Projet 7

Effectuez une prédiction de revenus

Perform an income prediction

Notre objectif

Créer un modèle permettant de déterminer le potentiel de revenu d'un individu, sur la base de l'indice de Gini, du revenu des parents et du pays de résidence.

Notre jeu de données

Les données que nous allons utiliser sont:

- Données sur la distribution des revenus dans le monde
- Données sur l'indice de Gini
- Données sur la population mondiale

Les sources sont la Banque mondiale et diverses sources ouvertes (kaggle, wiki).

```
world_income.head(3)
```

	country	year_survey	quantile	nb_quantiles	income	gdpppp
0	ALB	2008	1	100	728.90	7,297.00
1	ALB	2008	2	100	916.66	7,297.00
2	ALB	2008	3	100	1,010.92	7,297.00

Notre jeu de données

```python
# add missing data for Palestine and Kosovo from the same source
# https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.CD
missing_data.head(3)
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2012	2013	2014
0	Aruba	ABW	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	33567.5500169846	36829.0327743518	36779.429429343
1	Africa Eastern and Southern	AFE	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	3235.16335913109	3362.86880917301	3499.13287771861
2	Afghanistan	AFG	GDP per capita, PPP (current international $)	NY.GDP.PCAP.PP.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	1914.77422837964	2015.51477466948	2069.42402167356

TWN est Taiwan.
Selon une recherche sur Internet, TWN comptait 23 037 031 habitants en 2008.
Nous pouvons insérer les données brutes directement (méthode plus simple par rapport à PSE et XKX).

TWN is Taiwan.
according to internet search, TWN in 2008 had 23,037,031 inhabitants.
We can insert raw data directly (simpler way compared to PSE and XKX).
https://en.wikipedia.org/wiki/Demographics_of_Taiwan#Population_census

```python
df_global.loc[df_global['country'] == 'TWN', 'country_name'] = 'Taïwan'
df_global.loc[df_global['country'] == 'TWN', 'population'] = 23037031
df_global['population'] = df_global['population'].astype(int)
```

Notre jeu de données

Population covered by analysis ¶

```
# Population couverte dans df_global (toutes les années que nous avons)
# population covered in df_global (all years that we have)

print(str(our_pop * 100 / tp_avg) + ' %')
```

```
91.7358339312185 %
```

years of data used

```
# Nombre d'années de données utilisées
world_income.groupby('year_survey').nunique()
```

year_survey	country	quantile	nb_quantiles	income	gdpppp
2004	1	100	1	100	1
2006	5	100	1	500	5
2007	15	100	1	1500	15
2008	76	100	1	7599	76
2009	12	100	1	1200	12
2010	6	100	1	600	6
2011	1	100	1	100	1

2008 has the most amout of countries (76)

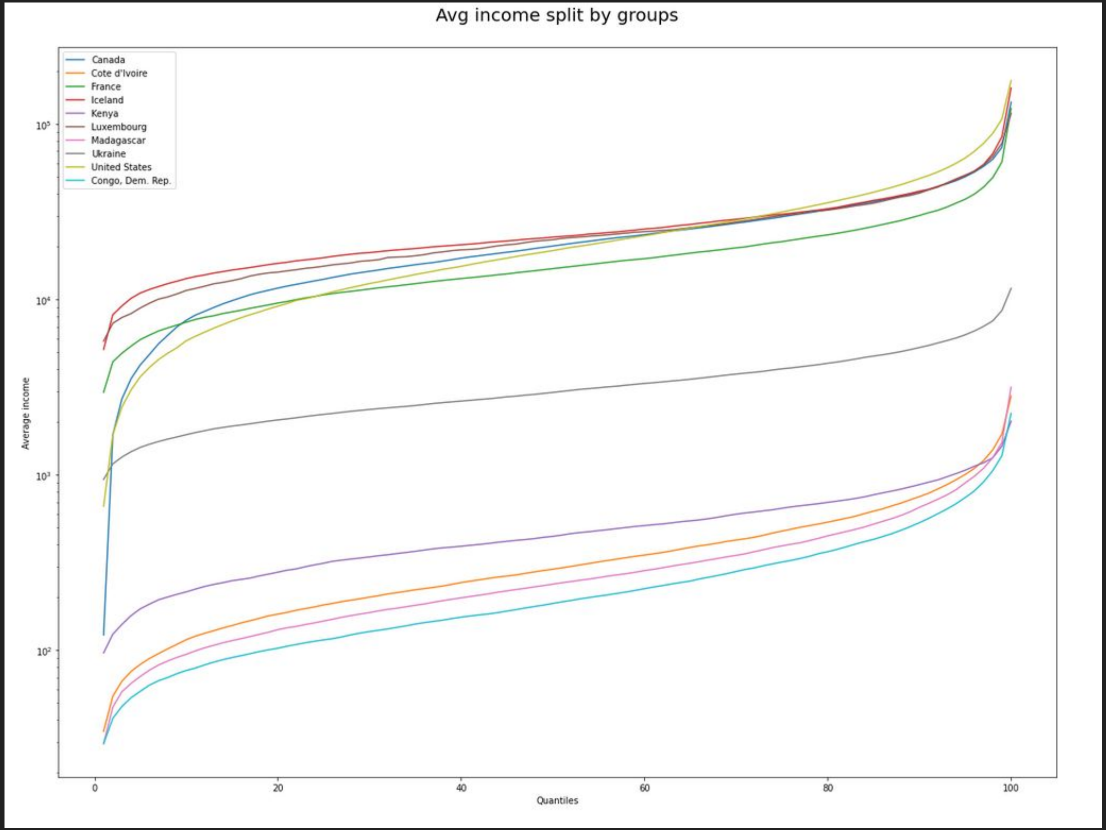
Number of countries

```
world_income['country'].nunique()
```

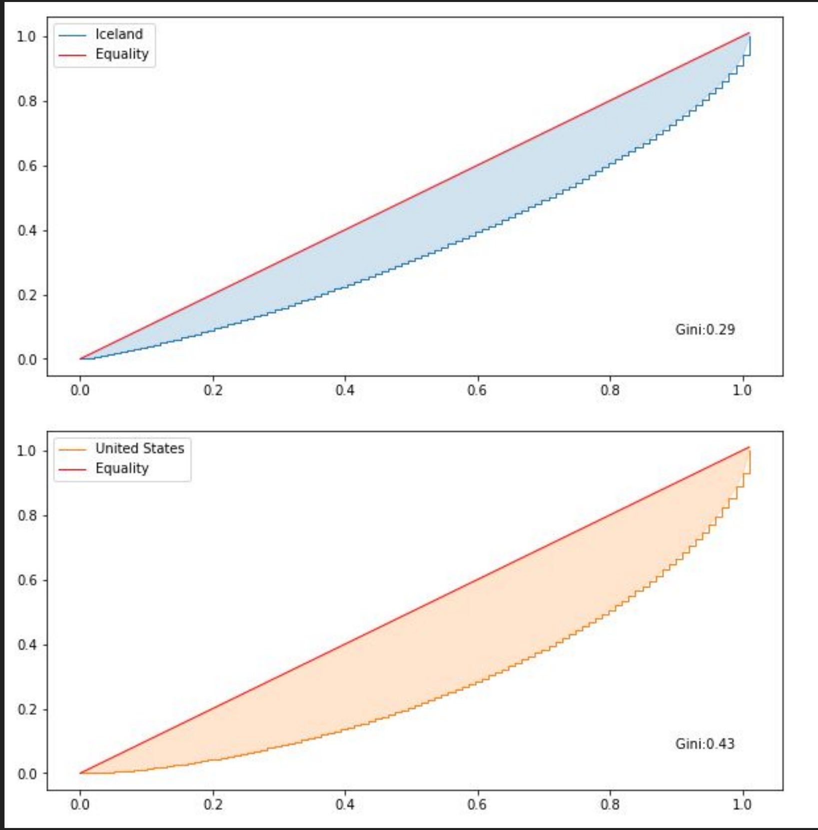
```
116
```

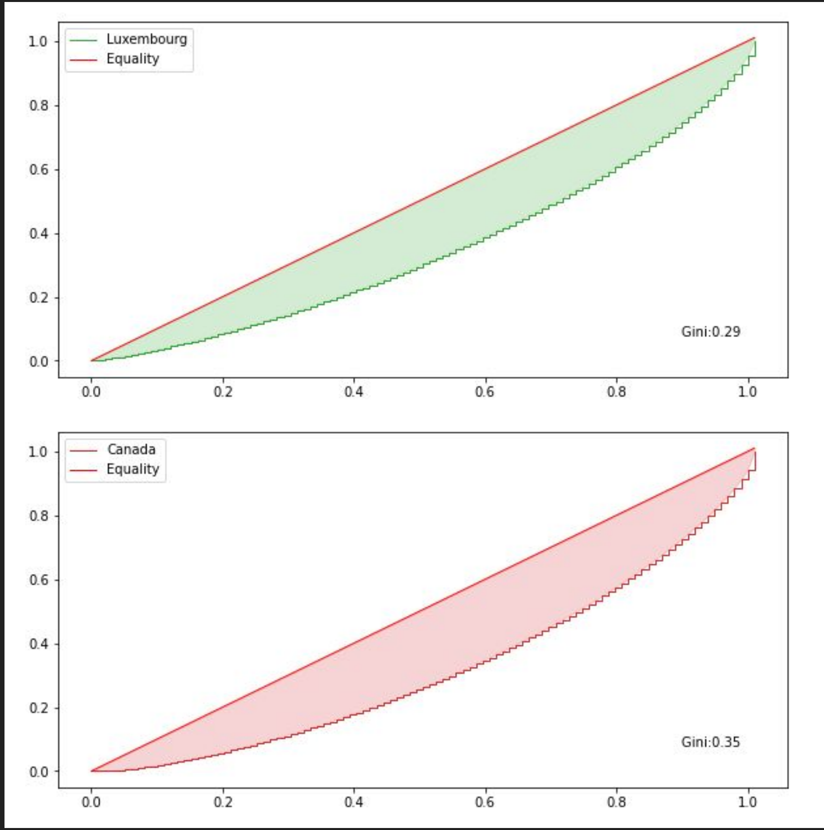
We have 116 countries. 100 quantiles per country means 11600 rows. We miss one row.

Diversité des pays en termes de distribution de revenus

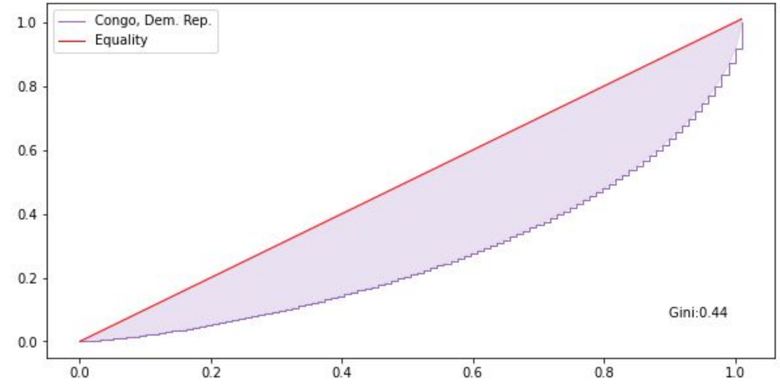


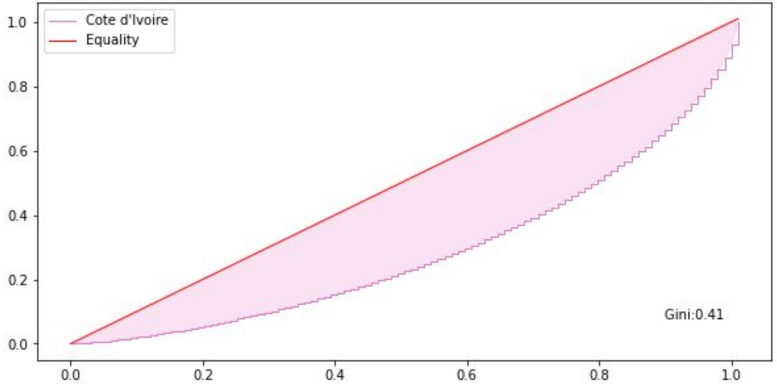
Courbe de Lorenz

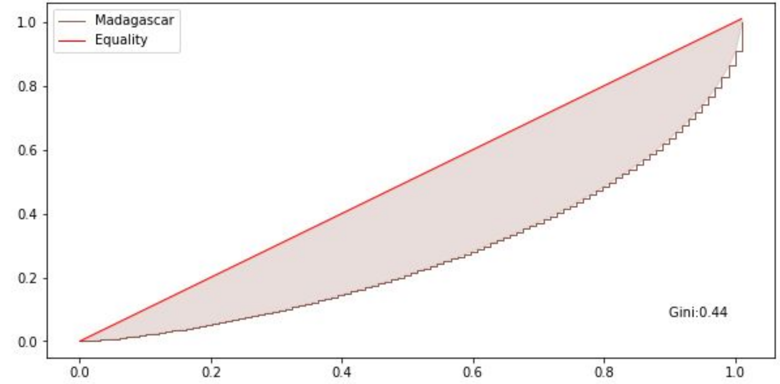


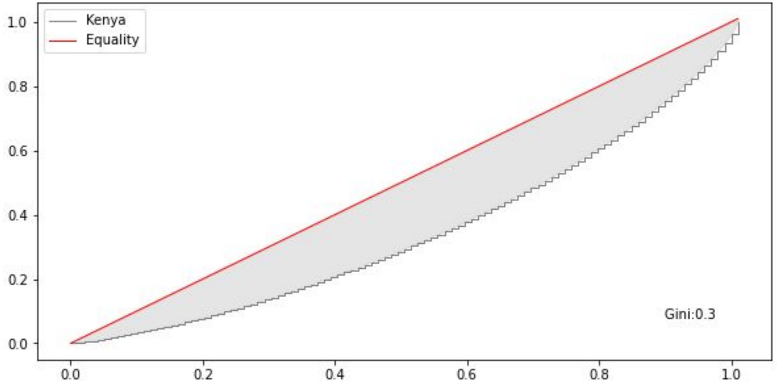


Courbe de Lorenz

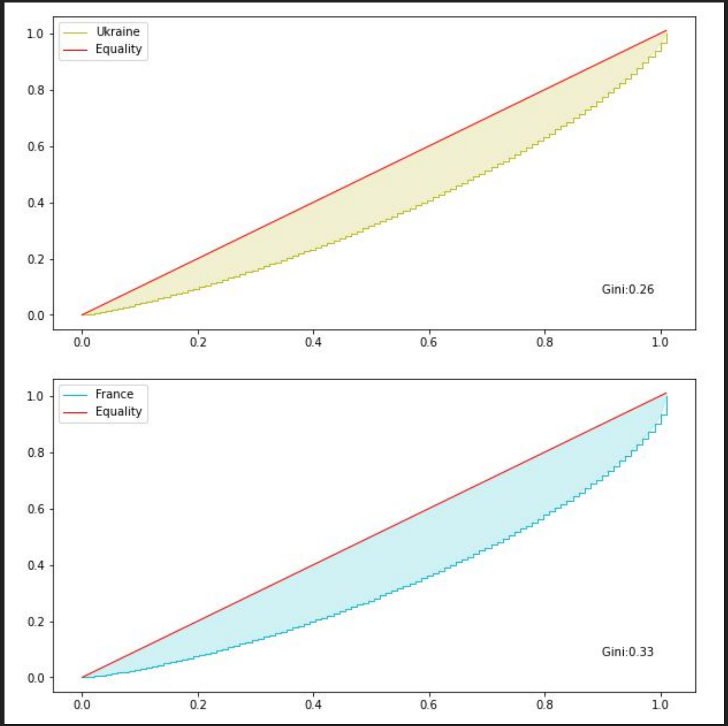


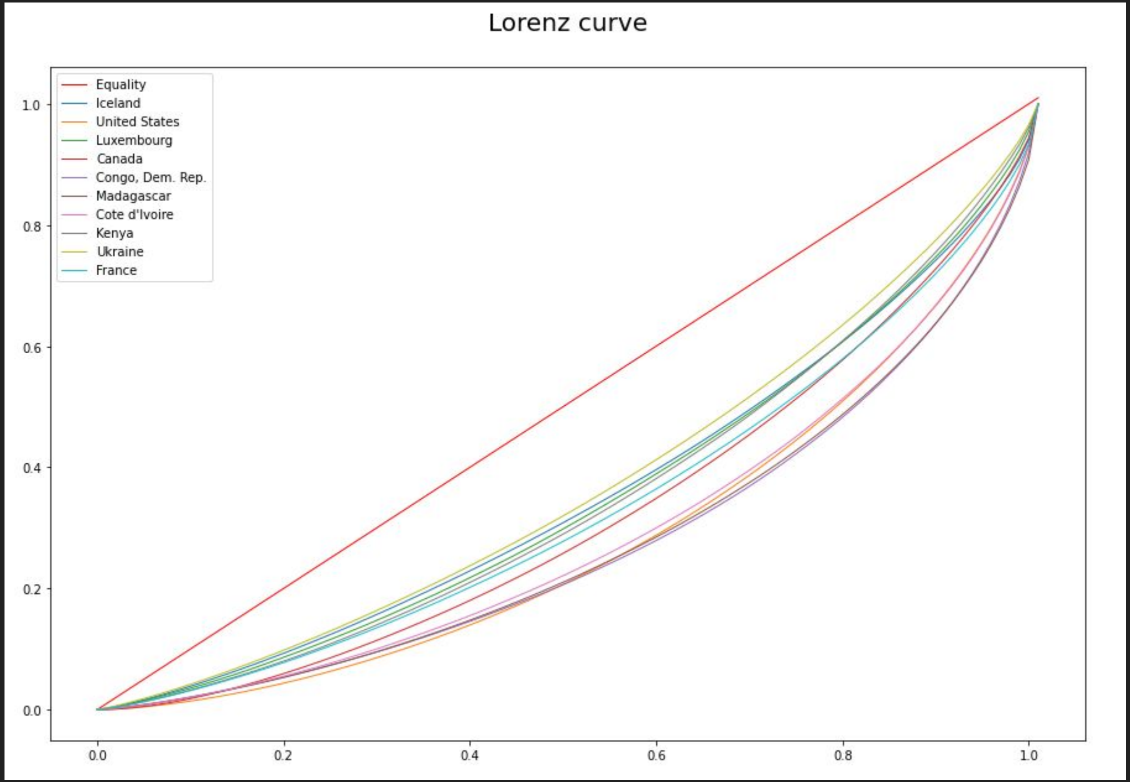




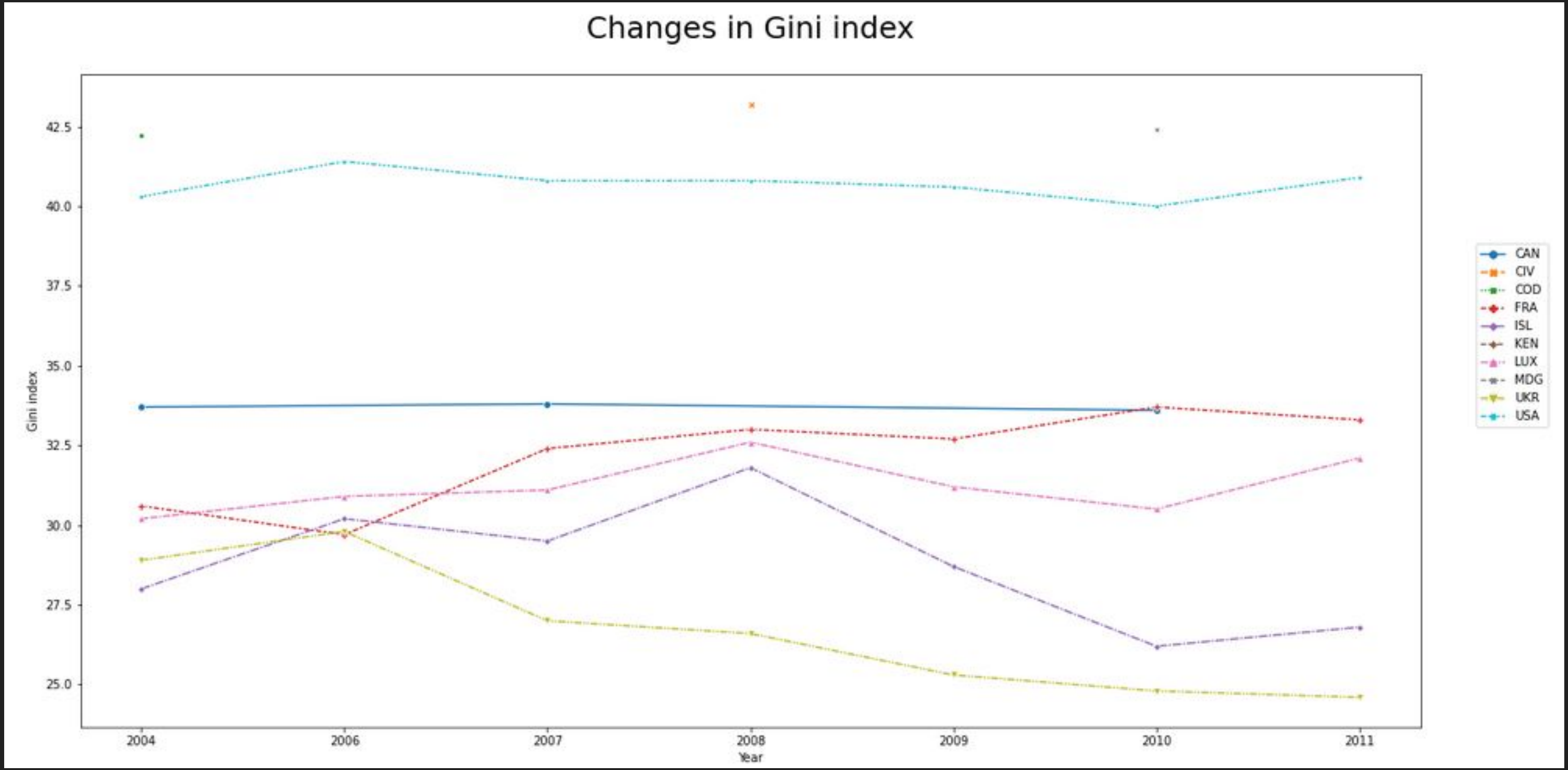


Courbe de Lorenz





L'évolution de l'indice de Gini



L'indice de Gini

Les 5 pays les plus égalitaires, les 5 pays les plus inégalitaires et la France

	gini_manual
country_name	
Slovenia	24.82
Slovak Republic	26.46
Czech Republic	27.02
Sweden	27.22
Ukraine	27.24

	gini_manual
country_name	
South Africa	68.29
Honduras	61.55
Colombia	58.34
Guatemala	58.25
Central African Republic	57.60

	country_name	gini_manual
36	France	34.56

Task 3

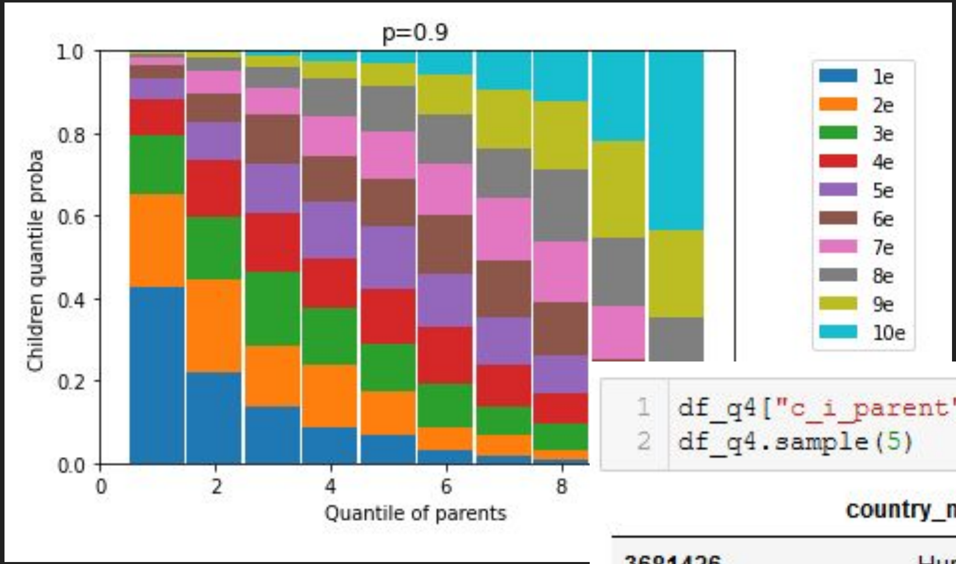
```
sample = compute_quantiles(y_child, y_parents, nb_quantiles)
sample.head()
```

	y_child	y_parents	c_i_child	c_i_parent
0	2.42	3.84	75	92
1	1.36	1.73	59	71
2	0.62	1.14	36	55
3	0.38	0.82	24	42
4	2.32	4.80	74	95

```
cd = conditional_distributions(sample, nb_quantiles)

c_i_child = 5
c_i_parent = 8

p = proba_cond(c_i_parent, c_i_child, cd)

print("P(c_i_parent = {} | c_i_child = {}, pj = {}) = {}".format(c_i_parent, c_i_child, pj, p))
```

```
P(c_i_parent = 8 | c_i_child = 5, pj = 0.9) = 0.031
```

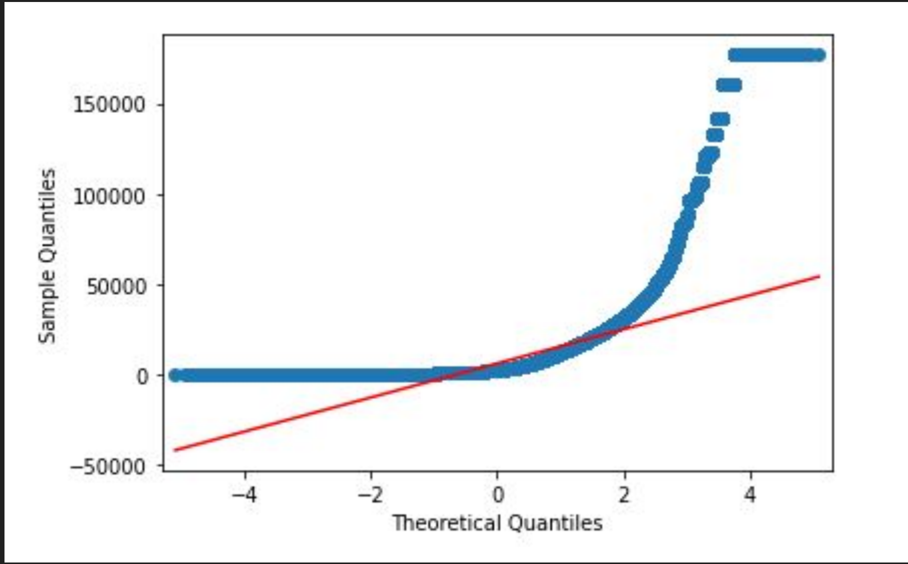
Task 3

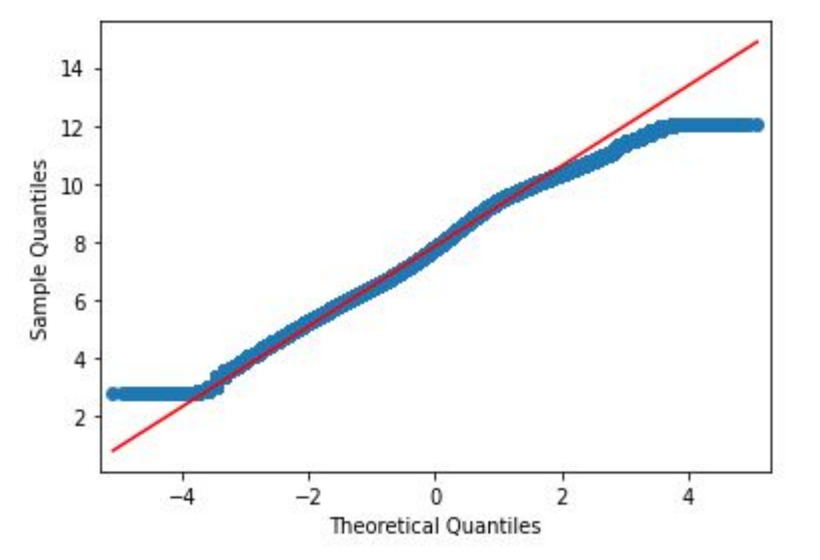



```
df_q4["c_i_parent"] = centiles_parent
df_q4.sample(5)
```

	country_name	country	region	c_i_child	y_child	elasticity	c_i_parent
3681426	Hungary	HUN	Europe	27	4,016.00	0.39	81
3179806	Central African Republic	CAF	Africa	7	113.78	0.71	5
4511410	Uganda	UGA	Africa	11	323.37	1.03	25
175716	China	CHN	Asia	17	666.41	0.40	9
455363	Spain	ESP	Europe	64	13,904.82	0.42	81

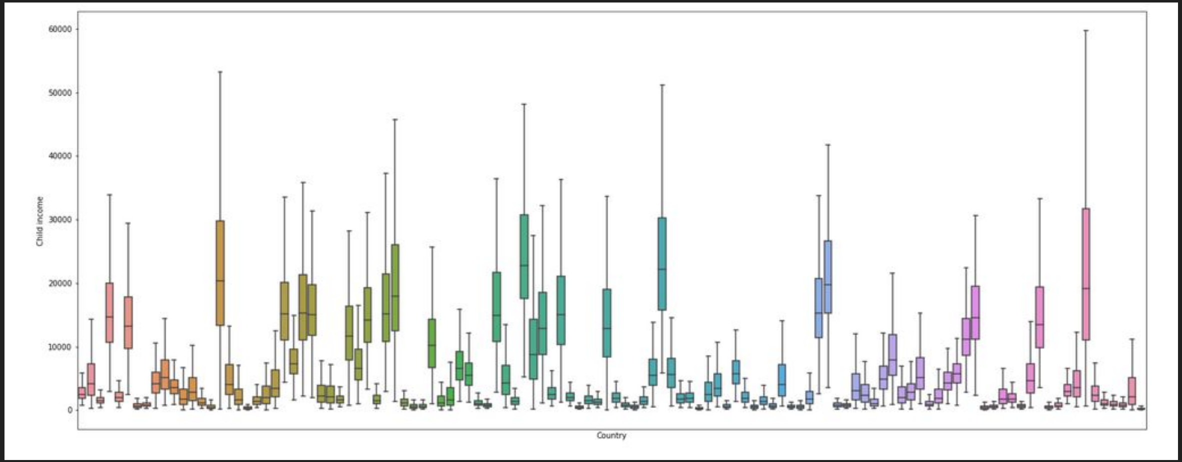
Modèle de prédiction

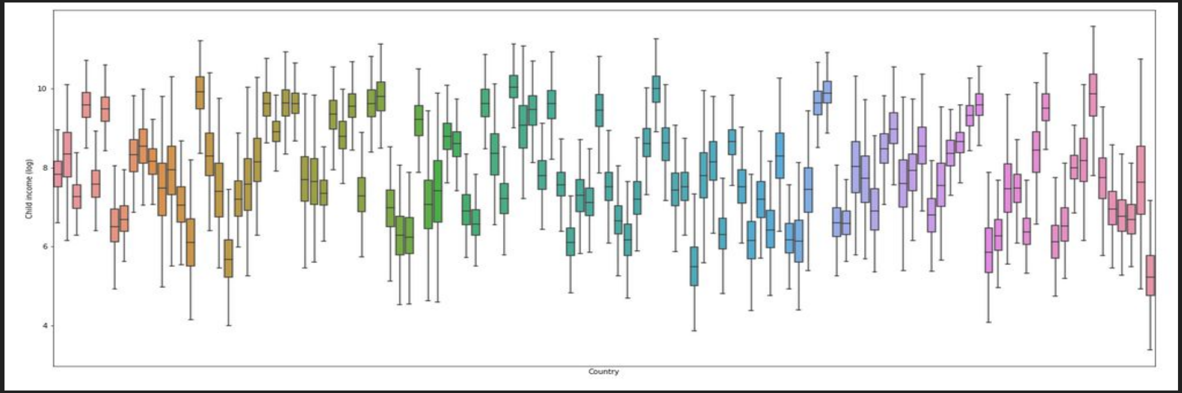




	W	pval	equal_var
levene	12,688.87	0.00	False

Modèle de prédiction





Modèle de prédiction

ANOVA

```
# ANOVA with country as single explanatory variable
pg.anova(data=df_q4, dv='y_child', between='country_name', detailed=True)

# ANOVA summary:
# 'Source': Factor names
# 'SS': Sums of squares
# 'DF': Degrees of freedom
# 'MS': Mean squares
# 'F': F-values
# 'p-unc': uncorrected p-values
# 'np2': Partial eta-square effect sizes
```

	Source	SS	DF	MS	F	p-unc	np2
0	country_name	255,118,762,151,916.69	115	2,218,424,018,712.32	49,710.76	0.00	0.50
1	Within	258,829,321,031,696.16	5799884	44,626,637.54	NaN	NaN	NaN

Modèle de prédiction

Linear regression v1

```python
# v1 is predicting y_child using Gj (gini) and mj (income)

regr_v1 = smf.ols('y_child ~ Gj + mj', data=df_q4).fit()
regr_v1.summary()
```

OLS Regression Results

Dep. Variable:	y_child	**R-squared:**	0.496
Model:	OLS	**Adj. R-squared:**	0.496
Method:	Least Squares	**F-statistic:**	2.858e+06
Date:	Mon, 28 Feb 2022	**Prob (F-statistic):**	0.00
Time:	03:44:06	**Log-Likelihood:**	-5.9310e+07
No. Observations:	5800000	**AIC:**	1.186e+08
Df Residuals:	5799997	**BIC:**	1.186e+08
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.167e-08	14.699	-1.47e-09	1.000	-28.809	28.809
Gj	4.163e-10	0.335	1.24e-09	1.000	-0.657	0.657
mj	1.0000	0.000	2234.874	0.000	0.999	1.001

Omnibus:	7299083.278	**Durbin-Watson:**	0.685
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2103715771.076
Skew:	6.739	**Prob(JB):**	0.00
Kurtosis:	95.322	**Cond. No.**	4.77e+04

```python
decomposition_of_variances(regr_v1)
```

	Explained Variance (%)
Gj	6.27
mj	43.37
Residual	50.36

```python
decomposition_of_variances(regr_v2)
```

	Explained Variance (%)
log_mj	71.84
Gj	1.04
Residual	27.12

Linear regression v2

```python
# transform data for the log scale
df_q4['log_y_child'] = np.log(df_q4['y_child'])
df_q4['log_mj'] = np.log(df_q4['mj'])
```

```python
regr_v2 = smf.ols('log_y_child ~ log_mj + Gj', data=df_q4).fit()
regr_v2.summary()
```

OLS Regression Results

Dep. Variable:	log_y_child	**R-squared:**	0.729
Model:	OLS	**Adj. R-squared:**	0.729
Method:	Least Squares	**F-statistic:**	7.793e+06
Date:	Mon, 28 Feb 2022	**Prob (F-statistic):**	0.00
Time:	03:44:11	**Log-Likelihood:**	-6.3181e+06
No. Observations:	5800000	**AIC:**	1.264e+07
Df Residuals:	5799997	**BIC:**	1.264e+07
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.4961	0.003	170.994	0.000	0.490	0.502
log_mj	0.9864	0.000	3651.055	0.000	0.986	0.987
Gj	-0.0165	3.5e-05	-471.886	0.000	-0.017	-0.016

Omnibus:	372790.841	**Durbin-Watson:**	0.390
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1752380.549
Skew:	-0.081	**Prob(JB):**	0.00
Kurtosis:	5.688	**Cond. No.**	402.

Modèle de prédiction

Linear regression v3

```
regr_v3 = smf.ols('log_y_child ~ log_mj + Gj + c_i_parent', data=df_q4).fit()
regr_v3.summary()
```

OLS Regression Results

Dep. Variable:	log_y_child	**R-squared:**	0.800
Model:	OLS	**Adj. R-squared:**	0.800
Method:	Least Squares	**F-statistic:**	7.732e+06
Date:	Mon, 28 Feb 2022	**Prob (F-statistic):**	0.00
Time:	03:44:16	**Log-Likelihood:**	-5.4354e+06
No. Observations:	5800000	**AIC:**	1.087e+07
Df Residuals:	5799996	**BIC:**	1.087e+07
Df Model:	3		
Covariance Type:	nonrobust		

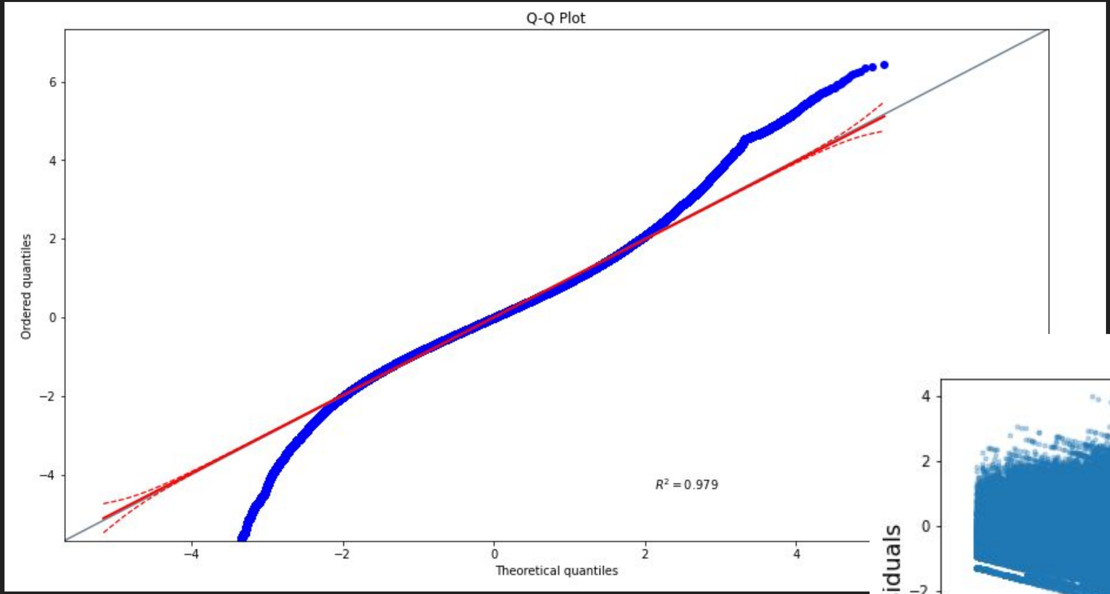
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.1479	0.003	-58.432	0.000	-0.153	-0.143
log_mj	0.9862	0.000	4250.672	0.000	0.986	0.987
Gj	-0.0165	3.01e-05	-548.965	0.000	-0.017	-0.016
c_i_parent	0.0128	8.88e-06	1436.507	0.000	0.013	0.013

Omnibus:	423568.092	**Durbin-Watson:**	0.826
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2077330.002
Skew:	-0.168	**Prob(JB):**	0.00
Kurtosis:	5.913	**Cond. No.**	682.

```
decomposition_of_variances(regr_v3)
```

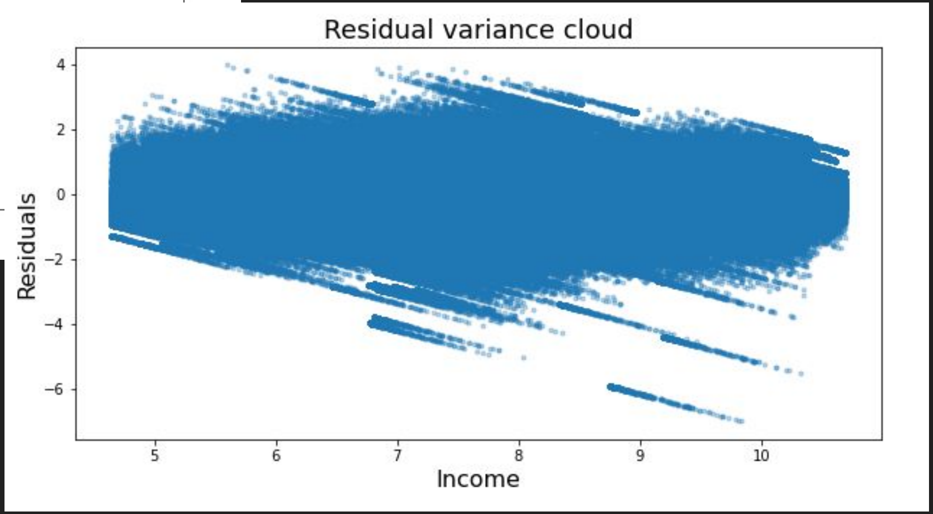
	Explained Variance (%)
log_mj	71.84
Gj	1.04
c_i_parent	7.12
Residual	20.00

Modèle de prédiction




```
VIF(regr_v3, ['log_mj', 'Gj', 'c_i_parent'])
```

	VIF
log_mj	1.08
Gj	1.08
c_i_parent	1.00



Conclusions générales après l'analyse effectuée

Globalement, nous pouvons voir que les données brutes ne sont pas très performantes avec nos modèles. Après normalisation, les performances peuvent être améliorées.
Le troisième modèle de régression linéaire a la meilleure performance (0.8). Les variables ont des valeurs p de 0, ce qui signifie qu'elles fonctionnent pour notre modèle.
Cependant, en termes d'impact, nous pouvons voir que le revenu est le facteur principal pour prédire le revenu d'un enfant, l'indice de Gini a un impact minimal, le quantile parental n'en est pas loin, le reste dépend d'autres facteurs, selon le modèle.

Overall we can see that raw data is not performing well with our models. After normalization performance can be improved.
Third linear regression model has the best performance (0.8). Variables have pvalues of 0 meaning they do work for our model.
However in terms of impact we can see that income is the major factor in predicting income of a child, gini index has minimal impact, parental quantile is not far from it, the rest depends on other factors, according to the model.