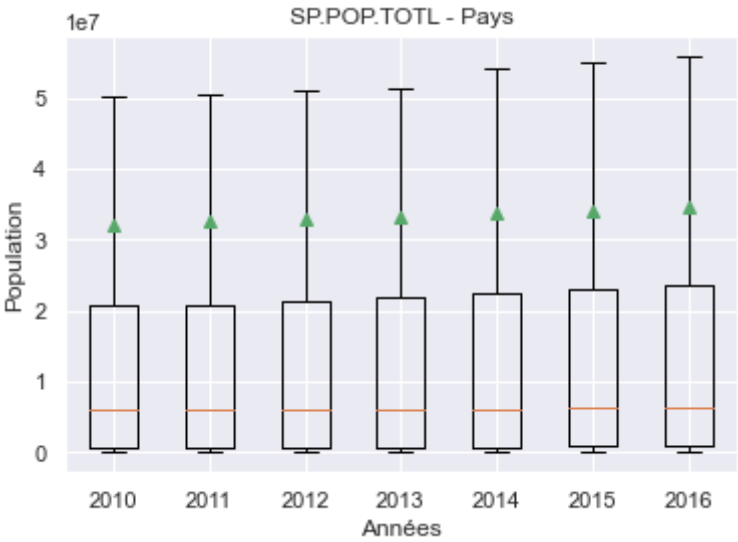
Out[109…

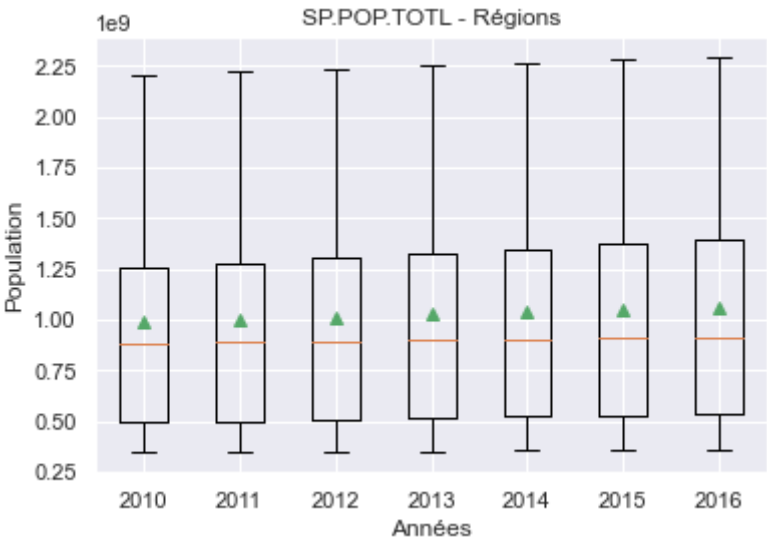
	2010	2011	2012	2013	2014	2015	
count	2.220000e+02	2.220000e+02	2.220000e+02	2.220000e+02	2.220000e+02	2.220000e+02	2.220(

	2010	2011	2012	2013	2014	2015	
mean	6.233340e+07	6.307390e+07	6.383568e+07	6.460563e+07	6.538173e+07	6.615870e+07	6.694
std	2.386456e+08	2.410239e+08	2.434485e+08	2.458925e+08	2.483603e+08	2.508371e+08	2.533
min	1.053100e+04	1.062800e+04	1.072500e+04	1.081900e+04	1.090800e+04	1.100100e+04	1.109
25%	8.827385e+05	8.988870e+05	9.205370e+05	9.434525e+05	9.664790e+05	9.858068e+05	9.992
50%	6.352497e+06	6.472590e+06	6.590636e+06	6.708318e+06	6.827780e+06	6.879675e+06	6.907
75%	2.406775e+07	2.460478e+07	2.504960e+07	2.589284e+07	2.675193e+07	2.741617e+07	2.805
max	2.205615e+09	2.220358e+09	2.235430e+09	2.250552e+09	2.265898e+09	2.281203e+09	2.296

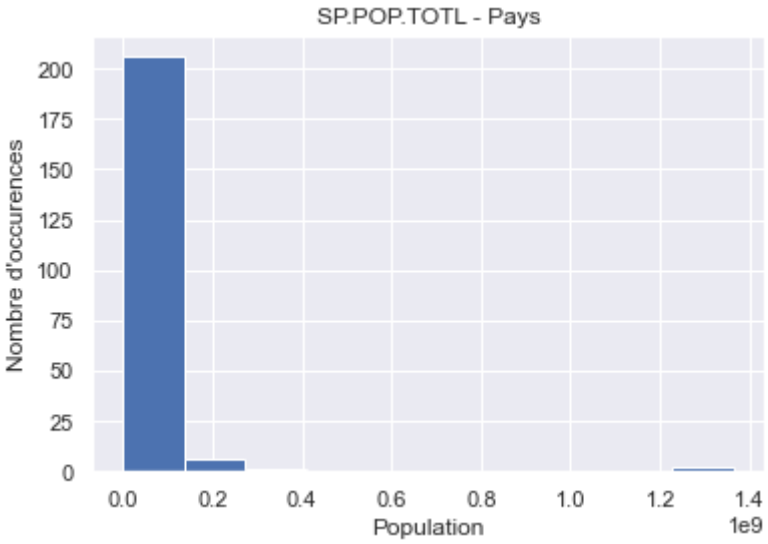
In [110…

```
plot_boxplot(ind, data_clean, unit = "Population", years = years, showmeans = True,
print_info_year('2014', data_clean, unit = "Population", ind = ind)
print_info_year('2016', data_clean, unit = "Population", ind = ind)
```

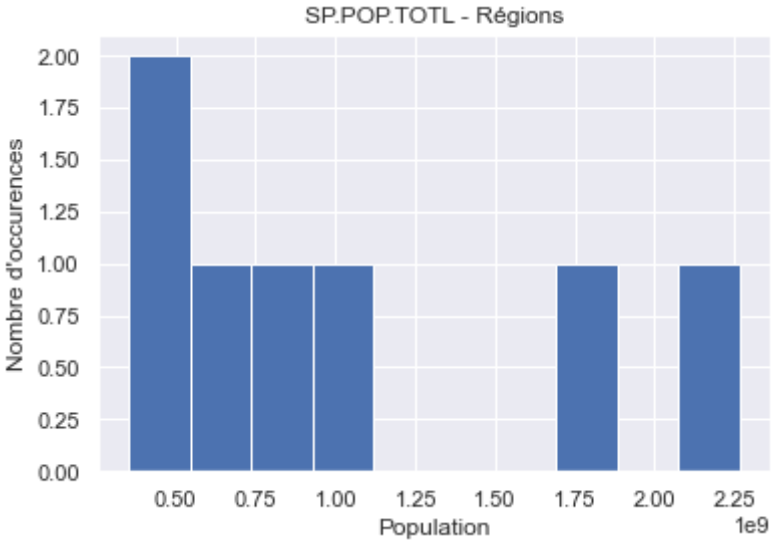




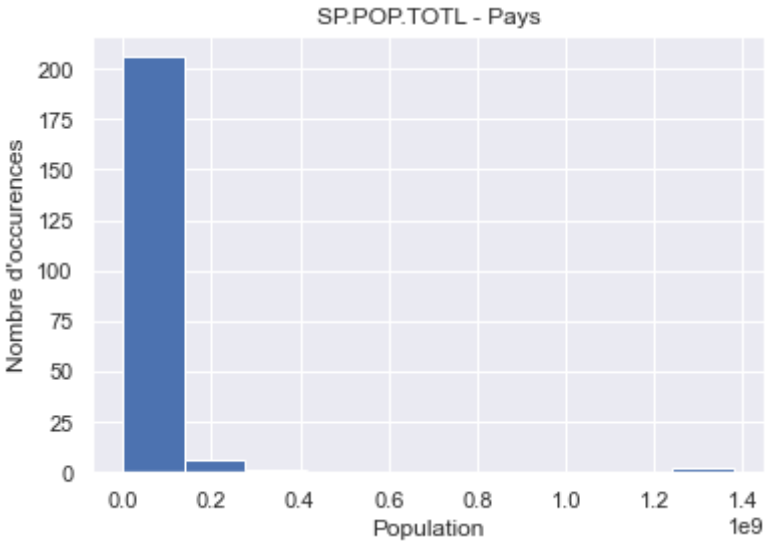

```
2014
Pays
Median:  6204108.0
Mean:  33701202.1243682
Std:  132762074.00437006
```



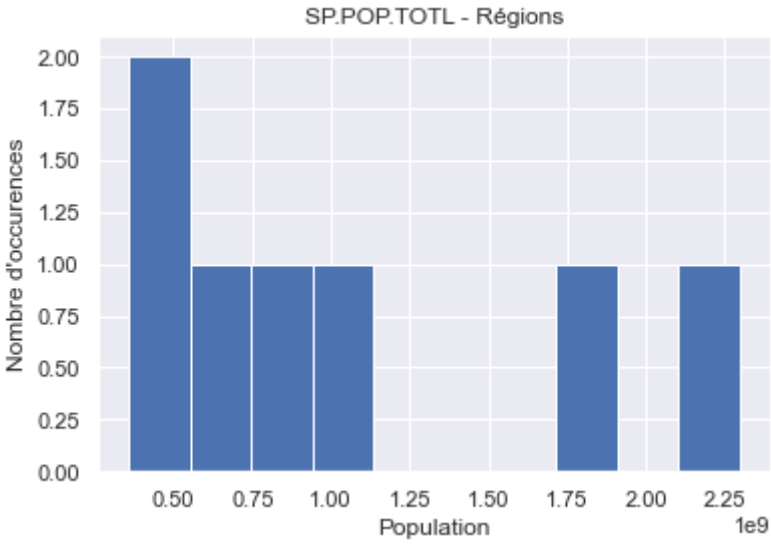

```
Régions
Median:  903094668.0
Mean:  1038426596.5341661
Std:  709048234.7089626
```




```
2016
Pays
Median:  6293253.0
Mean:  34506809.739534885
Std:  135048310.4802969
```




```
Régions
Median:  911995305.0
Mean:  1063162225.4285715
Std:  719809003.3983943
```



Résumé: SP.POP.TOTL

- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 6.29, 34.5 et 135 million Personnes pour les pays**
- Pour l'année 2016 **la médian, le moyenne et l'écart-type sont 911, 1 060 et 720 million Personnes pour les régions**
- Il y a une grande écarte entre les différents pays / régions.
- Cette variable donne une indication sur la taille du marché du pays / régions (par personne).

In [111…

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEAN = DATA_CLEAN.drop_duplicates()
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

4.2.5 Future potentiel

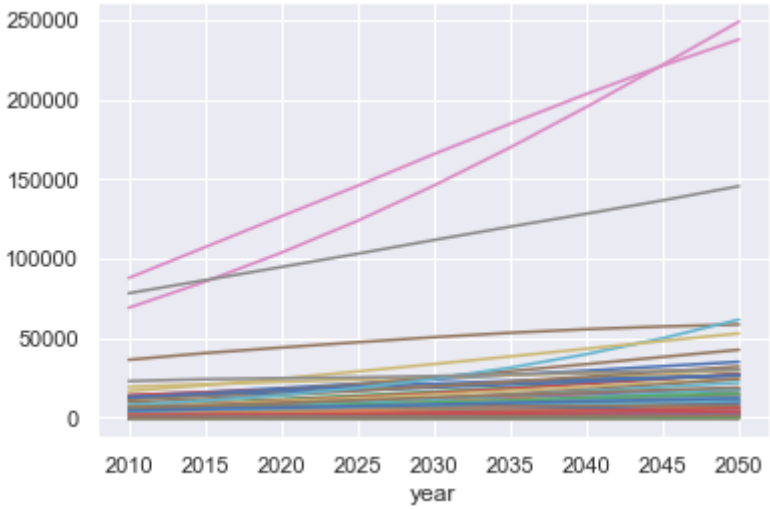
- Population avec ....
- PRJ.POP.ALL.4.MF , PRJ.ATT.25UP.4.MF

PRJ.POP.ALL.4.MF

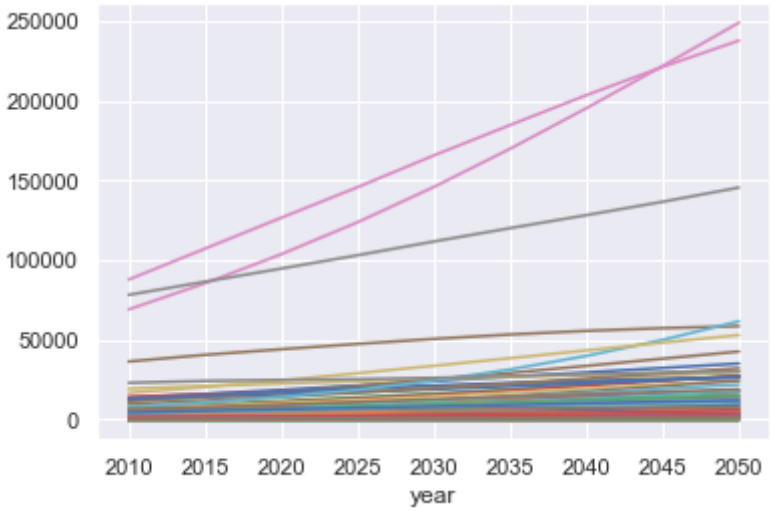
In [112…

```
ind = 'PRJ.POP.ALL.4.MF'
initial_investigation(ind, is_projection_data = True )[0].T.info()
```

```
**************************************************
PRJ.POP.ALL.4.MF
```




```
No data to plot.
We have data between 2010 and 2050.
The exact years are :
['2010' '2015' '2020' '2025' '2030' '2035' '2040' '2045' '2050']

We have data for the 166 countries.
The countries are :
['ABW' 'ALB' 'ARE' 'ARG' 'ARM' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL' 'BEN' 'BFA'
 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BOL' 'BRA' 'BTN' 'CAF' 'CAN'
 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB'
 'CYP' 'CZE' 'DEU' 'DNK' 'DOM' 'DZA' 'ECU' 'EGY' 'ESP' 'EST' 'ETH' 'FIN'
 'FRA' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GTM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KOR' 'KWT' 'LAO' 'LBN'
 'LBR' 'LCA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEX'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MUS' 'MWI' 'MYS' 'NAM' 'NCL'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'POL'
 'PRI' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAU' 'SDN' 'SEN'
 'SGP' 'SLE' 'SLV' 'SOM' 'SRB' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYR'
 'TCD' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'VCT' 'VEN' 'VNM' 'VUT' 'WSM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'AGO', 'BWA', 'SYC', 'DMA', 'CHI', 'LBY',
'MHL', 'PLW', 'MNP', 'CYM', 'TGO', 'BRB', 'NRU', 'KNA', 'CUW', 'AFG', 'ERI', 'SLB',
'GRD', 'BMU', 'PRK', 'SSD', 'FJI', 'TUV', 'SMR', 'GRL', 'MCO', 'MRT', 'LIE', 'FRO',
'DJI', 'LKA', 'ASM', 'OMN', 'VIR', 'FSM', 'KIR', 'ATG', 'BRN', 'TCA', 'IMN', 'VGB',
'XKX', 'SXM', 'PNG', 'UZB', 'YEM', 'GUM', 'AND', 'MAF', 'GIB'}
These regions have absolutely no data : {'MEA', 'LCN', 'SAS', 'ECS', 'EAS', 'NAC',
'SSF'}
**************************************************
**************************************************
PRJ.POP.ALL.4.MF
```




```
No data to plot.
```

```
We have data between 2010 and 2050.
The exact years are :
['2010' '2015' '2020' '2025' '2030' '2035' '2040' '2045' '2050']

We have data for the 166 countries.
The countries are :
['ABW' 'ALB' 'ARE' 'ARG' 'ARM' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL' 'BEN' 'BFA'
 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BOL' 'BRA' 'BTN' 'CAF' 'CAN'
 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB'
 'CYP' 'CZE' 'DEU' 'DNK' 'DOM' 'DZA' 'ECU' 'EGY' 'ESP' 'EST' 'ETH' 'FIN'
 'FRA' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GTM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KOR' 'KWT' 'LAO' 'LBN'
 'LBR' 'LCA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEX'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MUS' 'MWI' 'MYS' 'NAM' 'NCL'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'POL'
 'PRI' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAU' 'SDN' 'SEN'
 'SGP' 'SLE' 'SLV' 'SOM' 'SRB' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYR'
 'TCD' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'VCT' 'VEN' 'VNM' 'VUT' 'WSM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'AGO', 'BWA', 'SYC', 'DMA', 'CHI', 'LBY',
'MHL', 'PLW', 'MNP', 'CYM', 'TGO', 'BRB', 'NRU', 'KNA', 'CUW', 'AFG', 'ERI', 'SLB',
'GRD', 'BMU', 'PRK', 'SSD', 'FJI', 'TUV', 'SMR', 'GRL', 'MCO', 'MRT', 'LIE', 'FRO',
'DJI', 'LKA', 'ASM', 'OMN', 'VIR', 'FSM', 'KIR', 'ATG', 'BRN', 'TCA', 'IMN', 'VGB',
'XKX', 'SXM', 'PNG', 'UZB', 'YEM', 'GUM', 'AND', 'MAF', 'GIB'}
These regions have absolutely no data : {'MEA', 'LCN', 'SAS', 'ECS', 'EAS', 'NAC',
'SSF'}
**************************************************
Completeness for the years 2010-2050: 1.0

year           2010           2015           2020           2025  \
count    166.000000     166.000000     166.000000     166.000000
mean    3642.273976    4294.490241    4977.415542    5698.443193
std    11190.550180   13210.955716   15326.409313   17542.132794
min        0.870000       1.050000       1.250000       1.490000
25%       85.405000     110.642500     141.612500     184.862500
50%      527.930000     598.785000     739.360000     890.730000
75%     1893.540000    2384.447500    2645.472500    3322.790000
max    87715.740000  107118.130000  126496.960000  145660.860000

year            2030           2035           2040           2045  \
count     166.000000     166.000000     166.000000     166.000000
mean     6476.520361    7295.463675    8146.653193    9018.287169
std     19925.264340   22375.954613   24905.087511   27475.998172
min         1.800000       2.180000       2.620000       3.120000
25%       237.647500     288.072500     355.997500     412.030000
50%      1055.615000    1232.295000    1421.220000    1538.570000
75%      3805.522500    4361.190000    5007.547500    5599.827500
max    165561.250000  184640.780000  203381.680000  221695.020000

year            2050
count     166.000000
mean     9902.461386
std     30073.763725
min         3.690000
25%       456.782500
50%      1623.100000
75%      5982.977500
max    248880.500000
<class 'pandas.core.frame.DataFrame'>
Index: 166 entries, ABW to ZWE
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    166 non-null    float64
 1   2015    166 non-null    float64
 2   2020    166 non-null    float64
 3   2025    166 non-null    float64
```

```
 4   2030    166 non-null    float64
 5   2035    166 non-null    float64
 6   2040    166 non-null    float64
 7   2045    166 non-null    float64
 8   2050    166 non-null    float64
dtypes: float64(9)
memory usage: 13.0+ KB
```

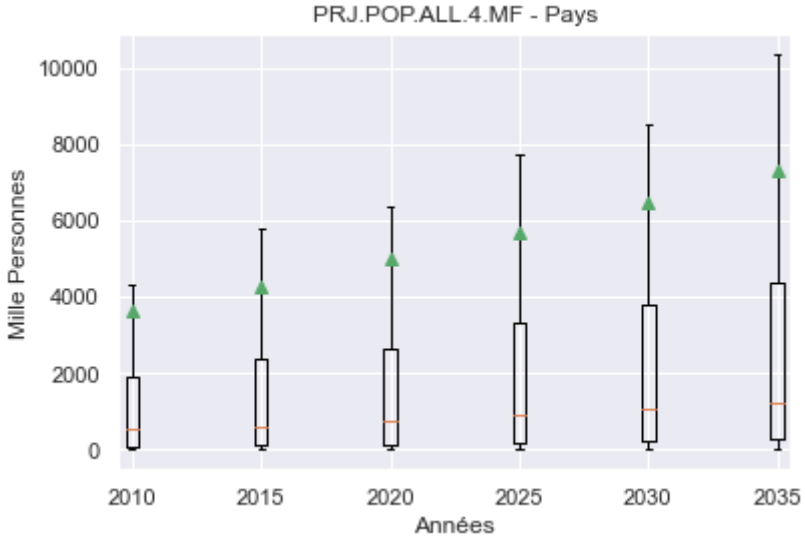
Résumé: PRJ.POP.ALL.4.MF

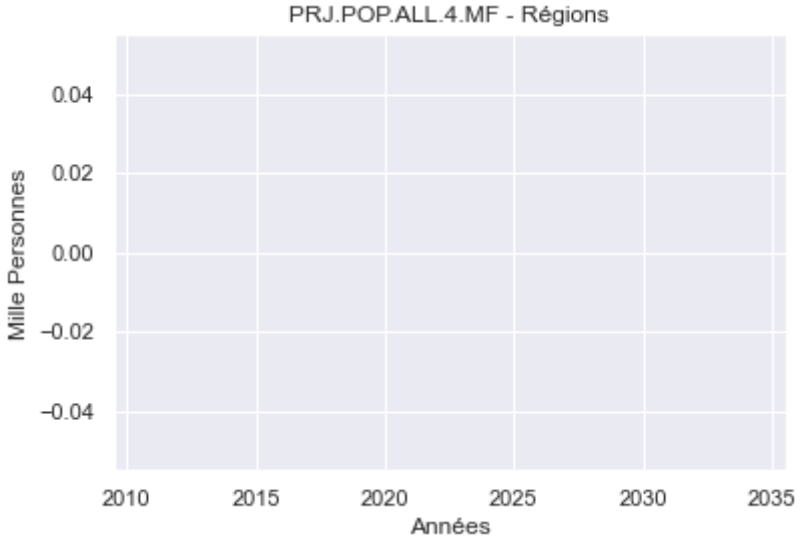
- Pour 166 pays, il y a de la data pour les années 2010-2050 (en intervales de 5 ans).
  - 51 des 217 pays n'ont pas de data pour cette variable.
- Il n'y a pas de la data pour les régions.
- **La data pour cette variable est complète à 100% (par construction)**

In [113…

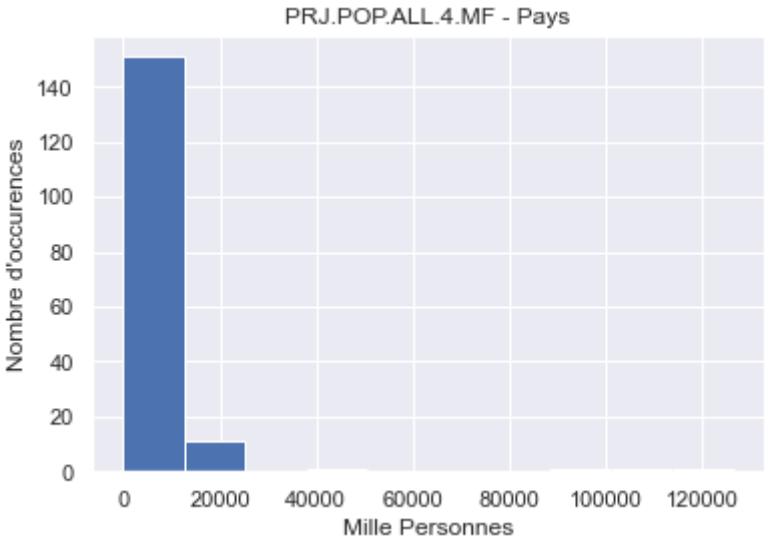
```
years = [str(y) for y in range(2010, 2036, 5)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
data_clean = drop_countries(data, 3)
plot_boxplot(ind, data_clean, unit = "Mille Personnes", years = years, showmeans = T
print_info_year('2020', data_clean, unit = "Mille Personnes", ind = ind)
print_info_year('2025', data_clean, unit = "Mille Personnes", ind = ind)
```

```
The following countries were dropped :
['AFG', 'AGO', 'AND', 'ASM', 'ATG', 'BMU', 'BRB', 'BRN', 'BWA', 'CHI', 'CUW', 'CYM',
'DJI', 'DMA', 'EAS', 'ECS', 'ERI', 'FJI', 'FRO', 'FSM', 'GIB', 'GRD', 'GRL', 'GUM',
'IMN', 'KIR', 'KNA', 'LBY', 'LCN', 'LIE', 'LKA', 'MAF', 'MCO', 'MEA', 'MHL', 'MNP',
'MRT', 'NAC', 'NRU', 'OMN', 'PLW', 'PNG', 'PRK', 'SAS', 'SLB', 'SMR', 'SSD', 'SSF',
'SXM', 'SYC', 'TCA', 'TGO', 'TUV', 'UZB', 'VGB', 'VIR', 'XKX', 'YEM']
```

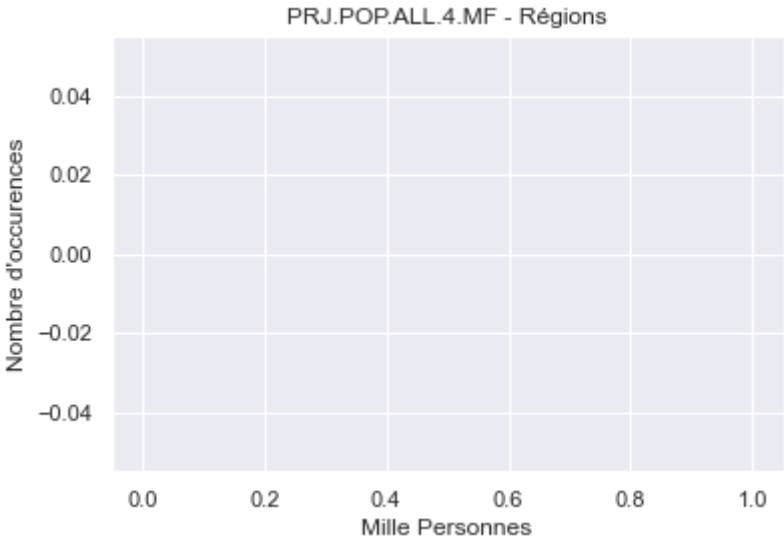




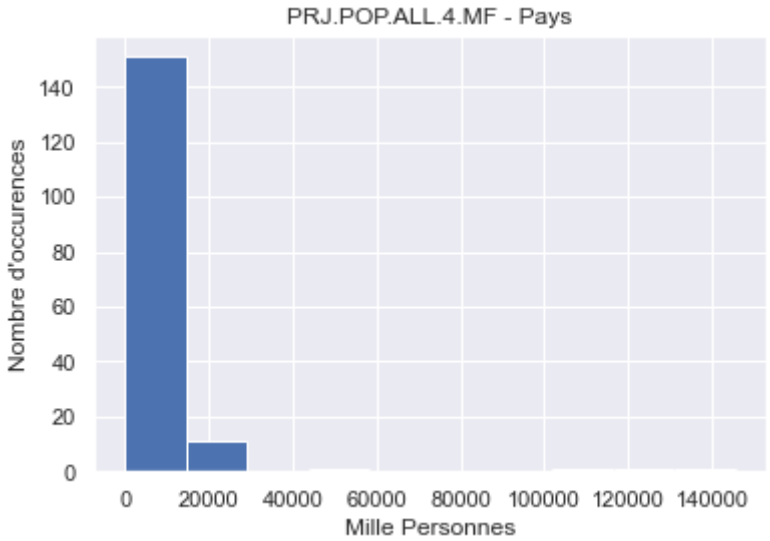
```
2020
Pays
Median:  739.36
Mean:  4977.415542168675
Std:  15326.409312893655
```




```
Régions
Median:  nan
Mean:  nan
Std:  nan
```




```
2025
Pays
Median:  890.73
Mean:  5698.443192771085
Std:  17542.132794187102
```




```
Régions
Median:  nan
Mean:  nan
Std:  nan
```

Résumé: PRJ.POP.ALL.4.MF

- Pour l'année 2020 **la médian, le moyenne et l'écart-type sont 739, 4 980 et 15 300 mille Personnes pour les pays**
- Pour l'année 2025 **la médian, le moyenne et l'écart-type sont 891, 5 700 et 17 500 mille Personnes pour les pays**
- Il y a une grande écarte entre les différents pays.
- Cette variable donne une indication sur l'evolution de la population avec une degrée d'education de BAC+

In [114…

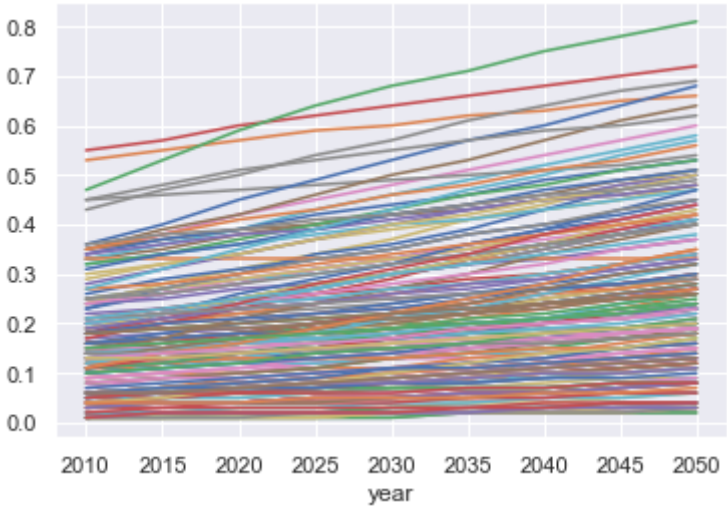
```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEAN = DATA_CLEAN.drop_duplicates()
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

PRJ.ATT.25UP.4.MF

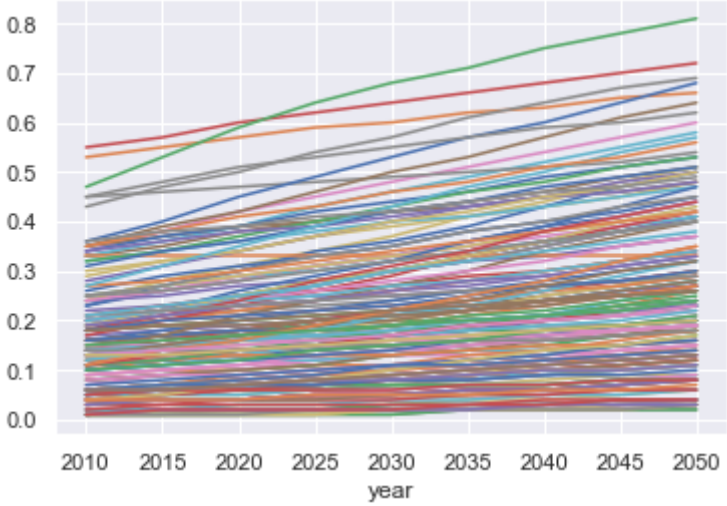
In [115…

```
ind = 'PRJ.ATT.25UP.4.MF'
initial_investigation(ind, is_projection_data = True )[0].T.info()
```

```
**************************************************
PRJ.ATT.25UP.4.MF
```




```
No data to plot.
We have data between 2010 and 2050.
The exact years are :
['2010' '2015' '2020' '2025' '2030' '2035' '2040' '2045' '2050']

We have data for the 166 countries.
The countries are :
['ABW' 'ALB' 'ARE' 'ARG' 'ARM' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL' 'BEN' 'BFA'
 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BOL' 'BRA' 'BTN' 'CAF' 'CAN'
 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB'
 'CYP' 'CZE' 'DEU' 'DNK' 'DOM' 'DZA' 'ECU' 'EGY' 'ESP' 'EST' 'ETH' 'FIN'
 'FRA' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GTM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KOR' 'KWT' 'LAO' 'LBN'
 'LBR' 'LCA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEX'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MUS' 'MWI' 'MYS' 'NAM' 'NCL'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'POL'
 'PRI' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAU' 'SDN' 'SEN'
 'SGP' 'SLE' 'SLV' 'SOM' 'SRB' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYR'
 'TCD' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'VCT' 'VEN' 'VNM' 'VUT' 'WSM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'AGO', 'BWA', 'SYC', 'DMA', 'CHI', 'LBY',
'MHL', 'PLW', 'MNP', 'CYM', 'TGO', 'BRB', 'NRU', 'KNA', 'CUW', 'AFG', 'ERI', 'SLB',
'GRD', 'BMU', 'PRK', 'SSD', 'FJI', 'TUV', 'SMR', 'GRL', 'MCO', 'MRT', 'LIE', 'FRO',
'DJI', 'LKA', 'ASM', 'OMN', 'VIR', 'FSM', 'KIR', 'ATG', 'BRN', 'TCA', 'IMN', 'VGB',
'XKX', 'SXM', 'PNG', 'UZB', 'YEM', 'GUM', 'AND', 'MAF', 'GIB'}
These regions have absolutely no data : {'MEA', 'LCN', 'SAS', 'ECS', 'EAS', 'NAC',
'SSF'}
**************************************************
**************************************************
PRJ.ATT.25UP.4.MF
```




```
No data to plot.
We have data between 2010 and 2050.
The exact years are :
['2010' '2015' '2020' '2025' '2030' '2035' '2040' '2045' '2050']

We have data for the 166 countries.
The countries are :
['ABW' 'ALB' 'ARE' 'ARG' 'ARM' 'AUS' 'AUT' 'AZE' 'BDI' 'BEL' 'BEN' 'BFA'
 'BGD' 'BGR' 'BHR' 'BHS' 'BIH' 'BLR' 'BLZ' 'BOL' 'BRA' 'BTN' 'CAF' 'CAN'
 'CHE' 'CHL' 'CHN' 'CIV' 'CMR' 'COD' 'COG' 'COL' 'COM' 'CPV' 'CRI' 'CUB'
 'CYP' 'CZE' 'DEU' 'DNK' 'DOM' 'DZA' 'ECU' 'EGY' 'ESP' 'EST' 'ETH' 'FIN'
 'FRA' 'GAB' 'GBR' 'GEO' 'GHA' 'GIN' 'GMB' 'GNB' 'GNQ' 'GRC' 'GTM' 'GUY'
 'HKG' 'HND' 'HRV' 'HTI' 'HUN' 'IDN' 'IND' 'IRL' 'IRN' 'IRQ' 'ISL' 'ISR'
 'ITA' 'JAM' 'JOR' 'JPN' 'KAZ' 'KEN' 'KGZ' 'KHM' 'KOR' 'KWT' 'LAO' 'LBN'
 'LBR' 'LCA' 'LSO' 'LTU' 'LUX' 'LVA' 'MAC' 'MAR' 'MDA' 'MDG' 'MDV' 'MEX'
 'MKD' 'MLI' 'MLT' 'MMR' 'MNE' 'MNG' 'MOZ' 'MUS' 'MWI' 'MYS' 'NAM' 'NCL'
 'NER' 'NGA' 'NIC' 'NLD' 'NOR' 'NPL' 'NZL' 'PAK' 'PAN' 'PER' 'PHL' 'POL'
 'PRI' 'PRT' 'PRY' 'PSE' 'PYF' 'QAT' 'ROU' 'RUS' 'RWA' 'SAU' 'SDN' 'SEN'
 'SGP' 'SLE' 'SLV' 'SOM' 'SRB' 'STP' 'SUR' 'SVK' 'SVN' 'SWE' 'SWZ' 'SYR'
 'TCD' 'THA' 'TJK' 'TKM' 'TLS' 'TON' 'TTO' 'TUN' 'TUR' 'TZA' 'UGA' 'UKR'
 'URY' 'USA' 'VCT' 'VEN' 'VNM' 'VUT' 'WSM' 'ZAF' 'ZMB' 'ZWE']

These countries have absolutely no data : {'AGO', 'BWA', 'SYC', 'DMA', 'CHI', 'LBY',
'MHL', 'PLW', 'MNP', 'CYM', 'TGO', 'BRB', 'NRU', 'KNA', 'CUW', 'AFG', 'ERI', 'SLB',
'GRD', 'BMU', 'PRK', 'SSD', 'FJI', 'TUV', 'SMR', 'GRL', 'MCO', 'MRT', 'LIE', 'FRO',
'DJI', 'LKA', 'ASM', 'OMN', 'VIR', 'FSM', 'KIR', 'ATG', 'BRN', 'TCA', 'IMN', 'VGB',
'XKX', 'SXM', 'PNG', 'UZB', 'YEM', 'GUM', 'AND', 'MAF', 'GIB'}
These regions have absolutely no data : {'MEA', 'LCN', 'SAS', 'ECS', 'EAS', 'NAC',
'SSF'}
**************************************************
Completeness for the years 2010-2050: 1.0

year         2010        2015        2020        2025        2030        2035  \
count  166.000000  166.000000  166.000000  166.000000  166.000000  166.000000
mean     0.150301    0.164157    0.177892    0.191627    0.205542    0.219819
std      0.113802    0.121332    0.129483    0.137291    0.144333    0.151515
min      0.010000    0.010000    0.010000    0.010000    0.010000    0.020000
25%      0.060000    0.060000    0.070000    0.080000    0.090000    0.100000
50%      0.130000    0.140000    0.160000    0.170000    0.175000    0.190000
75%      0.200000    0.220000    0.247500    0.270000    0.300000    0.320000
max      0.550000    0.570000    0.600000    0.640000    0.680000    0.710000

year         2040        2045        2050
count  166.000000  166.000000  166.000000
mean     0.234217    0.249699    0.265602
std      0.159150    0.166006    0.172877
min      0.020000    0.020000    0.020000
25%      0.110000    0.120000    0.130000
50%      0.205000    0.220000    0.235000
75%      0.347500    0.377500    0.400000
max      0.750000    0.780000    0.810000
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 166 entries, ABW to ZWE
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2010    166 non-null    float64
 1   2015    166 non-null    float64
 2   2020    166 non-null    float64
 3   2025    166 non-null    float64
 4   2030    166 non-null    float64
 5   2035    166 non-null    float64
 6   2040    166 non-null    float64
 7   2045    166 non-null    float64
 8   2050    166 non-null    float64
dtypes: float64(9)
memory usage: 13.0+ KB
```

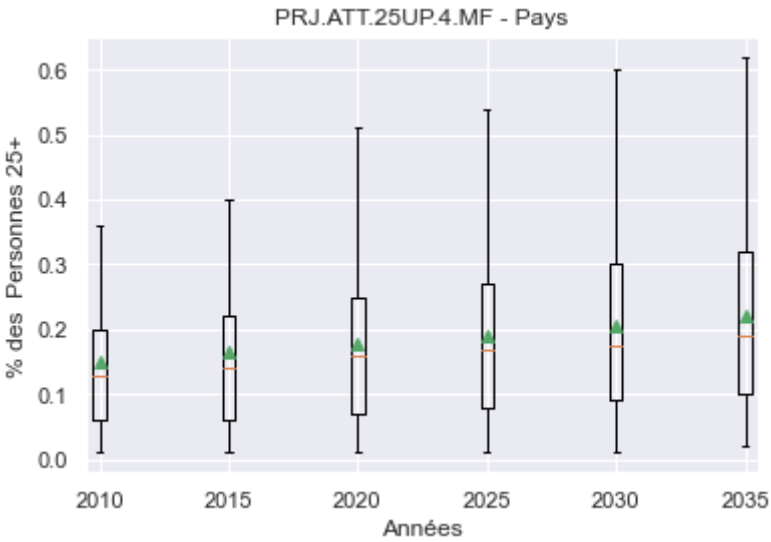
Résumé: PRJ.ATT.25UP.4.MF

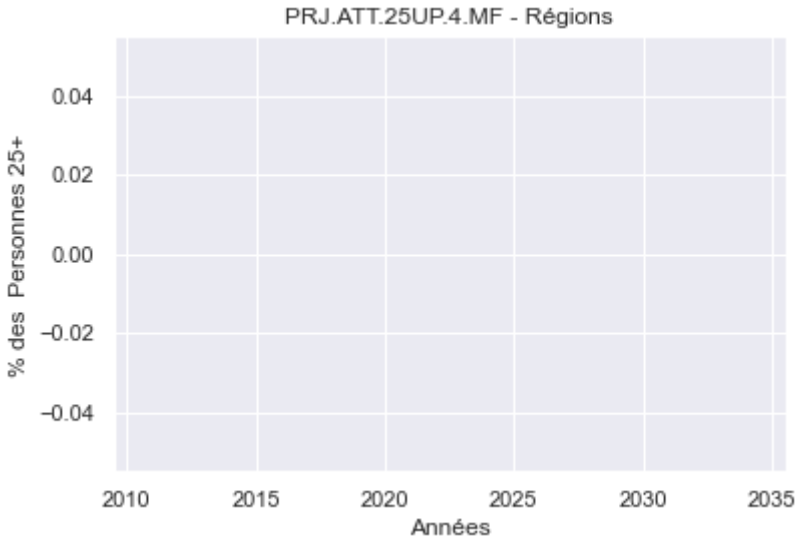
- Pour 166 pays, il y a de la data pour les années 2010-2050 (en intervales de 5 ans).
  - 51 des 217 pays n'ont pas de data pour cette variable.
- Il n'y a pas de la data pour les régions.
- **La data pour cette variable est complète à 100% (par construction)**

In [116…

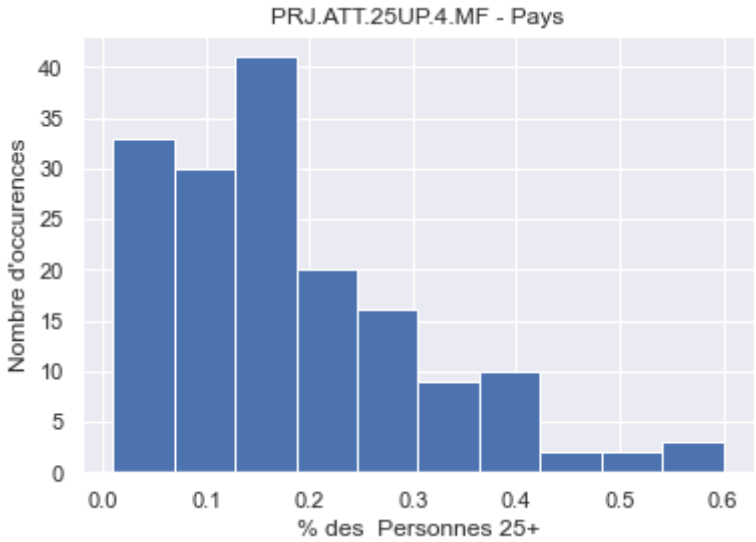
```
years = [str(y) for y in range(2010, 2036, 5)]
data = make_2D_map(DATA, ind, ['indicator', 'year'], 'country').drop(columns = ['ind
data_clean = drop_countries(data, 3)
plot_boxplot(ind, data_clean, unit = "% des  Personnes 25+", years = years, showmean
print_info_year('2020', data_clean, unit = "% des  Personnes 25+", ind = ind)
print_info_year('2025', data_clean, unit = "% des  Personnes 25+", ind = ind)
```

```
The following countries were dropped :
['AFG', 'AGO', 'AND', 'ASM', 'ATG', 'BMU', 'BRB', 'BRN', 'BWA', 'CHI', 'CUW', 'CYM',
'DJI', 'DMA', 'EAS', 'ECS', 'ERI', 'FJI', 'FRO', 'FSM', 'GIB', 'GRD', 'GRL', 'GUM',
'IMN', 'KIR', 'KNA', 'LBY', 'LCN', 'LIE', 'LKA', 'MAF', 'MCO', 'MEA', 'MHL', 'MNP',
'MRT', 'NAC', 'NRU', 'OMN', 'PLW', 'PNG', 'PRK', 'SAS', 'SLB', 'SMR', 'SSD', 'SSF',
'SXM', 'SYC', 'TCA', 'TGO', 'TUV', 'UZB', 'VGB', 'VIR', 'XKX', 'YEM']
```

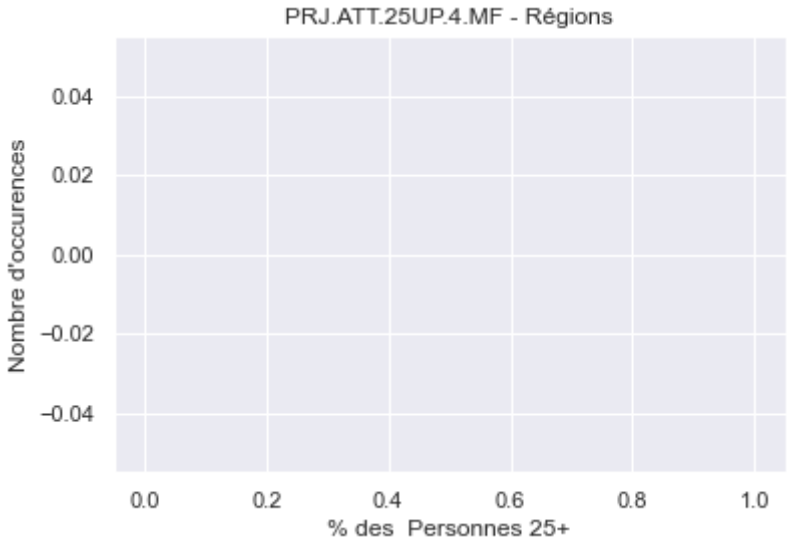





```
2020
Pays
Median:  0.16
Mean:  0.17789156626506025
Std:  0.1294829355363389
```

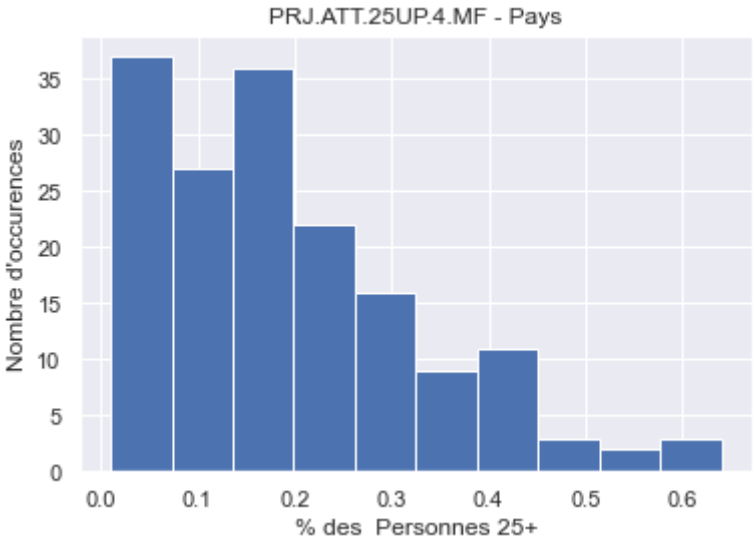



```
Régions
Median:  nan
Mean:  nan
Std:  nan
```

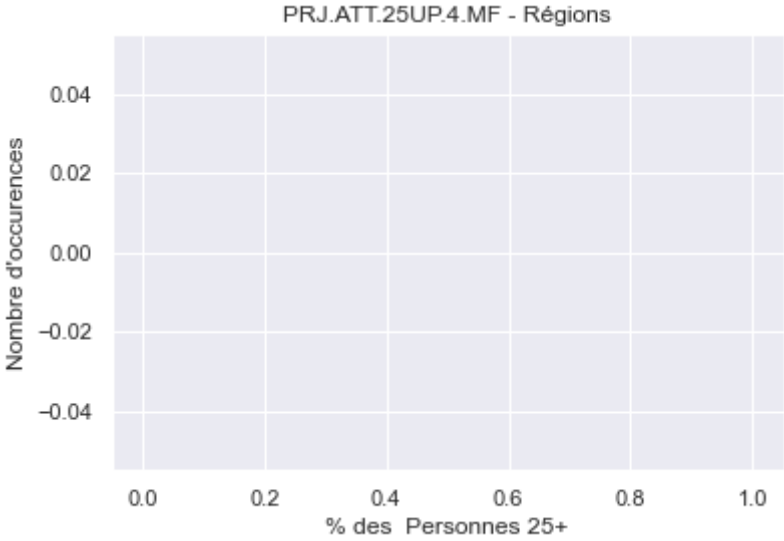



```
2025
Pays
Median:  0.17
```

```
Mean:  0.1916265060240964
Std:  0.13729112715759528
```




```
Régions
Median:  nan
Mean:  nan
Std:  nan
```



Résumé: PRJ.ATT.25UP.4.MF

- Pour l'année 2020 **la médian, le moyenne et l'écart-type sont 16.0%, 17.8% et 12.9% des Personnes 25+ pour les pays**
- Pour l'année 2025 **la médian, le moyenne et l'écart-type sont 17.0%, 19.2% et 13.7% des Personnes pour les pays**
- Cette variable donne une indication sur le niveau d'education d'un pays - le percentages des personnes 25+ avec une degrée d'education de BAC+ (post secondary)

In [117…

```
#save the cleaned data
data_clean['indicator'] = ind
DATA_CLEAN = DATA_CLEAN.append(deconstruct_df(data_clean.copy().reset_index().rename
DATA_CLEAN = DATA_CLEAN.drop_duplicates()
DATA_CLEANED = DATA.set_index(INDEX_NAMES).fillna(value = DATA_CLEAN.set_index(INDEX
```

4.3 Re-filtrer les pays

Choisir les pays qui ont assez de données et qui ont une population plus large que 500 000 inhabitants.

In [118…

```
my_indicators = ['IT.NET.USER.P2',
                 'SP.TER.TOTL.IN', 'SE.TER.ENRL', 'SE.TER.ENRR',
                 'SP.SEC.UTOT.IN', 'UIS.E.3', 'SE.SEC.ENRR.UP',
                 'NY.GDP.PCAP.CD',
                 'SP.POP.TOTL']
my_indicators_proj = ['PRJ.POP.ALL.4.MF', 'PRJ.ATT.25UP.4.MF']

my_years = [str(y) for y in range(2010, 2017)]
my_years_proj = [str(y) for y in range(2010,2036,5)]

select =  [i and j for i, j in zip(DATA.indicator.isin(my_indicators),DATA.year.isin
select_proj = [i and j for i, j in zip(DATA.indicator.isin(my_indicators_proj),DATA.

DATA_MIN = DATA[select]
DATA_MIN_CLEANED = DATA_CLEANED[select]

DATA_MIN_LAST = DATA_MIN.set_index(['country', 'indicator']).sort_values(by=['year']
DATA_MIN_LAST = DATA_MIN_LAST[[not d for d in DATA_MIN_LAST.index.duplicated(keep='l
DATA_MIN_PROJ = DATA[select_proj].dropna()
```

In [119…

```
def indicators_per_country(my_indicators, frac = 0.5, output = True):
    data = DATA_MIN_LAST[DATA_MIN_LAST.indicator.isin(my_indicators)]
    num_ind_cc = {}
    for cc in data.country.unique():
        num_ind_cc[cc] = data[data.country == cc].value.count().sum()

    if output:
        print("Number of countries with at least one indicator:",len(data.country.un
        cc_complete = []
        cc_morefrac = []
        cc_lessfrac = []
        num_ind = len(my_indicators)
        for cc, n in num_ind_cc.items() :
            if n == num_ind :
                cc_complete.append(cc)
            elif n >= num_ind*frac :
                cc_morefrac.append(cc)
            else :
                cc_lessfrac.append(cc)
        print("For these countries we have data for all the indicators:\n", cc_compl
        print(f"For these countries we have more than {frac*100}% of the indicators:
        print(f"For these countries we have less than {frac*100}% of the indicators:

    return data, num_ind_cc


def fill_in_education_indicator(my_indicators, output = False) :
    '''function used only in two very specific cases'''
    data, num_ind_cc = indicators_per_country(my_indicators, frac = 0.5, output = ou
    cc_list1 = [cc for cc, n in num_ind_cc.items() if n == len(my_indicators)]
    data_frame = pd.DataFrame()
    for cc in cc_list1 :
        #print(cc, num_ind_cc[cc], sep=": ")
        country_indicators = data[data.country == cc].indicator.values
        #print("The indicators available are", country_indicators, sep=":\n")
        data_row = {}
        for ind in my_indicators:
            if ind not in country_indicators:
                data_row['indicator'] = ind
```

```
                data_row['country'] = cc
                data_row['year'] = 'XXXX'
                if ind == my_indicators[1]:
                    val = (data[data.country == cc][data.indicator == my_indicators[
                           data[data.country == cc][data.indicator == my_indicators[
                elif ind == my_indicators[2]:
                    val = (data[data.country == cc][data.indicator == my_indicators[
                           data[data.country == cc][data.indicator == my_indicators[
                elif ind == my_indicators[0]:
                    val = (data[data.country == cc][data.indicator == my_indicators[
                           data[data.country == cc][data.indicator == my_indicators[
                data_row['value'] = val
                data_frame = data_frame.append(data_row, ignore_index = True)
    return DATA_MIN_LAST.append(data_frame, ignore_index = True)
```

In [120…

```
#These indicators are linked, if only one is missing use the other two to fill in th
my_indicators = ['SP.TER.TOTL.IN', 'SE.TER.ENRL', 'SE.TER.ENRR']
DATA_MIN_LAST = fill_in_education_indicator(my_indicators)
my_indicators = ['SP.SEC.UTOT.IN', 'UIS.E.3', 'SE.SEC.ENRR.UP']
DATA_MIN_LAST = fill_in_education_indicator(my_indicators)
```

In [121…

```
my_indicators = ['IT.NET.USER.P2', 'SE.TER.ENRR','SE.TER.ENRL','NY.GDP.PCAP.CD', 'SP
data, num_ind_cc = indicators_per_country(my_indicators, output = False)
my_countries = [cc for cc, n in num_ind_cc.items() if n == len(my_indicators) and cc
my_regions = [cc for cc, n in num_ind_cc.items() if n == len(my_indicators) and cc i
data = data[data.country.isin(my_countries)]
data = data[data.indicator == 'SP.POP.TOTL']
my_countries = data[data.value >= 500000].country.unique()
print(len(my_countries))
print(len(my_regions))
```

```
143
7
```

4.4 Le Score

- Définir un score sur une échelle de 0 et 10 pour les variables pertiantes.
- Faire un score total et classer les pays.

Le Score d'Attractivité

- L'usage d'internet et le niveau d'education d'un pays.
  - Personnes qui ont utilisé l'internet par 100 inhabitants (**IT.NET.USER.P2**). Par construction cette variable est entre 0 et 100.
  - Etudiants inscrits dans l'education tertaire par la population avec la tranche d'âge correspondent (**SE.TER.ENRR**). Cette variable peut être plus grande que 100, parce qu'il peut avoir des personnes en dehors de la tranche d'âge inscrites. Cependent la valeur maximum va être "proche" de 100, i.e. avoir la même ordre de magnitude.
- Ces deux variables sont normalisées.
- Utiliser un score linéaire entre 0 et 100.
  - Pour les valeurs plus grandes que 100 on va donner un 10. (pour SE.TER.ENRR)

Le Score de Marché

- Le nombre de potentiels clients et leur pouvoir d'âchat

  - L'nombre de personnes inscrites dans l'education tertiaire (**SE.TER.ENRL**). Cette variable n'est pas normalisées et varie beaucoup entre les différents pays.
  - La PIB per inhabitant (**NY.GDP.PCAP.CD**). Cette variable est normalisée par inhabitant, mais peut prendre n'importe quel valeur et varie beaucoup entre les differents pays.
- Utiliser un score exponentiel

4.4.1 Quels fonctions

In [122…

```
def score_lin(data_series, maxvalue=None, minvalue=None, name = None):
    data_values = data_series.values

    if maxvalue == None:
        maxvalue = max(data_values)
    if minvalue == None:
        minvalue = min(data_values)

    vals = [v if isinstance(v,(float, int)) and v >= minvalue else minvalue for v in
    vals = [v if v <= maxvalue else maxvalue for v in vals]
    scale = 10/(maxvalue-minvalue)
    scores = [(v-minvalue)*scale for v in vals]
    #for v in vals :
     #   print(v*scale, (v-minvalue)*scale, (((v-minvalue)*scale) // 10) )


    #scores = [(v-minvalue)*scale
    #          if 0 <= (v-minvalue)*scale <= 10
    #          else 10*( (((v-minvalue)*scale) // 10) >= 1 )
    #          for v in vals]

    data_scores = pd.Series(scores, index = data_series.index, name = name)
    return data_scores

def score_pow(data_series, maxvalue=None, minvalue=None, name = None):
    data_values = data_series.values

    if maxvalue == None :
        maxvalue = max(data_values)
    if minvalue == None :
        minvalue = maxvalue/1024
        base = 2
    else :
        base = (maxvalue/minvalue)**(0.1)
    norm = 10 - math.log(maxvalue, base)

    vals = [v if v >= minvalue and isinstance(v,(float, int)) else minvalue for v in
    vals = [v if v <= maxvalue else maxvalue for v in vals]
    scores = [math.log(v,base) + norm for v in vals]

    #scores = [math.log(v, base) + norm
    #          if 0 <= math.log(v,base) + norm <= 10
    #         else 10*( ((math.log(v, base) + norm) // 10) >= 1 )
    #         for v in vals]

    data_scores = pd.Series(scores, index = data_series.index, name = name)
    return data_scores
```

In [123…

```
def output_data(years, countries) :
    data_df = DATA_MIN_LAST
    data_df['year'] = 'LAST'
    data_df = data_df.set_index(INDEX_NAMES)
    data_df = pd.concat([data_df, DATA_MIN_CLEANED.set_index(INDEX_NAMES)]).sort_val
```

```
    select = [i and j for i, j in zip(data_df.country.isin(countries), data_df.year.
    data_df = data_df[select]
    return data_df

def output_data_scores(data_df, score_dict):
    data_all_scores = pd.DataFrame()
    groups = []
    suffix = '_Score'

    for ind,p in score_dict.items() :
        data_series = data_df[data_df.indicator == ind].drop(columns = ('indicator')

        p0 = p[1:3]
        name0 = ind + suffix
        score_func = score_lin if p[0] else score_pow

        data_series_score = data_series.copy()
        for y in data_df.year.unique():
            df_tmp = pd.DataFrame(score_func(data_series.xs(y, level='year').value,
            df_tmp['year'] = y
            df_tmp = df_tmp.set_index('year', append = True)
            data_series_score.update(df_tmp)
        data_series_score = data_series_score.rename(columns = {'value' : name0})
        data_all_scores = pd.concat([data_all_scores, data_series_score], axis = 1)

        groups.append(p[3])

    groups = list(set(groups))
    for g in groups :
        ind_in_group = [ind + suffix for ind, p in score_dict.items() if p[3] == g]
        data_series = pd.DataFrame(data_all_scores.loc[:,ind_in_group].T.mean(), col
        data_all_scores = pd.concat([data_all_scores, data_series], axis = 1)

    score_groups = [g + suffix for g in groups]
    data_series = pd.DataFrame(data_all_scores.loc[:,score_groups].T.mean(), columns
    data_all_scores = pd.concat([data_all_scores, data_series], axis = 1)

    return data_all_scores
```

In [124…

```
def make_2D_dataarray(data_df, x, x_column):
    ind = x
    data_df = data_df[data_df[x_column]== ind].drop(columns = [x_column])
    data_out = pd.DataFrame()
    y_column = INDEX_NAMES[(INDEX_NAMES.index(x_column) + 1) % 3]
    z_column = INDEX_NAMES[(INDEX_NAMES.index(x_column) + 2) % 3]

    for z in data_df[z_column].unique() :
        data_series = data_df[data_df[z_column] == z]
        data_tmp = pd.concat([data_series[data_series[y_column] == y].set_index(z_co
                              for y in data_series[y_column].unique()], axis = 1)
        data_out = pd.concat([data_out, data_tmp])

    return data_out
```

In [125…

```
def plot_boxplot2(ind, data_df, unit = "", savefigures = False, **kwags) :

    for countries, titlename in zip([COUNTRIES, REGIONS], ["Pays","Régions"]) :

        data_plot = data_df[data_df.country.isin(countries)]
        data_plot = make_2D_dataarray(data_plot, ind, 'indicator').T

        fig, ax = plt.subplots()
```

```
        data_years = data_plot.columns.values
        ax.boxplot(data_plot.values, **kwags)
        #ax.plot(data_years, data_plot.mean().values,'o', alpha=0.5, color = 'black'
        ax.set_xlabel("Années")
        ax.set_xticklabels(data_years)
        ax.set_ylabel(unit)
        title = " - ".join((ind, titlename))
        ax.set_title(title)
        if savefigures :
            plt.savefig(os.getcwd() + '\\Figures\\' + title.replace('.','_') + '_Box
        plt.show()
```

In [126…

```
def output_info(ind, data_df, saveinfo = False):
    data_all = pd.DataFrame()
    for countries, titlename in zip([COUNTRIES, REGIONS], ["Countries","Regions"]) :
        #print(ind, titlename, sep = ' - ')
        data_out = make_2D_dataarray(data_df[data_df.country.isin(countries)], ind,
        data_out = pd.concat([data_out.median(),data_out.mean(), data_out.std()], ax
                             keys = ['Median - '+titlename, 'Mean - '+titlename, 'St
        data_all = pd.concat([data_all, data_out], axis = 1)

        if saveinfo :
            data_all.to_csv(os.getcwd() + '\\Output\\' + ind.replace('.','_'))


    return (data_all)
```

In [127…

```
def plot_histogramm(ind, data_df, unit = '', savefigures = False, **kwags):

        for y in data_df.year.unique() :
            data_tmp = data_df[data_df.year == y]

            for countries, titlename in zip([COUNTRIES, REGIONS], ["Pays","Régions"]

                data_plot = data_tmp[data_tmp.country.isin(countries)]
                data_plot = make_2D_dataarray(data_plot, ind, 'indicator').T

                fig, ax = plt.subplots()
                data_years = data_plot.columns.values
                ax.hist(data_plot.values, **kwags)

                ax.set_xlabel(unit)
                ax.set_ylabel("Fréquence")
                title = " - ".join(("_".join((ind, y)), titlename))
                ax.set_title(title)
                if savefigures :
                    plt.savefig(os.getcwd() + '\\Figures\\' + title.replace('.','_')
                plt.show()
```

In [ ]:

Les indicateurs résumées

- Donner des informations statistiques sur les différents variables
- Illustrer le jeu de données avec des graphs
- Sauvegarder les informations (si souhaité)

In [128…

```
years = ['LAST','2014','2016']
```

```
countries = list(my_countries) + list(my_regions)

data_df = output_data(years = ['LAST','2014','2016'], countries = list(my_countries)

#regarder soit les minimum / maximum valeurs pour déterminer les limites pour le sco
data_tmp = data_df[data_df.indicator == 'NY.GDP.PCAP.CD'].sort_values('value', ascen
data_tmp[data_tmp.country.isin(my_countries)].head(20)
```

Out[128…

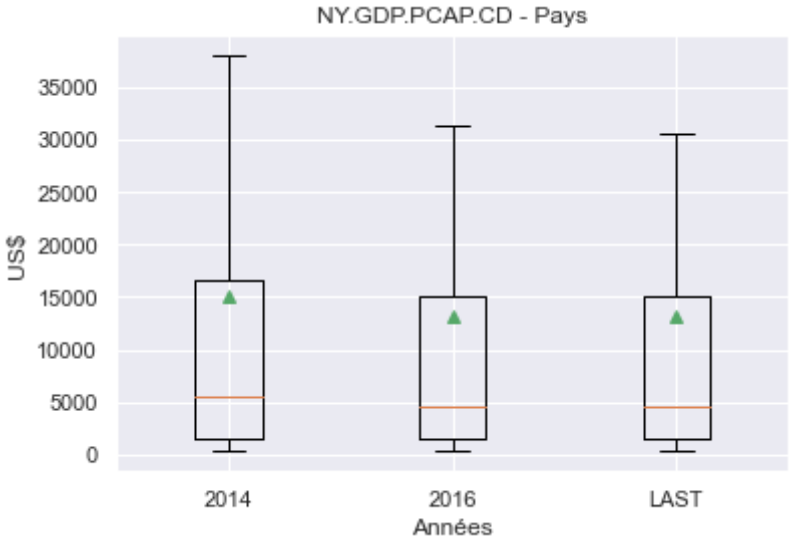
	indicator	country	year	value
2731	NY.GDP.PCAP.CD	LUX	2014	119225.380023
2734	NY.GDP.PCAP.CD	LUX	LAST	100573.139978
2733	NY.GDP.PCAP.CD	LUX	2016	100573.139978
2977	NY.GDP.PCAP.CD	NOR	2014	97199.919096
2747	NY.GDP.PCAP.CD	MAC	2014	94004.389829
3118	NY.GDP.PCAP.CD	QAT	2014	86852.710189
2047	NY.GDP.PCAP.CD	CHE	2014	86605.563376
2050	NY.GDP.PCAP.CD	CHE	LAST	79890.524005
2049	NY.GDP.PCAP.CD	CHE	2016	79890.524005
2750	NY.GDP.PCAP.CD	MAC	LAST	73186.960143
2749	NY.GDP.PCAP.CD	MAC	2016	73186.960143
2980	NY.GDP.PCAP.CD	NOR	LAST	70911.757159
2979	NY.GDP.PCAP.CD	NOR	2016	70911.757159
2517	NY.GDP.PCAP.CD	IRL	2016	63861.921982
2518	NY.GDP.PCAP.CD	IRL	LAST	63861.921982
2204	NY.GDP.PCAP.CD	DNK	2014	62548.985017
1863	NY.GDP.PCAP.CD	AUS	2014	62214.609121
3120	NY.GDP.PCAP.CD	QAT	2016	59324.338773
3121	NY.GDP.PCAP.CD	QAT	LAST	59324.338773
3286	NY.GDP.PCAP.CD	SWE	2014	59180.198978

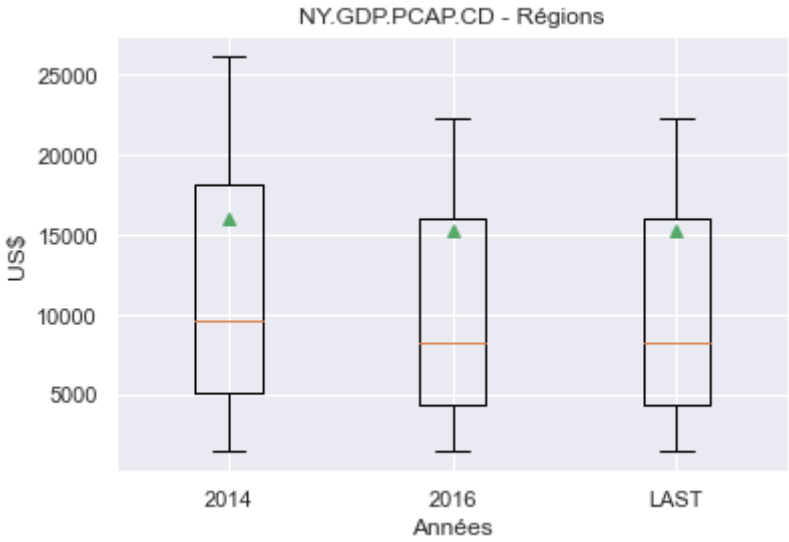
In [129…

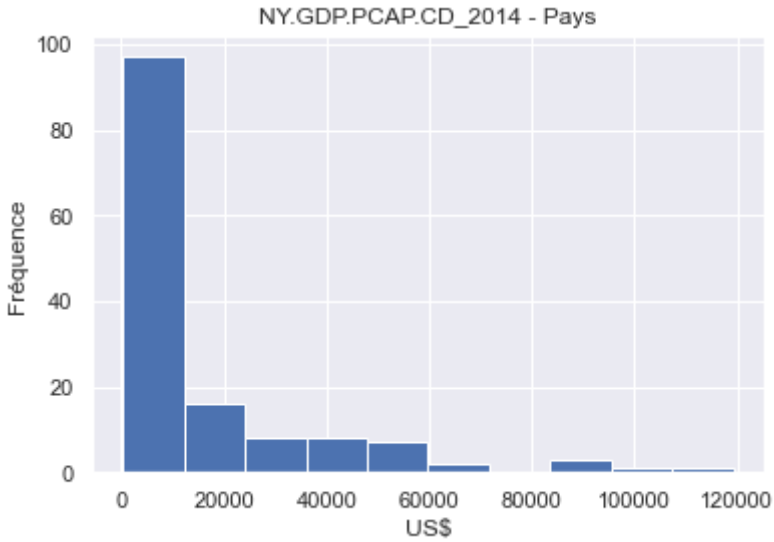
```
score_dict = {'NY.GDP.PCAP.CD' : (False, 5e4, 500, 'Market', 'US$'), 'SE.TER.ENRL' :
              'IT.NET.USER.P2' : (True, 100, 0, 'Attr', '%'), 'SE.TER.ENRR' : (True,

#afficher (et si souhaité sauvgarder) les informations statistiques sur les indicate
for ind in score_dict :
    print(50*'*')
    print(ind)
    print()
    print(output_info(ind, data_df))
    plot_boxplot2(ind, data_df, unit = score_dict[ind][4],showmeans = True, showflie
    plot_histogramm(ind, data_df, unit = score_dict[ind][4])
    print(50*'*')
```

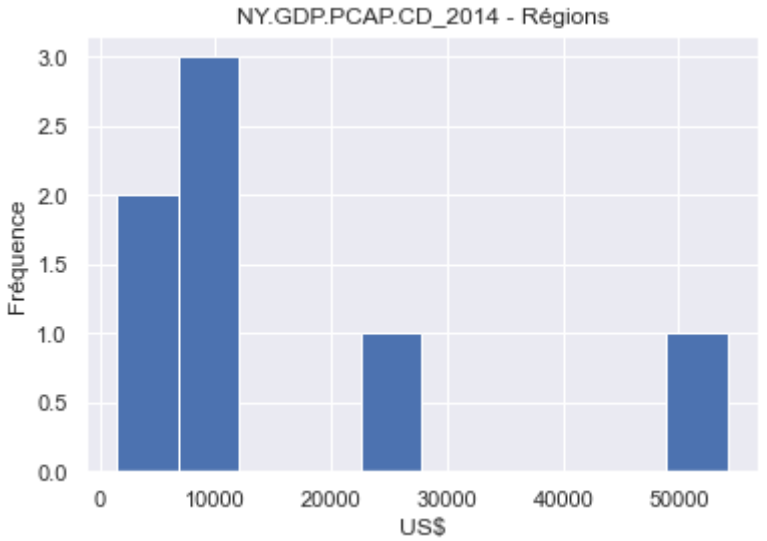
```
**************************************************
NY.GDP.PCAP.CD

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
year
```

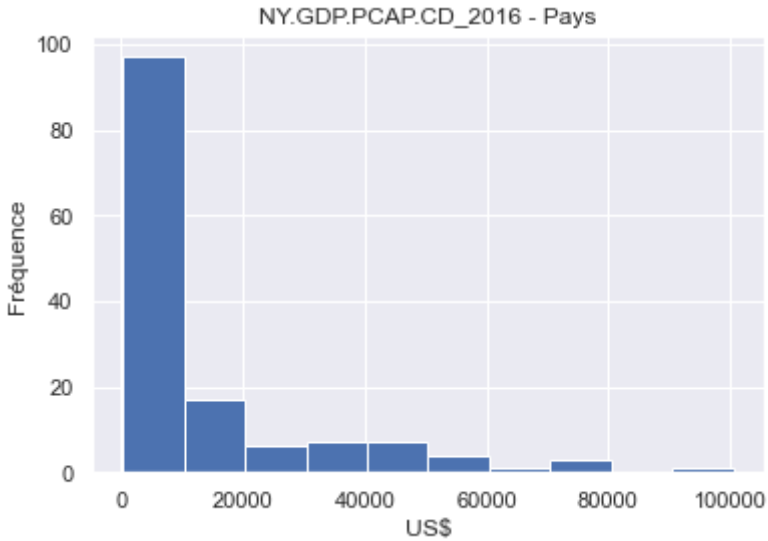
```
2014         5466.425778      15066.700239      22230.759086       9665.056079
2016         4529.139412      13224.698980      18935.682184       8311.370178
LAST         4529.139412      13197.756712      18922.843114       8311.370178

      Mean - Regions  Std - Regions
year
2014    16009.026713   18724.941880
2016    15244.838505   19296.324162
LAST    15244.838505   19296.324162
```

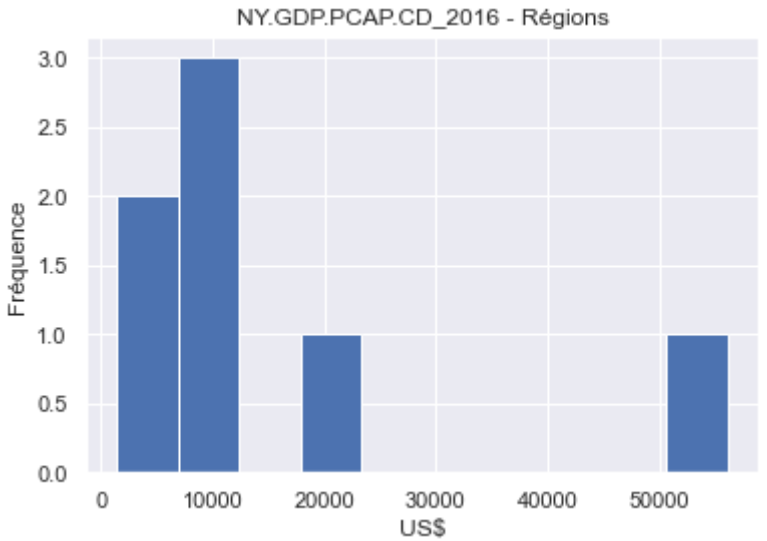


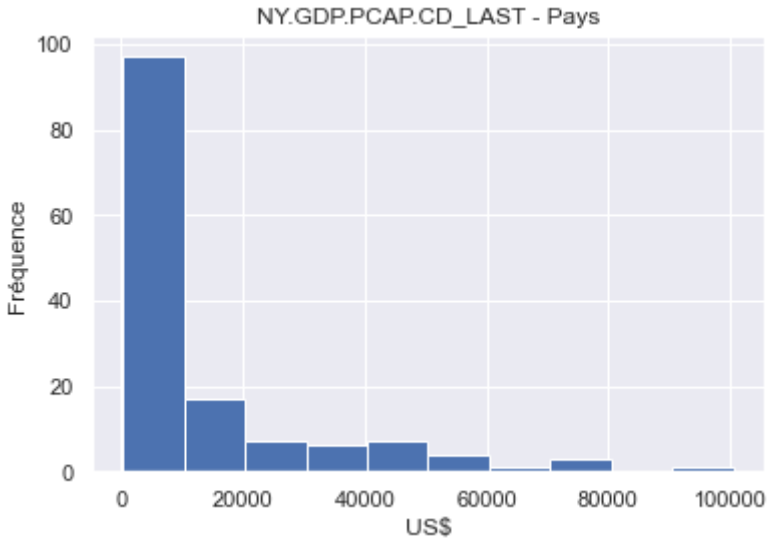


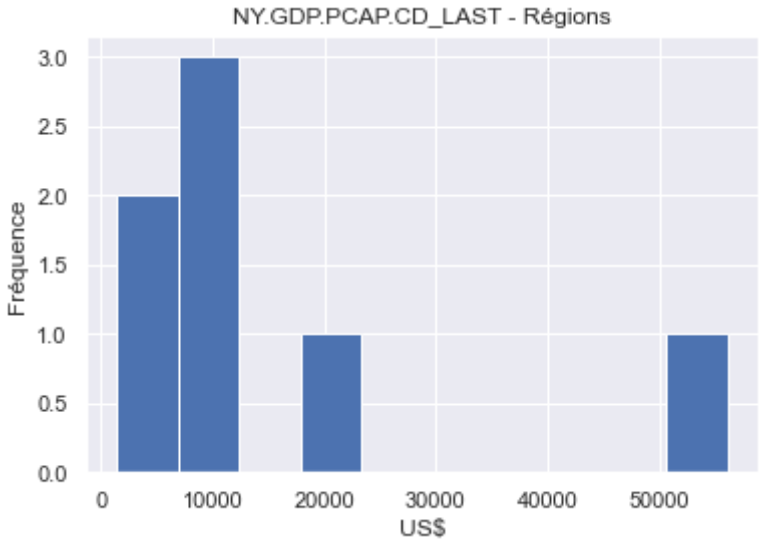



NY.GDP.PCAP.CD_2014 - Régions
Fréquence
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
10000
20000
30000
40000
50000
US$

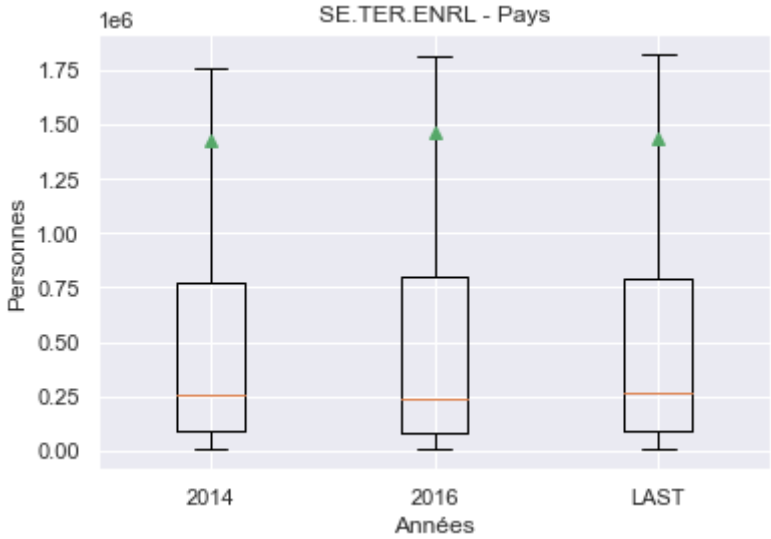

NY.GDP.PCAP.CD_2016 - Pays
Fréquence
100
80
60
40
20
0
0
20000
40000
60000
80000
100000
US$

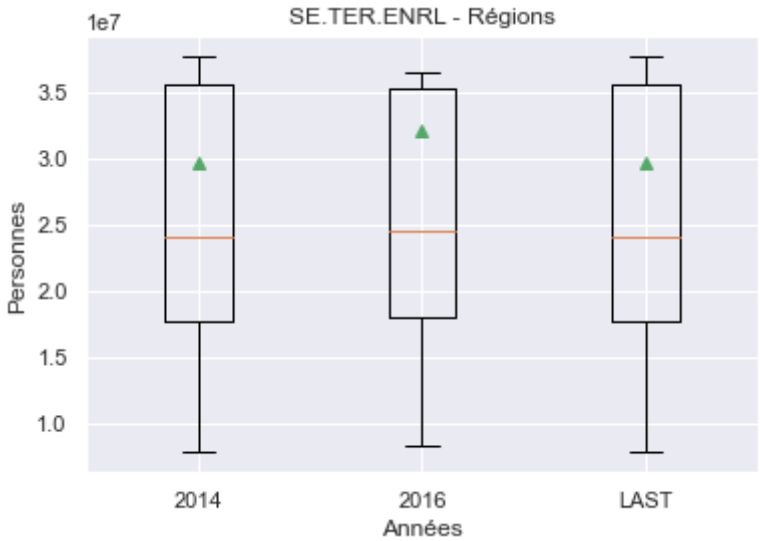

NY.GDP.PCAP.CD_2016 - Régions
Fréquence
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
10000
20000
30000
40000
50000
US$

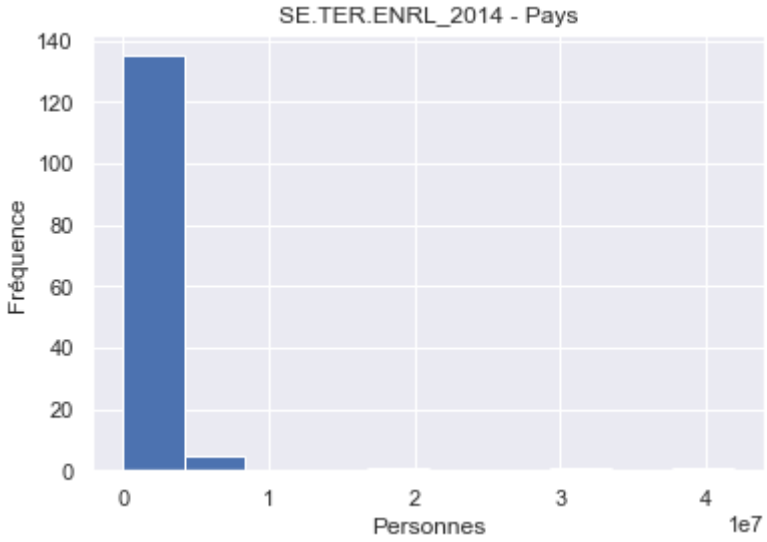


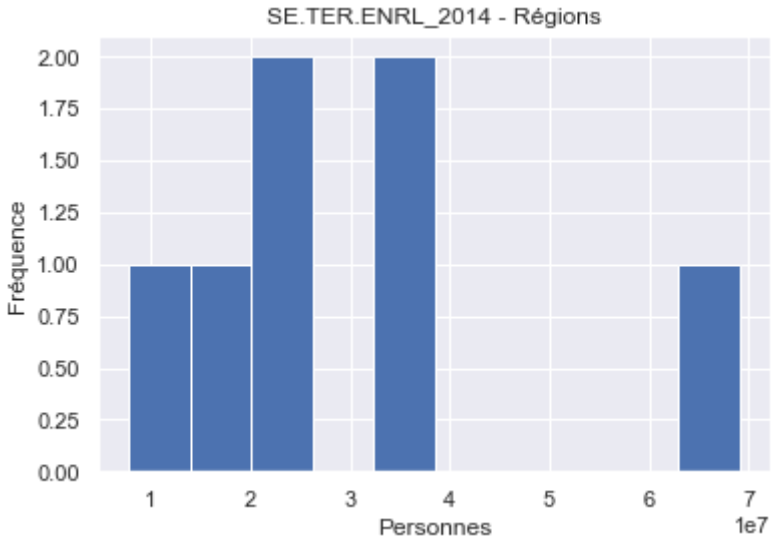


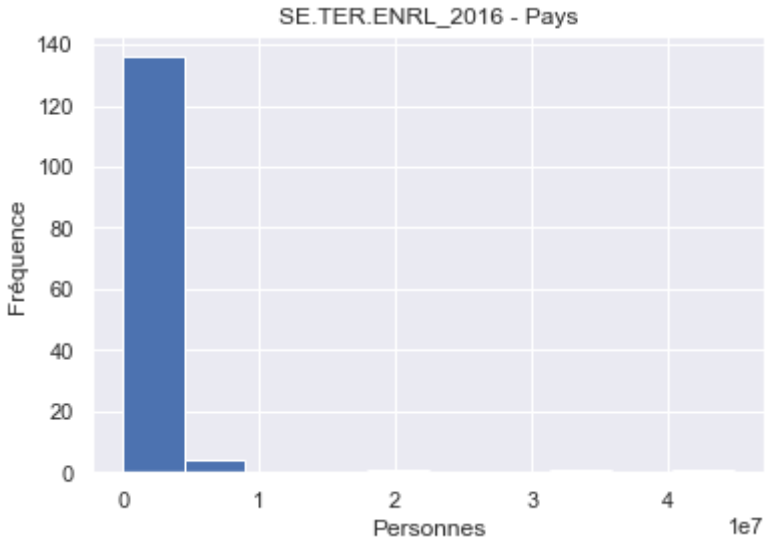

```
**************************************************
**************************************************
SE.TER.ENRL

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
year
2014            253602.0      1.425259e+06     4.695593e+06        24087544.0
2016            239260.0      1.460406e+06     5.018008e+06        24503504.0
LAST            261413.0      1.435061e+06     4.851943e+06        24087544.0

      Mean - Regions  Std - Regions
year
2014    2.964508e+07   2.021267e+07
2016    3.210922e+07   2.550535e+07
LAST    2.964508e+07   2.021267e+07
```

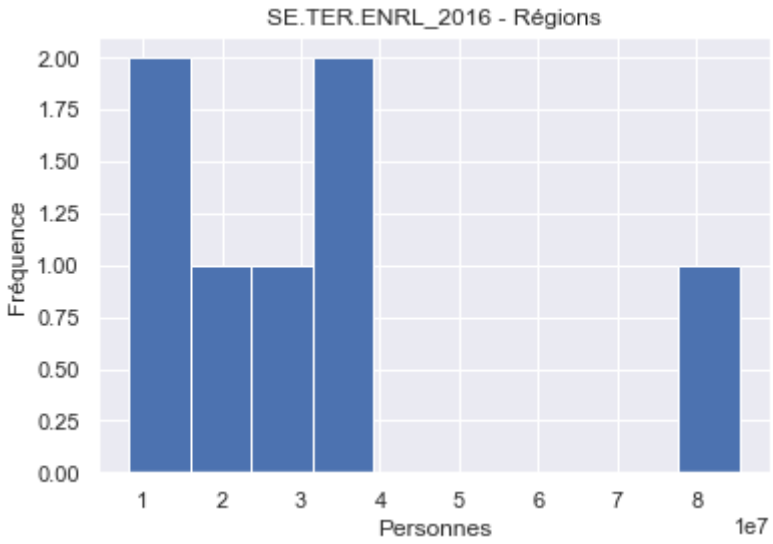

SE.TER.ENRL - Pays
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Personnes
2014
2016
LAST
Années

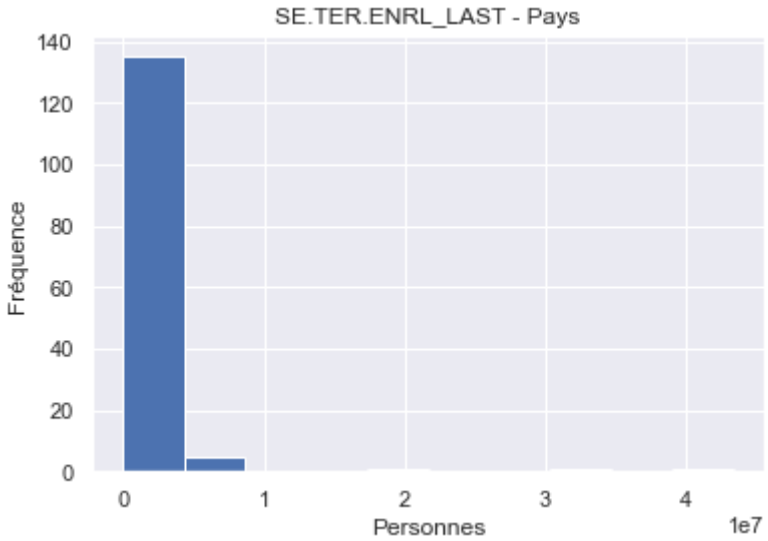

SE.TER.ENRL - Régions
1e7
3.5
3.0
2.5
2.0
1.5
1.0
Personnes
2014
2016
LAST
Années

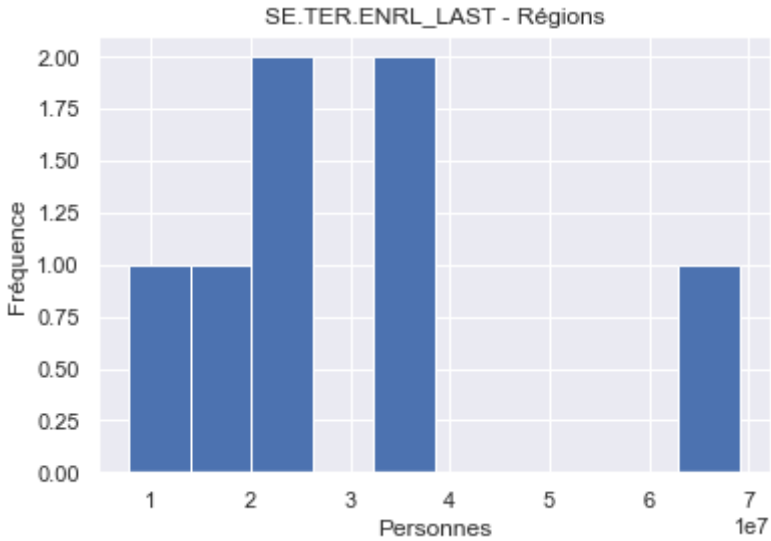

SE.TER.ENRL_2014 - Pays
140
120
100
80
60
40
20
0
Fréquence
0
1
2
3
4
1e7
Personnes


SE.TER.ENRL_2014 - Régions
Fréquence
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
Personnes
1e7

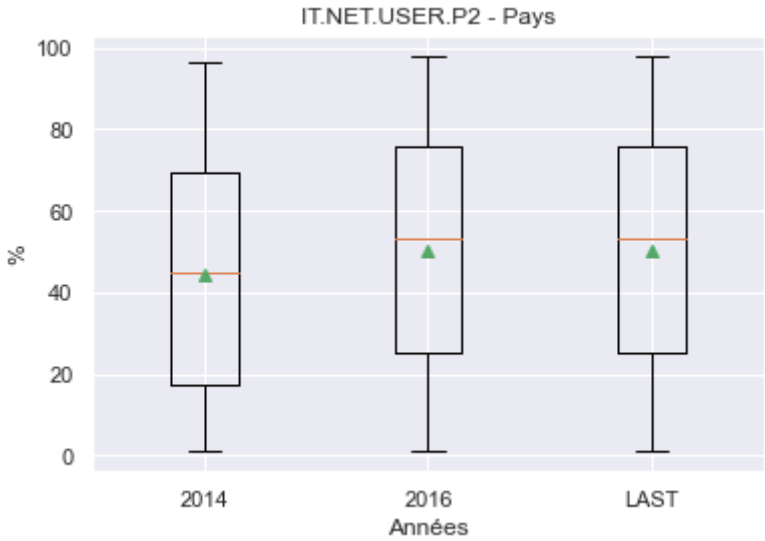

SE.TER.ENRL_2016 - Pays
Fréquence
140
120
100
80
60
40
20
0
0
1
2
3
4
Personnes
1e7

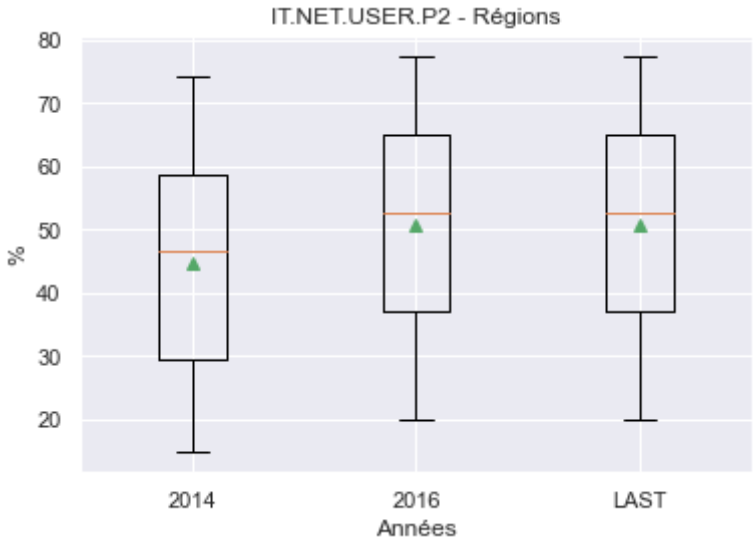

SE.TER.ENRL_2016 - Régions
Fréquence
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Personnes
1e7

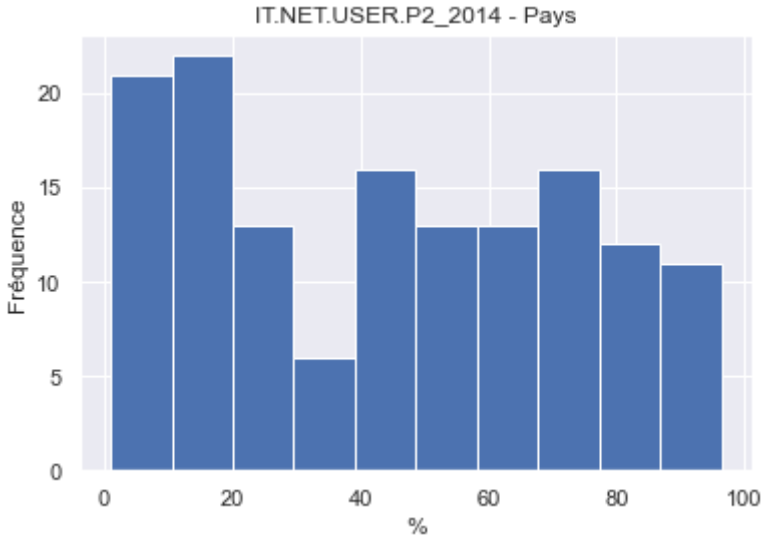


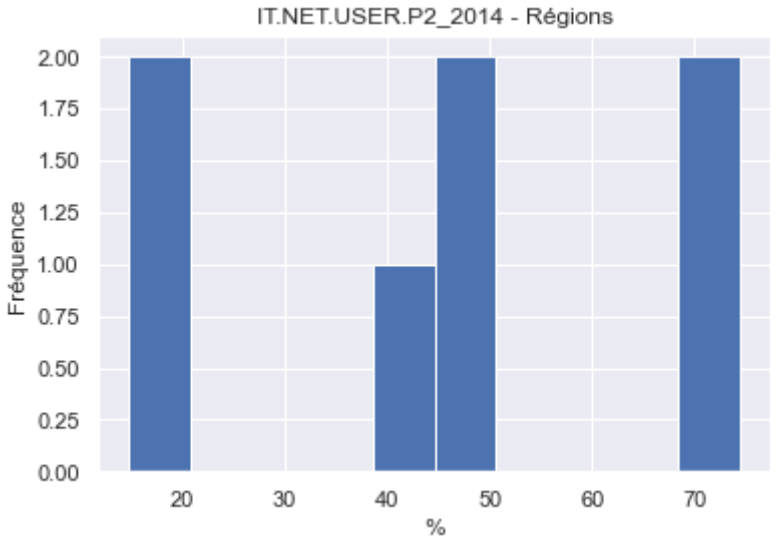


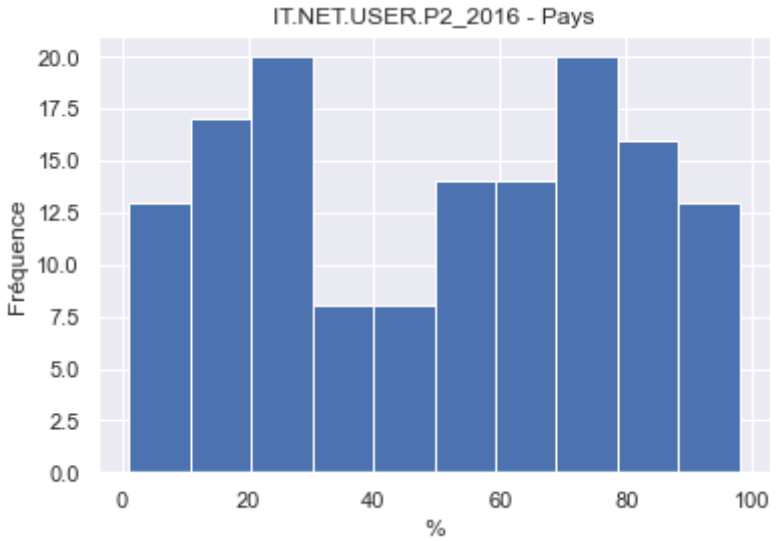

```
**************************************************
**************************************************
IT.NET.USER.P2

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
year
2014           44.920000         44.470066        28.757639         46.812103
2016           53.226178         50.191637        28.448887         52.796321
LAST           53.226178         50.191637        28.448887         52.796321

      Mean - Regions  Std - Regions
year
2014       44.701409      22.600324
2016       50.672946      21.738012
LAST       50.672946      21.738012
```

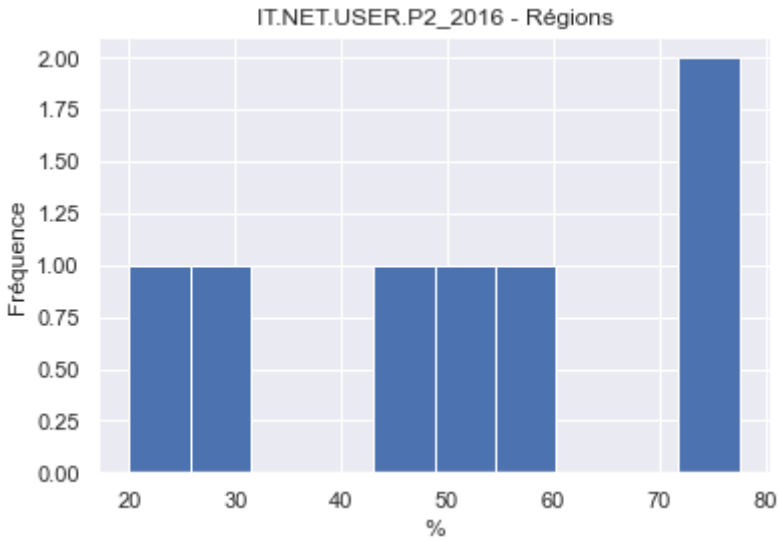

IT.NET.USER.P2 - Pays
100
80
60
40
20
0
%
2014
2016
LAST
Années

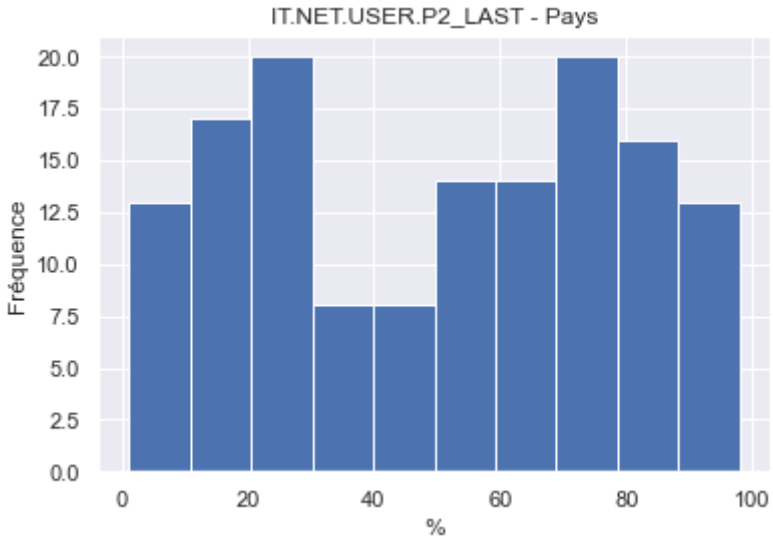

IT.NET.USER.P2 - Régions
80
70
60
50
40
30
20
%
2014
2016
LAST
Années

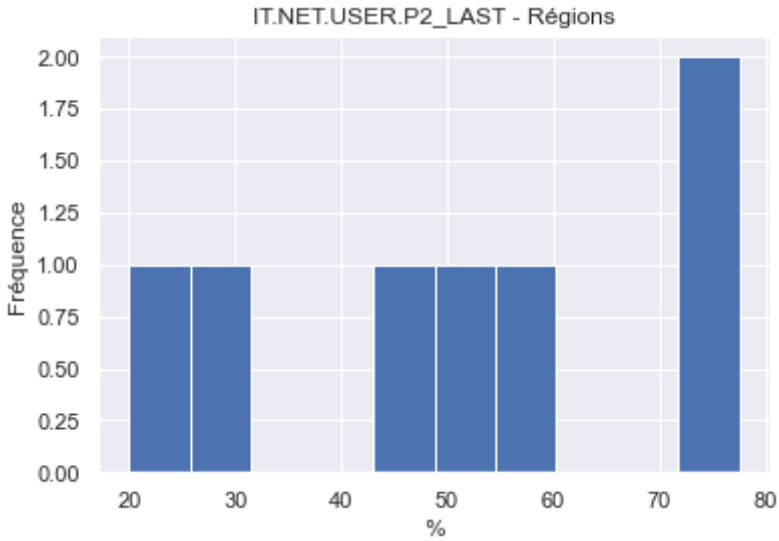

IT.NET.USER.P2_2014 - Pays
20
15
10
5
0
Fréquence
0
20
40
60
80
100
%


IT.NET.USER.P2_2014 - Régions
Fréquence
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
20
30
40
50
60
70
%

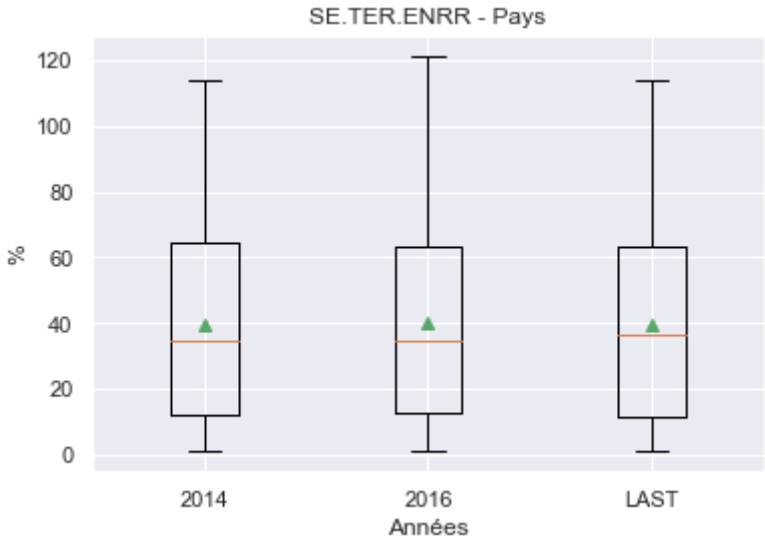

IT.NET.USER.P2_2016 - Pays
Fréquence
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100
%

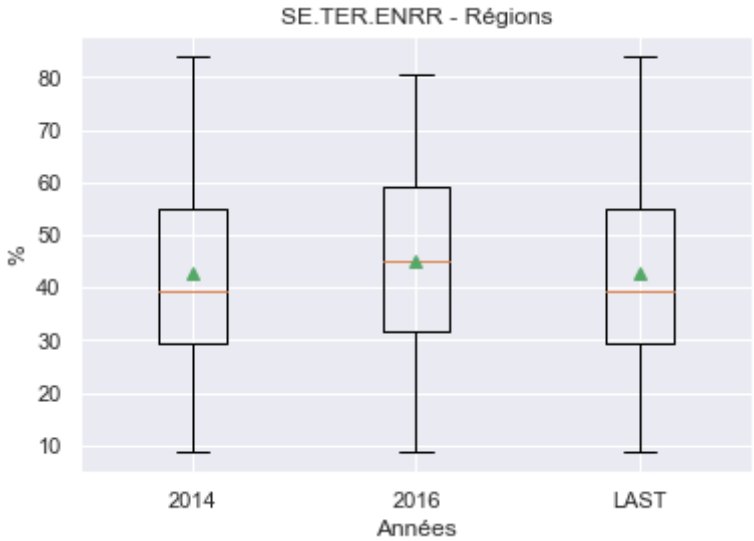

IT.NET.USER.P2_2016 - Régions
Fréquence
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
20
30
40
50
60
70
80
%

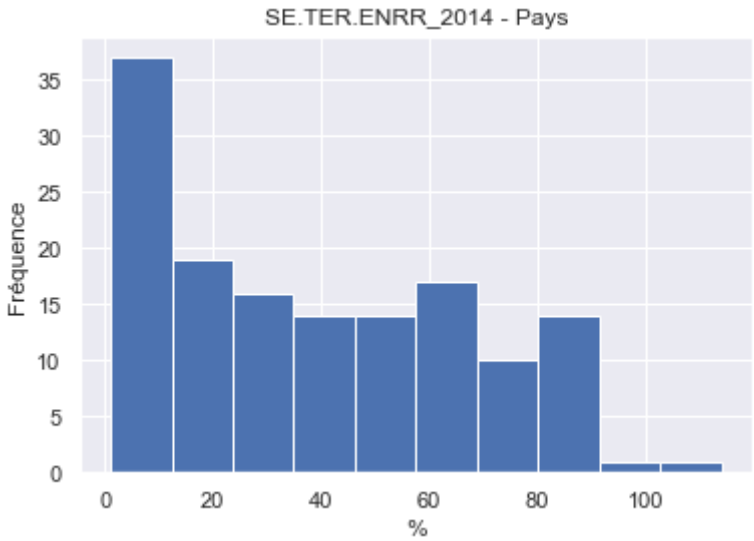


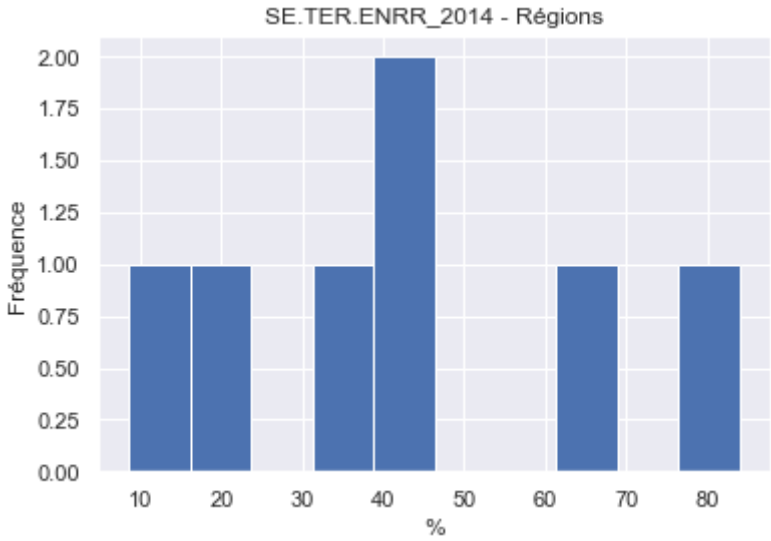


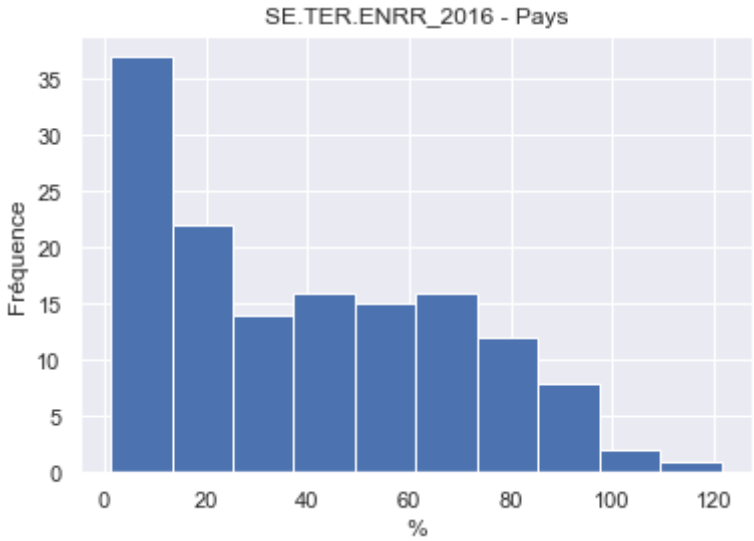

```
**************************************************
**************************************************
SE.TER.ENRR

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
year
2014           34.593811         39.230695        28.292627         39.147720
2016           34.608745         40.028632        29.183303         44.970909
LAST           36.280399         39.448677        28.579777         39.147720

      Mean - Regions  Std - Regions
year
2014       42.896155      25.460831
2016       45.127248      24.706675
LAST       42.896155      25.460831
```

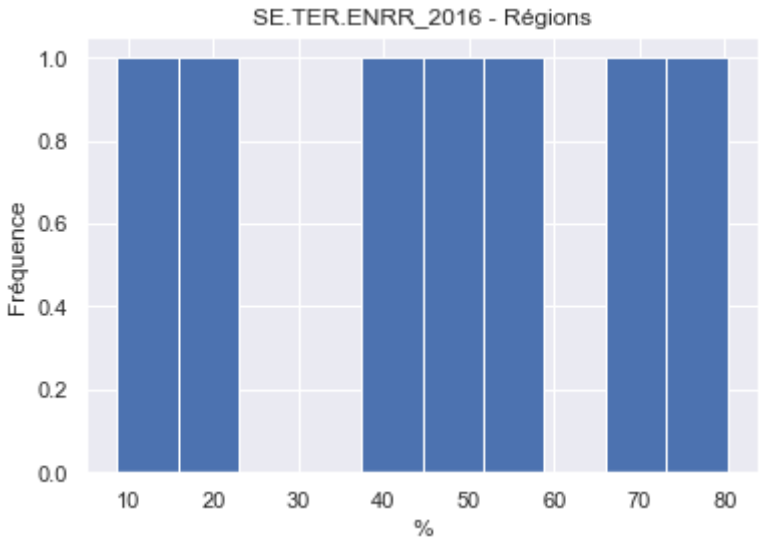

SE.TER.ENRR - Pays
120
100
80
60
40
20
0
%
2014
2016
LAST
Années

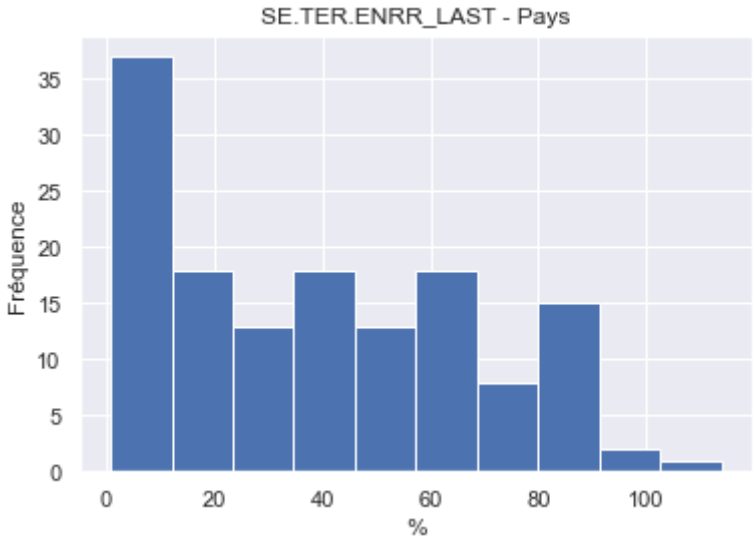

SE.TER.ENRR - Régions
80
70
60
50
40
30
20
10
%
2014
2016
LAST
Années

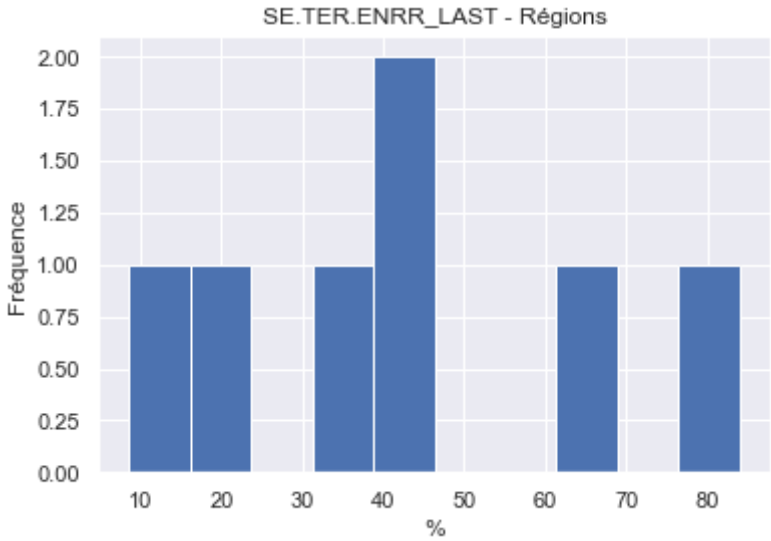

SE.TER.ENRR_2014 - Pays
35
30
25
20
15
10
5
0
Fréquence
0
20
40
60
80
100
%


SE.TER.ENRR_2014 - Régions
Fréquence
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
10
20
30
40
50
60
70
80
%


SE.TER.ENRR_2016 - Pays
Fréquence
35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
%


SE.TER.ENRR_2016 - Régions
Fréquence
1.0
0.8
0.6
0.4
0.2
0.0
10
20
30
40
50
60
70
80
%






```
**************************************************
```

4.4.3 Le Score

Faire un scoring des pays et choisir les 15 top pays

In [130…

```
#faire le scoring avec les parameter dans score_dict
data_scores_df = output_data_scores(data_df, score_dict)
print(data_scores_df)
```

```
              NY.GDP.PCAP.CD_Score  SE.TER.ENRL_Score  IT.NET.USER.P2_Score  \
country year
AFG     2014              0.439154           8.065825              0.700000
        2016              0.252977           8.572831              1.059573
        LAST              0.252977           8.065825              1.059573
AGO     2014              4.869937           7.807141              1.020000
        2016              4.103437           7.822584              1.300000
...                            ...                ...                   ...
ZMB     2016              2.023439           5.935927              2.550658
        LAST              2.023439           5.844766              2.550658
ZWE     2014              1.571664           6.867040              1.636474
        2016              1.567389           7.313243              2.311999
        LAST              1.567389           7.107265              2.311999

              SE.TER.ENRR_Score  Market_Score  Attr_Score  Total_Score
country year
AFG     2014           0.866280      4.252490    0.783140     2.517815
        2016           1.108928      4.412904    1.084250     2.748577
        LAST           0.866280      4.159401    0.962926     2.561164
AGO     2014           0.961557      6.338539    0.990779     3.664659
```

```
        2016          0.902250      5.963011    1.101125     3.532068
...                        ...           ...         ...          ...
ZMB     2016          0.383137      3.979683    1.466897     2.723290
        LAST          0.398141      3.934103    1.474399     2.704251
ZWE     2014          0.715680      4.219352    1.176077     2.697714
        2016          0.970512      4.440316    1.641255     3.040786
        LAST          0.843327      4.337327    1.577663     2.957495

[450 rows x 7 columns]
```

In [131…

```
#selectionner les top 15 pays et afficher leurs noms
data_scores_df_sorted = data_scores_df.xs('2016', level='year').sort_values('Total_S
top_fifteen = [top for top in data_scores_df_sorted.index.values[:22]
               if top in my_countries][:15]
list_countrynames(top_fifteen)
data_scores_df_sorted[data_scores_df_sorted.index.get_level_values('country').isin(t
```

```
AUS Australia
KOR Korea
NLD Netherlands
DNK Denmark
USA United States
ESP Spain
DEU Germany
NOR Norway
FIN Finland
AUT Austria
JPN Japan
NZL New Zealand
IRL Ireland
BEL Belgium
FRA France
```

Out[131…

country	Market_Score	Attr_Score	Total_Score
AUS	9.998431	9.364418	9.681425
KOR	9.352438	9.242790	9.297614
NLD	9.791560	8.734016	9.262788
DNK	9.188947	9.048235	9.118591
USA	10.000000	8.055345	9.027673
ESP	9.316400	8.542451	8.929425
DEU	9.812496	8.040827	8.926662
NOR	9.058293	8.695528	8.876910
FIN	8.972189	8.681106	8.826648
AUT	9.268092	8.371516	8.819804
JPN	9.727465	7.867732	8.797598
NZL	8.818556	8.773177	8.795866
IRL	8.921436	8.620904	8.771170
BEL	9.308454	8.164522	8.736488
FRA	9.668815	7.734754	8.701784

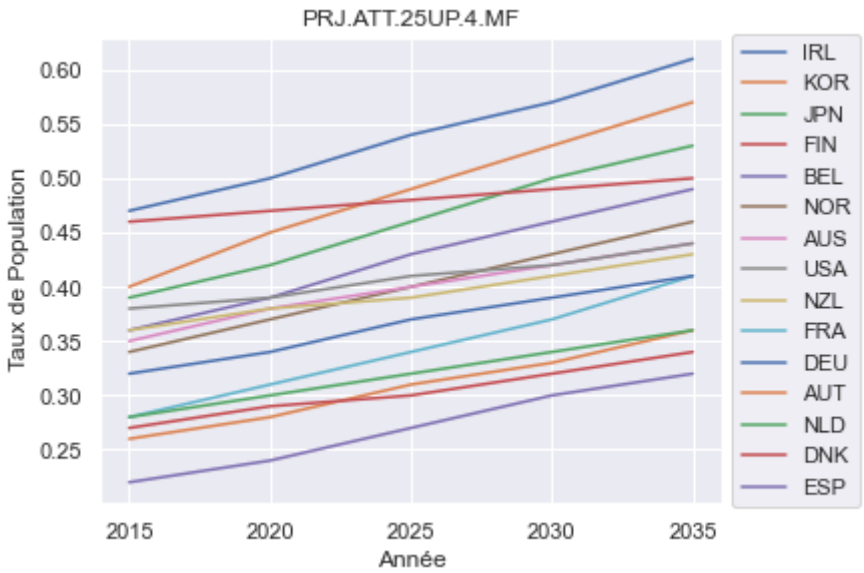
In [132…

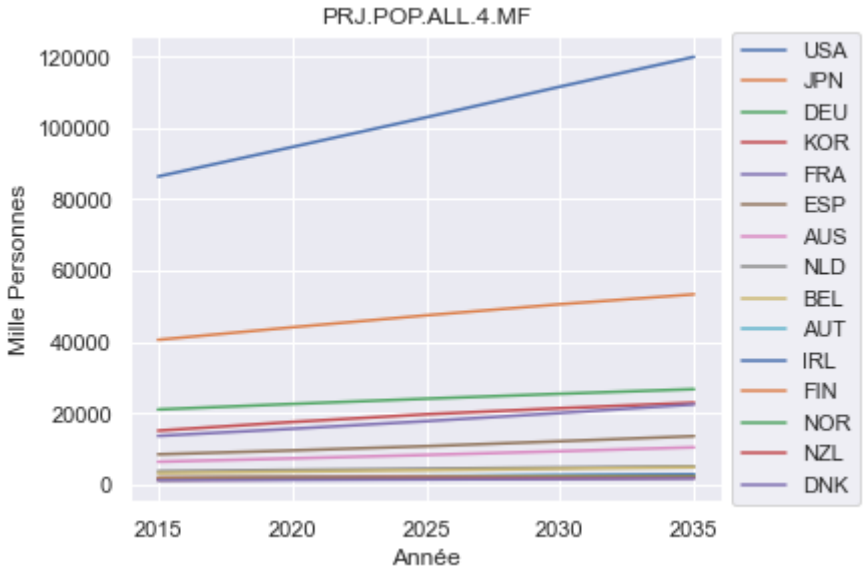
```
def plot_countryindictors_vs_years(ind, data_df, unit = '', savefigures = False, *kw
    data_plot = make_2D_dataarray(data_df, ind, 'indicator')
```

```
    data_plot = data_plot.sort_values(max(data_plot.index.values), axis = 1, ascendi
    #print(data_plot)
    fig, ax = plt.subplots()
    ax.plot(data_plot)

    ax.set_xlabel("Année")
    ax.set_ylabel(unit)
    title = ind
    ax.set_title(title)
    ax.legend(data_plot.columns.values, bbox_to_anchor=(1, 0.5), loc='center left')
    plt.tight_layout(pad=0.4, w_pad=1, h_pad=0.5)

    if savefigures :
        plt.savefig(os.getcwd() + '\\Figures\\' + title.replace('.','_') + '_Plots',
    plt.show()
```

In [133…

```
years_proj = ['2015','2020','2025','2030', '2035']
proj_inds = [('PRJ.ATT.25UP.4.MF',' Taux de Population'), ('PRJ.POP.ALL.4.MF', 'Mill

data_df_proj = DATA_MIN_PROJ
select = [i and j for i, j in zip(data_df_proj.country.isin(top_fifteen),
                                  data_df_proj.year.isin(years_proj))]
data_df_proj = data_df_proj[select]

for t in proj_inds :
    plot_countryindictors_vs_years(t[0], data_df_proj, savefigures=True, unit = t[1]
```





In [134…

```
#look at the projected growth rate
data_df_proj_growth = pd.DataFrame()
for ind in ['PRJ.ATT.25UP.4.MF','PRJ.POP.ALL.4.MF', ] :
    data_tmp = data_df_proj[data_df_proj.indicator == ind].drop(columns = ['indicato
    data_tmp[data_tmp.year == '2035'].values
    data_tmp1 = (data_tmp[data_tmp.year == '2035'].drop(columns = ['year']) - data_t
    data_tmp1 = data_tmp1 / data_tmp[data_tmp.year == '2015'].drop(columns = ['year'
    data_tmp1 = data_tmp1.rename(columns = {'value' : ind + "_GrowthRate"})
    data_df_proj_growth = pd.concat([data_df_proj_growth, data_tmp1], axis = 1)

data_df_proj_growth_score = data_df_proj_growth.T.mean().rename('Proj_GrowthRate_Sco
data_df_proj_growth = pd.concat([data_df_proj_growth, data_df_proj_growth_score], ax
print(data_df_proj_growth)
data_df_proj_growth.sort_values('Proj_GrowthRate_Score',ascending = False)
```

```
         PRJ.ATT.25UP.4.MF_GrowthRate  PRJ.POP.ALL.4.MF_GrowthRate  \
country
AUS                          0.257143                     0.647854
AUT                          0.384615                     0.479164
BEL                          0.361111                     0.524315
DNK                          0.259259                     0.434003
FIN                          0.086957                     0.193212
FRA                          0.464286                     0.652579
DEU                          0.281250                     0.272528
IRL                          0.297872                     0.631855
JPN                          0.358974                     0.315325
KOR                          0.425000                     0.523447
NLD                          0.285714                     0.374852
NZL                          0.194444                     0.505462
NOR                          0.352941                     0.633566
ESP                          0.454545                     0.610575
USA                          0.157895                     0.388915

         Proj_GrowthRate_Score
country
AUS                   0.452499
AUT                   0.431889
BEL                   0.442713
DNK                   0.346631
FIN                   0.140084
FRA                   0.558433
DEU                   0.276889
IRL                   0.464863
JPN                   0.337150
KOR                   0.474224
NLD                   0.330283
NZL                   0.349953
```

```
NOR                    0.493254
ESP                    0.532560
USA                    0.273405
```

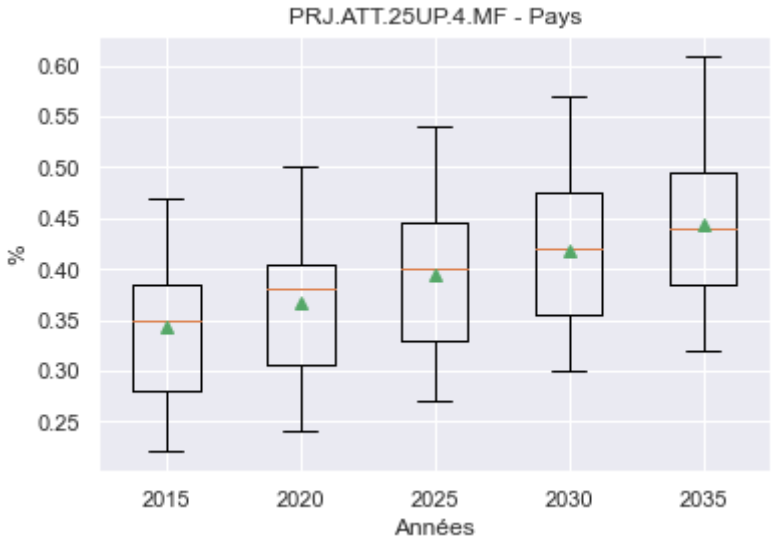
Out[134…

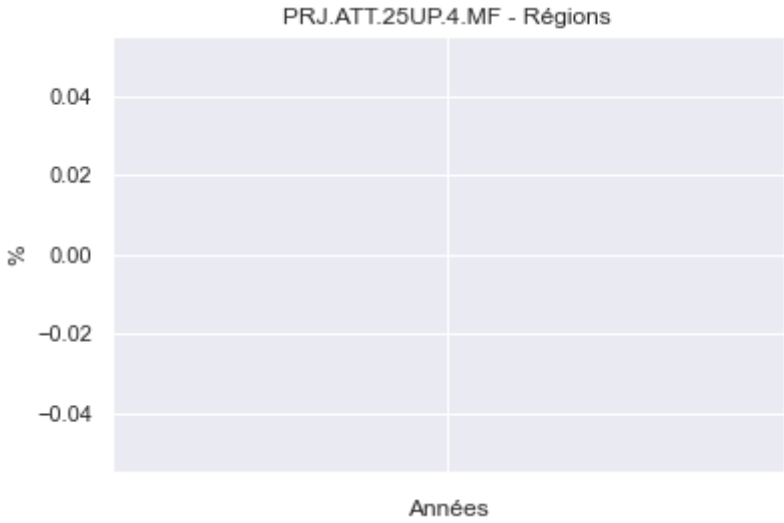
country	PRJ.ATT.25UP.4.MF_GrowthRate	PRJ.POP.ALL.4.MF_GrowthRate	Proj_GrowthRate_Score
FRA	0.464286	0.652579	0.558433
ESP	0.454545	0.610575	0.532560
NOR	0.352941	0.633566	0.493254
KOR	0.425000	0.523447	0.474224
IRL	0.297872	0.631855	0.464863
AUS	0.257143	0.647854	0.452499
BEL	0.361111	0.524315	0.442713
AUT	0.384615	0.479164	0.431889
NZL	0.194444	0.505462	0.349953
DNK	0.259259	0.434003	0.346631
JPN	0.358974	0.315325	0.337150
NLD	0.285714	0.374852	0.330283
DEU	0.281250	0.272528	0.276889
USA	0.157895	0.388915	0.273405
FIN	0.086957	0.193212	0.140084

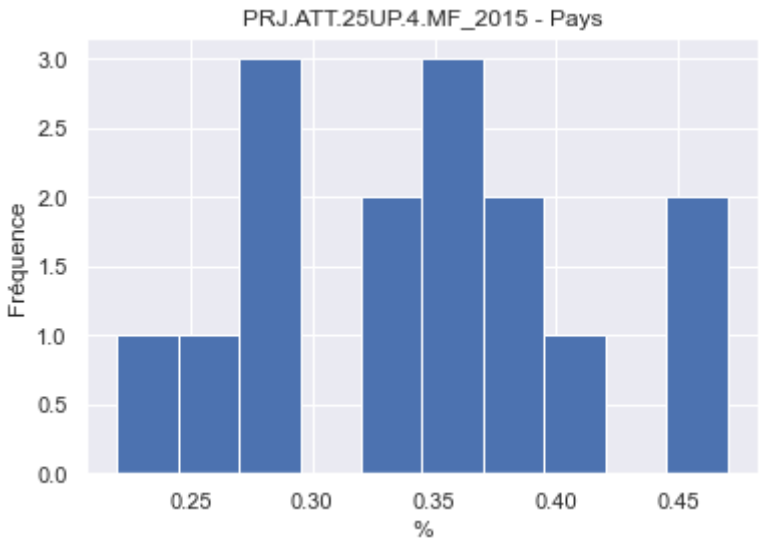
In [135…

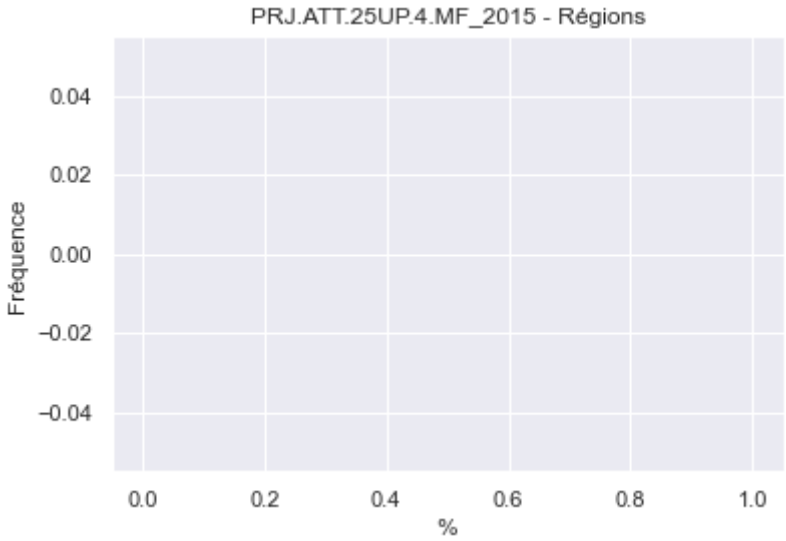
```
score_dict_proj = {'PRJ.ATT.25UP.4.MF' : (True, 0.6, 0, 'Proj', '%'), 'PRJ.POP.ALL.4
for ind, vals in score_dict_proj.items() :
    print(50*'*')
    print(ind)
    print()
    print(output_info(ind, data_df_proj, saveinfo = False))
    plot_boxplot2(ind, data_df_proj, unit = vals[4], savefigures = False, showmeans
    plot_histogramm(ind, data_df_proj, unit = vals[4], savefigures = False)
    print(data_df_proj[data_df_proj.indicator == ind])
    print(50*'*')
```

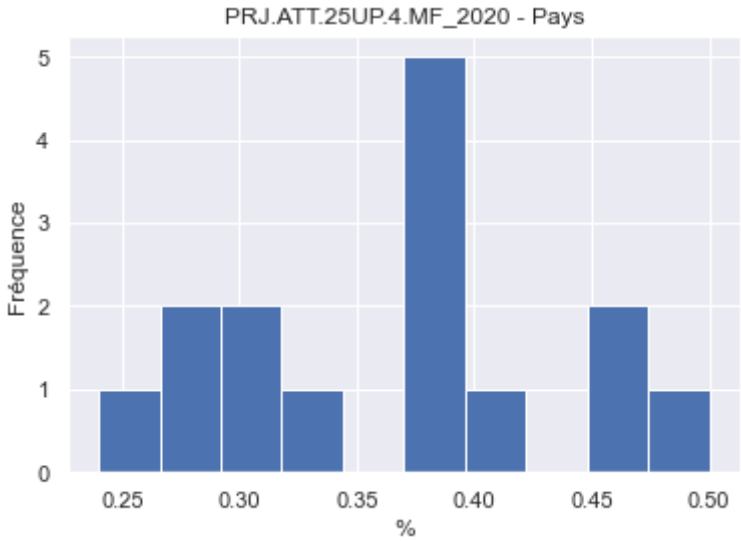
```
**************************************************
PRJ.ATT.25UP.4.MF

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
2015                0.35          0.342667         0.072256               NaN
2020                0.38          0.367333         0.074399               NaN
2025                0.40          0.394000         0.077533               NaN
2030                0.42          0.418667         0.080166               NaN
2035                0.44          0.444667         0.084165               NaN

      Mean - Regions  Std - Regions
2015             NaN            NaN
2020             NaN            NaN
2025             NaN            NaN
2030             NaN            NaN
2035             NaN            NaN
```

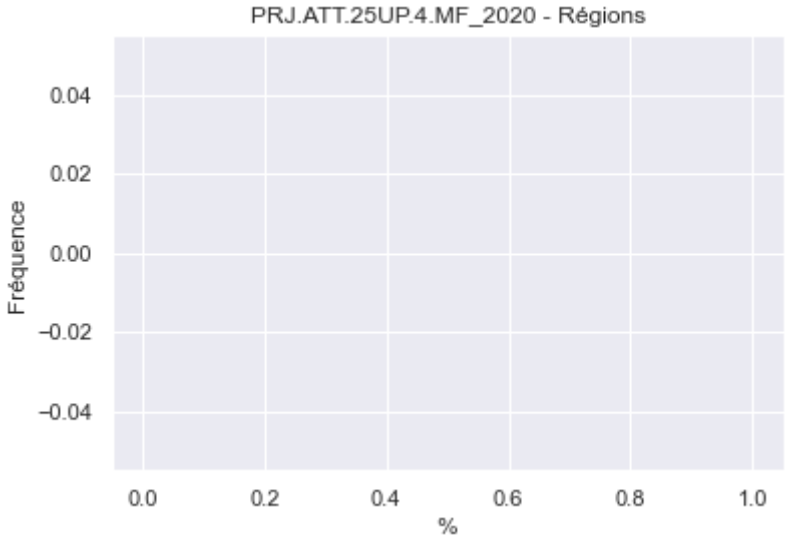

PRJ.ATT.25UP.4.MF - Pays
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
%
2015
2020
2025
2030
2035
Années

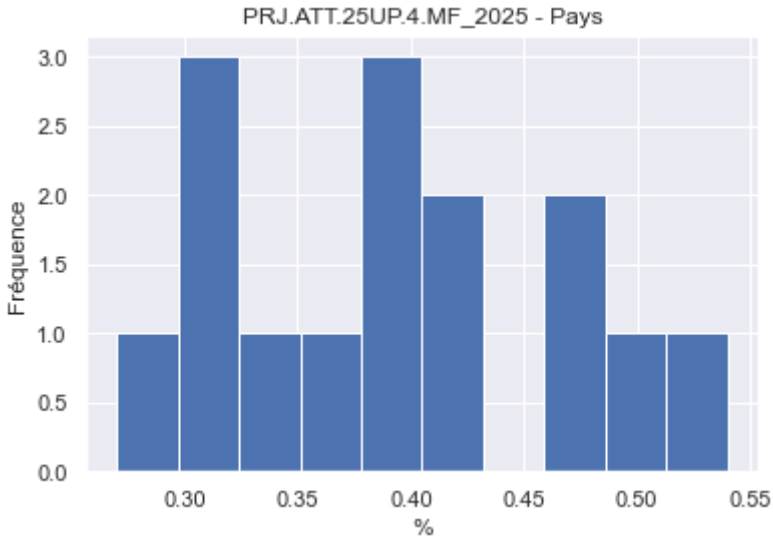

PRJ.ATT.25UP.4.MF - Régions
0.04
0.02
0.00
−0.02
−0.04
%
Années

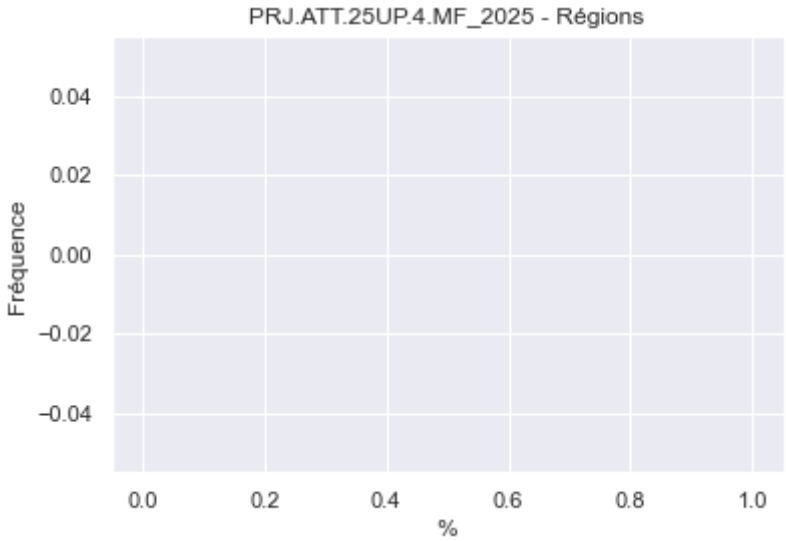

PRJ.ATT.25UP.4.MF_2015 - Pays
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Fréquence
0.25
0.30
0.35
0.40
0.45
%

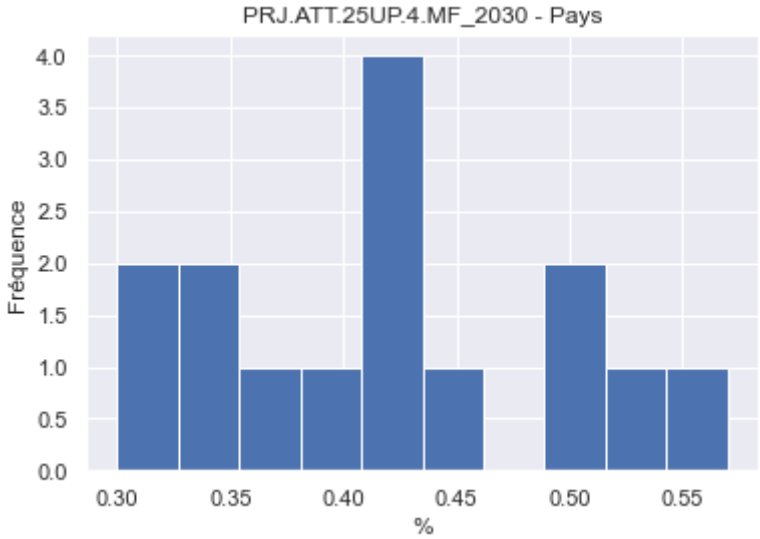

PRJ.ATT.25UP.4.MF_2015 - Régions
Fréquence
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
%

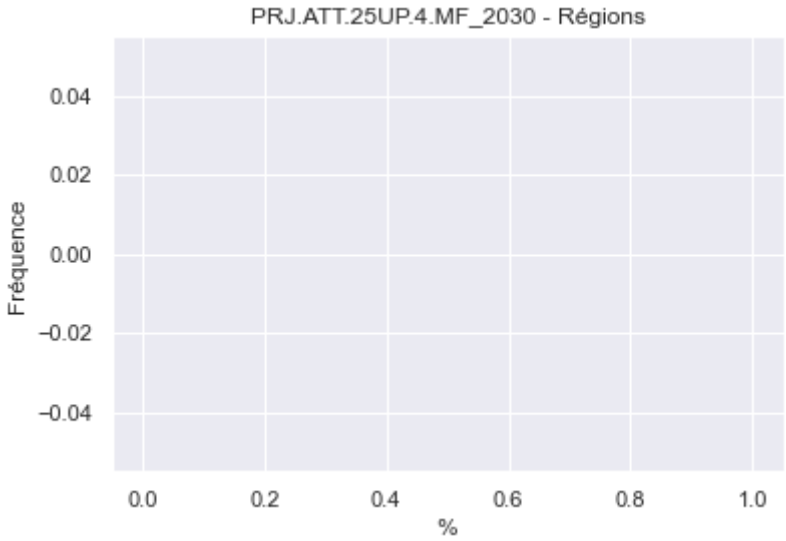

PRJ.ATT.25UP.4.MF_2020 - Pays
Fréquence
5
4
3
2
1
0
0.25
0.30
0.35
0.40
0.45
0.50
%

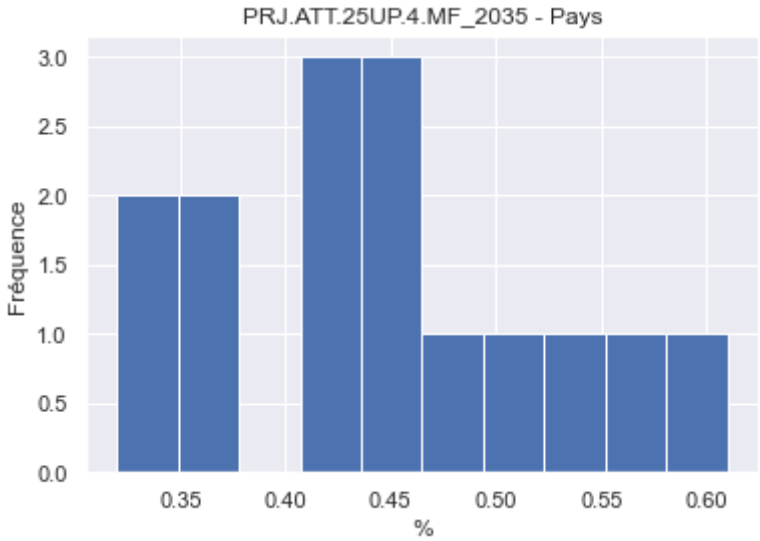

PRJ.ATT.25UP.4.MF_2020 - Régions
Fréquence
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
%

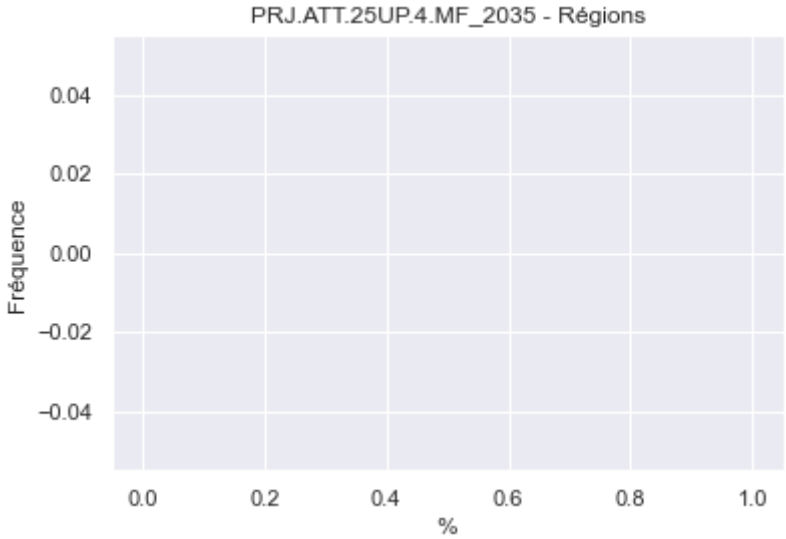

PRJ.ATT.25UP.4.MF_2025 - Pays
Fréquence
%
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.30
0.35
0.40
0.45
0.50
0.55


PRJ.ATT.25UP.4.MF_2025 - Régions
Fréquence
%
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


PRJ.ATT.25UP.4.MF_2030 - Pays
Fréquence
%
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0.30
0.35
0.40
0.45
0.50
0.55

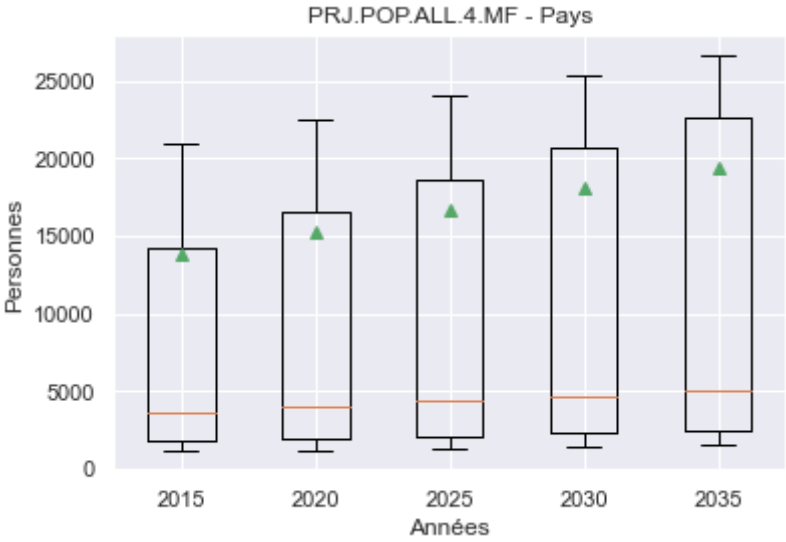


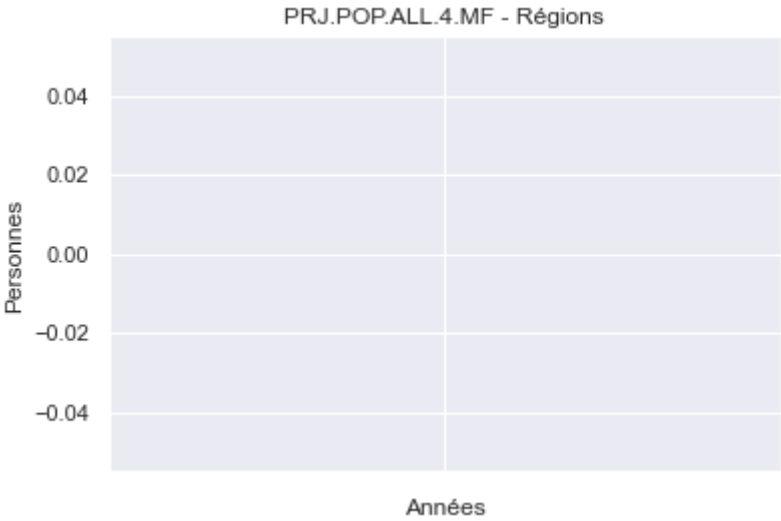


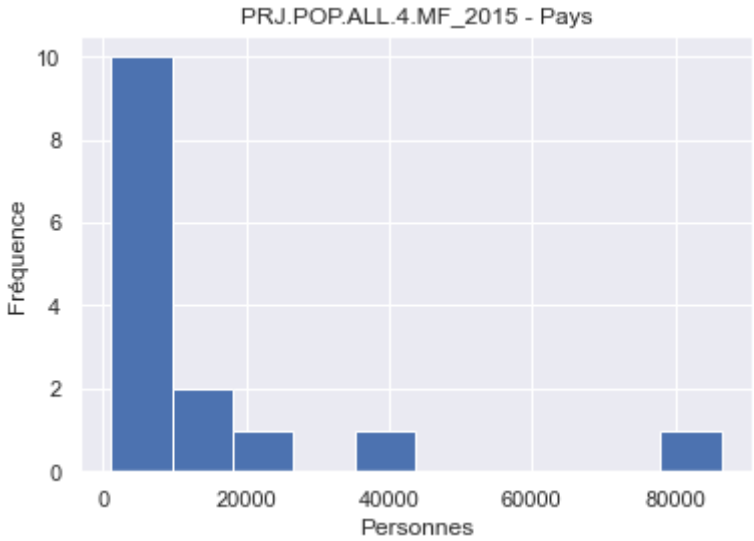


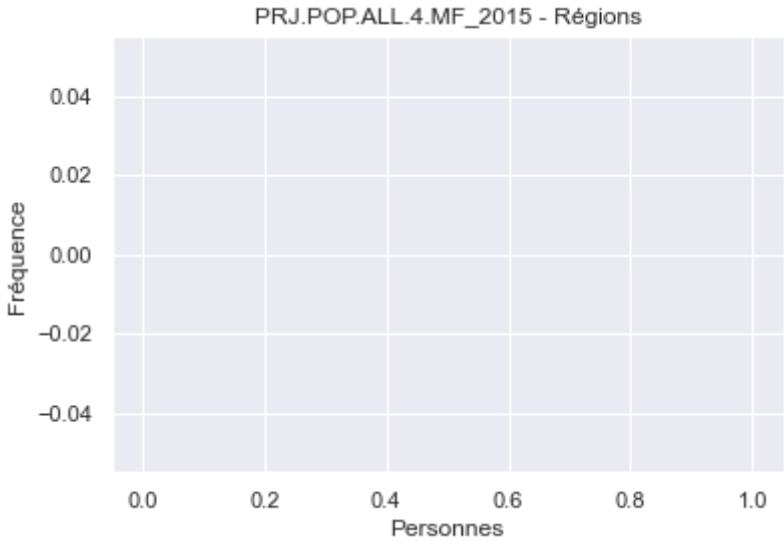

```
                indicator country  year  value
131763  PRJ.ATT.25UP.4.MF     AUS  2015   0.35
135428  PRJ.ATT.25UP.4.MF     AUT  2015   0.26
161083  PRJ.ATT.25UP.4.MF     BEL  2015   0.36
289358  PRJ.ATT.25UP.4.MF     DNK  2015   0.27
337003  PRJ.ATT.25UP.4.MF     FIN  2015   0.46
...                   ...     ...   ...    ...
604548  PRJ.ATT.25UP.4.MF     NLD  2035   0.36
611878  PRJ.ATT.25UP.4.MF     NZL  2035   0.43
630203  PRJ.ATT.25UP.4.MF     NOR  2035   0.46
747483  PRJ.ATT.25UP.4.MF     ESP  2035   0.32
850103  PRJ.ATT.25UP.4.MF     USA  2035   0.44
```

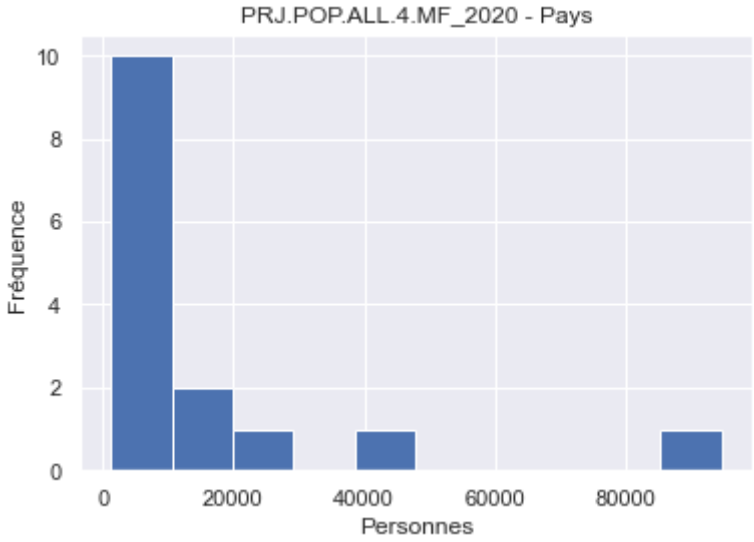
```
[75 rows x 4 columns]
**************************************************
**************************************************
PRJ.POP.ALL.4.MF

      Median - Countries  Mean - Countries  Std - Countries  Median - Regions  \
2015             3635.41      13777.822667     22755.978931               NaN
2020             3979.74      15214.305333     24877.310269               NaN
2025             4334.84      16640.942667     26997.879271               NaN
2030             4682.39      18066.725333     29158.268444               NaN
2035             4998.15      19450.951333     31235.590333               NaN

      Mean - Regions  Std - Regions
2015             NaN            NaN
2020             NaN            NaN
2025             NaN            NaN
2030             NaN            NaN
2035             NaN            NaN
```

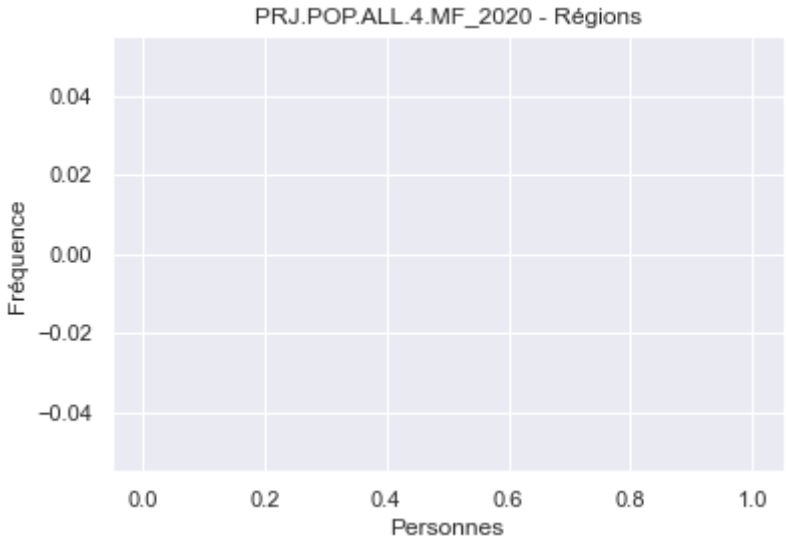


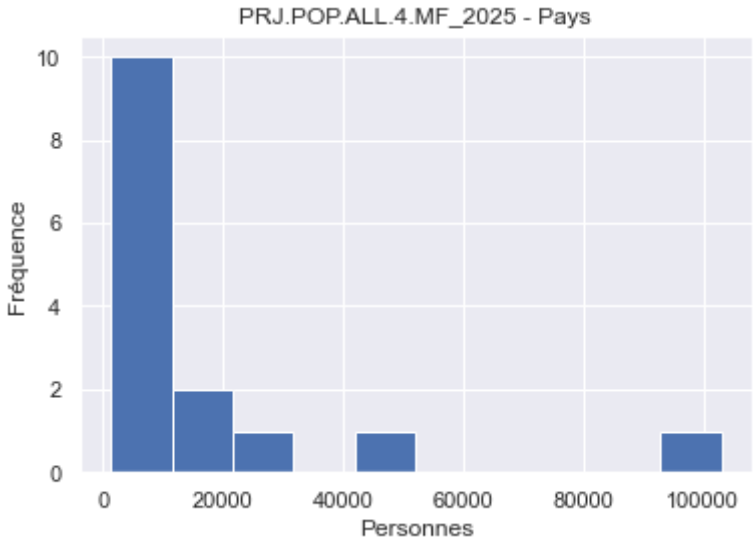


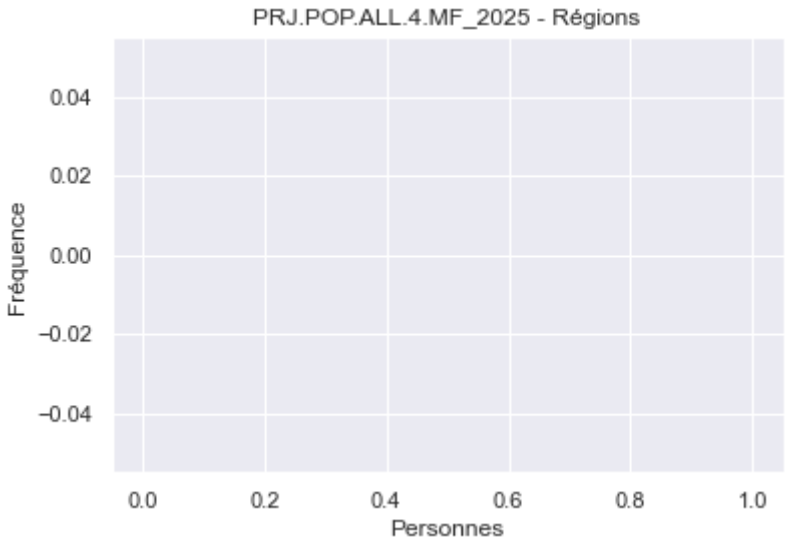

PRJ.POP.ALL.4.MF_2015 - Pays
Fréquence
Personnes
0
2
4
6
8
10
0
20000
40000
60000
80000

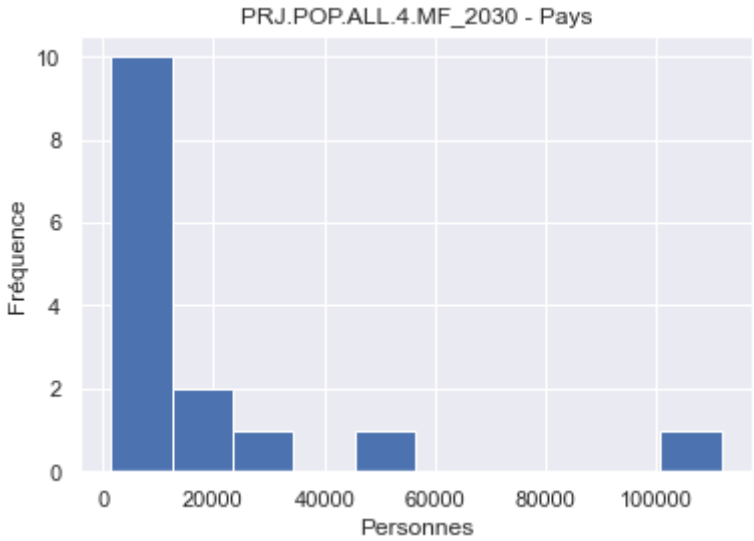

PRJ.POP.ALL.4.MF_2015 - Régions
Fréquence
Personnes
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

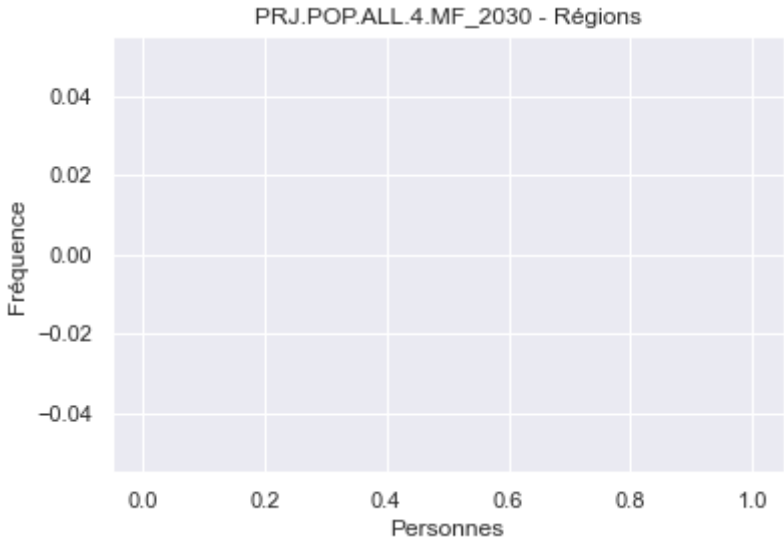

PRJ.POP.ALL.4.MF_2020 - Pays
Fréquence
Personnes
0
2
4
6
8
10
0
20000
40000
60000
80000

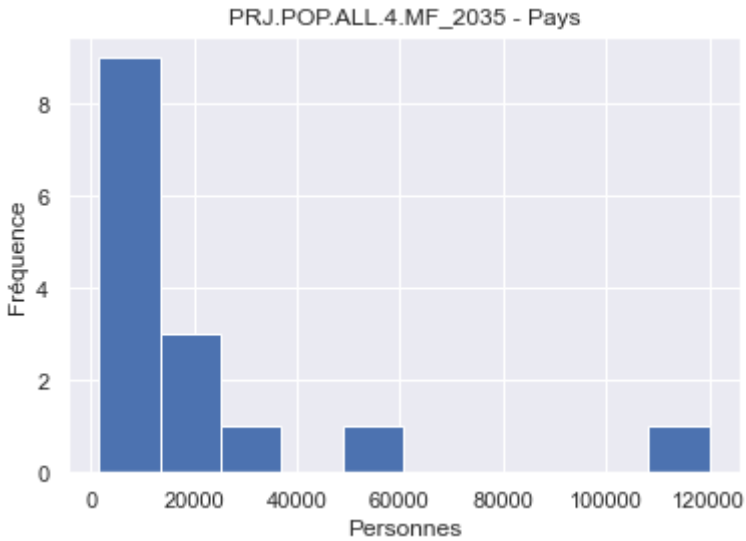

PRJ.POP.ALL.4.MF_2020 - Régions
Fréquence
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
Personnes

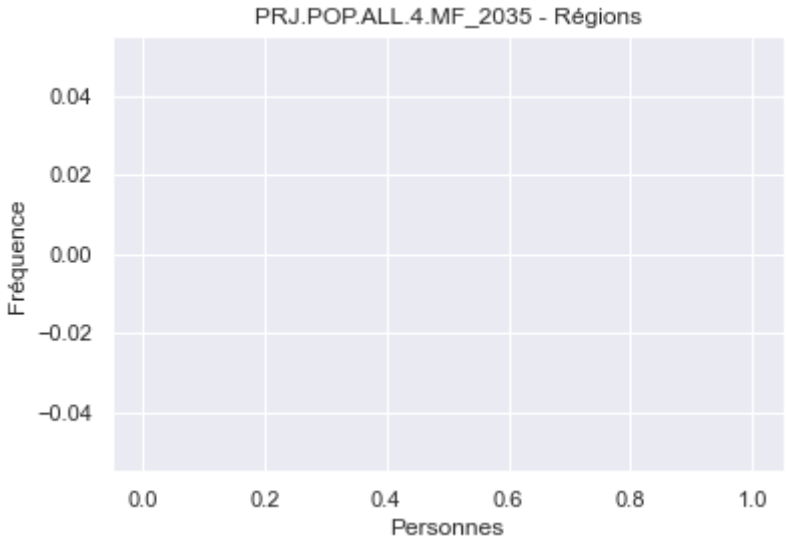

PRJ.POP.ALL.4.MF_2025 - Pays
Fréquence
10
8
6
4
2
0
0
20000
40000
60000
80000
100000
Personnes


PRJ.POP.ALL.4.MF_2025 - Régions
Fréquence
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
Personnes


PRJ.POP.ALL.4.MF_2030 - Pays
Fréquence
Personnes
0
2
4
6
8
10
0
20000
40000
60000
80000
100000


PRJ.POP.ALL.4.MF_2030 - Régions
Fréquence
Personnes
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0


PRJ.POP.ALL.4.MF_2035 - Pays
Fréquence
Personnes
0
2
4
6
8
0
20000
40000
60000
80000
100000
120000



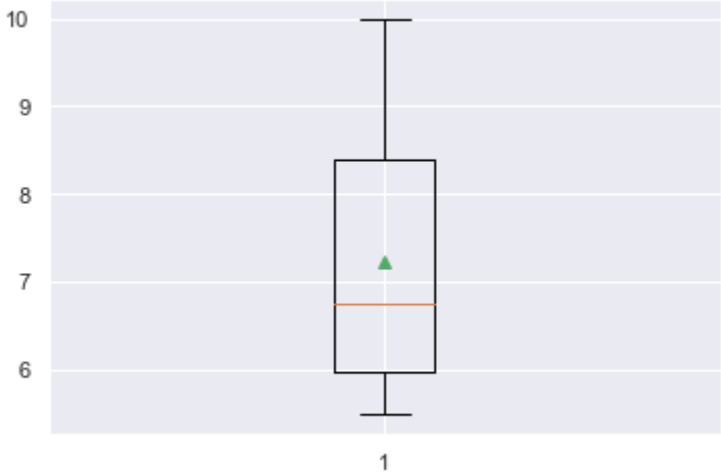

```
               indicator country  year      value
131925  PRJ.POP.ALL.4.MF     AUS  2015    6276.89
135590  PRJ.POP.ALL.4.MF     AUT  2015    1817.25
161245  PRJ.POP.ALL.4.MF     BEL  2015    3107.79
289520  PRJ.POP.ALL.4.MF     DNK  2015    1084.90
337165  PRJ.POP.ALL.4.MF     FIN  2015    1843.21
...                  ...     ...   ...        ...
604710  PRJ.POP.ALL.4.MF     NLD  2035    4998.15
612040  PRJ.POP.ALL.4.MF     NZL  2035    1827.51
630365  PRJ.POP.ALL.4.MF     NOR  2035    2089.07
747645  PRJ.POP.ALL.4.MF     ESP  2035   13469.87
850265  PRJ.POP.ALL.4.MF     USA  2035  119972.68

[75 rows x 4 columns]
**************************************************
```

In [136…

```
data_scores_df_proj = output_data_scores(data_df_proj, score_dict_proj).reset_index(
#print(data_scores_df_proj)
plt.boxplot(data_scores_df_proj.loc[data_scores_df_proj.year == '2035', 'PRJ.POP.ALL

data_tmp = data_scores_df_proj.set_index(['country', 'year']).xs('2035', level='year
clist = data_tmp.head(20).index.values
print(data_tmp)
list_countrynames(clist)
```

```
         PRJ.ATT.25UP.4.MF_Score  PRJ.POP.ALL.4.MF_Score  Proj_Score  \
country
JPN                     8.833333               9.316828    9.075081
KOR                     9.500000               8.398036    8.949018
USA                     7.333333              10.000000    8.666667
IRL                    10.000000               6.066746    8.033373
DEU                     6.833333               8.565042    7.699187
FRA                     6.833333               8.374394    7.603864
AUS                     7.333333               7.536658    7.434996
BEL                     8.166667               6.688816    7.427741
FIN                     8.333333               5.855731    7.094532
NOR                     7.666667               5.799882    6.733275
ESP                     5.333333               7.823409    6.578371
NZL                     7.166667               5.654649    6.410658
NLD                     6.000000               6.747023    6.373512
AUT                     6.000000               6.073577    6.036789
DNK                     5.666667               5.479850    5.573258

         Total_Score
country
JPN         9.075081
KOR         8.949018
USA         8.666667
IRL         8.033373
```

```
DEU         7.699187
FRA         7.603864
AUS         7.434996
BEL         7.427741
FIN         7.094532
NOR         6.733275
ESP         6.578371
NZL         6.410658
NLD         6.373512
AUT         6.036789
DNK         5.573258
JPN Japan
KOR Korea
USA United States
IRL Ireland
DEU Germany
FRA France
AUS Australia
BEL Belgium
FIN Finland
NOR Norway
ESP Spain
NZL New Zealand
NLD Netherlands
AUT Austria
DNK Denmark
```



In [137…

```
CURRENT_STOP
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-137-e88427ca3d30> in <module>
----> 1 CURRENT_STOP

NameError: name 'CURRENT_STOP' is not defined
```