Canazei 2018
Winter School on Inequality and Social Welfare Theory

R TUTORIAL

paolo.brunori@unifi.it[1]

[1] Thanks to Daniel G. Mahler for useful comments.

1.1 Handle your data

We can easily import data of different formats in R. Economists often have data written by EXCEL, STATA, SAS or SPSS. R comes with a set of basic functions, but for many tasks you need to download ‘packages’ (similar to .ado-files in Stata). The package “foreign” does the job of important different kinds of data (use the package “readstata13” for STATA 13 or 14).

```
library(foreign) # load the package
data<-read.dta("/Users/paolobrunori/Downloads/Brunori_IT93.dta") #
 Italian income data 1993

fix(data) # you can look in it

summary(data) # or get a summary
```

The package “ineq” written by Achim Zeileis and Christian Kleiber does almost everything you may need do perform distributive analysis. Inequality indexes, poverty indexes, plot of Lorenz and other curves. It is extremely useful if you need to quickly get an inequality estimate.

```
library(ineq)
ineq(data$y, type="Gini")

# mean log deviation
ineq(data$y, type="entropy", parameter=0)
```

The package contains also rather sophisticated tools such as the Mehran’s bounds for the Lorenz curve of grouped data and calculates the most common poverty indexes.

You can find out more information about what ‘ineq’ can do, by typing ?ineq (equivalent to “help xxx” in Stata).

It may be useful to also have a look at the package“IC2” written by Didier Plat which also performs decomposition of a number of inequality indexes.

```
# IC2
library(IC2)
decompSGini(data$y, data$birth_area)
```

We get between region of birth, within, and overlap decomposition of the Gini index (Bhattacharya and Mahalanobis’s version).

1.2 Write your own function

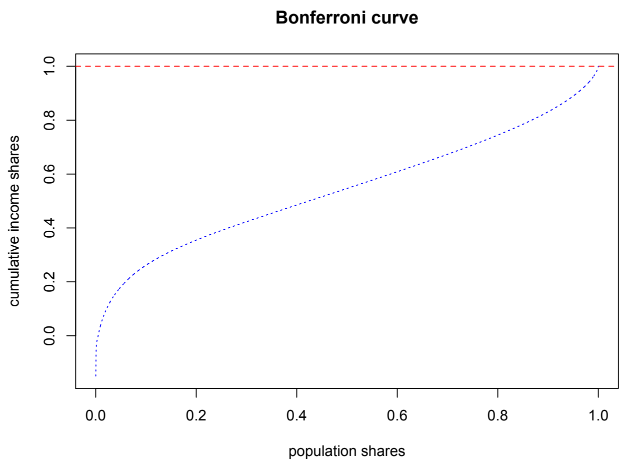
We want to estimate our own inequality index. Assume my index is the Bonferroni inequality index:

$$I = \frac{\left[\bar{y} - \frac{1}{n}\sum_{i=1}^{n} \bar{y}_l\right]}{\bar{y}}$$

Where $\bar{y}_l$ is the mean of the first l incomes ranked by increasing values.

Despite it has a number of nice properties (as nice as the Gini, see for example Chakravarty (2007)) there is no package that calculates this index.

First note that the Bonferroni index measures the area below the horizontal line with intercept = 1 and the Bonferroni curve. The Bonferroni curve plots cumulative population shares (x-axis) against the ration of the mean income of the same population shares over the mean income of the entire population (y-axis).



This curve is obtained with the following code:

```
Y<-data$y[order(data$y)] # vector of ordered incomes

pop.shares<-((1:length(Y))/length(Y)) # ignoring sampling weights

income.shares<-Y
 for (i in 1:length(Y)){
     income.shares[i]<-(sum(Y[1:i])/i)
 }

income.shares<-income.shares/mean(Y)
```

```
plot(pop.shares, income.shares, type="l", col=4, lty=3,
 main="Bonferroni curve", xlab="population shares", ylab="cumulative
 income shares") # there are many possible options for plot in R
abline(h=1, col="red", lty="dashed") # dashed red horizontal line
```

We can write our own function and use it then to estimate the Bonferroni index.

The general structure of a function in R is:

```
function.name <- function(x) {
      some code
      return(z)
}
```

Where "x" is the argument of the function and "z" is the output of the function.
For example if you try:

```
Mean <- function(x) {
      z<-sum(x)/length(x)
      return(z)
}
```

You will get exactly what the built-in function "mean" does:

```
Mean(data$y)==mean(data$y)
```

The Bonferroni index is slightly more difficult to write:

```
bonferroni <- function(x) {
      m<-mean(x)
      x<-x[order(x)]
      mi<-rep(0,length(x))
      for (i in 1:length(x)){
            mi[i]<-mean(x[1:i])
      }
      return(sum(m-mi)/m*(1/length(mi)))
 }

bonferroni(data$y)
```

An R function can have more than one argument. Assume you want to estimate the weighted mean of the incomes:

```r
w.mean<-function(x,f){
 wi<-f/sum(f)
 return(sum(wi*x))
}
```

The use of sampling weights (*pesofit*) will then result in a different mean.

```r
mean(data$y)
w.mean(data$y, data$pesofit)
```

Exercise: tray to write the Bonferroni index taking into account the sampling weights.[2]

1.3 Bootstrap your function

To obtain standard errors and confidence intervals for the Bonferroni index we can load the "boot" package written by Angelo Canty and Brian Ripley. The workhorse of the package is the function "boot" which generates bootstrap replicates of a statistic applied to data.
The basic arguments of the function boot are: the data, the function you want to bootstrap, and the number of replications (*R*)

If you try to use the bonferroni function as argument for the bootstrap function you get an error message:

```r
library(boot)
boot(data$y, bonferroni, 10)
```

When willing to bootstrap a function with the package boot you need to specify, after the argument, an indicator used for the resampling. For example:

```r
b.mean<-function(x, d){
 y<-x[d]
 return(mean(y))
}

my.b.mean<-boot(data$y, b.mean, R=100)
my.b.mean
```

[2] *Hint*: one way to do it very quickly is to use the function *rep()*.

Similarly:

```
b.bonferroni <- function(y,d) {
 y<-y[d]
 m<-mean(y)
 y<-y[order(y)]
 mi<-rep(0,length(y))
 for (i in 1:length(y)){
      mi[i]<-mean(y[1:i])
 }
 return(sum(m-mi)/m*(1/length(mi)))
}

bonf.b<-boot(data$y, b.bonferroni, R=20)
bonf.b
bonferroni(data$y)
```

A boot object is made of a number of elements that can be used to estimate confidence intervals:

```
summary(bonf.b)
bonf.b[2]

# bias
mean(bonf.b$t)-bonferroni(data$y)

# standard error
sd(bonf.b$t)
```

The function boot.ci() simplifies the job producing a number of possible confidence intervals based on an object produced by the function boot().

```
bonf.b<-boot(data$y, b.bonferroni, R=100)

# confidence interval
boot.ci(bonf.b, type="perc")
quantile(bonf.b$t, c(0.025,0.975))

boot.ci(bonf.b, type="norm")
bonf.b$t0-sd(bonf.b$t)*1.96
bonf.b$t0+sd(bonf.b$t)*1.96
```

References

Chakravarty, S. R. 2007. 'A deprivation-based axiomatic characterization of the absolute Bonferroni index of inequality,' Journal of Economic Inequality, 5: 339-351.