Estimating Croatia's Gini Coefficient using Lagrange Interpolation Method for Lorenz Curve Approximation

Student: Mariana Ávalos Arce
Professor: Bruno Klajn
Numerical Methods
Spring 2020

Zagreb, Croatia

1 Theoretical background

Income

Income is a monetary receipt that an individual/household earns during a certain period of time, usually a year. It is constituted by **labor income**, such as salaries, and **capital income**, such as earnings from ownership of financial capital or land. **Disposable income** is the result of the addition of transfers and substraction of taxes to an individual's gross income.

Inequality of Income

Income distribution shows how income is spread among a society, usually a country's population. In order to express this income distribution, a nation's population is divided into households. All households must be ordered, starting with the poorest and finishing with the richest household in terms of income. Then, all households are divided into **income groups**. Statistic offices commonly divide the ordered population into five or ten groups. Each income group must contain the same amount of households, where the first tenth represent the poorest 10% of population and the last tenth represents the richest 10% percent. Finally, for each income group the share of income with respect to total national income (GDP) is calculated.

In a perfectly equal distribution of income, each group earns the same share of national income, say 10%. This is a hypothetical and ideal case called **Absolute Equality**. The other extreme case is when the last tenth earns the whole 100% of total national income, whereas the remaining 90% of households earn none. This case is called **Absolute Inequality**. A nation's actual inequality occurs somewhere between these two cases.

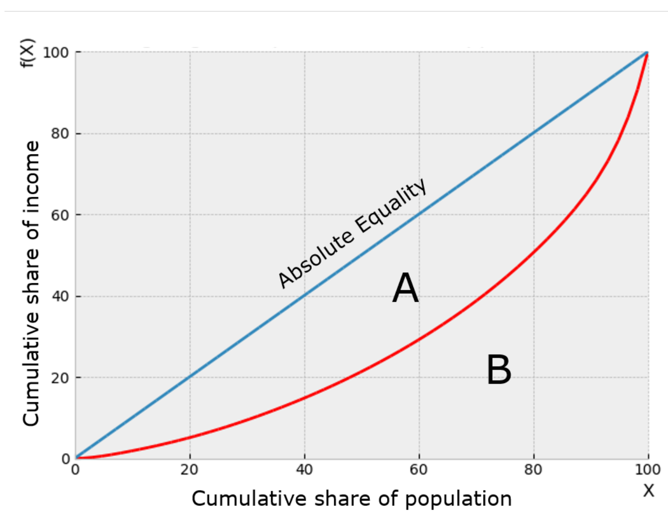

Figure 1: Lorenz Curve diagram

If we take the cumulative share of income (y axis) for each cumulative share of population (x axis), **Lorenz Curve** is plotted. This curve is a representation of the actual distribution of income in a country. **Inequality** is then the deviation of the Lorenz curve from the Perfect Equality line. The bigger the area enclosed between the Equality and Lorenz curves (A), the higher inequality there will be.

Gini Coefficient

The **Gini Coefficient** is the numerical value that functions as a measure of inequality. Taking into account Figure 1, the Gini coefficient is calculated as:

$$GINI = \frac{A}{A+B} \tag{1}$$

Therefore, this coefficient's value goes from 0 to 1, where 0 is **perfect equality** and 1 is **perfect inequality**. The ideal for a country is to have the lowest Gini coefficient as possible, assuring income equality. This coefficient can also be expressed using functions and integrals, as it involves curves and areas enclosed by said curves.

$$A = \int_0^{100} (x - L(x))dx \tag{2}$$

$$B = \int_0^{100} xdx \tag{3}$$

The purpose of this project was to estimate Croatia's Gini Coefficient in 2018 (0.293) using **Lagrange Interpolation Method** with the lorenz curve data points. The language chosen is Python 3.6.2.

2 Generating Lorenz curve points

The first step to estimate Gini Coefficient requires data for Croatia's distribution of income among the income groups. *Eurostat* provides statistics for Croatia in 2018 as the latest, given that this coefficient does not change significantly each year. Data was provided in tenths as the following:

Income Distribution Croatia 2018	
Income group	Income (€)
1st tenth	2,894
2nd tenth	4,052
3rd tenth	4,984
4th tenth	5,876
5th tenth	6,659
6th tenth	7,620
7th tenth	8,711
8th tenth	10,152
9th tenth	12,423
10th tenth	17,874
Total (GDP)	81,245

The program reads the data from a csv file for easier testing, and stores the **income column** in a list. For each value, the cumulative ratio of the income over total GDP is calculated to generate the Lorenz curve points, following the algorithm shown below.

```
for i in range(len(Lx)):
    yValue = (Ly[i] * 1.0) / (GDP * 1.0) * 100.0
    if i != 0:
        yValue += yPos[i - 1]
    print("X: ", Lx[i], "Y: ", yValue)
    yPos.append(yValue)
```

Figure 2: Y axis data calculation

The resulting coordinates are the data points with which the Lorenz curve is plotted:

Lorenz curve Croatia 2018	
X	Y
0.0	0.0
10.0	3.5620653578681765
20.0	8.549449196873653
30.0	14.683980552649393
40.0	21.916425626192378
50.0	30.112622315219397
60.0	39.491661025293865
70.0	50.21355160317558
80.0	62.709089790140936
90.0	77.9998769155025
100.0	100.0

3 Lagrange Interpolation Method

Lagrange Interpolation Method is an approach to find a polynomial approximation of n-1 degree that connects or interpolates the n given sets of points, which is expressed as the following:

$$\sum_{j=1}^{n} P_j(x), \text{ where } P_j(x) = \prod_{k=1,k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

Given this expression and using Python's Sympy library, the algorithm for the calculation of a Lagrange polynomial approximation in terms of x can be coded for a given set of n points as the parameter of the function.

```python
def Lagrange (Lx, Ly):
    X = sympy.symbols('X')
    if  len(Lx) != len(Ly):
        print ("Error data set")
        return 1
    y = 0
    for i in range(len(Lx)):
        t = 1
        for j in range(len(Lx)):
            if j != i:
                t *= ((X - Lx[j]) / (Lx[i] - Lx[j]))
        y += t * Ly[i]
    return y
```

Figure 3: Function that computes a Lagrange polynomial

As mentioned before, the input parameter for the above function would be a **set of n points**, which in this case will be the set of coordinates that plot Croatia's Lorenz curve. After the list of points is sent to the function call, the resulting polynomial approximation is the following:

$$\begin{aligned} L(x) =& 2.92888733459338 \times 10^{-16}x^{10} - 1.47066776487513 \times 10^{-13}x^{9} \\ &+ 3.17698630863945 \times 10^{-11}x^{8} - 3.85505917173149 \times 10^{-9}x^{7} \\ &+ 2.87809556086541 \times 10^{-7}x^{6} - 1.3608014410573 \times 10^{-5}x^{4} \\ &+ 0.000404140485432608x^{3} - 0.00722530516190734x^{2} \\ &+ 0.0764988150674863x + 0.0204593786968275 \end{aligned} \tag{4}$$

This polynomial was also plotted graphically using Python's Matplotlib library, producing the Lorenz curve (red line) diagram shown bellow:

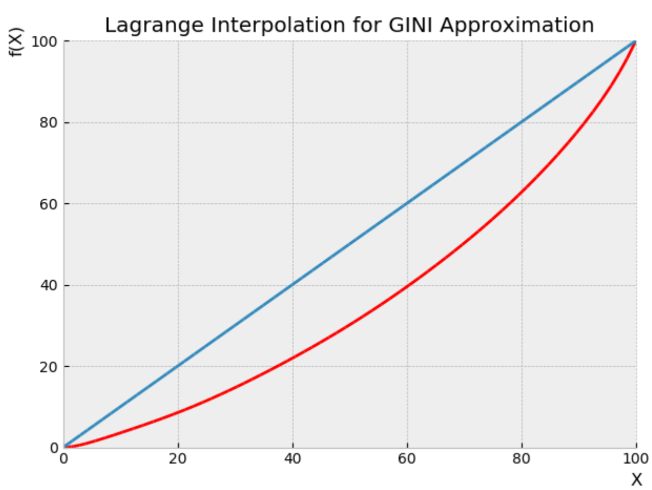

Figure 4: Lorenz curve Croatia 2018

The diagram included also the Absolute Equality line, f(x) = x, so as to give a better idea of the dimensions of Croatia's equality. Numerically speaking, the measure of inequality is done by the calculation of the Gini Coefficient, as mentioned before. Therefore, the integral in (2) was calculated now that the function L(x) was determined, resulting in the following enclosed area:

$$A = \int_0^{100} (x - L(x))dx = 1431.52827736660 \tag{5}$$

And knowing the equality area is this,

$$B = \int_0^{100} xdx = 5000 \tag{6}$$

The Gini Coefficient is estimated now as:

$$GINI = \frac{A}{A+B} = 0.286305655473319 \tag{7}$$

Which gives an accuracy of 97.72% when is compared to the known Gini for that year (0.293), after 1.825127 seconds of computation time. For further conclusions, the Lorenz curve for Mexico in 2018 was also plotted with the program (red), alongside a curve joining each of the points with a straight line (gray). The results of Croatia (left) and Mexico (right) are shown below, giving also a Gini of 0.286 and 0.435, respectively.

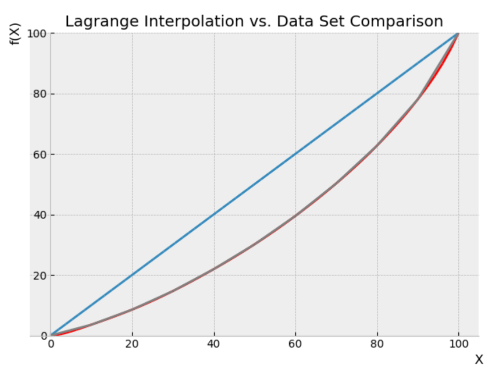

Figure 5: Approximation for Croatia

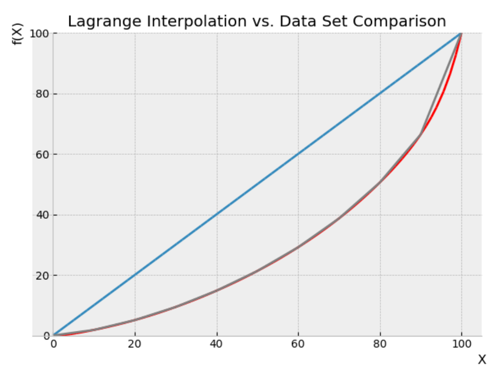

Figure 6: Approximation for Mexico

4 Conclusions

After observing the differences between Croatia's and Mexico's approximated polynomial for their respective Lorenz curves, one can see that Croatia's red curve is closer to the original data set curve, given that Mexico's Lorenz curve is steeper towards the end of the last tenth of population, making the lagrange function differ more and more as it gets steeper. Professor Klajn suggests experimenting with Padé Approximation for a better result when we have steeper curve sections.