Data Analytics

[Welfare and Inequality Analysis]

Hye-Jin Cho-Drugeon
Choh9323@gmail.com

June, 2023

Table of content

Introduction

Business Use Case

Goal: This project studies intergenerational transfers such as tax and social security in the field of welfare and inequality for Euro-currency using countries with the OECD distribution data. It intends to explain the behavior of working age generation of 18-65 and old generation of above 65.

Domain: Economics and Statistics

Output

In Github [DAFT_0410/module5 at main · chatlapin/DAFT_0410 · GitHub]

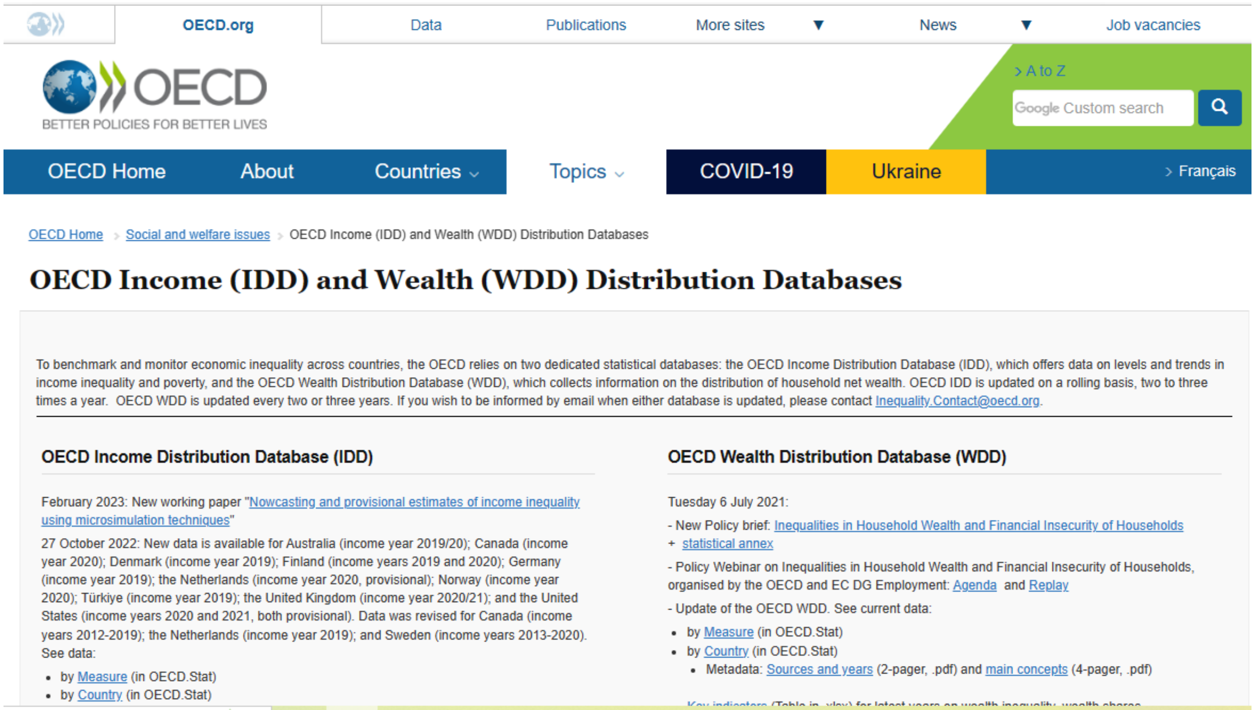
Erdiagram_OECD.drawio.pdf
Jira
OECD Data Cleaning and EDA.ipynb
OECD.sql
OECD.API31.ipynb
OECDRawData.ipynb
Project Toymodel.ipynb
UnitRootTestOECD meandi.ipynb
UnitRootTestOECDCI.ipynb
UnitRootTestOECDCPI.ipynb
UnitRootTestOECDP90P10.ipynb
UnitRootTestOECDTE.ipynb
UnitRootTestOECDtaxsecu.ipynb
UnitRootTestOECD transfer.ipynb
France.csv
Welfare_DataVisualisation.xlsx

Plan

1. Planning of my project in Jira
2. Code in Python for Data collection and cleaning
3. ER Diagram
4. Data source and Meta Data
5. Database Script
6. Report (10 pages)
7. Slides

Jira https://chatlapin.atlassian.net/jira/software/projects/CHAT/boards/1

Data and data sources



RAW data from: https://www.oecd.org/social/income-distribution-database.htm#:~:text=The%20OECD%20Income%20Distribution%20Database%20provides%20information%20on%20the%20equivalised,households%20on%20a%20comparable%20basis.

Data collection

Income components, disposable, market and primary income

From Term Reference of OECD in 2017-2018 (ref: IDD-ToR.pdf (oecd.org), OECD project on the distribution of household incomes studies the relationship between welfare and inequality. Income distributions refer to a particular year, which should be indicated in the Excel spreadsheet "Metadata". All income components should be reported on an annual basis and in nominal prices. Five main components of household disposable income are identified in the OECD questionnaire:

-E: employee income, including wages and salaries, cash bonuses and gratuities, commissions and tips, directors' fees, profit sharing bonuses and other forms of profit-related pay, shares offered as part of employee remuneration, free and subsidized goods and services from an employer, severance and termination pay.1 Sick pay paid by social security should also be included.

-KI 2: capital and property income, including income from financial assets (net of expenses), income from non-financial assets (net of expenses) and royalties. Regular receipts from voluntary individual private pension plans and life insurance schemes should also be included in this income component. In line with the 2011 Canberra Handbook, capital gains should not be included in KI.

-SEI 3: income from self-employment, including profits and losses from unincorporated enterprises, as well as goods produced for own consumption (net of the costs of inputs). [The inclusion of this latter variable aims to adjust the OECD income concept to the realities of middle-income countries (such as Brazil, South Africa and others), where subsistence agriculture represents a significant income source for people at the bottom of the distribution. Countries that do not collect information on this income item should indicate so in the metadata sheet of the OECD questions

-TRR: current transfers received, including transfers from social security (including accident and disability benefits, old-age cash benefits, unemployment benefits, maternity allowances, child and/or family allowances, all income-tested and means-tested benefits that are part of social assistance, including quasi-cash transfers given for a specific purpose such as food stamps); transfers from employment related social insurance; as well as cash transfers from both non-profit institutions and other households.

-• TRP: current transfers paid, including direct taxes on income and wealth, social security contributions paid by households, contributions to employment-related social insurance, current transfers paid to both other households and non-profit institutions. Taxes on realised capital gains should be excluded from wealth taxes when possible. [Values for transfers paid should be reported in the OECD questionnaire with a negative sign].

While relevance and data availability for the sub-components of current transfers will vary across countries (depending on the structure of their social protection system and on features of their micro-data), this more detailed breakdown allows better reflecting the situation of countries with an important employment-related pension pillar.

• In the case of current transfers received (TRR):

-TRRSS: current transfers received from social security.

– TRRER: current transfers received from employment-related social insurance schemes (e.g. occupational pensions), where such schemes meet at least one of the following conditions: i) participation is obligatory; ii) the scheme is collective; and iii) the employer makes a contribution on behalf of an employee. 4

-– TRROT: current transfers received from non-profit institutions and other private households, e.g. alimonies.

• In the case of current transfers paid (TRP):

TA: direct taxes on income and wealth paid by households (net of refunds), as well as contributions paid by households to public social security schemes.

– TRPER: contributions paid by households to employment-related social insurance schemes (as defined above). – TRPOT: current transfers paid by households to non-profit institutions and other households, e.g. alimonies.

```
df.sample(4)
```

	LOCATION	Country	MEASURE	Measure	AGE	Age group	DEFINITION	Definition	METHODO	Methodology	...	Year
19190	ESP	Spain	PVTAA5	Age group 51-65: Poverty rate after taxes and ...	TOT	Total population	CURRENT	Current definition	METH2012	New income definition since 2012	...	2011
48845	NOR	Norway	GINIG	Gini (gross income, before taxes)	OLD	Retirement age population: above 65	CURRENT	Current definition	METH2012	New income definition since 2012	...	2020
35389	POL	Poland	TRROTCTOTAL	Current transfers received from non-profit ins...	WA	Working age population: 18-65	CURRENT	Current definition	METH2012	New income definition since 2012	...	2017

Essential equations related to this study are:

(1) Disposable income from primary income, market income, gross income (ref: IDD-ToR.pdf (oecd.org). P.5)

[1] **Equivalised primary income**: $PI_{ij} = E_{ij} + KI_{ij} + SEI_{ij} + (TRROTij - TRPOTij)$

[2] **Equivalised market income**: $MI_{ij} = PI_{ij} + TRRERij$

[3] **Equivalised gross income**: $GI_{ij} = MI_{ij} + TRRSS_{ij} - TRPERij$

[4] **Equivalised disposable income**: $DI_{ij} = GI_{ij} - TA_{ij}$

$$DI_{ij} = E_{ij} + KI_{ij} + SEI_{ij} + TRR_{ij} - TRP_{ij} =$$

$$= (EH_{ij} + ES_{ij} + EO_{ij}) + KI_{ij} + (SE_{ij} + OC_{ij}) + (TRRSS_{ij} + TRRER_{ij} + TRROT_{ij}) - (TA_{ij} + TRPER_{ij} + TRPOT_{ij})$$

(2) Gini index (ref: IDD-ToR.pdf (oecd.org). P.8)

Gini index		
	$$Gini = \left(\frac{2}{\mu . n^2} \cdot \sum_{k=1}^{n} k.W_k\right) - \frac{n+1}{n} = \frac{2\operatorname{cov}\left(W_k, \frac{k}{n}\right)}{\mu} = \frac{\frac{2}{n}\sum_{k=1}^{n}(W_k - \mu)\cdot\left(\frac{k}{n} - \frac{1}{n^2}\sum_{k=1}^{n} k\right)}{\mu}$$	Household incomes per equivalent household members (W_k) are ranked in ascending order (such as k = 1, 2, …, n). Individuals falling in each of the three population groups (entire population, population of working age and population of retirement age) should be ranked separately. n is the total number of individuals; μ is the arithmetic mean of disposable incomes: $\mu = \frac{\sum_k W_k}{n}$.

Data cleaning and Exploratory data analysis

Starting to import all necessary libraries, This code imports necessary modules for data analysis and machine learning, including pandas for data manipulation, matplotlib and seaborn for data visualization, re for regular expressions, numpy for numerical computing, and scikit-learn for machine learning. • The **%matplotlib inline** command sets the backend of matplotlib to the 'inline' backend, which allows the plots to be displayed in the Jupyter notebook. • The **sns.set()** command sets the default style of the plots to the Seaborn style. • The code then imports the training and test data from CSV files using pandas' **read_csv()** function and stores them in dataframes **df_train** and **df_test**, respectively.

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

Data Representation in Scikit-Learn

```
import sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    precision_recall_curve, roc_curve, roc_auc_score

)
from sklearn.model_selection import GridSearchCV
```

(1) Toy model: France

```
data31.head()
```

	Unnamed: 0	totalearning	capitalincome	transferrec	transferpaid	priceindex2015	meandi	saving
0	2012	20240	2710	5730	-4590	98.60500	26170	2617.00
1	2013	20520	2340	5750	-4800	99.45667	25850	2197.25
2	2014	20840	2410	5740	-5000	99.96167	26060	2319.34
3	2015	21040	2310	5780	-4830	99.99917	26260	2179.58
4	2016	21460	2070	5700	-4870	100.18250	26300	2156.60

```python
data31 = data31.drop('Unnamed: 0', axis=1)
```

```python
data31.columns
```

```
Index(['totalearning', 'capitalincome', 'transferrec', 'transferpaid',
       'priceindex2015', 'meandi', 'saving'],
      dtype='object')
```

```python
import matplotlib.pyplot as mp
data31.corr()
```

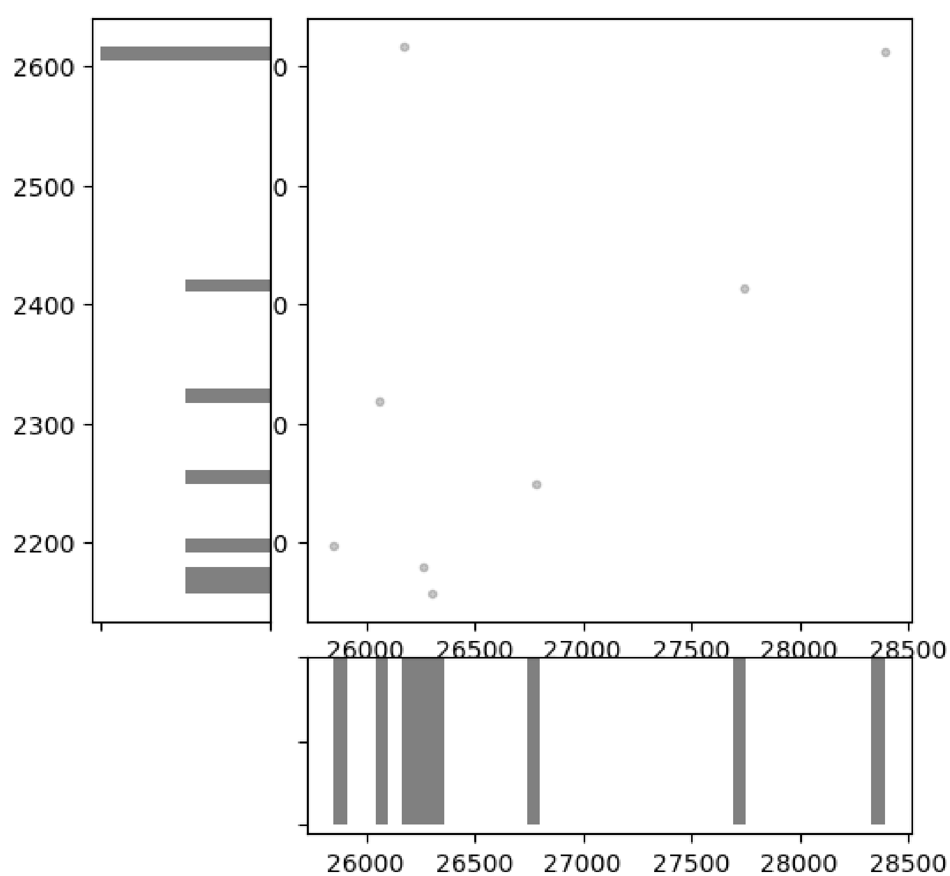
	totalearning	capitalincome	transferrec	transferpaid	priceindex2015	meandi	saving
totalearning	1.000000	-0.279288	-0.757628	-0.937070	0.991709	0.968509	0.350294
capitalincome	-0.279288	1.000000	0.304340	0.185130	-0.230677	-0.045563	0.720740
transferrec	-0.757628	0.304340	1.000000	0.630508	-0.707577	-0.716054	-0.316815
transferpaid	-0.937070	0.185130	0.630508	1.000000	-0.963940	-0.898802	-0.308936
priceindex2015	0.991709	-0.230677	-0.707577	-0.963940	1.000000	0.965564	0.362460
meandi	0.968509	-0.045563	-0.716054	-0.898802	0.965564	1.000000	0.559111
saving	0.350294	0.720740	-0.316815	-0.308936	0.362460	0.559111	1.000000

The data with key variables in France have variance with different range of mean values. Except for paid transfers, variables are greater than 49. For price index in 2015, the index is defined as percentage but mean values are 100 which is ideal for accommodating current market prices.

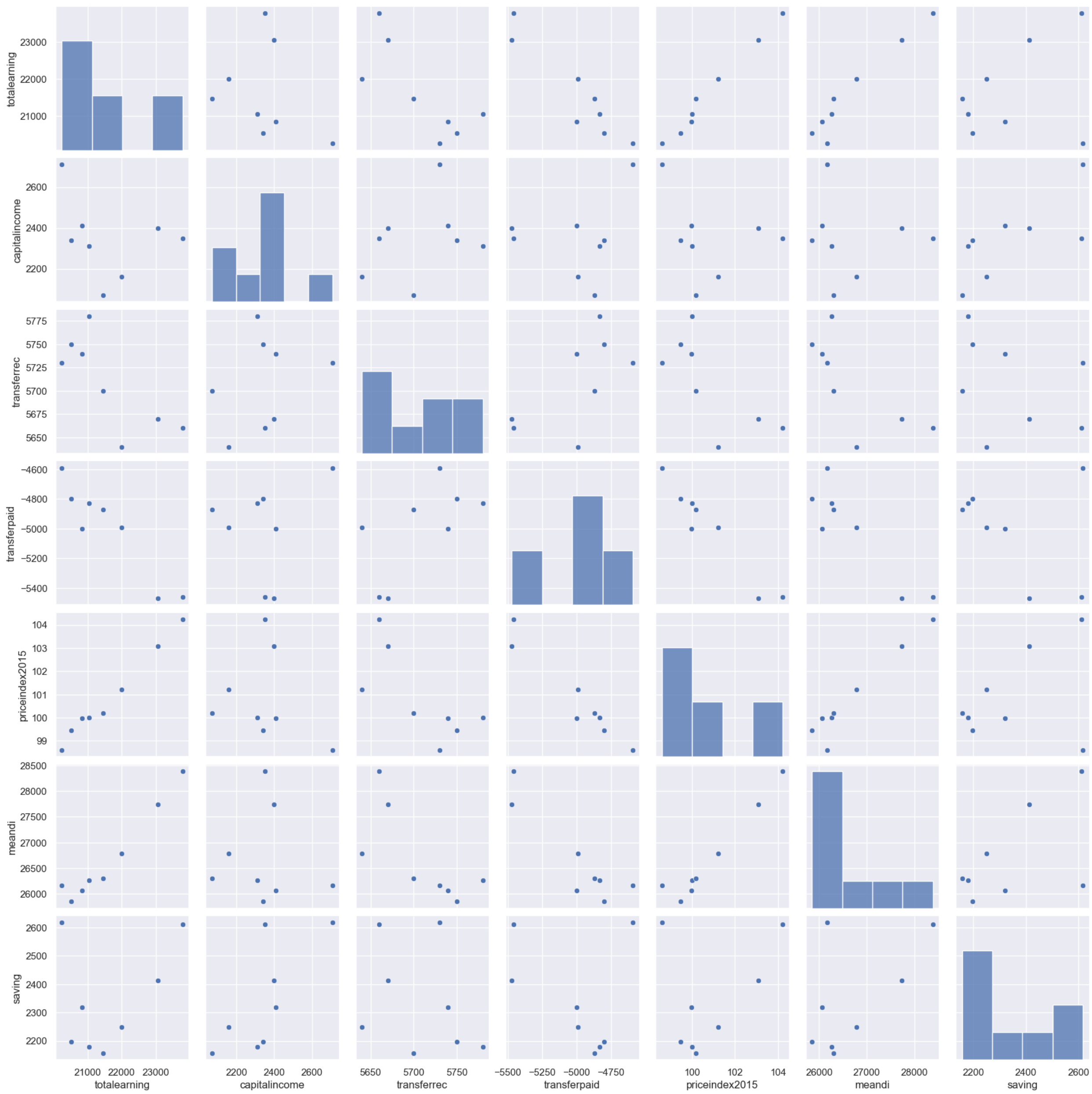
```python
data31.describe()
```

	totalearning	capitalincome	transferrec	transferpaid	priceindex2015	meandi	saving
count	8.000000	8.000000	8.000000	8.00000	8.000000	8.000000	8.00000
mean	21618.750000	2343.750000	5708.750000	-5001.25000	100.843026	26693.750000	2343.06875
std	1254.232121	189.581306	49.117207	313.16073	1.909861	903.199669	186.82237
min	20240.000000	2070.000000	5640.000000	-5470.00000	98.605000	25850.000000	2156.60000
25%	20760.000000	2272.500000	5667.500000	-5115.00000	99.835420	26142.500000	2192.83250
50%	21250.000000	2345.000000	5715.000000	-4930.00000	100.090835	26280.000000	2284.43000
75%	22265.000000	2402.500000	5742.500000	-4822.50000	101.685025	27020.000000	2463.00500
max	23790.000000	2710.000000	5780.000000	-4590.00000	104.232500	28390.000000	2617.00000

```
x=data31['meandi']
y=data31['saving']

# Set up the axes with gridspec
fig = plt.figure(figsize=(6, 6))
grid = plt.GridSpec(4, 4, hspace=0.2, wspace=0.2)
main_ax = fig.add_subplot(grid[:-1, 1:])
y_hist = fig.add_subplot(grid[:-1, 0], xticklabels=[], sharey=main_ax)
x_hist = fig.add_subplot(grid[-1, 1:], yticklabels=[], sharex=main_ax)

# scatter points on the main axes
main_ax.plot(x, y, 'ok', markersize=3, alpha=0.2)

# histogram on the attached axes
x_hist.hist(x, 40, histtype='stepfilled',
            orientation='vertical', color='gray')
x_hist.invert_yaxis()

y_hist.hist(y, 40, histtype='stepfilled',
            orientation='horizontal', color='gray')
y_hist.invert_xaxis()
```



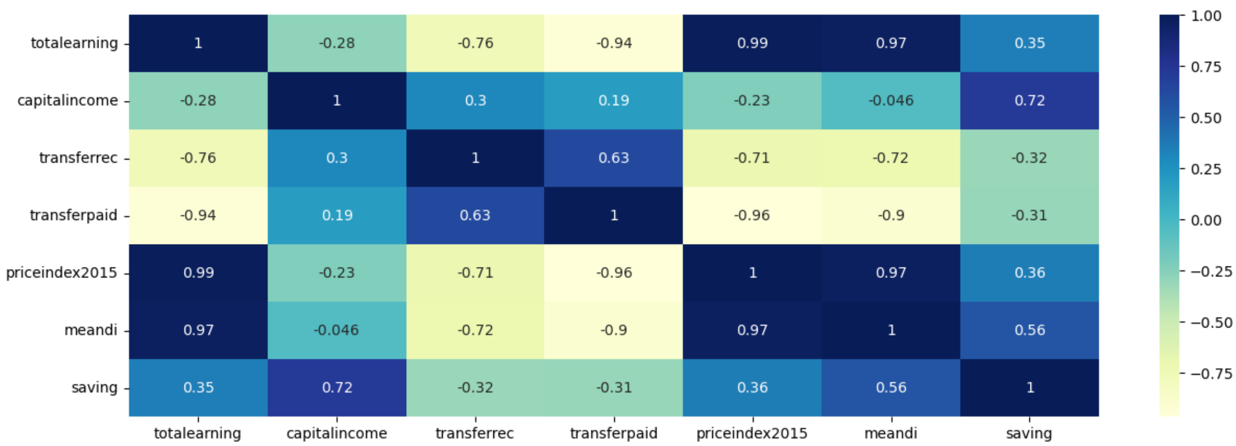
```
%matplotlib inline
import seaborn as sns; sns.set()
sns.pairplot(data31)
```



I may compare the heatmap with or without the index as percentage as below:

With price index 2015:

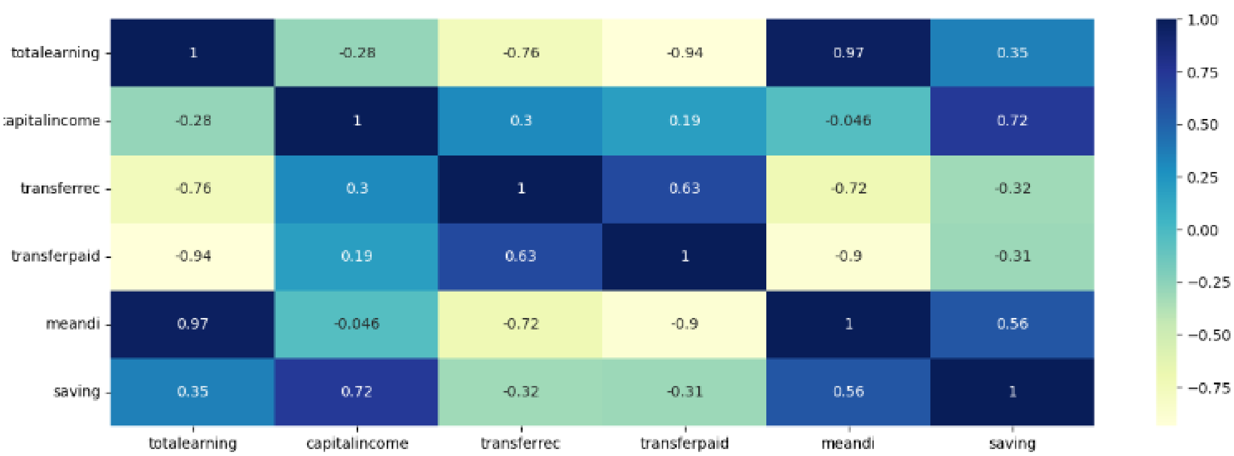
```
import seaborn as sb
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(15, 5))
dataplot = sb.heatmap(data31.corr(), cmap="YlGnBu", annot=True)
mp.show()
```



Without price index 2015

```
data31a = data31.drop('priceindex2015', axis=1)
```

```
import seaborn as sb
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(15, 5))
dataplot = sb.heatmap(data31a.corr(), cmap="YlGnBu", annot=True)
mp.show()
```



Machine Learning (ML)

Supervised learning example: Simple linear regression

```
import matplotlib.pyplot as plt
X = data31a[['totalearning','capitalincome','transferrec','meandi']]
y = data31a['saving']
```

1. Choose a class of model

```
from sklearn.linear_model import LinearRegression
```

2. Choose model hyperparameters. Would we like to fit for the offset (i.e., y-intercept)?

```
model = LinearRegression(fit_intercept=True)
model
```

```
LinearRegression()
```

3.A generate regression dataset

```
from sklearn.datasets import make_regression
X, y = make_regression(n_samples=100, n_features=2, noise=0.1, random_state=1)
```

4. Fit the model to your data

```
model.fit(X, y)
```

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
X.shape
```

```
(100, 2)
```

```
y.shape
```

```
(100,)
```

```
model.coef_
```

```
array([32.25850999, 86.42024677])
```

```
model.intercept_
```

```
0.007303861711072557
```

5. Predict labels for unknown data

```
# new instances where we do not know the answer
Xfit, _ = make_regression(n_samples=3, n_features=2, noise=0.1, random_state=1)
# make a prediction
yfit = model.predict(Xfit)
# show the inputs and predicted outputs
for i in range(len(Xnew)):
 print("X=%s, Predicted=%s" % (Xfit[i], yfit[i]))
```

```
X=[-1.07296862 -0.52817175], Predicted=-80.2497983168563
X=[-0.61175641  1.62434536], Predicted=120.649280643451
X=[-2.3015387   0.86540763], Predicted=0.5518357031231957
```

```
yfit = model.predict(Xfit)
```

Data base type selection

World data from a toy model of France

Summary: Stationary Test by Dickey-Fuller test: Except for Capital Income, variables are stationary.

data31['saving']	data31['meandi']	data31['priceindex2015']	data31['transferpaid']
ADF Statistic: -2.09 Critial Values: 1%, -6.05 Critial Values: 5%, -3.93 Critial Values: 10%, -2.99 p-value: 0.25 Stationary	ADF Statistic: 1.32 Critial Values: 1%, -6.05 Critial Values: 5%, -3.93 Critial Values: 10%, -2.99 p-value: 1.00 Stationary	ADF Statistic: -0.44 Critial Values: 1%, -6.05 Critial Values: 5%, -3.93 Critial Values: 10%, -2.99 p-value: 0.90 Stationary	ADF Statistic: -0.72 Critial Values: 1%, -4.94 Critial Values: 5%, -3.48 Critial Values: 10%, -2.84 p-value: 0.84 Stationary

data31['transferrec']	data31['capitalincome']	data31['totalearning']
ADF Statistic: -0.93 Critial Values: 1%, -4.94 Critial Values: 5%, -3.48 Critial Values: 10%, -2.84 p-value: 0.78 Stationary	ADF Statistic: -8.90 Critial Values: 1%, -6.05 Critial Values: 5%, -3.93 Critial Values: 10%, -2.99 p-value: 0.00 Stationary p-value: 0.00 Non-Stationary	ADF Statistic: 1.85 Critial Values: 1%, -5.35 Critial Values: 5%, -3.65 Critial Values: 10%, -2.90 p-value: 1.00 Stationary

SQL or No SQL

For analyzing entity relation, SQL is useful to check which kind of joints are possibly applied. For cross-country data, it is not easy to make left-join or outer join, the reason is the number of row and variables are reduced due to duplicated elements. When we did pivot in Python, it's the delicate method from the same reason.

Though, when we use SQL, the entity relation of data is automatically captured. In addition, by using functions such as group by, order by, sum, avg, min, max, it was useful to check the data structure as below.

```sql
#function 1: Group by
#show the table with selected Euro-using countries with measures.
SELECT *
FROM OECD.`monde (1)`
GROUP BY Country;

#total generations
SELECT *
FROM OECD.`monde (1)`
WHERE Age='TOT';

#average values for total generations (young and old)
SELECT avg('values')
from (select *
FROM OECD.`monde (1)`
group by Age='TOT'
limit 5) summary;

# function 2: Order by
#working-aging generations
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
ORDER BY Country;
```

```sql
# function 3: UNION
SELECT *
FROM OECD.`monde (1)`
WHERE Age='OLD'
UNION
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA';

#(1) measure: GINI
SELECT *
FROM OECD.`monde (1)`
WHERE Age='OLD'
and Measure='GINI';

#GROUP FUNCTIONS: MAX(4), MIN(5), AVG(6), SUM(7), COUNT(7)
SELECT MAX('values') as max,
MIN('values') as min,
AVG('values') as average,
SUM('values') as total,
COUNT('values') as NUM_columns
FROM OECD.`monde (1)`
WHERE Age='OLD'
and Measure='GINI';
```

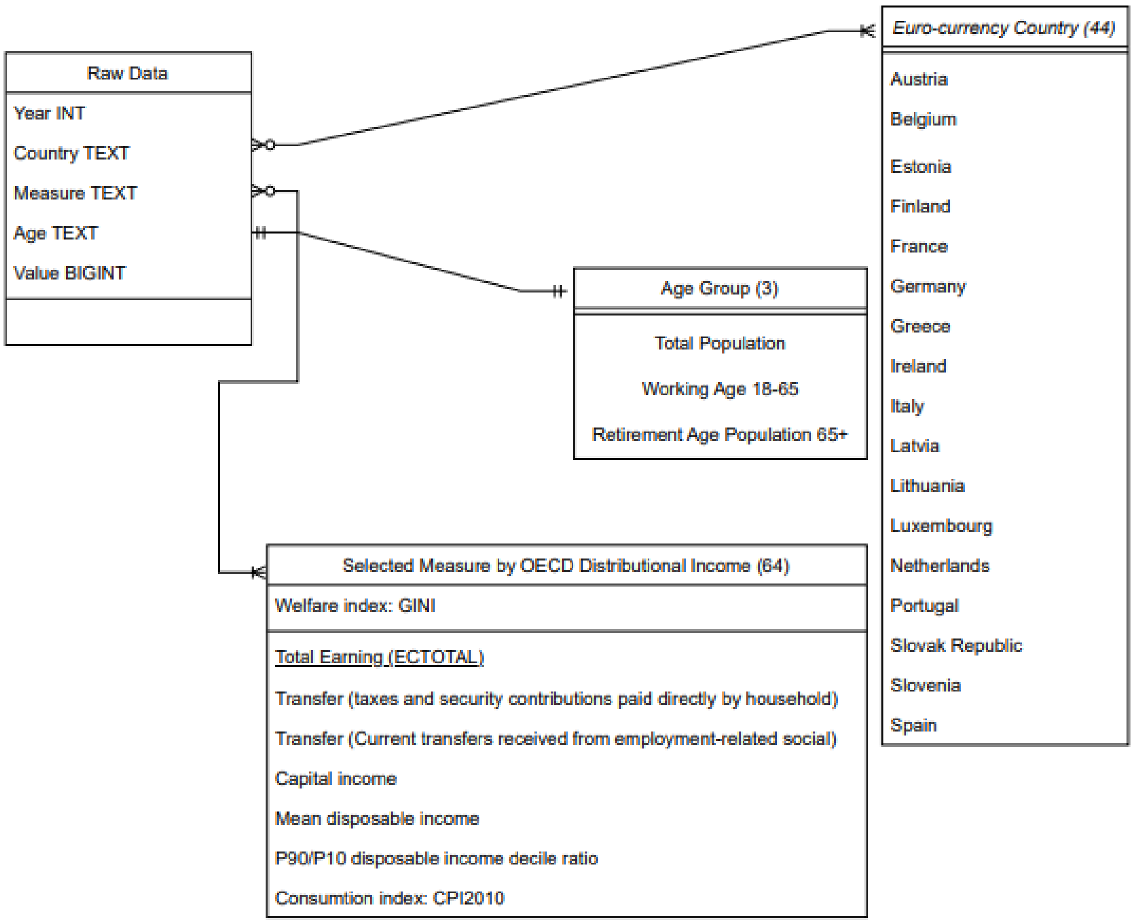
```sql
SELECT *
FROM OECD.`monde (1)`
WHERE Age='OLD'
and Measure='GINI'
ORDER by Measure='GINI' desc
limit 5 ;

# (2) measure: CPI2010
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='CPI2010';

# (3) measure: Total Earning (ECTOTAL)
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='ECTOTAL';

# (4) measure: taxes and security contributions paid directly by households (TACTOTAL)
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='TACTOTAL';
```

```sql
# (5) measure: Current transfers received from employment-related social insurance schemes (Current prices)
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='TRRERCTOTAL';

# (6) measure: Capital income
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='KICTOTAL';

# (7) Mean disposable income
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='INCCTOTAL';

# (8) P90/P10 disposable income decile ratio
SELECT *
FROM OECD.`monde (1)`
WHERE Age='WA'
and Measure='P90P10';
```

Entities. ERD

```
Raw Data. Shape(77182, 21)
(#Number) : Unique at Raw Data)
```

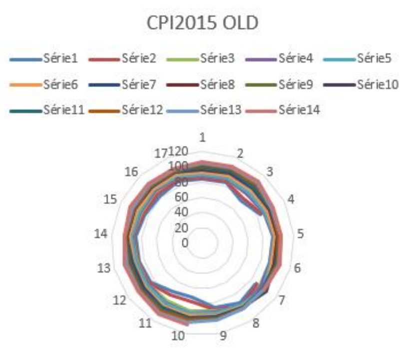
https://chatlapin.atlassian.net/jira/software/projects/CHAT/boards/1

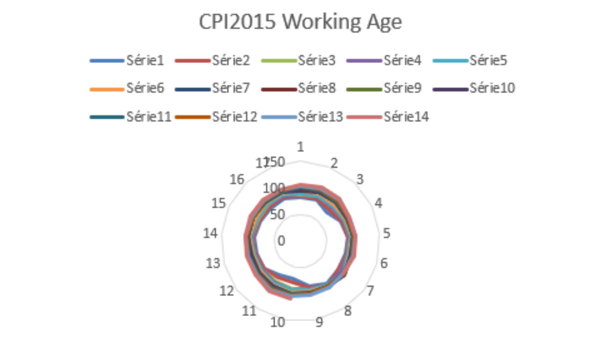


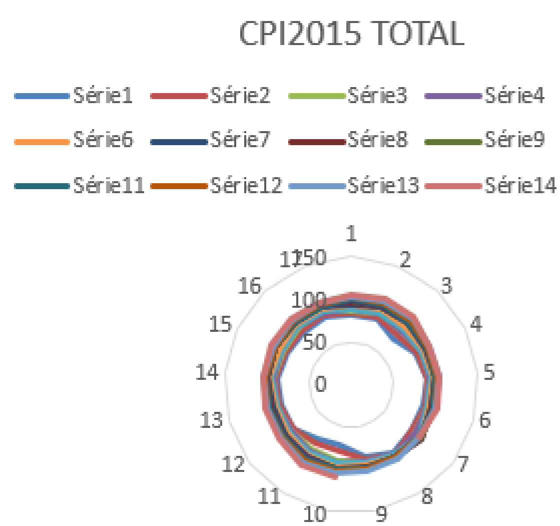
DATA Visualization

The comparison of Old generation and Working Age (18-65) generation

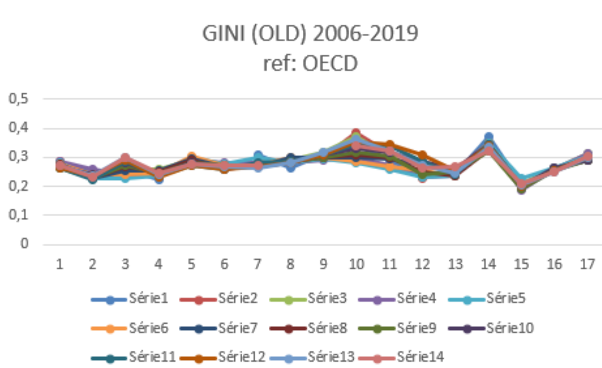
Consumption: Consumer Price Index (CPI)

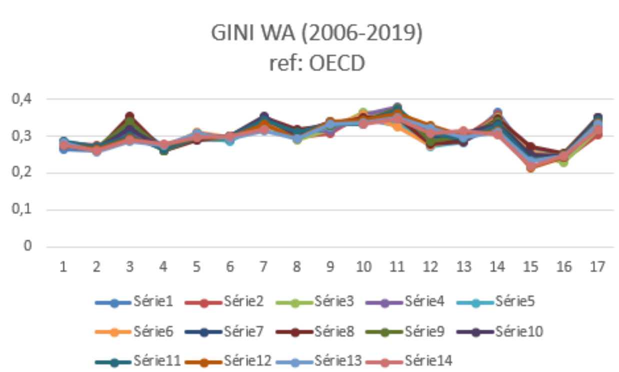




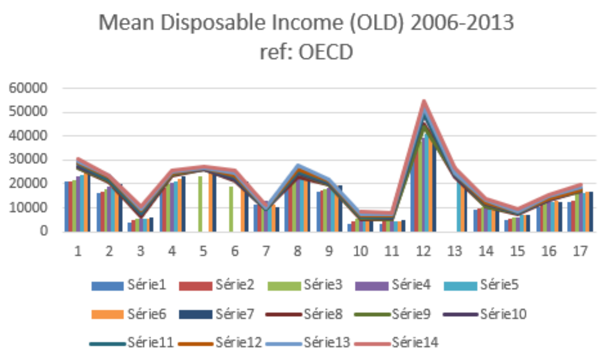


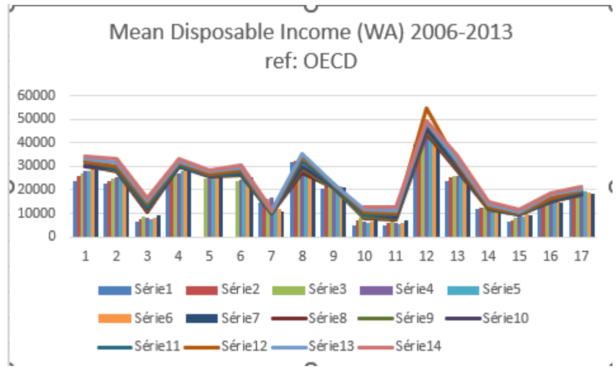
Inequality: Gini Index



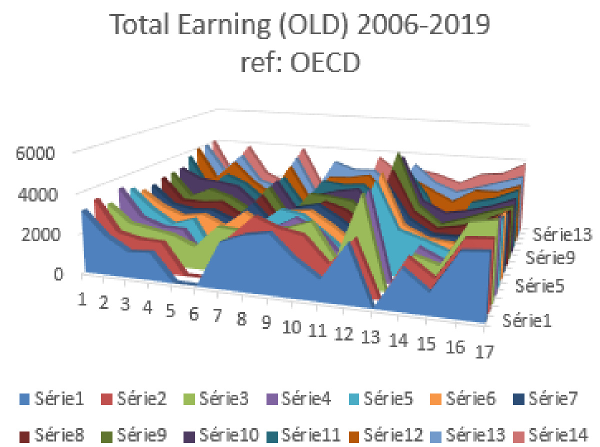


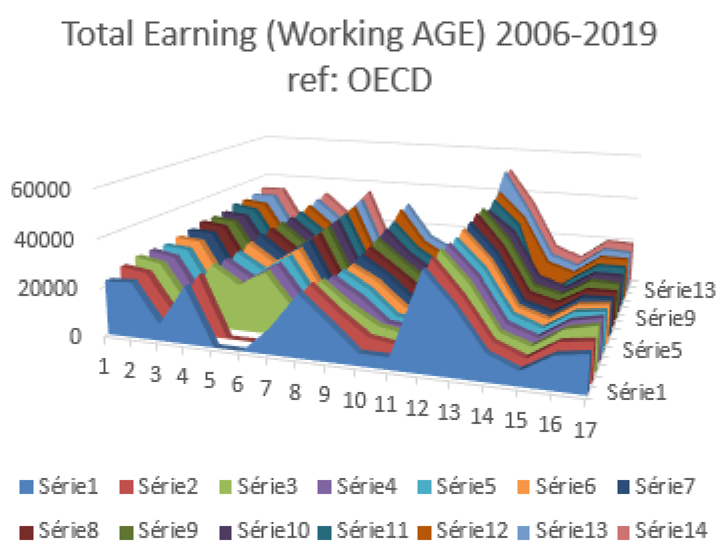
Mean Disposable Income: estimated as the greater amount among variables. Target Variable



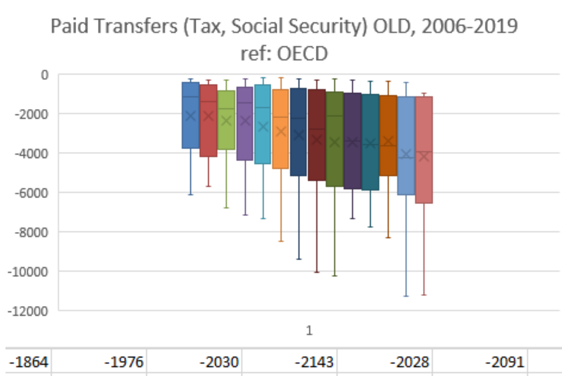


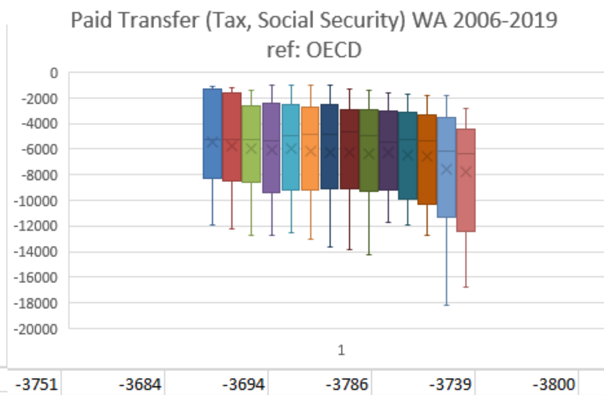
Total Yearly Earning per person



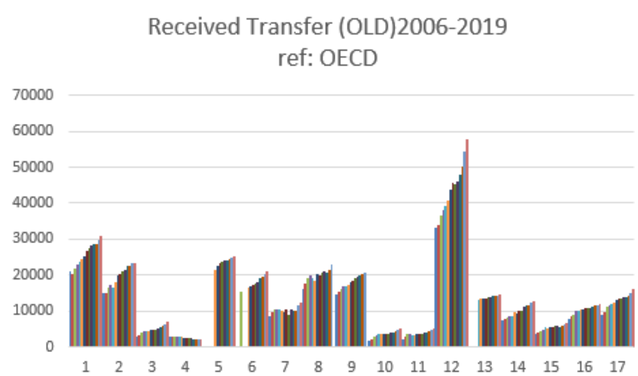


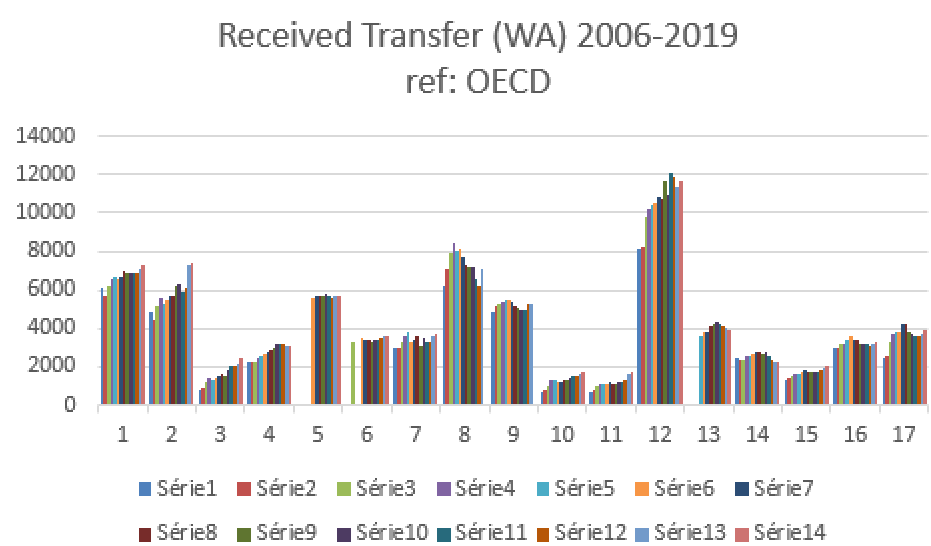
Paid Transfer (Tax, Insurance…)





Received Transfer





The P90/P10 ratio compares the income at the 90th percentile to the one at the tenth percentile