LaboratorioBonus-MD

Leislie R. Manjarrez O.

2023-03-19

Hecho con gusto por Leislie R. Manjarrez O.

Laboratorio Bonus- Ejemplos EconGeo

Instalamos la paquetria install.packages(“devtools”) install.packages(“EconGeo”)

Llamamos a las librerias

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(EconGeo)
```

```
##
## Please cite EconGeo in publications as:
```

```
##  Balland, P.A. (2017) Economic Geography in R: Introduction to the EconGeo Package, Papers in Evolutionary Eco
nomic Geography, 17 (09): 1-75
```

Example 4: Plot a Hoover curve (pag. 7) Esta funcion traza una curva de Hoover (1936) a partir de matrices de regiones-industrias.Generamos vectores de conteo industrial y poblacional

```
ind <- c(0,10,10,30,50)
pop <- c(10,15,20,25,30)
```

Verificamos el vector ind

```
ind
```

```
## [1]  0 10 10 30 50
```

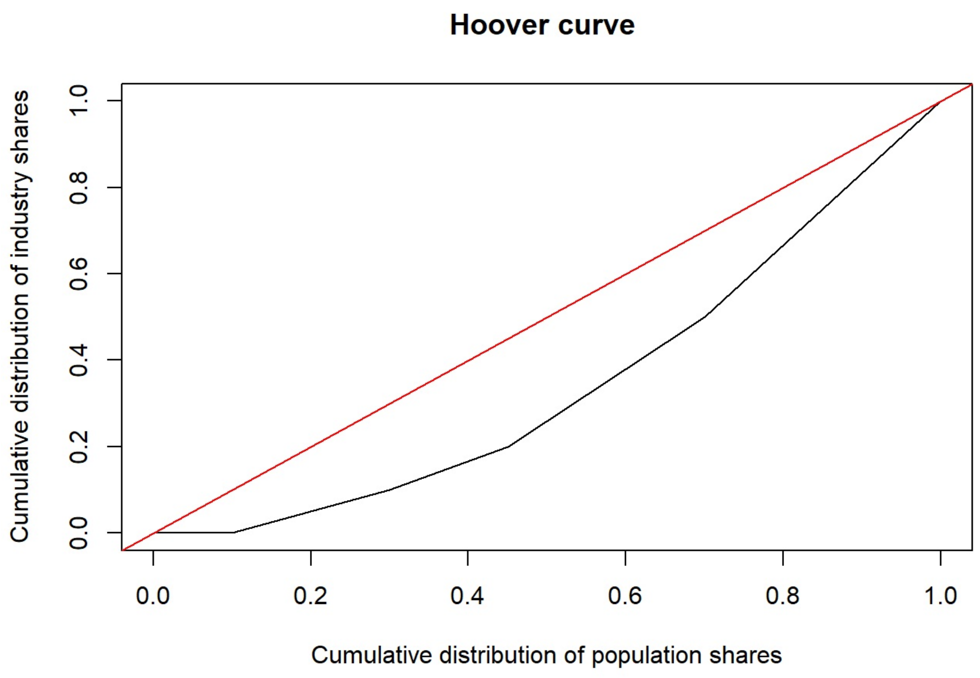
Verificamos el vector pop

```
pop
```

```
## [1] 10 15 20 25 30
```

Ejecute la funcion (30% de la poblacion produce el 50% de la produccion industrial)

```
Hoover.curve (ind,pop)
```



Calcule el Gini de Hoover correspondiente

```
Hoover.Gini(ind,pop)
```

```
## [1] 0.31
```

Gini (pag.14) La funcion calcula el coeficiente de Gini. El indice de Gini mide la desigualdad espacial. Va desde 0 (perfecta igualdad de ingresos) a 1 (perfecta desigualdad de ingresos) y se deriva de la curva de Lorenz Ejemplo: Generamos los vectores

```
ind <- c(0,10,10,30,50)
```

Corremos la funcion

```
Gini(ind)
```

```
## [1] 0.48
```

Generamos una matriz industria-region

```
mat = matrix (
      c(0,1,0,0,
        0,1,0,0,
        0,1,0,0,
        0,1,0,1,
        0,1,1,1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4")
```

Corremos la funcion

```
Gini(mat)
```

```
##   Industry Gini
## 1       I1  NaN
## 2       I2  0.0
## 3       I3  0.8
## 4       I4  0.6
```

Corremos la funcion y agregamos todas las industrias

```
Gini(rowSums(mat))
```

```
## [1] 0.25
```

Corremos la funcion para la industria 1 solamente (igualdad perfecta)

```
Gini(mat[,1])
```

```
## [1] NaN
```

Corremos la funcion para la industria 2 solamente (igualdad perfecta)

```
Gini(mat[,2])
```

```
## [1] 0
```

Corremos la funcion para la industria 3 solamente (desigualdad perfecta: max Gini = (5-1)/5)

```
Gini(mat[,3])
```

```
## [1] 0.8
```

Corremos la funcion para la industria 4 solamente (el 40% superior produce el 100% de la producción)

```
Gini(mat[,4])
```

```
## [1] 0.6
```

Hoover Gini (pag.26) Esta funcion calcula el Gini de Hoover (la cual lleva el nombre de Hedgar Hoover). El indice de Hoover es una medida de desigualdad espacial. Va desde 0 (perfecta igualdad de ingresos) a 1 (perfecta desigualdad de ingresos) y se calcula a partir de la curva de Hoover asociada a una determinada distribución de población, industrias o tecnologías y una categoría de referencia Ejemplo: Generamos los vectores de conteo industrial y poblacional

```
ind <- c(0,10,10,30,50)
pop <- c(10,15,20,25,30)
```

Corremos la funcion (30% de la poblacion produce el 50% de la produccion industrial)

```
Hoover.Gini(ind, pop)
```

```
## [1] 0.31
```

Generamos una matriz region-industria

```
mat = matrix(
  c(0,10,0,0,
    0,15,0,0,
    0,20,0,0,
    0,25,0,1,
    0,30,1,1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4")
```

Corremos la funcion

```
Hoover.Gini(mat,pop)
```

```
##   Industry Hoover.Gini
## 1       I1         NaN
## 2       I2       0.000
## 3       I3       0.700
## 4       I4       0.475
```

Corremos la funcion agregando todas las industrias

```
Hoover.Gini(rowSums(mat), pop)
```

```
## [1] 0.015
```

Corremos la funcion con la industria 1 solamente

```
Hoover.Gini(mat[,1], pop)
```

```
## [1] NaN
```

Corremos la funcion con la industria 2 solamente (perfectamente proporcional con la pobalcion)

```
Hoover.Gini(mat[,2], pop)
```

```
## [1] 0
```

Corremos la funcion con la industria 3 solamente (30% de la poblacion produce 100% de la produccion)

```
Hoover.Gini(mat[,3], pop)
```

```
## [1] 0.7
```

Corremos la funcion con la industria 4 solamente (55% de la poblacion produce 100% de la produccion)

```
Hoover.Gini(mat[,4], pop)
```

```
## [1] 0.475
```

locational.Gini (pag. 34) Esta funcion calcula el coeficiente de Gini de ubicacion propuesto por Krugman a partir de matrices de regiones-industrias. Cuanto mayor sea el coeficiente (limite teorico = 0,5), mayor sera la concentracion industrial. El Gini de ubicacion de una industria que no esta localizada en absoluto (perfectamente distribuida) en proporcion al empleo total seria 0. Ejemplo: Generamos una matriz region-industria

```
mat = matrix(
  c(100,0,0,0,0,
    0,15,5,70,10,
    0,20,10,20,50,
    0,25,30,5,40,
    0,40,55,5,0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4","I5")
```

Corremos la funcion

```
locational.Gini(mat)
```

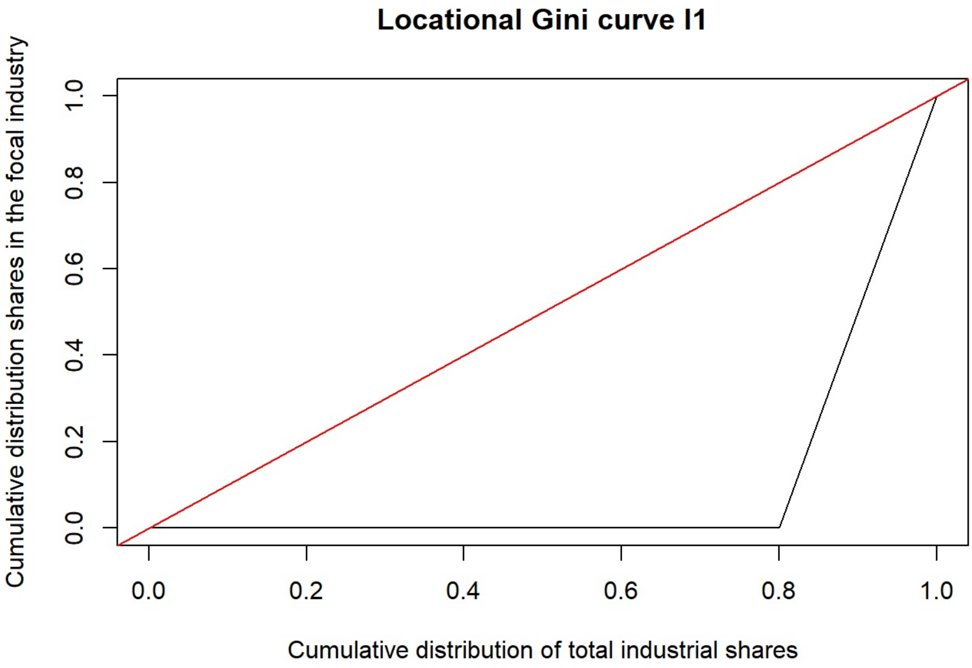
```
##   Industry Loc.Gini
## 1       I1     0.40
## 2       I2     0.18
## 3       I3     0.27
## 4       I4     0.31
## 5       I5     0.28
```

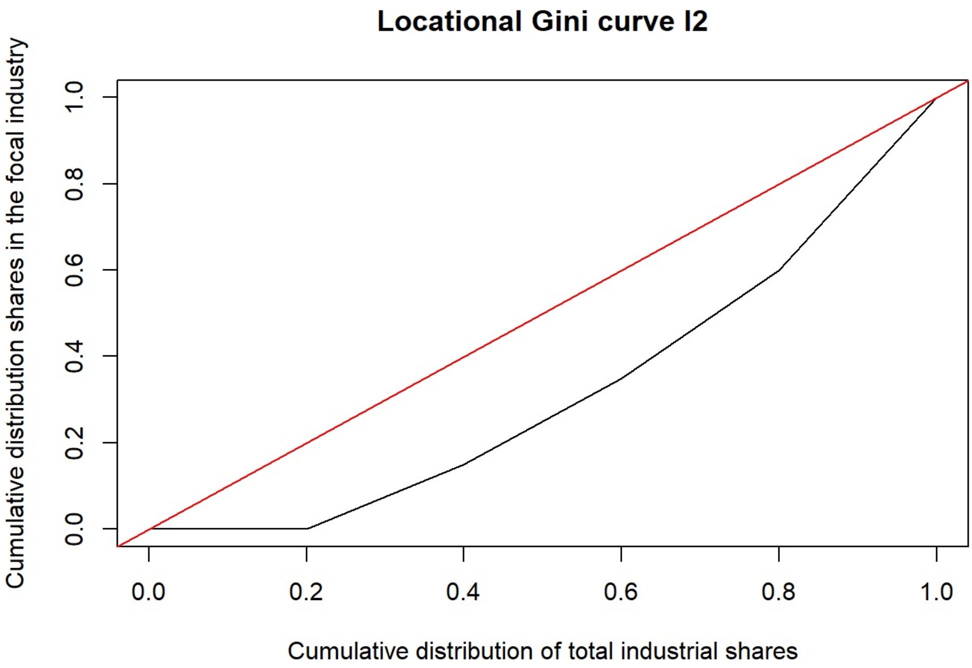
locational.Gini.curve (pag.35) Esta funcion traza una curva de Gini de ubicacion siguiendo a Krugman a partir de matrices de regiones-industrias Ejemplo: Generamos una matriz region-industria

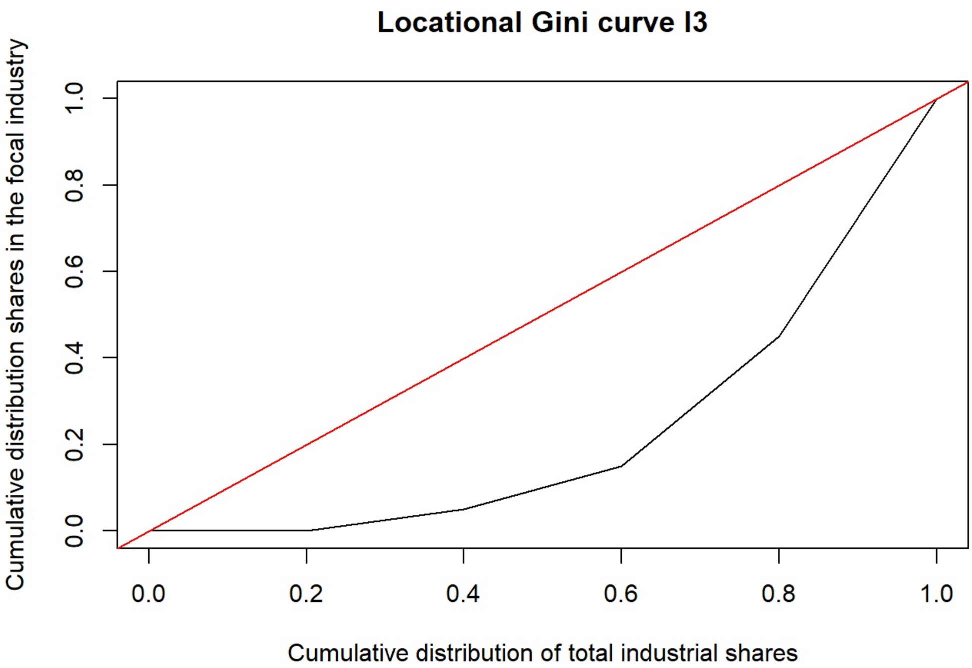
```
mat = matrix(
  c(100,0,0,0,0,
    0,15,5,70,10,
    0,20,10,20,50,
    0,25,30,5,40,
    0,40,55,5,0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4","I5")
```

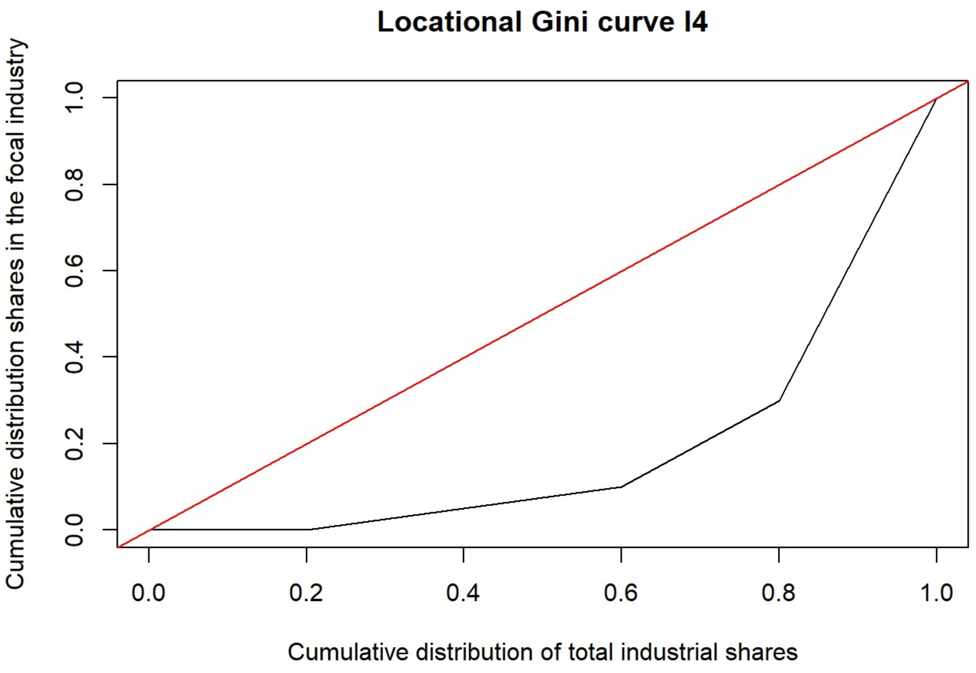
Corremos la funcion

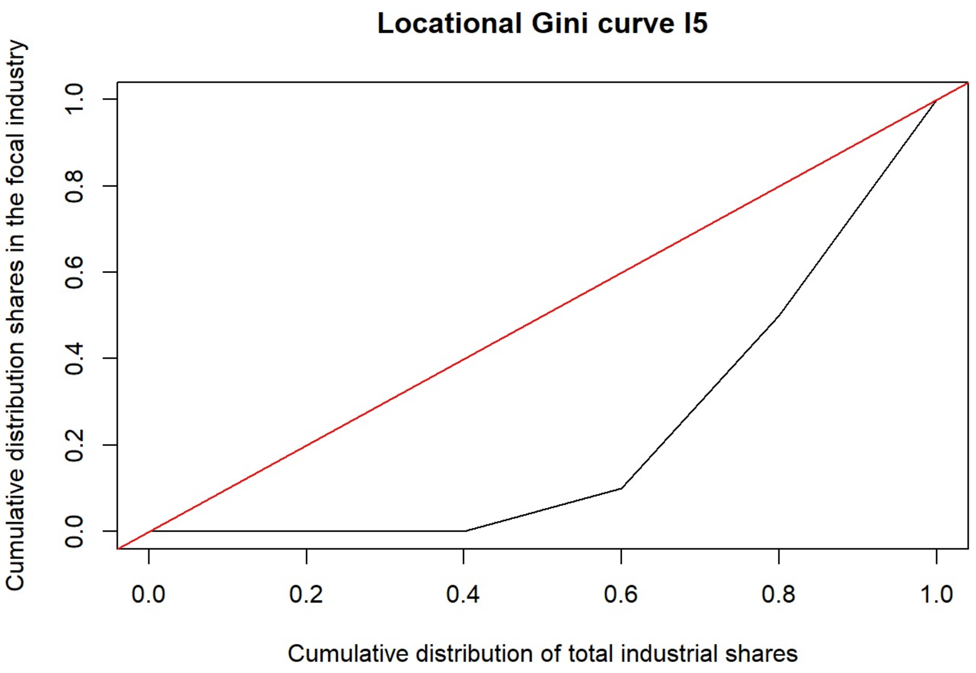
```
locational.Gini.curve(mat)
```


Locational Gini curve I1
Cumulative distribution shares in the focal industry
Cumulative distribution of total industrial shares
0.0
0.2
0.4
0.6
0.8
1.0


Locational Gini curve I2
Cumulative distribution shares in the focal industry
Cumulative distribution of total industrial shares
0.0
0.2
0.4
0.6
0.8
1.0


Locational Gini curve I3
Cumulative distribution shares in the focal industry
Cumulative distribution of total industrial shares
0.0
0.2
0.4
0.6
0.8
1.0






```
locational.Gini.curve(mat, pdf = TRUE)
```

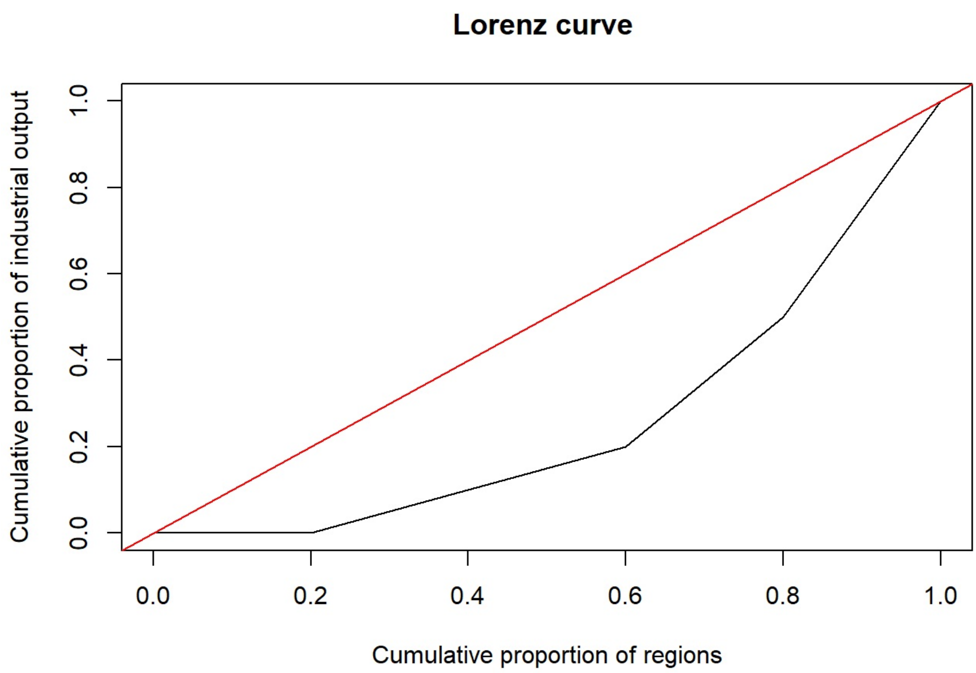
```
## [1] "locational.Gini.curve.pdf has been saved to your current working directory"
```

Lorenz.curve (pag.36) Esta funcion traza una curva de Lorenz a partir de recuentos industriales regionales. Esta curva indica la distribucion desigual de una industria entre regiones Ejemplo: Generamos vectores de recuentos industriales

```
ind <- c(0,10,10,30,50)
```

Corremos la funcion

```
Lorenz.curve(ind)
```




```
Lorenz.curve(ind, pdf = TRUE)
```

```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(ind, plot = FALSE)
```

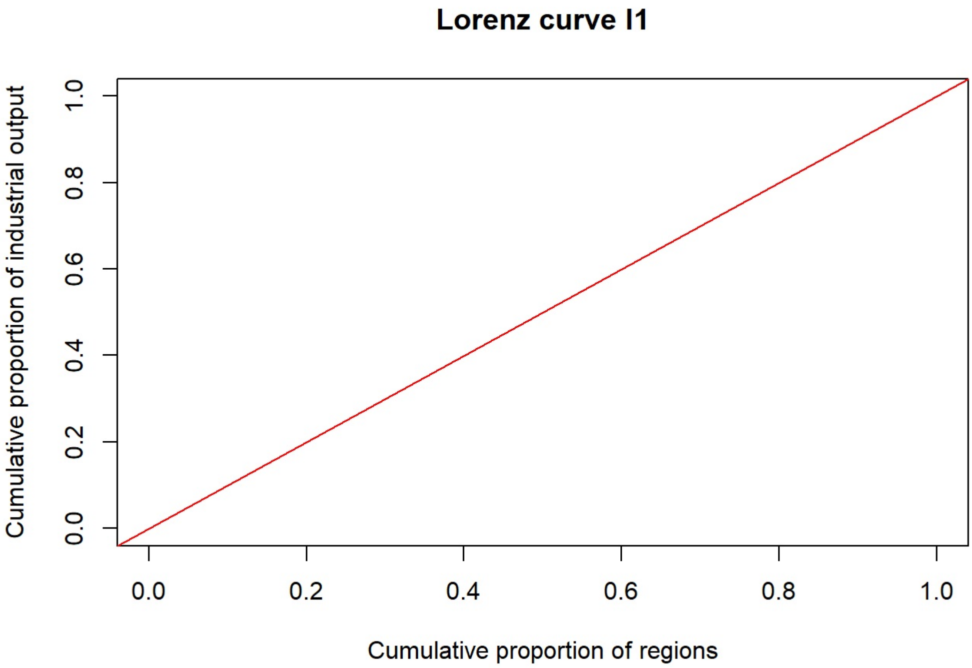
```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
## [1] 0.0 0.0 0.1 0.2 0.5 1.0
```

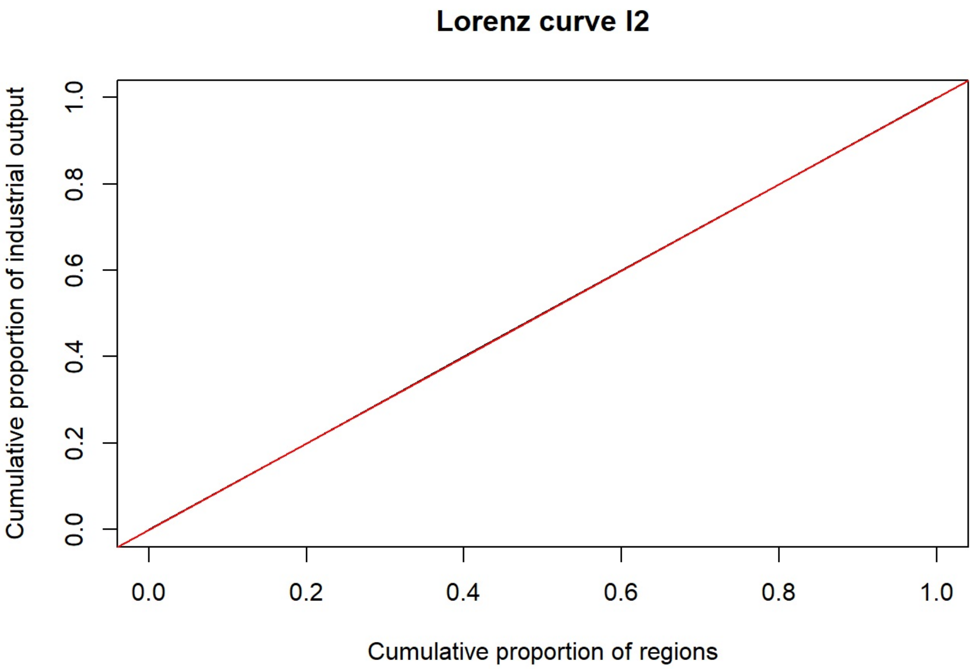
Generamos una matriz region-industria

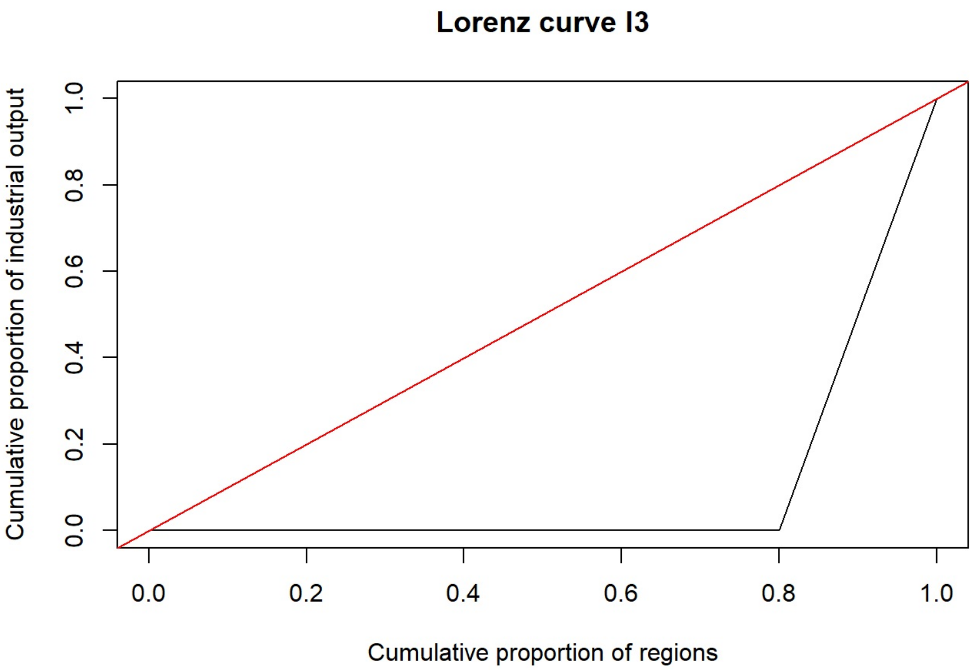
```
mat = matrix(
  c(0,1,0,0,
    0,1,0,0,
    0,1,0,0,
    0,1,0,1,
    0,1,1,1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4")
```

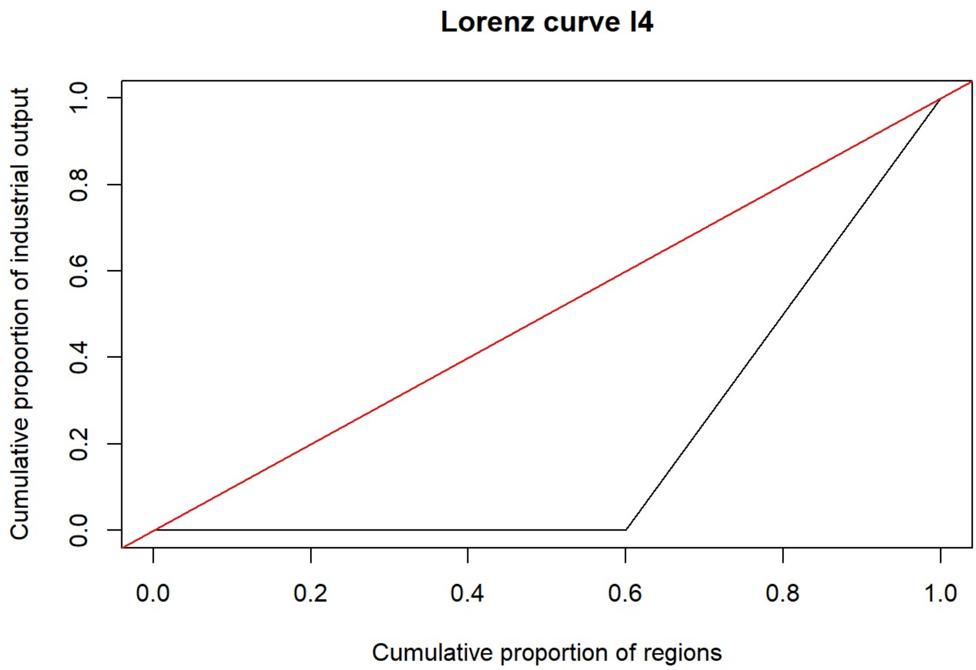
Corremos la funcion

```
Lorenz.curve(mat)
```


Lorenz curve I1
Cumulative proportion of industrial output
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative proportion of regions
0.0
0.2
0.4
0.6
0.8
1.0


Lorenz curve I2
Cumulative proportion of industrial output
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative proportion of regions
0.0
0.2
0.4
0.6
0.8
1.0


Lorenz curve I3
Cumulative proportion of industrial output
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative proportion of regions
0.0
0.2
0.4
0.6
0.8
1.0




```
Lorenz.curve(mat, pdf = TRUE)
```

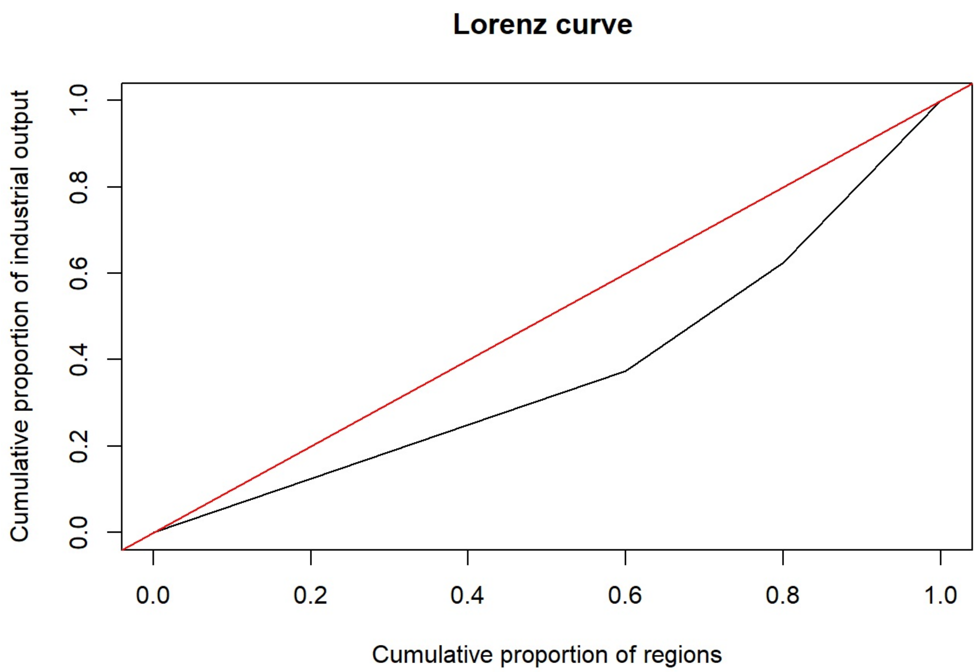
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(mat, plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
## 
## $cum.out
##      R1  R2  R3  R4  R5
##    0 NaN NaN NaN NaN NaN
```

Corremos la funcion agregando todas las industrias

```
Lorenz.curve(rowSums(mat))
```




```
Lorenz.curve(rowSums(mat), pdf = TRUE)
```

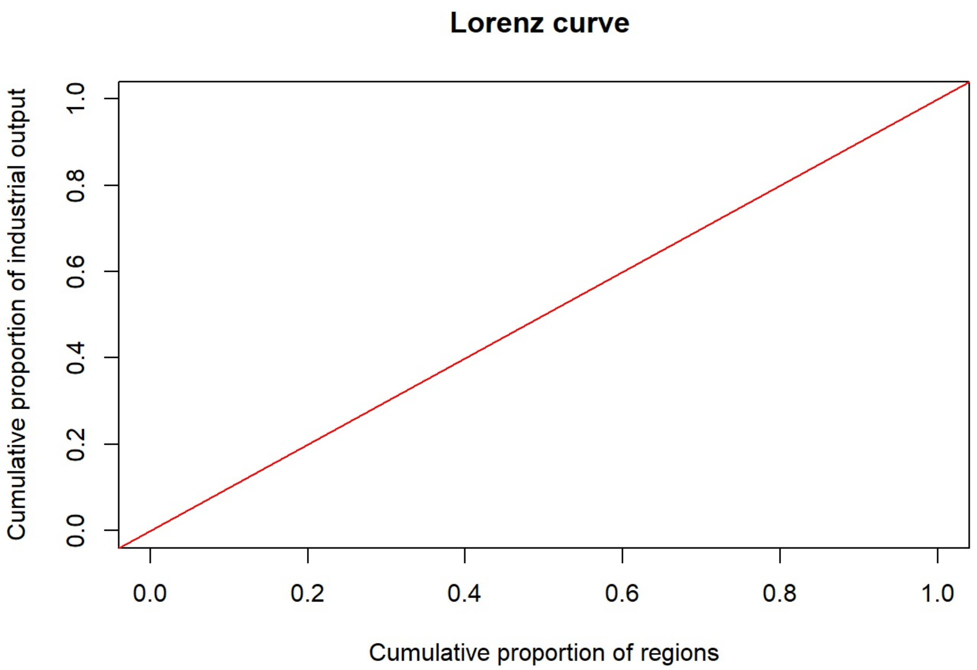
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(rowSums(mat), plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##          R1    R2    R3    R4    R5
## 0.000 0.125 0.250 0.375 0.625 1.000
```

Corremos la funcion solamente con la industria 1 (igualdad perfecta)

```
Lorenz.curve(mat[,1])
```




```
Lorenz.curve(mat[,1], pdf = TRUE)
```

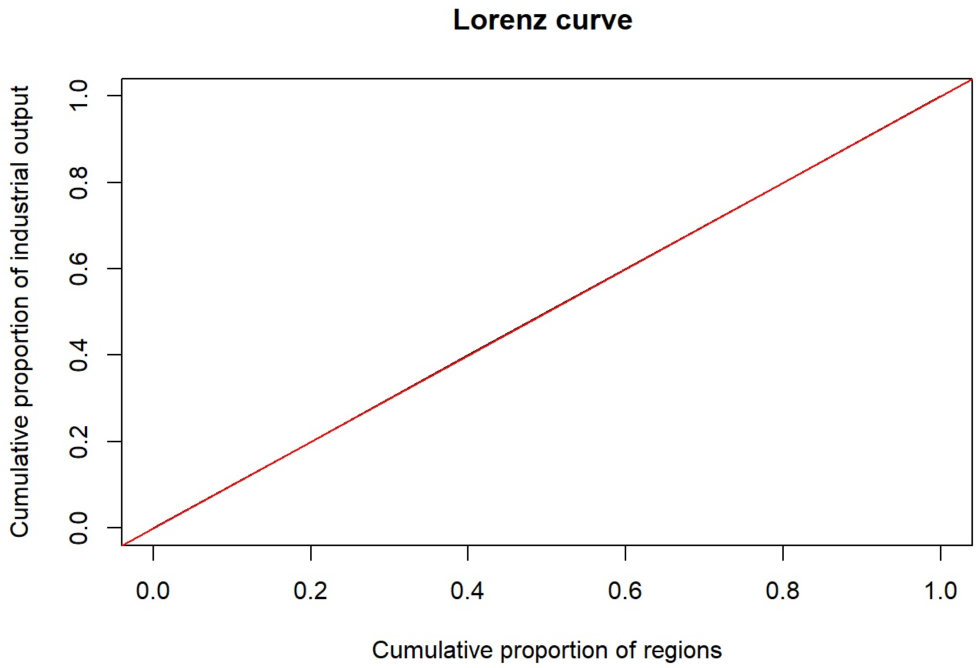
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(mat[,1], plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##      R1  R2  R3  R4  R5
##   0 NaN NaN NaN NaN NaN
```

Corremos la funcion solamente con la industria 2 (igualdad perfecta)

```
Lorenz.curve(mat[,2])
```




```
Lorenz.curve(mat[,2], pdf = TRUE)
```

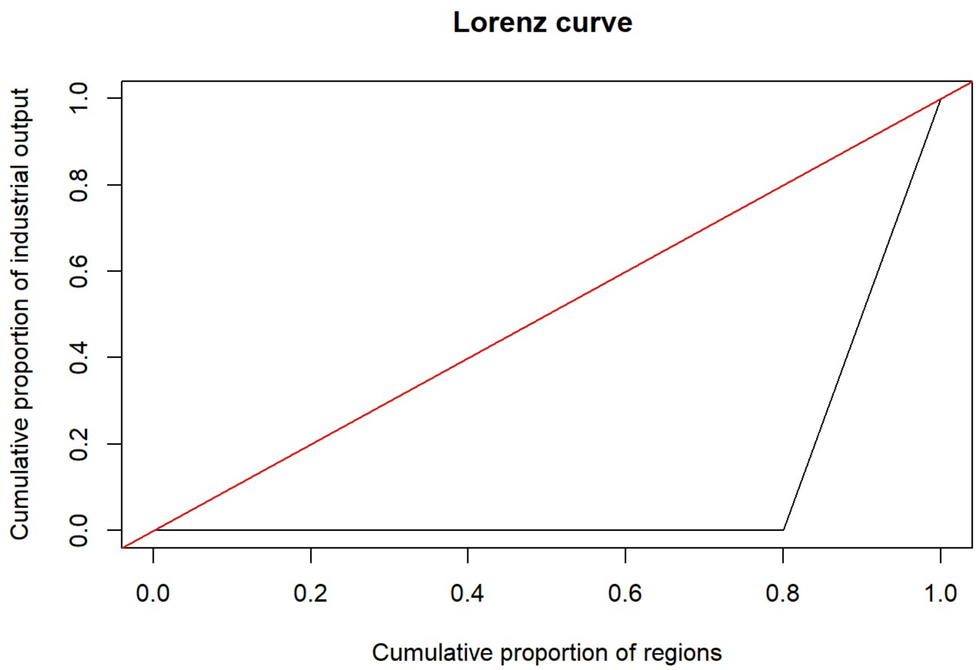
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(mat[,2], plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##      R1  R2  R3  R4  R5
## 0.0 0.2 0.4 0.6 0.8 1.0
```

Corremos la funcion solamente con la industria 3 (desigualdad perfecta)

```
Lorenz.curve(mat[,3])
```




```
Lorenz.curve(mat[,3], pdf = TRUE)
```

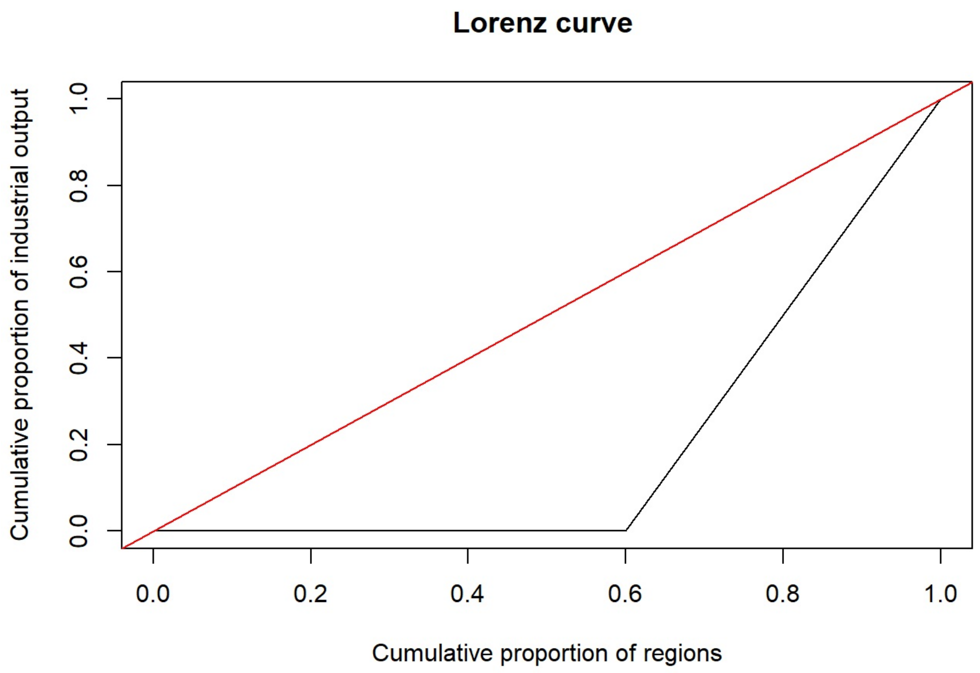
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(mat[,3], plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##    R1 R2 R3 R4 R5
##  0  0  0  0  0  1
```

Corremos la funcion solamente con la industria 4 (el 40% superior produce el 100% de la produccion)

```
Lorenz.curve(mat[,4])
```




```
Lorenz.curve(mat[,4], pdf = TRUE)
```

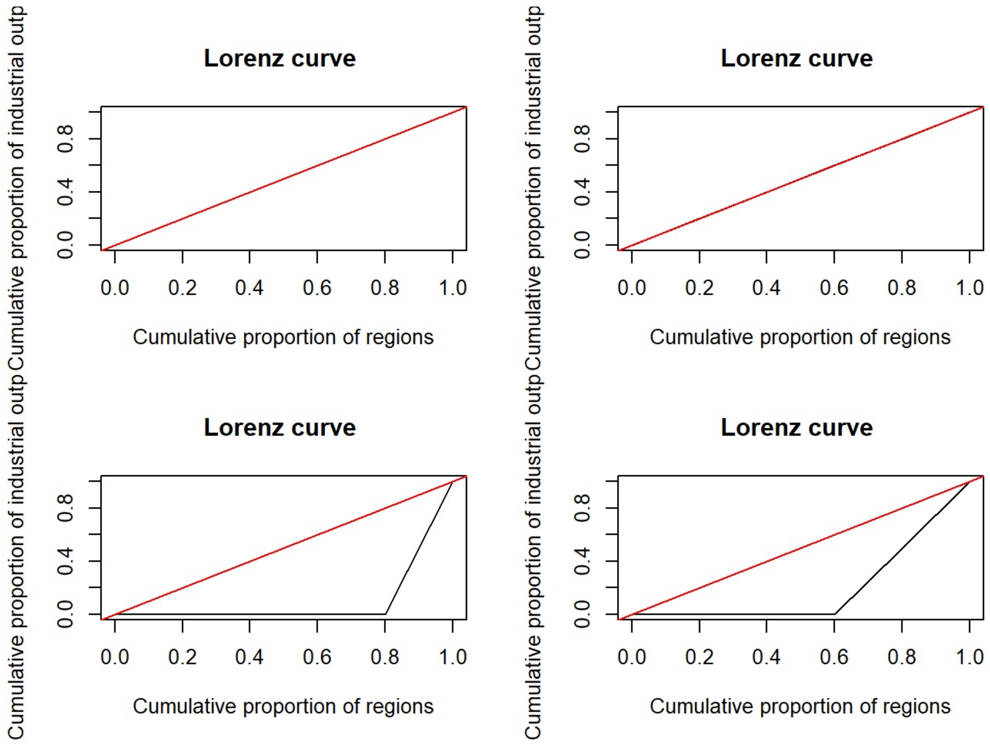
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
Lorenz.curve(mat[,4], plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##      R1  R2  R3  R4  R5
## 0.0 0.0 0.0 0.0 0.5 1.0
```

Comparamos la distribucion de las industrias

```
par(mfrow=c(2,2))
Lorenz.curve(mat[,1])
Lorenz.curve(mat[,2])
Lorenz.curve(mat[,3])
Lorenz.curve(mat[,4])
```


Lorenz curve
Lorenz curve
Lorenz curve
Lorenz curve
Cumulative proportion of industrial outp
Cumulative proportion of regions
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.4
0.8