Using Grouped Income Data to Calculate the Local Gini Coefficient

Chao Liu*

July 19, 2021

I compare two methods to use grouped income data to calculate the local Gini coefficient. The first method uses a user-written program called **inequal7**, while the second method applies the formula of the Gini coefficient to perform matrix calculations. The preferred method depends on the number of locations you have.[1]

I Grouped Income Data

The structure of grouped income data typically involves organizing individual income data into distinct income brackets or groups, which is particularly useful for analyzing income distribution and inequality. Below are key components and a general representation of such a data structure:

1. **Income Groups/Brackets:** These are ranges of income values, e.g., $0 - $10,000, $10,001 - $20,000, etc.

2. **Frequency:** The number of individuals or households in each income group.

3. **Income Group Midpoint:** The average income value of each income group, often used as a representative value.

4. **County or Location Identifier:** A unique identifier for different geographic areas if the data is being analyzed across multiple locations.

*Liu: Kellogg School of Management, Northwestern University. Email: chao.liu1@kellogg.northwestern.edu.

[1] Do files and data can be downloaded from https://github.com/chaoliu-kellogg/geodata_stata.

county	**income_group**	**pop**	**income**
1	$0 - $10,000	100	$5,000
1	$10,001 - $20,000	150	$15,000
...	...	...	...
2	$0 - $10,000	80	$5,000
2	$10,001 - $20,000	120	$15,000
...	...	...	...

Table 1. Data structure of grouped income data

In this structure:

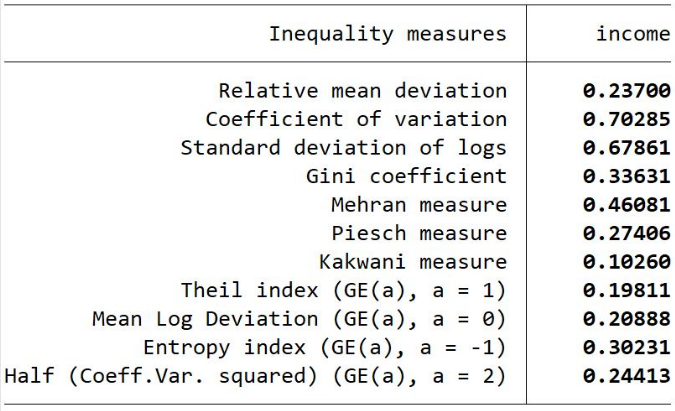
- Each row represents an income group within a particular county.
- The `county` column identifies the county.
- The `income_group` column specifies the income range for that group.
- The `pop` column shows the number of individuals or households in that income group.
- The `income` column provides a representative income value for that income group.

This organized data structure allows for a more straightforward analysis of income distribution within and across different geographic areas.

II Method 1: Using *inequal7*

To use this method, first install **inequal7** by executing the command `ssc install inequal7`. The following code iterates over all counties in your data set. For each location, **inequal7** can calculate several inequality measures, as illustrated in Figure 1.

```
use ”grouped_income.dta”, clear
gen gini = ””
glevelsof county, local(county)
foreach location of local county {
        inequal7 income [aw = pop] if county == `location'
        replace gini = r(gini) if county == `location'
}
```



```
                          Inequality measures |   income
----------------------------------------------+----------
                      Relative mean deviation |  0.23700
                     Coefficient of variation |  0.70285
                    Standard deviation of logs|  0.67861
                             Gini coefficient |  0.33631
                               Mehran measure |  0.46081
                               Piesch measure |  0.27406
                              Kakwani measure |  0.10260
                     Theil index (GE(a), a = 1)|  0.19811
             Mean Log Deviation (GE(a), a = 0)|  0.20888
                Entropy index (GE(a), a = -1) |  0.30231
      Half (Coeff.Var. squared) (GE(a), a = 2)|  0.24413
```

Figure 1. inequal7

```
gduplicates drop county, force
destring gini, replace force
keep county gini
save ”gini_method1.dta”, replace
```

III Method 2: Using Formula

The Gini coefficient using grouped income data is calculated as:

$$Gini_c = \frac{1}{2\bar{W}_c} \sum_{i=1}^{K} \sum_{j=1}^{K} f_{ic} f_{jc} \left| w_{ic} - w_{jc} \right|,$$

where $\bar{W}_c$ is the mean income in county c. f_{ic} and w_{ic} represent the share and income level of the income group i in county c, respectively. The following code applies this formula to calculate the local Gini coefficient:

```
use ”grouped_income.dta”, clear
gcollapse (mean) W = income [aw = pop], by(county) merge
bys county: gegen totalpop = sum(pop)
gen share = pop / totalpop
tempfile tmp
save `tmp', replace

rename * *1
rename county1 county
```

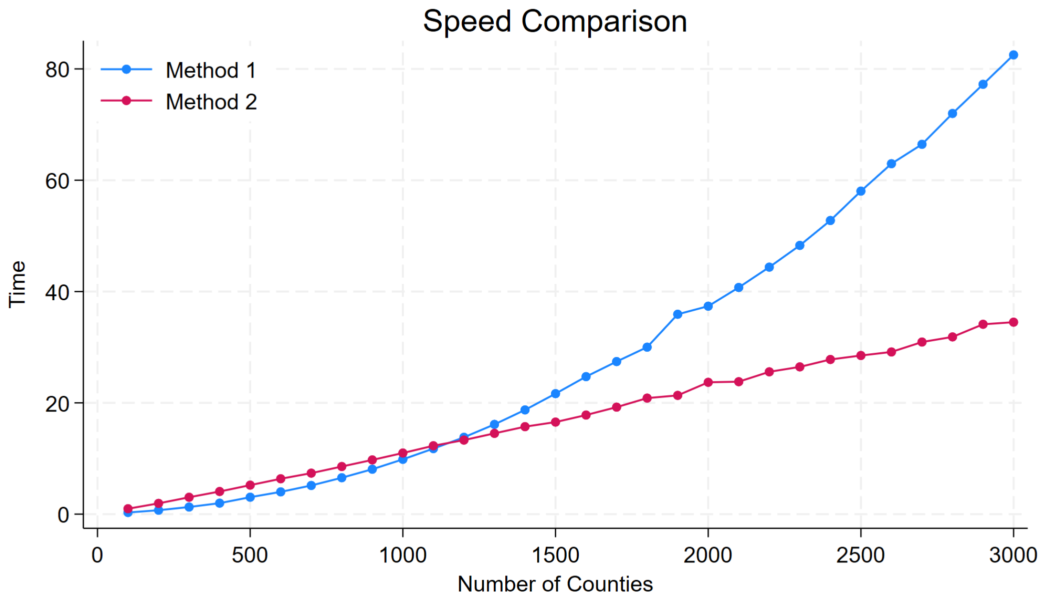


Figure 2. Speed Comparison

```
joinby county using `tmp'
gen aux = abs(income1 − income) * share * share1 / (2 * W1)
bys county: gegen gini = sum(aux)
gduplicates drop county, force
keep county gini
save ”gini_method2.dta”, replace
```

The above code snippet will generate the same local Gini coefficients as the first method.

IV Speed Comparison

Figure 2 compares the running time of two methods as a function of the number of counties. If the data set does not contain a large number of counties, running a loop is slightly faster than the second method. Otherwise, Method 2 is significantly faster than Method 1. Note that the `joinby` opeartion requires a substantial amount of RAM in your computer.