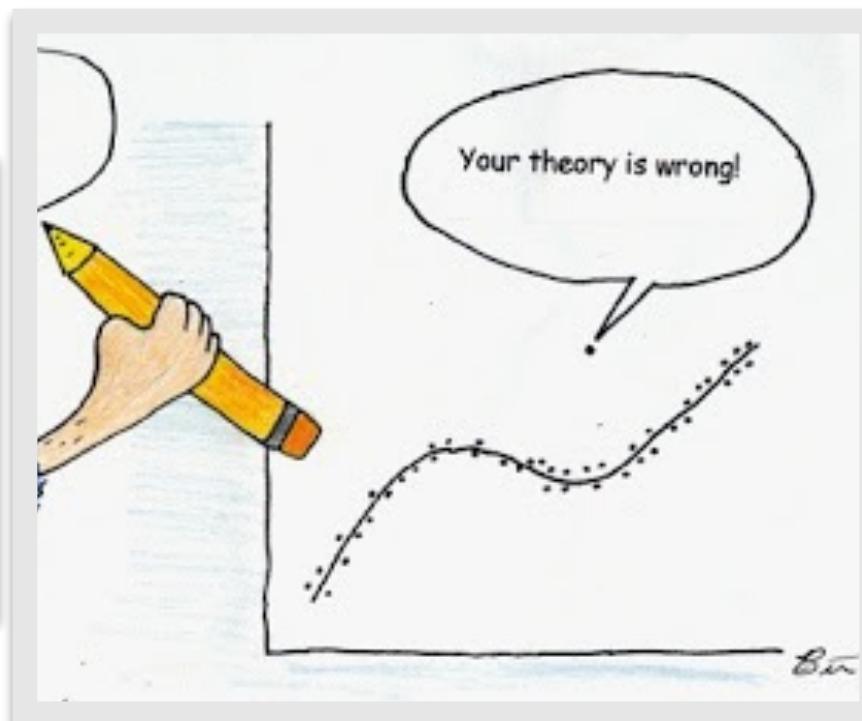


Projet 7 :Effectuer une prédition de revenus

Année 2018-2019



Sommaire



H₂

Mission 1

Problématique



SOUHAITE CIBLER DE NOUVEAUX JEUNES CLIENTS



EN CIBLANT DES FUTUR PERSONNES AVEC DES HAUTS REVENUS



CRÉER UN MODÈLE POUR DÉTERMINER LE REVENU POTENTIEL D'UNE PERSONNE



EMPLOYÉ DANS UNE BANQUE

Les données



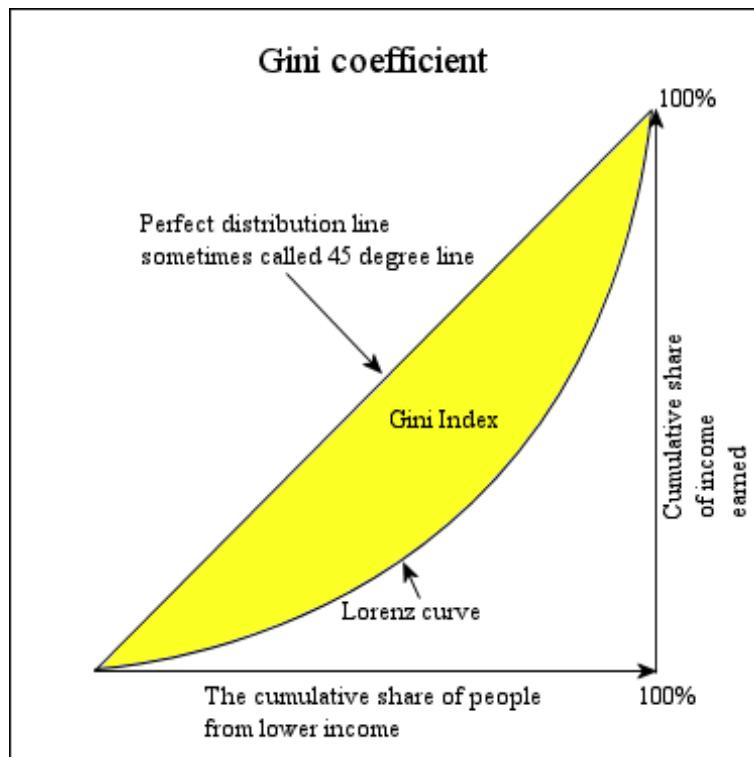
World inequality database

L'open data sur les inégalités globales du monde .

On y trouve :
les richesses, la qualité et les inégalités de revenus des personnes

Créé par :
Un consortium universitaire international

Les données



L'indice de gini :

Le coefficient de Gini, ou indice de Gini, est une mesure statistique permettant de se rendre compte de la répartition d'une variable (salaire, revenus, patrimoine) au sein d'une population

World income Distribution 2008

Contient diverses informations sur les pays :
-nombre de pays, distribution des revenus
etc...

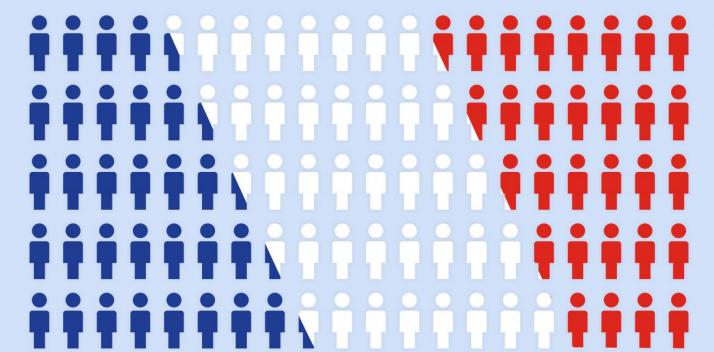
Nos données



Année(s) des données utilisées :
2004,2006,2007,2008,2009,2010,
2011

**Nombre de pays
présents :**
116

Les questions



Echantillonner une population est une bonne méthode ?



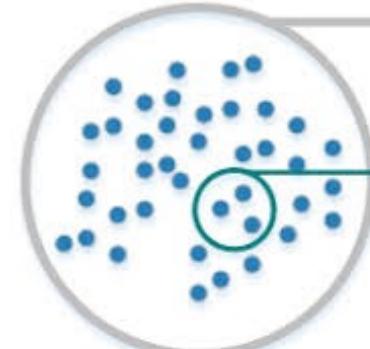
Plus facile à représenter

De quel type de quantile s'agit il ?

Quartiles

Déciles ?

Centiles ?

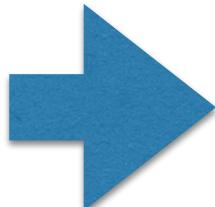


Nos données

De quel type de quantiles s'agit-il (quartiles, déciles, etc.) ? : centiles

	country	year_survey	quantile	nb_quantiles	income	gdpppp
0	ALB	2008	1	100	728.89795	7297.0
1	ALB	2008	2	100	916.66235	7297.0
2	ALB	2008	3	100	1010.91600	7297.0
3	ALB	2008	4	100	1086.90780	7297.0
4	ALB	2008	5	100	1132.69970	7297.0
5	ALB	2008	6	100	1171.14120	7297.0
6	ALB	2008	7	100	1201.13240	7297.0
7	ALB	2008	8	100	1240.89760	7297.0
8	ALB	2008	9	100	1285.69140	7297.0
9	ALB	2008	10	100	1325.25330	7297.0
10	ALB	2008	11	100	1351.31230	7297.0

Pays



100 quantiles

Les variables importantes

```
RangeIndex: 11599 entries, 0 to 11598  
Data columns (total 6 columns):  
country           11599 non-null object  
year_survey       11599 non-null int64  
quantile          11599 non-null int64  
nb_quantiles     11599 non-null int64  
income            11599 non-null float64  
gdpppp           11399 non-null float64  
dtypes: float64(2), int64(3), object(1)
```

Notre jeu de données :

-6 variables

-2 anomalies

Un centile manquant

Deux pays avec des gdpppp manquants

Centile manquant

Détection du pays :

-En réalisant la somme tous les pays devraient avoir 10 000

```
Erreurs=Annee08.groupby(['country']).sum()
#Ici on identifie qu'il manque un quartile pour LTU en 2008
Erreurs.loc[(Erreurs['nb_quantiles']<10000),:]
```

	year_survey	quantile	nb_quantiles	income	gdpppp
country					
LTU	198792	5009	9900	657483.5158	1739529.0

Détection du rang du quantile :

-On va chercher les valeurs en dessous de 76

On va devoir le ré-insérer :

-On fait le quantile 40+42 et on divise par 2

```
erreurs=Annee08.groupby(['quantile']).count()
erreurs.loc[(erreurs['country']<76),:]
```

	country	year_survey	nb_quantiles	income	gdpppp
quantile					
41	75	75	75	75	74

GDPPPP manquants

Notre jeu de données propre :

- Une anomalie : gdpppp 11400
- Mais le reste a été corrigé

```
RangeIndex: 11600 entries, 0 to 11599
Data columns (total 6 columns):
country           11600 non-null object
year_survey       11600 non-null int64
quantile          11600 non-null int64
nb_quantiles     11600 non-null int64
income            11600 non-null float64
gdpppp           11400 non-null float64
dtypes: float64(2), int64(3), object(1)
```

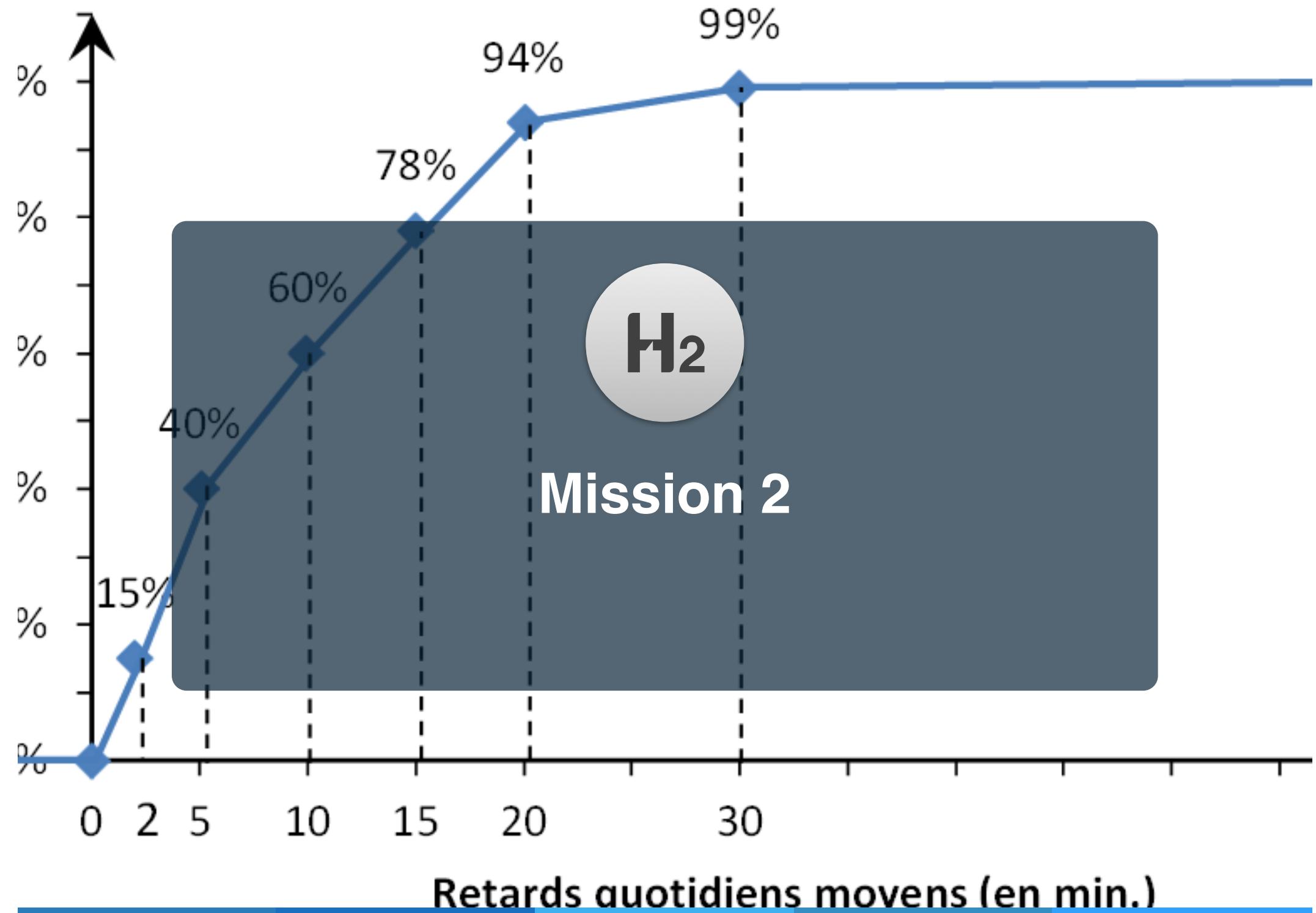
Détection des erreurs :

- loc avec une condition « .isnull « sur le gdpppp nous donne deux pays

Rectification des valeurs :

- J'ai cherché une source similaire de données pour les pays.

5824	XXX	2008	25	100	1264.56400	NaN
5825	XXX	2008	26	100	1287.11850	NaN
5826	XXX	2008	27	100	1306.18650	NaN
5827	XXX	2008	28	100	1326.08200	NaN
5828	XXX	2008	29	100	1354.35280	NaN
5829	XXX	2008	30	100	1377.85770	NaN
...
11269	PSE	2009	71	100	1212.43230	NaN
11270	PSE	2009	72	100	1236.57520	NaN
11271	PSE	2009	73	100	1253.49760	NaN
11272	PSE	2009	74	100	1275.80200	NaN
11273	PSE	2009	75	100	1297.43880	NaN
11274	PSE	2009	76	100	1330.10490	NaN
11275	PSE	2009	77	100	1358.09030	NaN



Mission 2

01

Réaliser différents graphiques

02

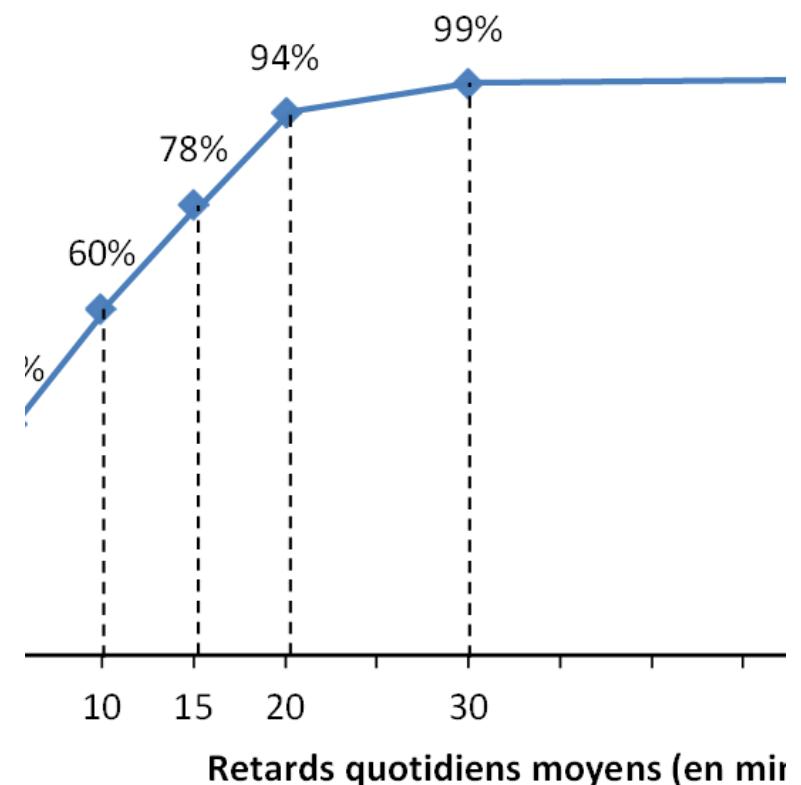
Sélectionner quelques pays

03

Mais des pays différents des uns des autres sur le plan économique

04

Utilisation d'un nouveau jeu de donnée



Préparation d'un nouveau jeu de données

analyse :

```
RangeIndex: 263 entries, 0 to 262
Data columns (total 5 columns):
Country Code    263 non-null object
Region          217 non-null object
IncomeGroup     217 non-null object
SpecialNotes    250 non-null object
TableName       263 non-null object
```

		1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2010	2011	2012	2013	2014
IncomeGroup	Country Code																
High income	PAN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	51.6	51.3	51.7	51.5	50.
	URY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	44.5	42.2	39.9	40.5	40.
Lower middle income	GEO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	39.5	39.6	39.0	38.6	37.
	HND	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	53.1	56.2	56.1	52.6	50.
Lower middle income	IDN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	36.4	39.7	39.6	39.9	39.
	KGZ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	30.1	27.8	27.4	28.8	26.
Lower middle income	MDA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	32.1	30.6	29.2	28.5	26.
	CIV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	42.5	42.2	41.8	40.4	41.



5 variables



4 catégories d'IncomeGroup :

- High income
- Upper middle income
- Low middle income
- Low income



On veut quelques pays avec un indice de gini



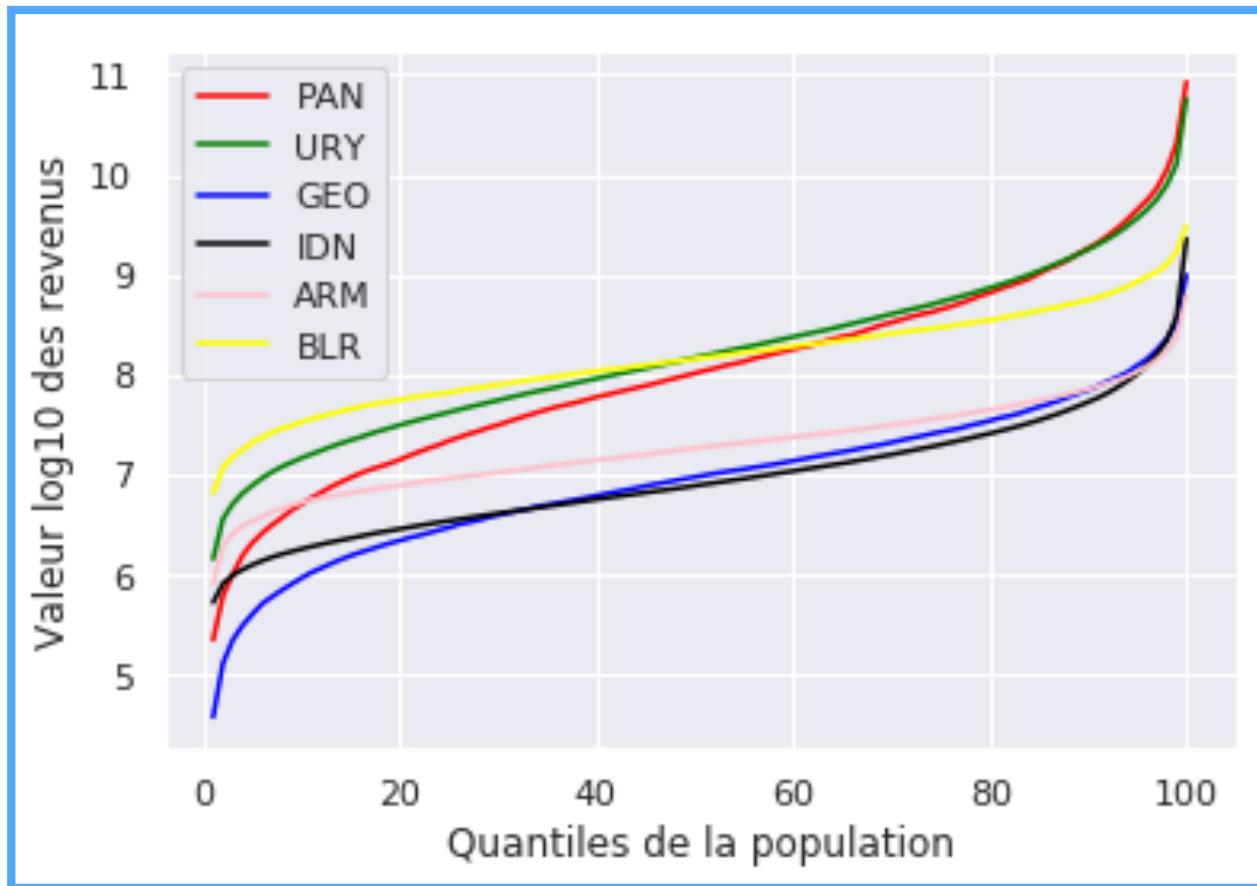
Avec aussi des classes différents

-High income : PAN, URY

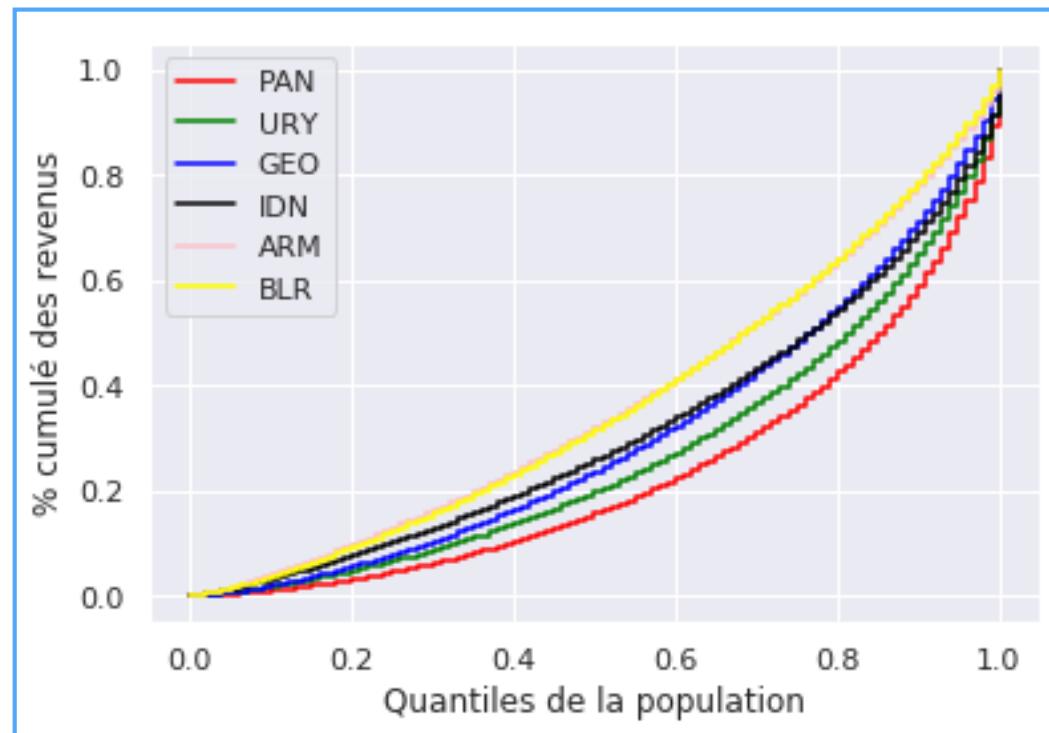
6 pays :
-Upper middle income : ARM, BLR

-Lower Middle Income : GEO, IDN

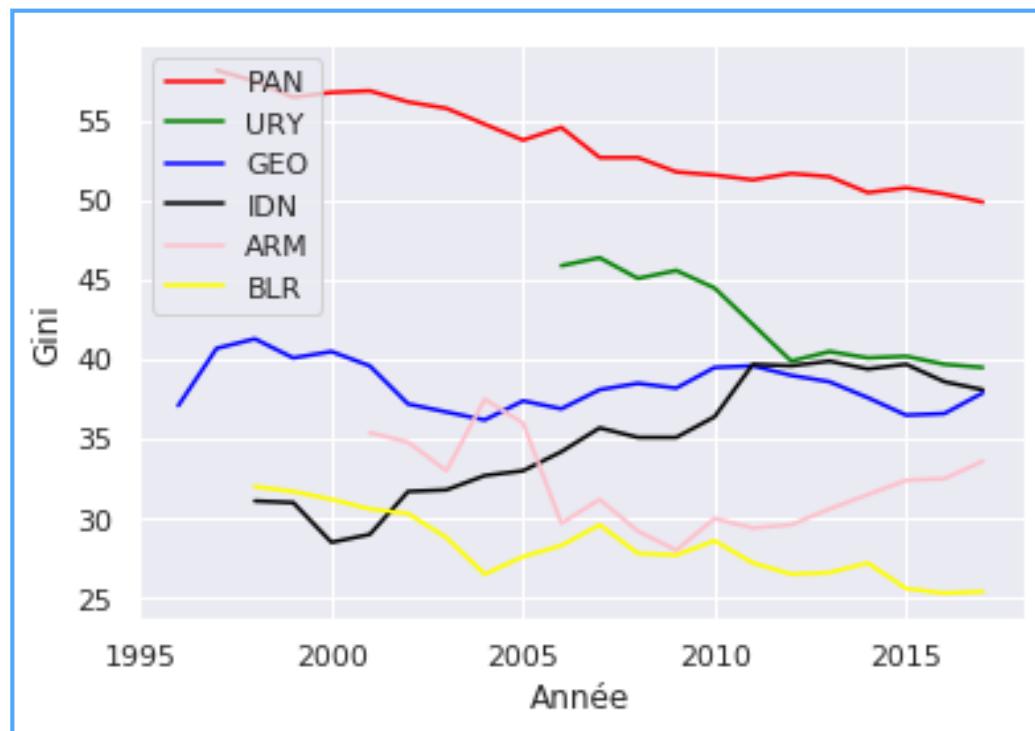
Diversité des pays en termes de distribution de revenus



Courbe de Lorenz

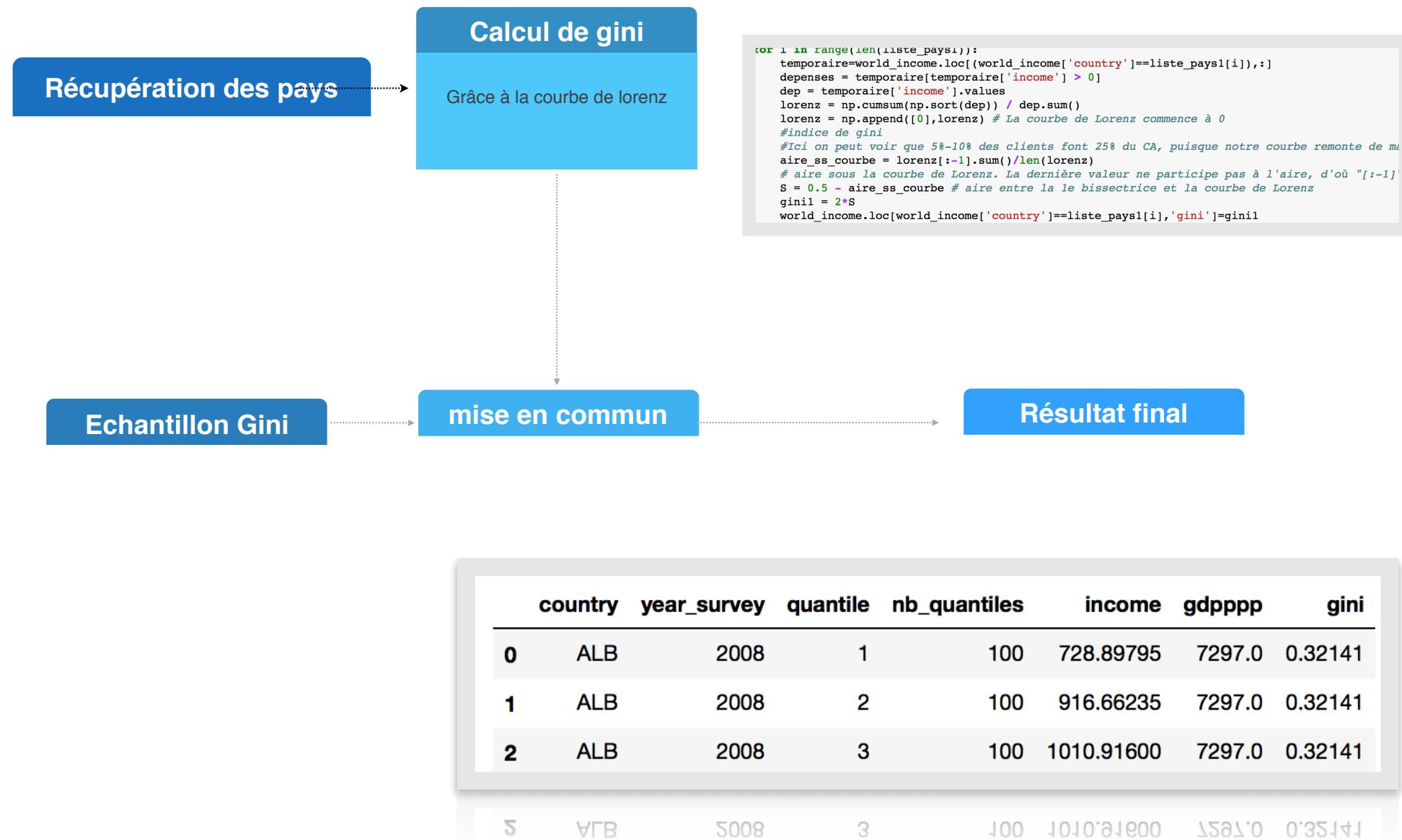


Indice de Gini





L'indice de Gini



Indice de gini

Les 5 premiers



	country	year_survey	quantile	nb_quantiles	income	gdpppp	gini	RANK
9599	SVN	2008	1	100	2814.95300	27197.0	24.824876	1.0
9499	SVK	2008	1	100	791.46204	20515.0	26.457318	2.0
2300	CZE	2008	1	100	1586.24710	23223.0	27.016228	3.0
9699	SWE	2008	1	100	2284.43290	34371.0	27.216580	4.0
10699	UKR	2008	1	100	942.38495	6721.0	27.241550	5.0

La france

La moyenne est de:
38,35

```
print(gini_ranking.loc[gini_ranking['country']=='FRA',:])
```

3300	FRA	2008	1	100	2958.304	30357.0
					gini	RANK
3300	34.563984				35.0	

	country	year_survey	quantile	nb_quantiles	income	gdpppp	gini	RANK
1100	BOL	2008	1	100	20.584948	3950.0	57.571868	72.0
1400	CAF	2008	1	100	40.928130	685.0	57.597241	73.0
2000	COL	2008	1	100	62.605060	8185.0	58.343686	74.0
4000	HND	2008	1	100	50.166843	3628.0	61.551164	75.0
11399	ZAF	2008	1	100	60.490383	9602.0	68.294901	76.0

Les 5 derniers

Coefficient d'élasticité

01

Il existe 5 années différentes

02

Sur les 5 années il n'y a qu'une valeur renseigné

	countryname	wbcode	iso3	region	incgroup2	incgroup4	fragile	survey	year	status	...	Cores2125_MACatC1	Shortfa
0	Afghanistan	AFG	AFG	South Asia	Developing economies	Low income	1	NRVA	1980	Co-residents only	...		NaN
1	Afghanistan	AFG	AFG	South Asia	Developing economies	Low income	1	NRVA	1980	Co-residents only	...		NaN
2	Afghanistan	AFG	AFG	South Asia	Developing economies	Low income	1	NRVA	1980	Co-residents only	...		NaN

03

Mais nous avons des infos utiles :
 -region
 -incgroup4
 -IGEincome

Coefficient d'élasticité

	region	incgroup4	IGEIncome
0	East Asia & Pacific	Lower middle income	0.527665
1	East Asia & Pacific	Upper middle income	0.469500
2	Europe & Central Asia	Lower middle income	0.424817
3	Europe & Central Asia	Upper middle income	0.477171
5	Latin America & Caribbean	Lower middle income	0.940737
6	Latin America & Caribbean	Upper middle income	0.878852
7	Middle East & North Africa	Lower middle income	0.916461
8	Middle East & North Africa	Upper middle income	0.517398
9	South Asia	Low income	0.436000
10	South Asia	Lower middle income	0.528167
11	South Asia	Upper middle income	Nan
12	Sub-Saharan Africa	Low income	0.673630
13	Sub-Saharan Africa	Lower middle income	0.629976
14	Sub-Saharan Africa	Upper middle income	0.677000

On a juste une petite valeur en NAN

On va utiliser le tableau qu'on nous a donné

	Base case	Optimistic (high mobility)	Pessimistic (low mobility)
Nordic European countries and Canada	0.2	0.15	0.3
Europe (except nordic countries)	0.4	0.3	0.5
Australia/New Zealand/USA	0.4	0.3	0.5
Asia	0.5	0.4	0.6
Latin America/Africa	0.66	0.5	0.9

Préparation du jeu de données pour la mission 3

coeff ancien

Correspond au coefficient renseigné à la base dans notre jeu de données

DataSet

Si coeff ancien n'existe pas alors on met le nouveau uniquement.

coeff nouveau

Coeff obtenu avec le merge de notre jeu de données

On multiplie par 500

index	country	quantile	nb_quantiles	gdpppp	income	gini	coeff_elas
0	ALB	1	100	7297.0	728.89795	0.32141	0.815874
11400	ALB	1	100	7297.0	728.89795	0.32141	0.815874
22800	ALB	1	100	7297.0	728.89795	0.32141	0.815874
34200	ALB	1	100	7297.0	728.89795	0.32141	0.815874
45600	ALB	1	100	7297.0	728.89795	0.32141	0.815874
57000	ALB	1	100	7297.0	728.89795	0.32141	0.815874
68400	ALB	1	100	7297.0	728.89795	0.32141	0.815874

Mission 3

```

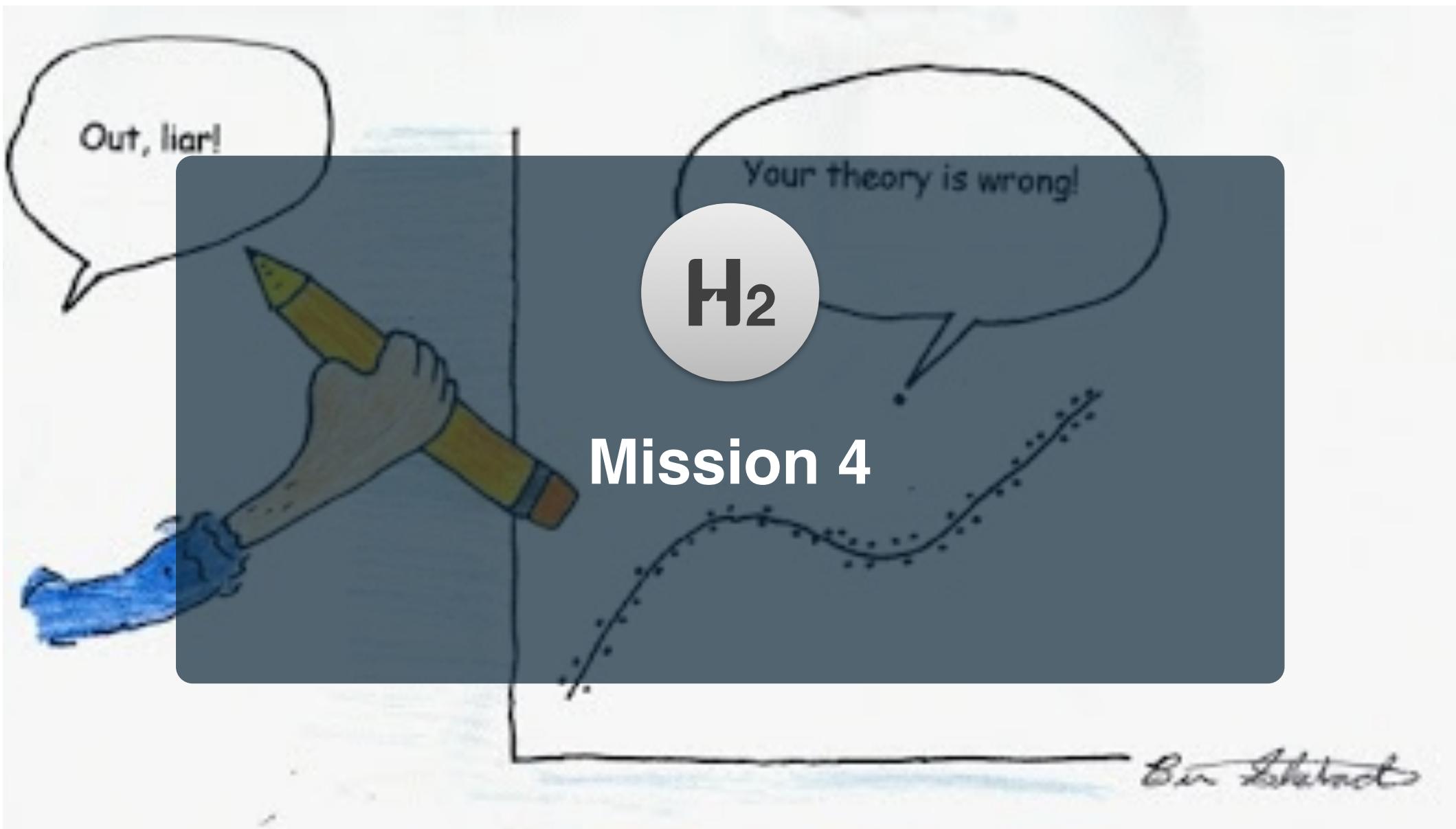
pays_list=big_data[ 'country' ].unique()
classe_parent=0
classe_enfant=0
list_prob=[ ]

for pays in pays_list:
    pj=big_data.loc[big_data[ 'country' ]==pays, 'coeff_elas'].iloc[0]
    nb_quantiles = 100 # nombre de quantiles (nombre de classes de revenu)
    n = 50000 # taille de l'échantillon
    y_child, y_parents = generate_incomes(n, pj)
    sample = compute_quantiles(y_child, y_parents, nb_quantiles)
    cd = conditional_distributions(sample, nb_quantiles)
    for c_i_child in range(100):
        for c_i_parent in range(100):
            p = proba_cond(c_i_parent, c_i_child, cd)
            #print("\nP(c_i_parent = {} | c_i_child = {}, pj = {}) = {}".format(c_i_parent,
            list_prob.append([c_i_parent+1]*(int(p*500)))

```

parent
enfant 8:5=>0,03

0,03x500=15
Donc on va assigner de 8 à 15 la classe 5



Anova

Mission 4

```
: question_1=big_data.groupby(['country','quantile']).mean().reset_index()
```

```
: import statsmodels.api as sm
import statsmodels.formula.api as smf
anova_variete = smf.ols('income~country', data=question_1).fit()
print(anova_variete.summary())
```

OLS Regression Results

```
=====
Dep. Variable:           income    R-squared:                 0.494
Model:                          OLS    Adj. R-squared:            0.489
Method:                         Least Squares    F-statistic:             97.62
Date:                Mon, 12 Aug 2019    Prob (F-statistic):        0.00
Time:                      11:30:22    Log-Likelihood:          -1.1667e+05
No. Observations:          11400    AIC:                  2.336e+05
Df Residuals:              11286    BIC:                  2.344e+05
Df Model:                           113
Covariance Type:            nonrobust
```

la variance expliquée =SCE/(SCE+SCR)

La variance expliquée du pays est de 45,8%

Analyse de l'anova

```
OLS Regression Results
=====
Dep. Variable: income    R-squared: 0.49
Model: OLS      Adj. R-squared: 0.48
Method: Least Squares F-statistic: 97.6
Date: Sat, 21 Sep 2019 Prob (F-statistic): 0.0
Time: 08:44:35 Log-Likelihood: -1.1667e+0
No. Observations: 11400 AIC: 2.336e+0
Df Residuals: 11286 BIC: 2.344e+0
Df Model: 113
Covariance Type: nonrobust
```

5.1 Test de fisher

Ce qui nous intéresse réellement, c'est le test de Fisher. La p-valeur de ce test (0) est très petite et largement inférieure à 5 %. On rejette donc l'hypothèse H₀ selon laquelle $\alpha_1=\alpha_2=\alpha_3=\alpha_4=0$.

Le pays a donc bien un effet sur les revenus

Mission 4

	country	quantile	nb_quantiles	gdpppp	income_normal	log	gini	coeff_elas	classe_parent	income_mean
0	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
1	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
2	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
3	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
4	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
5	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
6	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902
7	ALB	1	100	7297.0	728.89795	8.895219	0.32141	0.815874	1	2994.829902

Pour obtenir la moyenne
on réalise un groupe by puis un merge

Mission 4

2.1 régression linéaire

```
]: reg_multi = smf.ols('income_normal~income_mean+gini', data=regression_2).fit()  
print(reg_multi.summary())
```

OLS Regression Results

```
=====
```

Dep. Variable:	income_normal	R-squared:	0.494
Model:	OLS	Adj. R-squared:	0.494
Method:	Least Squares	F-statistic:	2.786e+06
Date:	Mon, 12 Aug 2019	Prob (F-statistic):	0.00
Time:	11:30:27	Log-Likelihood:	-5.8336e+07
No. Observations:	5700000	AIC:	1.167e+08
Df Residuals:	5699997	BIC:	1.167e+08
Df Model:	2		
Covariance Type:	nonrobust		

Regression pour l'income normal : 49,4%

variance

Income_mean	gini
45,89	8,5xe^-25

Mission 4

2.2 regression linéaire

```
[]: reg_multil = smf.ols('log_income~log_income_mean+gini', data=question3).fit()
print(reg_multil.summary())
```

OLS Regression Results

```
=====
Dep. Variable:          log_income    R-squared:                 0.728
Model:                          OLS    Adj. R-squared:             0.728
Method:                     Least Squares    F-statistic:            7.631e+06
Date:                Mon, 12 Aug 2019    Prob (F-statistic):       0.00
Time:                      11:30:31    Log-Likelihood:         -6.2259e+06
No. Observations:      5700000    AIC:                    1.245e+07
Df Residuals:           5699997    BIC:                    1.245e+07
Df Model:                           2
Covariance Type:            nonrobust
```

Regression pour log l'income normal : 72,8%

	log_Income_mean	gini
variance	69,4 %	8,5xe^-25

Mission 4

3.1 regression linéaire

```
4]: reg_multi2 = smf.ols('income_normal~income_mean+gini+classe_parent', data=question3).fit()  
print(reg_multi2.summary())
```

OLS Regression Results

```
=====
```

Dep. Variable:	income_normal	R-squared:	0.520
Model:	OLS	Adj. R-squared:	0.520
Method:	Least Squares	F-statistic:	2.058e+06
Date:	Mon, 12 Aug 2019	Prob (F-statistic):	0.00
Time:	11:30:35	Log-Likelihood:	-5.8187e+07
No. Observations:	5700000	AIC:	1.164e+08
Df Residuals:	5699996	BIC:	1.164e+08
Df Model:	3		
Covariance Type:	nonrobust		

Regression pour l'income normal : 52%

	Income_mean	gini	classe_parent
variance	45,89	8,5xe^-25	5,08

Mission 4

3.2 regression linéaire

```
7]: reg_multi3 = smf.ols('log_income~log_income_mean+gini+classe_parent', data=question3).fit()
print(reg_multi3.summary())
```

OLS Regression Results

Dep. Variable:	log_income	R-squared:	0.783
Model:	OLS	Adj. R-squared:	0.783
Method:	Least Squares	F-statistic:	6.862e+06
Date:	Mon, 12 Aug 2019	Prob (F-statistic):	0.00
Time:	11:30:38	Log-Likelihood:	-5.5807e+06
No. Observations:	5700000	AIC:	1.116e+07
Df Residuals:	5699996	BIC:	1.116e+07
Df Model:	3		
Covariance Type:	nonrobust		

Regression pour log income normal :

78%

	log_Income_mean	gini	classe_parent
variance	45,89	8,5xe^-25	0,5 %

Mission 4

**Regression pour l'income normal :
49,4%**

variance	Income_mean	gini
	45,89	≈0

**Regression pour log l'income normal :
72,8%**

variance	log_Income_mean	gini
	69,4 %	≈0

Regression pour l'income normal : 52%

variance	Income_me an	gini	classe_parent
	45,89	≈0	5,08

**Regression pour log income
normal : 78%**

variance	log_Incom e_mean	gini	classe_parent
	69,4	≈0	5,08

**En observant le coefficient de régression associé à l'indice de Gini,
peut-on affirmer que le
fait de vivre dans un pays plus inégalitaire favorise plus de personnes
qu'il n'en défavorise ?**

```
reg_multi3 = smf.ols('log_income~log_income_mean+gini+classe_parent', data=ques)
print(reg_multi3.summary())

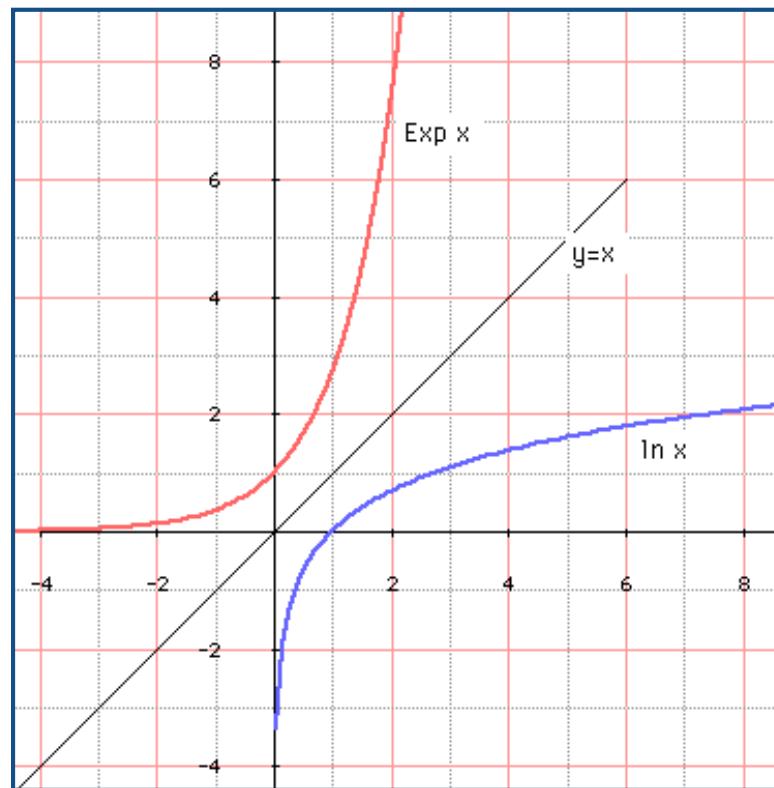
OLS Regression Results
-----
Dep. Variable: log_income R-squared: 0.783
Model: OLS Adj. R-squared: 0.783
Method: Least Squares F-statistic: 6.858e+06
Date: Sat, 21 Sep 2019 Prob (F-statistic): 0.00
Time: 08:44:49 Log-Likelihood: -5.5821e+06
No. Observations: 5700000 AIC: 1.116e+07
Df Residuals: 5699996 BIC: 1.116e+07
Df Model: 3
Covariance Type: nonrobust
-----
975]
-----
         coef  std err      t  P>|t|    [0.025    0.
Intercept -0.0711  0.003   -26.549  0.000   -0.076  -
log_income_mean 0.9864  0.000  4029.558  0.000    0.986
0.987
gini        -1.6527  0.003  -524.272  0.000   -1.659  -
1.646
classe_parent 0.0112  9.35e-06 1201.930  0.000    0.011
0.011
-----
Omnibus: 385852.822 Durbin-Watson: 0.007
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1819245.069
Skew: -0.133 Prob(JB): 0.00
Kurtosis: 5.755 Cond. No. 824.
-----
```

Si on fait

- Plus le gini augmente plus le revenu va diminuer à cause de son coefficient négatif.
- Un pays est plus inégalitaire si son indice de gini est élevé.
- Donc plus il y a un gini important, plus ça va défavoriser des personnes

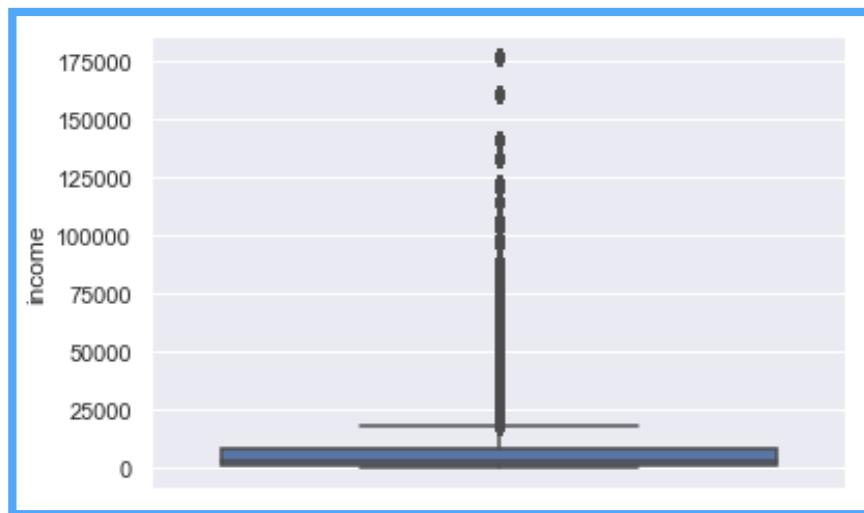
Mission 4

Linéarisation de modèle avec le log

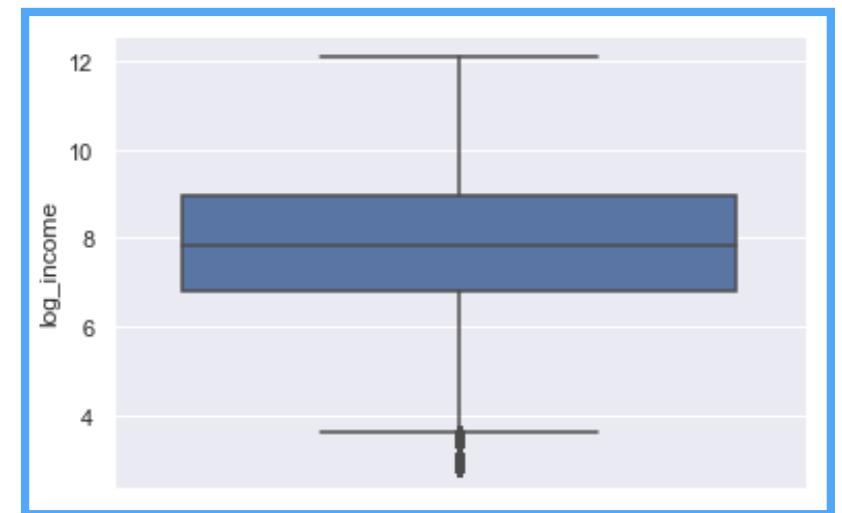


L'effet du logarithme sur les outliers

Avant



Après



Mission 4

Vérifier la colinéarité des variables

Une autre chose à vérifier est l'éventuelle colinéarité approchée des variables :

```
In[48]: variables = reg_multi1.model.exog  
[variance_inflation_factor(variables, i) for i in np.arange(1,variables.shape[1])]  
  
Out[48]: [1.0822115727471704, 1.082211572747236]
```

Ici, tous les coefficients sont inférieurs à 10, il n'y a donc pas de problème de colinéarité.

9.4 Vérifier la colinéarité des variables

Une autre chose à vérifier est l'éventuelle colinéarité approchée des variables :

```
variables = reg_multi3.model.exog  
[variance_inflation_factor(variables, i) for i in np.arange(1,variables.shape[1])]  
  
[1.082211572747219, 1.0822115727472146, 1.0]
```

Ici, tous les coefficients sont inférieurs à 10, il n'y a donc pas de problème de colinéarité.

Mission 4

Testez l'homoscédasticité

On peut également tester l'homoscédasticité (c'est-à-dire la constance de la variance) des résidus :

```
Entrée [49]: __, pval, __, f_pval = statsmodels.stats.diagnostic.het_breuschpagan(reg_multil.resid, variabl
print('p value test Breusch Pagan:', pval)

p value test Breusch Pagan: 0.0
```

9.5 Testez l'homoscédasticité ¶

On peut également tester l'homoscédasticité (c'est-à-dire la constance de la variance) des résidus

:

```
] __, pval, __, f_pval = statsmodels.stats.diagnostic.het_breuschpagan(reg_multil3.
print('p value test Breusch Pagan:', pval)

p value test Breusch Pagan: 0.0
```

La p-valeur ici est inférieure à 5 %, on rejette l'hypothèse H_0 selon laquelle les variances sont constantes (l'hypothèse d'homoscédasticité).

H0 est rejeté puisque notre P-value<5%

Mission 4

Testez la normalité des résidus

Si l'on veut tester la normalité des résidus, on peut faire un test de Shapiro-Wilk.

Entrée [50]: `shapiro(reg_multi1.resid)`

```
/home/arnaud/.local/lib/python3.5/site-packages/scipy/stats/morestats.py:1660: UserWarning: p  
-value may not be accurate for N > 5000.  
    warnings.warn("p-value may not be accurate for N > 5000.")
```

Out[50]: `(0.9774155616760254, 0.0)`

9.6 Testez la normalité des résidus

Si l'on veut tester la normalité des résidus, on peut faire un test de Shapiro-Wilk.

: `shapiro(reg_multi3.resid)`

```
C:\Users\Arnaud\Anaconda3\lib\site-packages\scipy\stats\morestats.py:1309: User  
Warning: p-value may not be accurate for N > 5000.  
    warnings.warn("p-value may not be accurate for N > 5000.")
```

`(0.9745134711265564, 0.0)`

Néanmoins, l'observation des résidus, le fait qu'ils ne soient pas très différents d'une distribution symétrique, et le fait que l'échantillon soit de taille suffisante (supérieure à 30) permettent de dire que les résultats obtenus par le modèle linéaire gaussien ne sont pas absurdes, même si le résidu n'est pas considéré comme étant gaussien

Nos résidus sont ils distribués selon une loi normale ?

```
sm.qqplot(residus_reg3,line='45')
```

