STAT 5361 Interim Resport

A Simple and Effective Inequality Measure

Chamundeswari Koppisetti* Shynggys Magzanov†

19 11 2020

Abstract

We'll write it later

1. Introduction

1.1. Measures of Inequality

There exist various types of metrics used by economists to measure the income inequality. The most widely exploited ones are the Lorenz curves and the Gini coefficient, each of which have their own advantages and limitations. Those obvious limitations include the requirement for the population mean and variance to exist, as well as down-weighting smaller incomes and that way stressing way more attention to the middle incomes. This paper analyzes another income inequality measure, namely using ratios of symmetric quantiles, that proves to be helpful to overcome previously mentioned disadvantages. In addition to that, it is proven that the given metric satisfies the median preserving principle and applicable to widely used income distributions. But the major benefit is that there is no need of parametric model assumption to work with the given inequality measure.

*chamundeswari.koppisetti@uconn.edu

†shynggys.magzanov@uconn.edu

1.2. Concepts

2. Examples with well-known income distributions

2.1. Inequality curves

2.2 Comparing I and G

3. Inference

3.1. Approximation of I

3.2 Confidence intervals

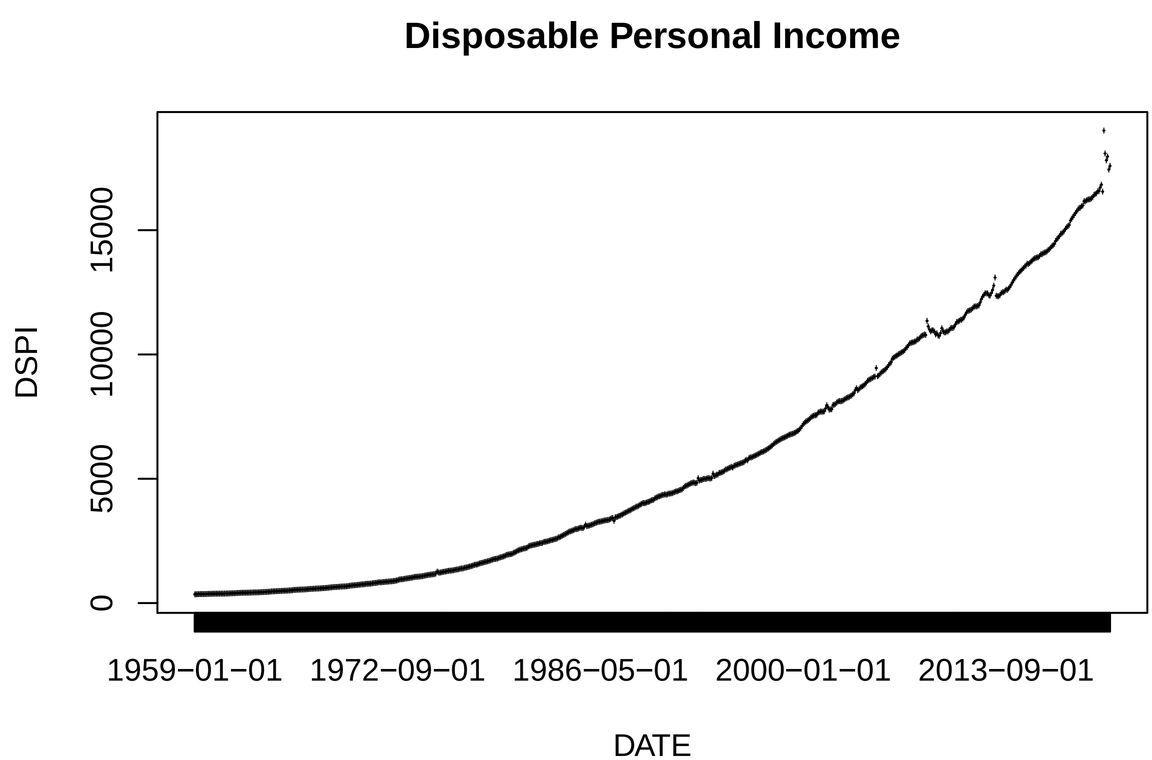
3.3 Resistance to outliers

4. Application

4.1. Disposable Personal Income

The data of Disposable Personal Income can be obtained at https://fred.stlouisfed.org/series/DSPI.

```r
DSPI <- read.csv("C:\\Users\\Asus\\Downloads\\DSPI.csv")
plot(DSPI, type = "l", main = "Disposable Personal Income")
```



```r
QOR.ln <- function(u){
  # QOR function for the log-normal
  q.n.0 <- 1/dnorm(qnorm(u))
  q.n.1 <- qnorm(u)*q.n.0^2
  q.n.2 <- (1 + 2*qnorm(u)^2)*q.n.0^3
  1/(q.n.0^2 + 3*q.n.1 + q.n.2/q.n.0)
}

QOR.gld <- function(u, lambda = NULL){
  # QOR function for the GLD FKML paramaterization
  if(is.null(lambda)) lambda <- fit.fkml(x)$lambda
  l3 <- lambda[3]
  l4 <- lambda[4]
  (u^(l3 - 1) + (1 - u)^(l4 - 1))/(u^(l3 - 3)*(l3 - 2)*(l3 - 1) +
                                   (1 - u)^(l4 - 3)*(l4 - 2)*(l4 - 1))
}

hatI <- function(x, J = 100, conf.level = 0.95, bw.correct = TRUE, QOR.FUN = QOR.ln, ...){

  n <- length(x)

  us <- ((1:J) - 0.5)/J
  Rs <- (xu2 <- quantile(x, us/2))/(x1u2 <- quantile(x, 1 - us/2))
  I <- sum(1 - Rs)/J

  if(!is.null(conf.level)){
    v <- c(us/2, 1 - us/2)
    qor <- QOR.FUN(v, ...)
    bw <- 15^(1/5)*abs(qor)^(2/5)/n^(1/5)
    if (bw.correct) bw[v <= bw] <- v[v <= bw]

    kernepach <- function(u) 3/4*(1 - u^2)*(abs(u) <= 1)
    m1 <- matrix(v, nrow = 2*J, ncol = n, byrow = FALSE)
    m2 <- matrix(1:n, nrow = 2*J, ncol = n, byrow = TRUE)

    consts <- kernepach((m1 - (m2 - 1)/n)*(1/bw))*(1/bw) -
      kernepach((m1 - m2/n)*(1/bw))*(1/bw)

    x.sorted <- sort(x)
    q.hat <- c(consts%*%x.sorted)
    q.hat.1 <- q.hat[1:(length(q.hat)/2)]
    q.hat.2 <- q.hat[-(1:(length(q.hat)/2))]

    rc <- matrix(Rs, ncol = J, nrow = J, byrow = FALSE)

    covm <- ((1/x1u2)%*%t(1/x1u2))*(((us/2)%*%t(1 - us/2))*(q.hat.1%*%t(q.hat.1) + Rs%*%t(Rs)*
```

```
                                        ((us/2)%*%t(us/2))*((q.hat.1%*%t(q.hat.2))*t(rc) + (q.ha

    sigma.p2 <- (us/2)*(1 - us/2)*q.hat.1^2
    sigma.q2 <- (1 - us/2)*(us/2)*q.hat.2^2
    sigma.pq <- (us/2)^2*q.hat.1*q.hat.2
    a0 <- sigma.p2/x1u2^2
    a1 <- -2*sigma.pq/x1u2^2
    a2 <- sigma.q2/x1u2^2
    Vs <- (a0 + a1*Rs + a2*Rs^2)/n

    V <- (sum(Vs) + 2*sum(covm[row(covm) < col(covm)]))/J^2
    SE <- sqrt(V)
    conf.int <- I + c(-1, 1)*qnorm(1 - (1 - conf.level)/2)*sqrt(V)
  } else{
    V <- NULL
    SE <- NULL
    conf.int <- NULL
  }

  list(I = I, SE = SE, conf.int = conf.int)
}

hatG <- function(x, conf.level = 0.95){
  indices <- 1:(n <- length(x))
  ordered.x <- sort(x)
  sx <- sum(ordered.x*(indices - 1/2))
  mu.hat <- mean(x)
  Gv <- 2/mu.hat/n^2*sx - 1

  Z.hat <- -(Gv + 1)*ordered.x + (2*indices - 1)/n*ordered.x - 2/n*cumsum(ordered.x)
  Z.bar <- mean(Z.hat)

  V <- 1/n^2/mu.hat^2*sum((Z.hat - Z.bar)^2)
  conf.int <- Gv + c(-1, 1)*qnorm(1 - (1 - conf.level)/2)*sqrt(V)

  list(G = Gv, SE = sqrt(V), conf.int = conf.int)
}
```

```
DSPI <- DSPI$DSPI
I <- hatI(DSPI)
I
```

```
## $I
## [1] 0.7560689
##
## $SE
## [1] 0.01064179
```

```
##
## $conf.int
## [1] 0.7352114 0.7769264
```

```
G <- hatG(DSPI)
G
```

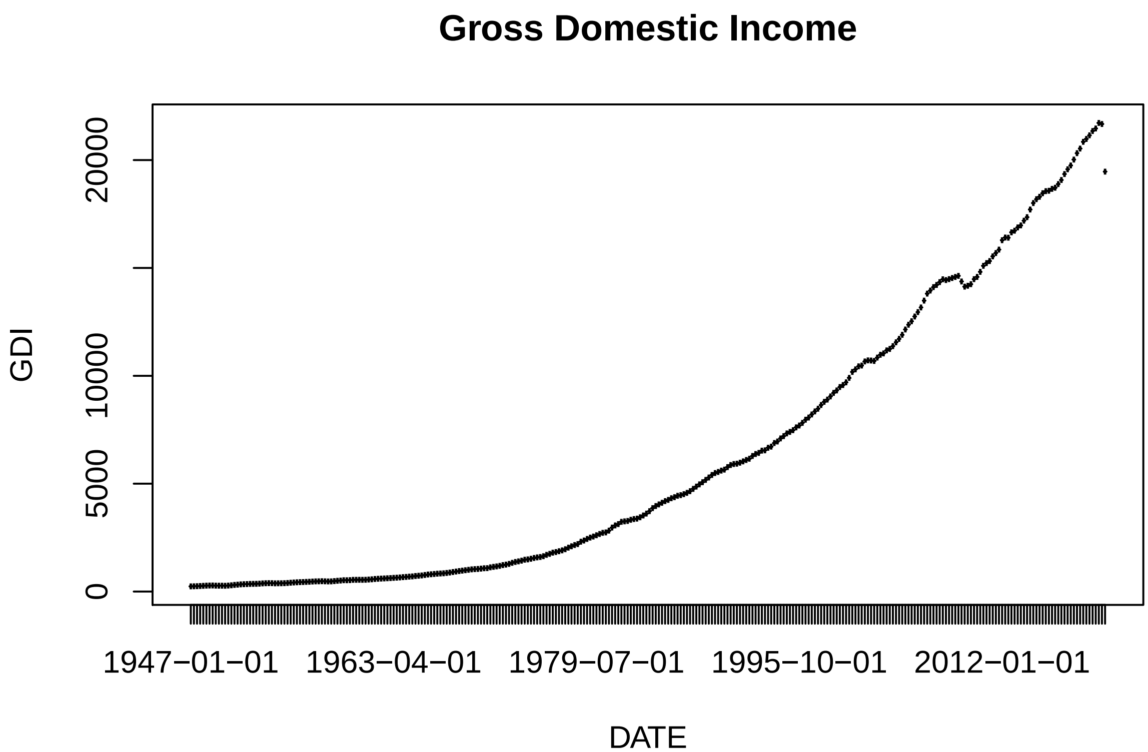
```
## $G
## [1] 0.487252
##
## $SE
## [1] 0.009500346
##
## $conf.int
## [1] 0.4686317 0.5058723
```

Firstly, I and G were calculated for the Disposable Personal Income Data. It appears that $\hat{I} = 0.7561$, which is an estimate of I, with a standard error of 0.0106. Additionally, 95% confidence interval of $\hat{I}$ was calculated to be $(0.7352, 0.7769)$. While $\hat{G} = 0.4872$ is significantly lower, with a standard error of 0.0095 and respective 95% confidence interval of $(0.4686, 0.5059)$.

4.2. Gross Domestic Income

The data on Gross Domestic Income can be found here: https://fred.stlouisfed.org/series/GDI.

```
GDI <- read.csv("C:\\Users\\Asus\\Downloads\\GDI.csv")
plot(GDI, type = "l", main = "Gross Domestic Income")
```




```r
GDI <- GDI$GDI
I_gdi <- hatI(GDI)
I_gdi
```

```
## $I
## [1] 0.8204182
##
## $SE
## [1] 0.01624378
##
## $conf.int
## [1] 0.7885809 0.8522554
```

```r
G_gdi <- hatG(GDI)
G_gdi
```

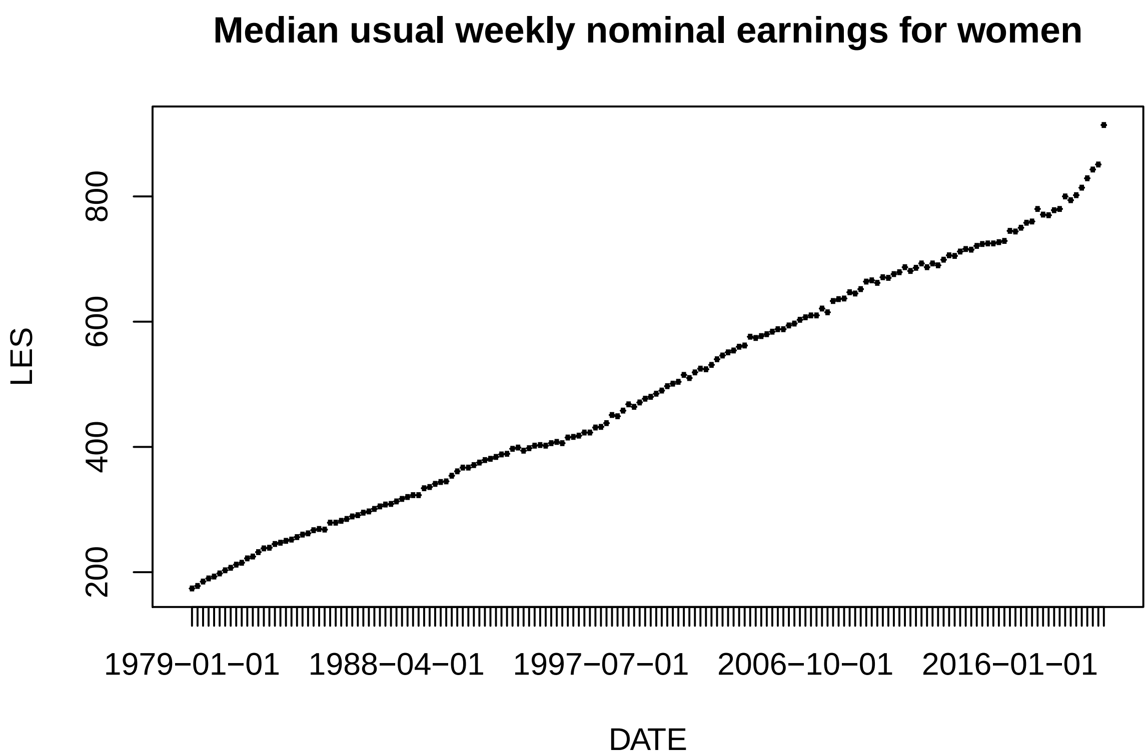
```
## $G
## [1] 0.549877
##
## $SE
## [1] 0.01613956
##
## $conf.int
## [1] 0.5182440 0.5815099
```

As a result, ratio of symmetric quantiles of the GDI is $\hat{I} = 0.8204$ with a standard error 0.0162 and confidence interval $(0.7886, 0.8523)$. At the same time $\hat{G} = 0.5499$ with a standard error of 0.0161 and CI $(0.5182, 0.5815)$.

4.3. Earnings data of women

The data on Earnings of women in the US between 1979 and 2020 can be otbained from: https://fred.stlouisfed.org/series/LES1252882700Q.

```
LES <- read.csv("C:\\Users\\Asus\\Downloads\\LES.csv")
plot(LES, type = "l", main = "Median usual weekly nominal earnings for women")
```




```
LES <- LES$LES
I_les <- hatI(LES)
I_les
```

```
## $I
## [1] 0.4665993
##
## $SE
## [1] 0.01831842
##
## $conf.int
## [1] 0.4306959 0.5025028
```

```r
G_les <- hatG(LES)
G_les
```

```
## $G
## [1] 0.2199384
##
## $SE
## [1] 0.009415175
##
## $conf.int
## [1] 0.2014850 0.2383918
```

Results reveal that $\hat{I} = 0.4666$ with a standard error 0.0183 and CI $(0.4307, 0.5025)$. While Gini coefficient is $\hat{G} = 0.2199 \pm 0.0094$ with CI $(0.2015, 0.2384)$.