



Munich Personal RePEc Archive

Gini Coefficients for the 2000 Census

Burkey, Mark L.

North Carolina AT State University

March 2006

Online at <https://mpra.ub.uni-muenchen.de/57900/>
MPRA Paper No. 57900, posted 15 Aug 2014 10:59 UTC

Gini Coefficients for the 2000 Census:
Data and Methods used for calculating Gini Coefficients for Census 2000 Data in R
by Mark L. Burkey
Professor of Economics
North Carolina A&T State University
March, 2006

Abstract: This note documents the method for calculating Gini Coefficients in R using 2000 Census data. Many people have use this method for computing Ginis for ZCTA's for example. You can find these estimates for both the 50 states, as well as all US counties and additional information about Gini coefficients at <https://sites.google.com/a/burkeyacademy.com/main/home/gini-coefficients>

Here is enough information so that you can learn to computer Gini coefficients and understand how these were calculated. These Ginis are from *Household Income*. All suggestions and corrections are welcome!

I am using a Windows-based computer and Microsoft Excel.

To compute gini coefficients with Census 2000 data: Get variables P52 and P54. Get rid of total households, and total income, and total income for households less than 200,000

- 1) Transpose data with midpoints of classes in left hand column, labeled "x"
- 2) end up with 16 midpoints like so (but in a column)

5000 12500 17500 22500 27500 32500 37500 42500 47500 55000 67500 87500 112500 137500 175000

The highest midpoint is unknown since it is for all households earning above \$200,000. Leave it blank(the category after 175000). Append row of total incomes for hholds making above 200000 as row 18 in Excel, which will become obs#17 when it is read into R

- 3) Put the frequencies of households in counties, etc. in different groups, and label the columns y1, y2,...

- 4) Read the data into R. From Excel the easiest way is to copy all of the data to the Clipboard (highlight, Ctrl-C). Then in R, type
`dataone=read.delim("clipboard")`
`attach(dataone)`

- 5) Now, you are ready to compute gini coefficients.

```
library(ineq)
ineq(rep(c(x[1:15],y1[17]/y1[16]),y1[1:16]))
```

Will compute the Gini coefficient for the region in column y1.

- 6) In order to compute gini coefficients for many counties or states, you need a function to automatically repeat the process for y1, y2, ... Type in the function as below:

```
burkeyineq=function(y) ineq(rep(c(x[1:15],y[17]/y[16]),y[1:16])) #define function
```

```
myginis=sapply(dataone[,2:101],burkeyineq) #apply to 100 counties in columns 2 to 101.
```

```
uscogini=sapply(uscogin3[,2:3217],burkeyineq) #apply to 3,216 US counties
```

Now, you can look at your Ginis by simply typing *myginis* at the > prompt. You can look at a list of all of the objects you have created in **R** by simply typing *ls()*. If you want to transfer your Ginis to Excel or another program, two easy methods are:

1) `write.csv(myginis,file="ginis.csv")` This will write a file to your R installation directory. You can put an a: or f:\data\gini or other location in the quotes with the file name if you like.

2) `write(myginis,file="clipboard",ncolumns=1)` This will write it to your windows clipboard so you can paste it into Excel with control-v or edit/paste.

7) Plot three Lorenz curves on the same graph, using observations 41,66, and 67.

```
plot(Lc(rep(c(x[1:15],y41[17]/y41[16]),y41[1:16])))
```

```
lines(Lc(rep(c(x[1:15],y67[17]/y67[16]),y67[1:16])),col=2)
```

```
lines(Lc(rep(c(x[1:15],y66[17]/y66[16]),y66[1:16])),col=3)
```

Data example below for first three counties in North Carolina:

x	y1	y2	y3
5000	5231	1183	782
12500	3113	725	389
17500	3483	909	391
22500	3958	930	366
27500	3718	1195	443
32500	3659	921	303
37500	3223	966	314
42500	3477	772	249
47500	2748	814	226
55000	5128	1540	346
67500	5571	1521	301
87500	4568	1009	258
112500	1783	344	76
137500	828	150	60
175000	593	76	23
	641	122	74
	223833700	53915200	27364100

y1, y2, for each county- you could have the county names instead if you like.

See below for some more information from a handout I made.

Lorenz Curves and Gini Coefficients in

The **ineq** library in R can be used to construct Lorenz Curves and calculate Gini Coefficients.

For the Gini coefficients I calculated, I used the Household Income counts from the Census 2000. First, I created a vector of incomes **x** (using the midpoint of the household income classes) and a vector of frequencies **y1** for each county or state. The numbers below represent the data from Alamance County, North Carolina:

x	y1
5000	5231
12500	3113
17500	3483
22500	3958
27500	3718
32500	3659
37500	3223
42500	3477
47500	2748
55000	5128
67500	5571
87500	4568
112500	1783
137500	828
175000	593
	641
	223833700

The last two entries in **y1** are 641 families making above \$200,000, and \$223,833,700 was the total earnings for these families, for an average of \$349,195 each. We will calculate and use the income \$349,195 and frequency 641 in the calculations, calculated on the fly since this will be different for each county.

```
> library(ineq)                                #load ineq library
> plot(Lc(rep(c(x[1:15],y1[17]/y1[16]),y1[1:16]))) #plot Lorenz Curve
> ineq(rep(c(x[1:15],y1[17]/y1[16]),y1[1:16]))   #Calculate Gini
```

R will also calculate other inequality and dissimilarity measures, by using the function:

```
> ineq(rep(c(x[1:15],y1[17]/y1[16]),y1[1:16]), type = " TYPE "), where TYPE is one of the following:
```

RS(x)	Ricci-Schutz coefficient (Pietra's measure)
Atkinson(x, parameter = 0.5)	Atkinson's Measure
Theil(x, parameter = 0)	Theil's Entropy Index
Kolm(x, parameter = 1)	Kolm's Measure
var.coeff(x, square = FALSE)	Coefficient of Variation
entropy(x, parameter = 0.5)	Generalized Entropy Index

`RS` is the the Ricci-Schutz coefficient (also called Pietra's measure), `Atkinson` gives Atkinson's measure and `Kolm` computes Kolm's measure.

If the parameter in `Theil` is 0 Theil's entropy measure is computed, for every other value Theil's second measure is computed.

`ineq(x, type="var")` and `var.coeff(x)` respectively compute the coefficient of variation, while `ineq(x, type="square.var")` and `var.coeff(x, square=TRUE)` compute the squared coefficient of variation.

`entropy` computes the generalized entropy, which is for parameter 1 equal to Theil's entropy coefficient and for parameter 0 equal to the second measure of Theil.

Credit goes to Achim Zeileis zeileis@ci.tuwien.ac.at, the author of the `ineq` function in R.