Ejemplos EconGeo

Luz Yolanda Rivera

10/23/2021

#llamar a Econgeo de biblioteca

```
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
```

generate vectors of industrial and population count

generar vectores de conteo industrial(ind) y poblacional(pop)

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

check the ind vector

#comprobar el vector ind y el vector pop

```
ind
```

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

```
pop
```

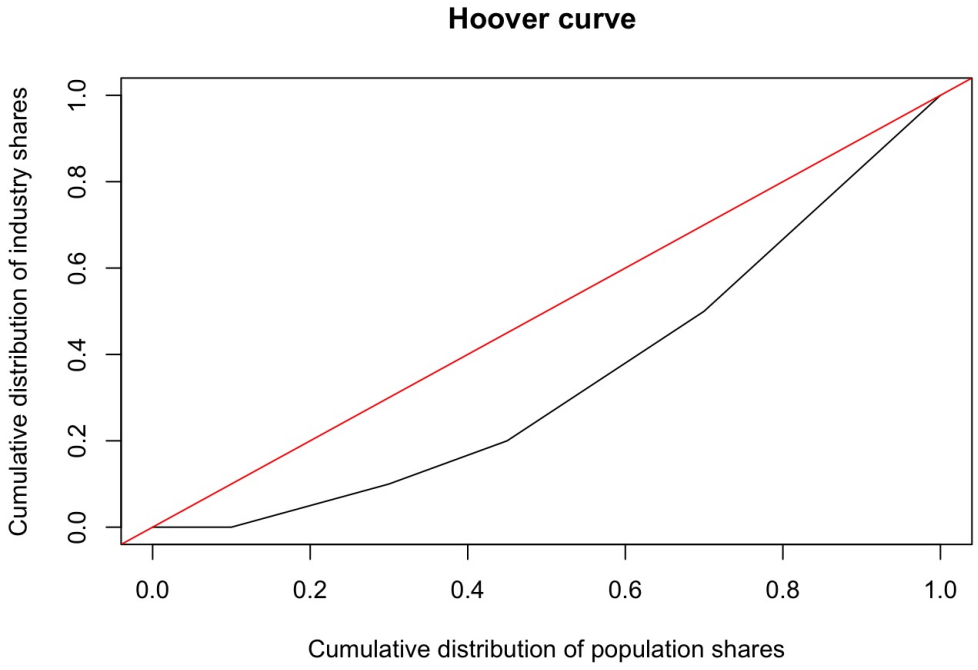
```
## [1] 10 15 20 25 30
```

run the function (30% of the population produces 50% of the industrial output)

se corre la función hoover.curve y apreciamos como el 30% de la población produce

#el 50% de la producción que genera la industria y se aprecia en la gráfica

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



compute the corresponding Hoover Gini

#Se calcula el coheficiente Gini correspondiente (es el 31%)

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

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

#Ejemplo 2. GINI

generate vectors of industrial count

generar vectores de conteo industrial(ind)

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

run the function

#corremos la función para calcuar Gini de la industria

```
Gini (ind)
```

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

generate a region - industry matrix

generamos una matriz de la industria de la región de 4 columnas(I1..I4) y cinco renglones de regiones (R1..R5)

```
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")
```

##revisamos la matriz que se generó

```
mat
```

```
##    I1 I2 I3 I4
## R1  0  1  0  0
## R2  0  1  0  0
## R3  0  1  0  0
## R4  0  1  0  1
## R5  0  1  1  1
```

run the function

#Hacemos el cálculo Gini de la matriz que generamos

```
Gini (mat)
```

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

run the function by aggregating all industries

##agregamos todas las industrias

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

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

run the function for industry #1 only (perfect equality)

##si ejecutamos Gini solo para la industria 1, obtendremos una igualdad perfecta

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

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

run the function for industry #2 only (perfect equality)

#Corremos la función para la Industria dos que también presenta igualdad perfecta=0

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

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

run the function for industry #3 only (perfect unequality: max Gini = (5-1)/5)

corremos la función para la industria 4 que presenta desigualdad = .8 ya que solo una región produce todo

#por eso el cálculo de 5-4/5 nos da desigualdad perfecta

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

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

run the function for industry #4 only (top 40% produces 100% of the output)

para la industria 4 es producida por el 40% de las regiones

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

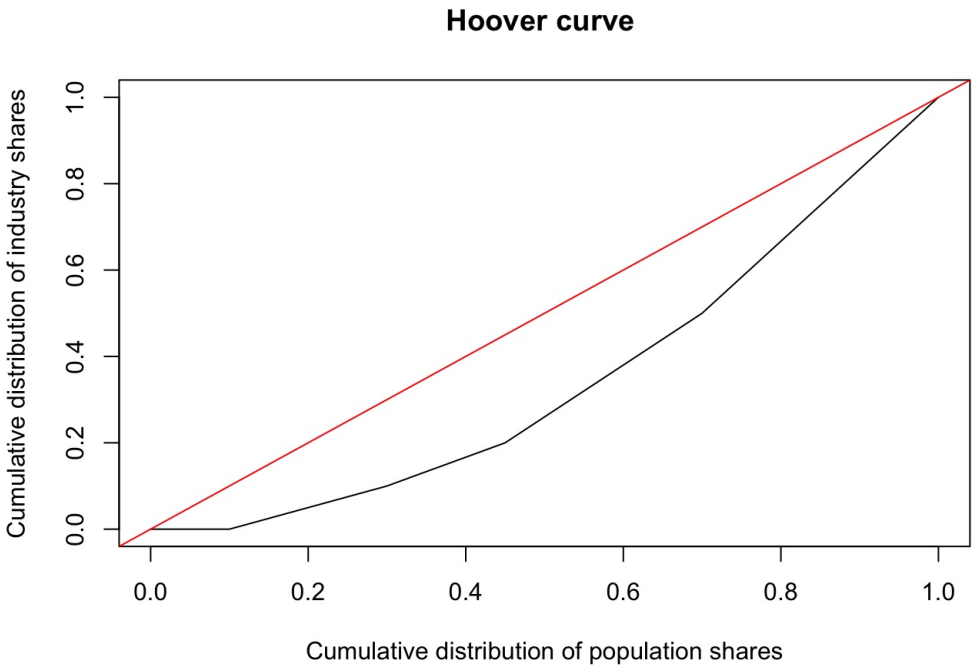
```
## [1] 0.6
```

#Ejemplo 3. HOOVER GINI

utilizamos los vectores que ya habíamos generado antes

##ind <- c(0, 10, 10, 30, 50) ##pop <- c(10, 15, 20, 25, 30) ## run the function (30% of the population produces 50% of the industrial output)
#corremos la función (se obtiene que el 30% de la población produce el 50% de la producción industrial)

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




```
Hoover.curve (ind, pop, pdf = TRUE)
```

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

```
Hoover.curve (ind, pop, plot = F)
```

```
## $cum.reg
## [1] 0.00 0.10 0.30 0.45 0.70 1.00
##
## $cum.out
## [1] 0.0 0.0 0.1 0.2 0.5 1.0
```

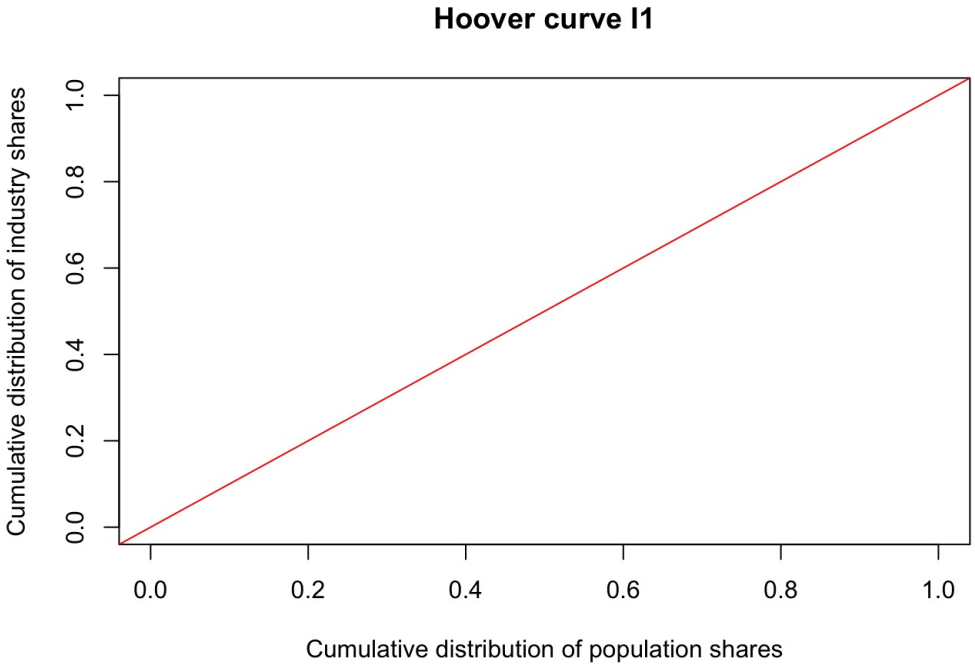
generate a region - industry matrix

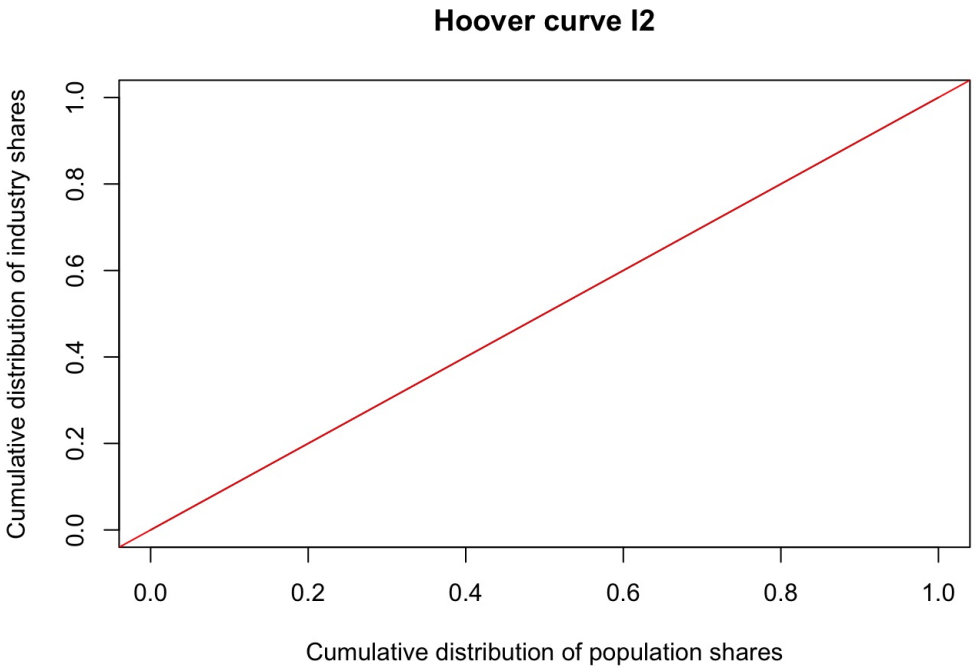
##generamos una matriz mat

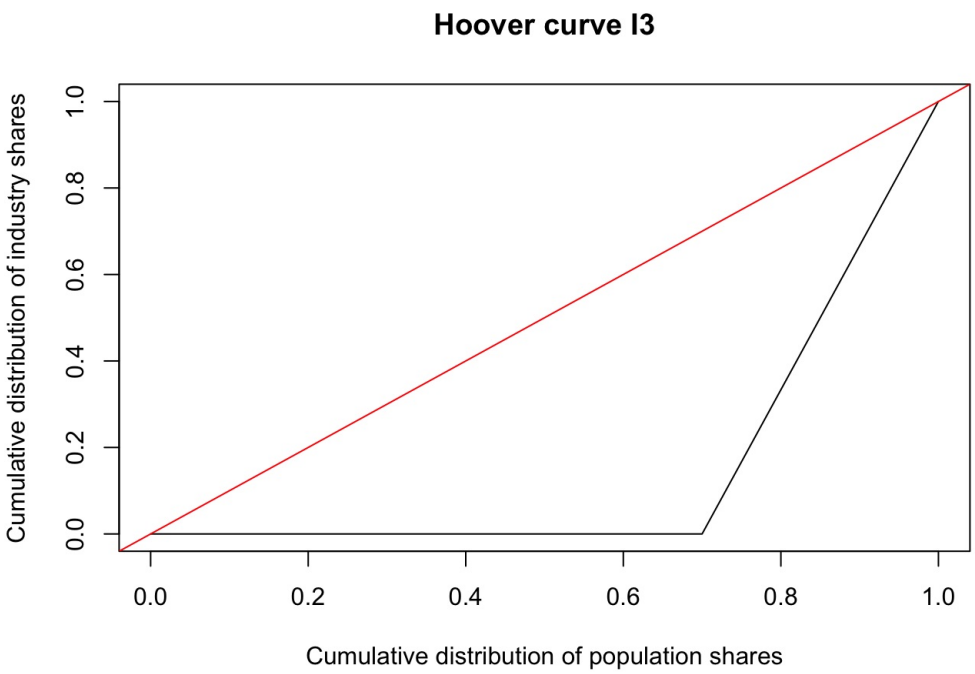
```
mat = matrix (
  c (0, 10, 0, 0,
     0, 15, 0, 0,
     0, 20, 0, 0,
     0, 25, 0, 1,
     0, 30, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")
```

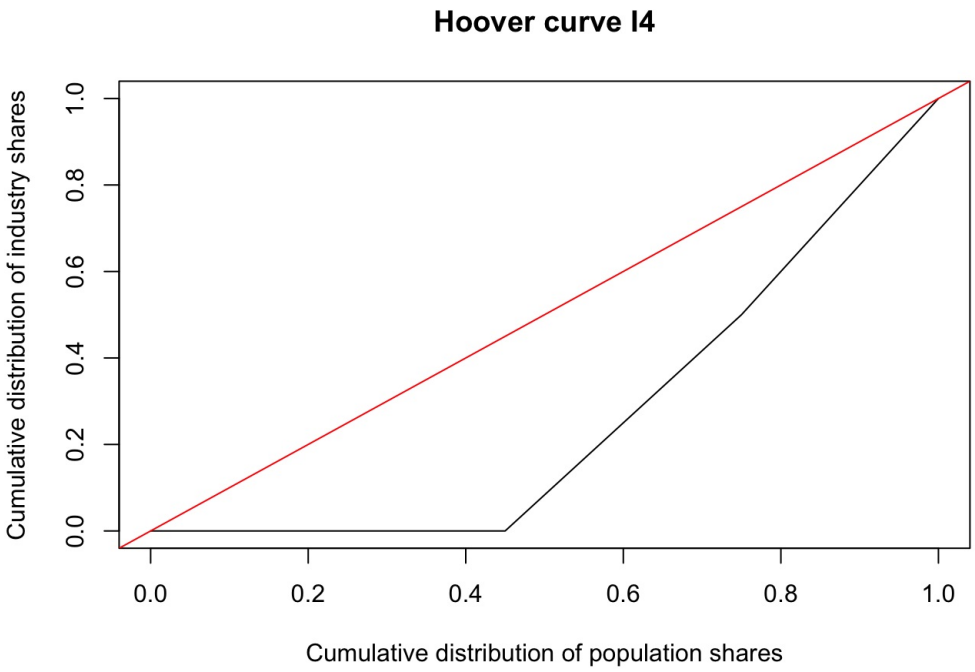
Generamos las curvas

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


Hoover curve l1
Cumulative distribution of industry shares
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of population shares


Hoover curve l2
Cumulative distribution of industry shares
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of population shares


Hoover curve l3
Cumulative distribution of industry shares
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of population shares



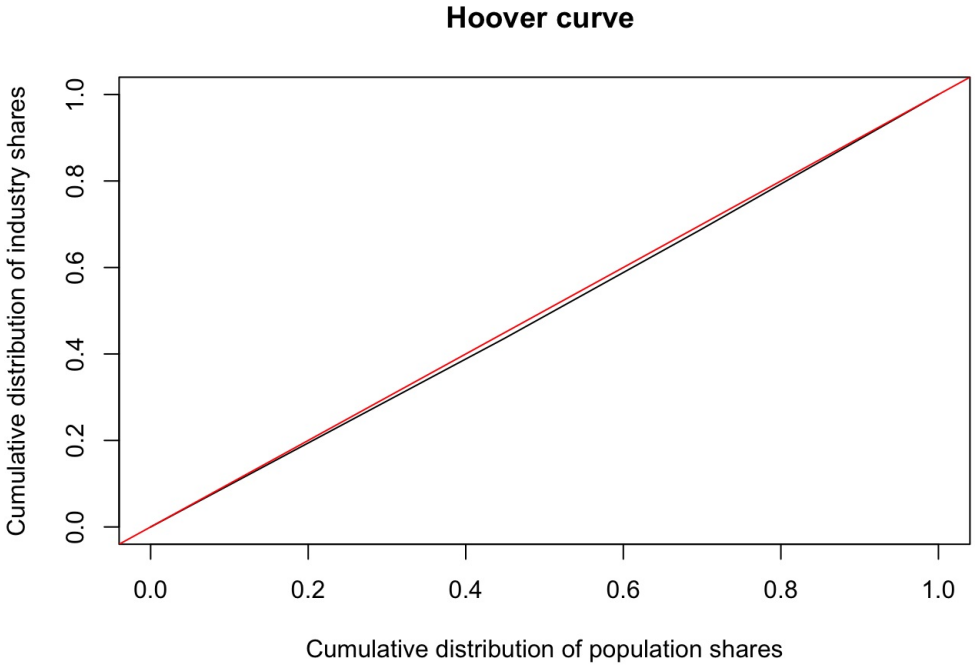

```
#Hoover.curve (mat, pop, pdf = TRUE) para generar el pdf
Hoover.curve (mat, pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
##
## $cum.out
## [1] NaN NaN NaN NaN NaN NaN
```

run the function by aggregating all industries

#agregamos todas las industrias

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



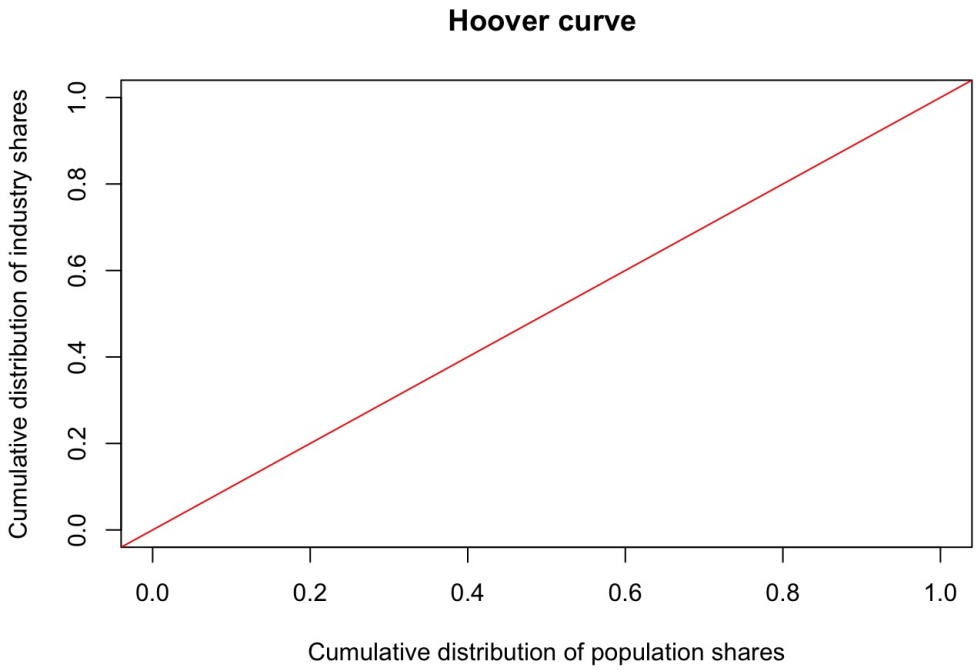

```
#Hoover.curve (rowSums(mat), pop, pdf = TRUE) si queremos generar el pdf
Hoover.curve (rowSums(mat), pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
##
## $cum.out
## [1] 0.00000000 0.09708738 0.24271845 0.43689320 0.68932039 1.00000000
```

run the function for industry #1 only

#sólo para la idustria uno

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



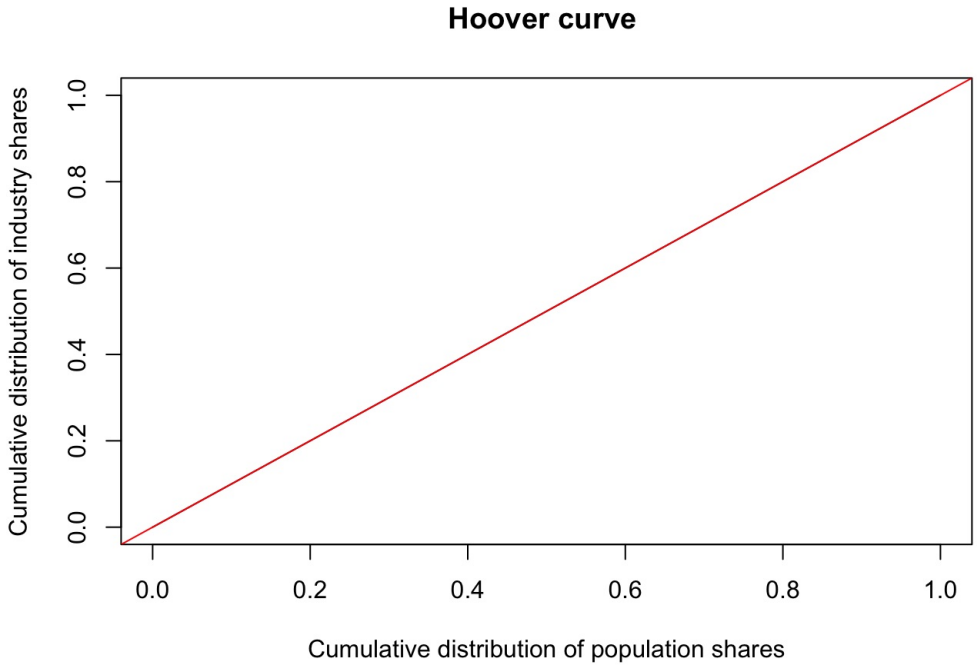

```
#Hoover.curve (mat[,1], pop, pdf = TRUE) sólo si queremos generar el pdf
Hoover.curve (mat[,1], pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
##
## $cum.out
## [1] NaN NaN NaN NaN NaN NaN
```

run the function for industry #2 only (perfectly proportional to population)

#para la industrai dos que es proporcional a la población

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



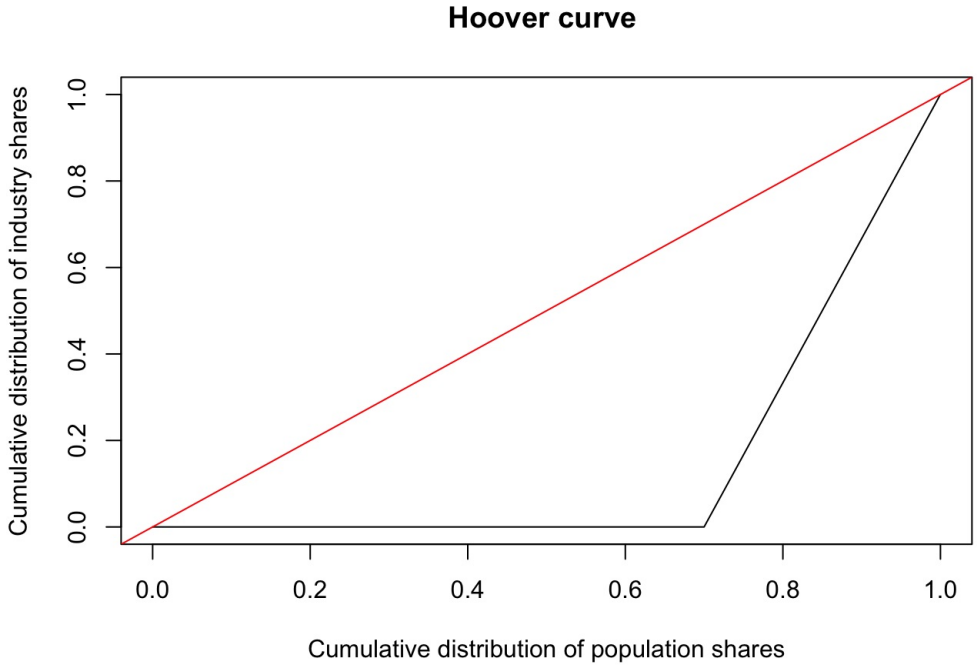

```
#Hoover.curve (mat[,2], pop, pdf = TRUE) para generar el pdf
Hoover.curve (mat[,2], pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
##
## $cum.out
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
```

run the function for industry #3 only (30% of the pop. produces 100% of the output)

#para la industria tres

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



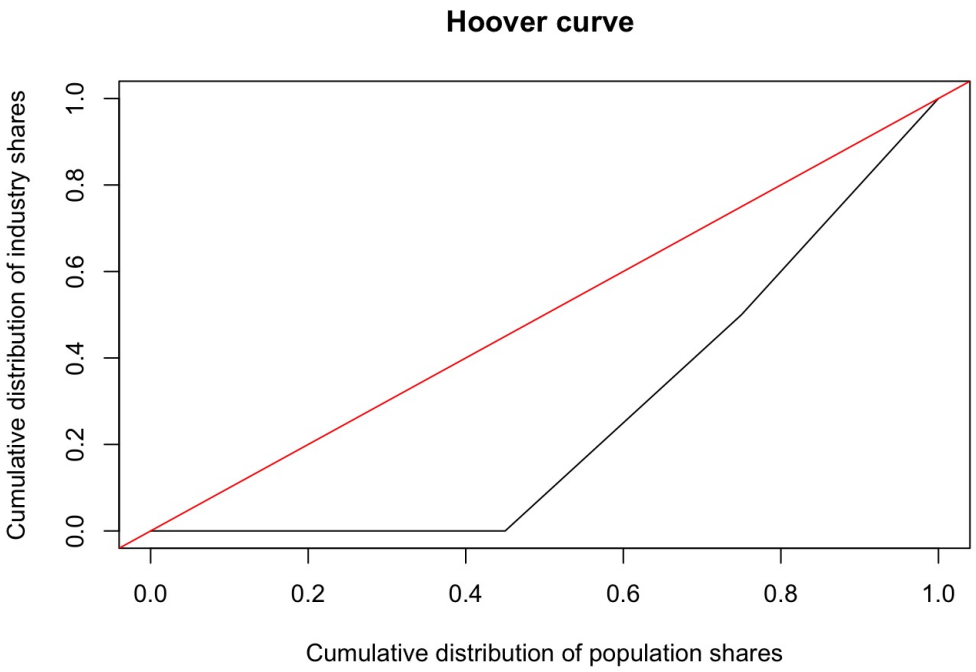

```
#Hoover.curve (mat[,3], pop, pdf = TRUE)
Hoover.curve (mat[,3], pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.70 1.00
##
## $cum.out
## [1] 0 0 0 0 0 1
```

run the function for industry #4 only (55% of the pop. produces 100% of the output)

##para la industria 4

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

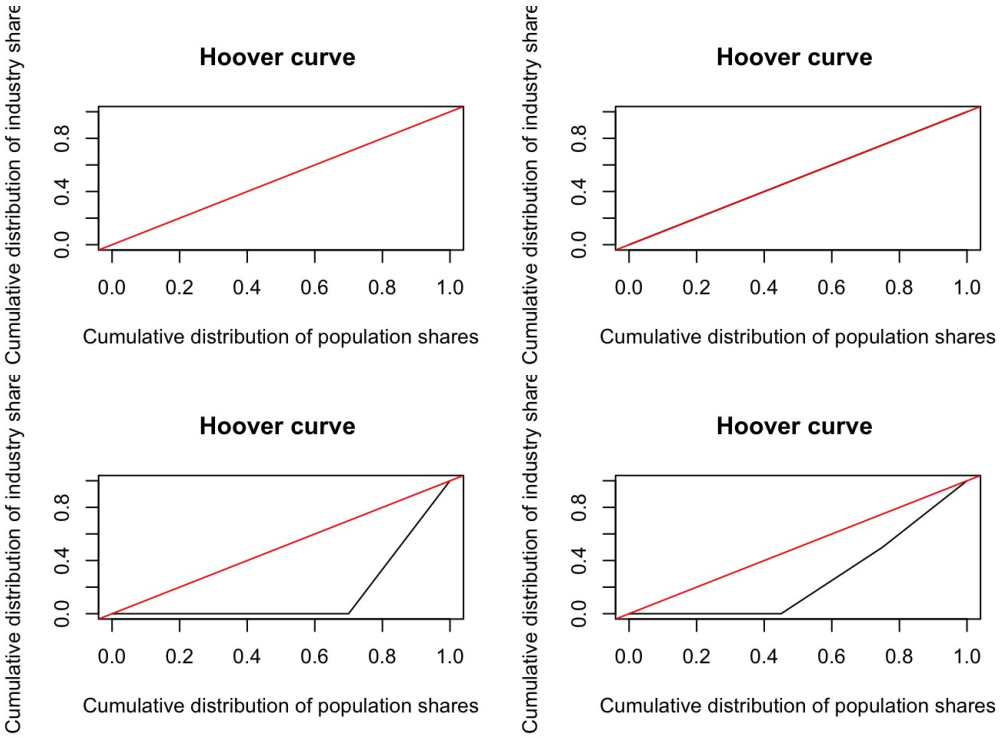



```
#Hoover.curve (mat[,4], pop, pdf = TRUE)
Hoover.curve (mat[,4], pop, plot = FALSE)
```

```
## $cum.reg
## [1] 0.00 0.10 0.25 0.45 0.75 1.00
##
## $cum.out
## [1] 0.0 0.0 0.0 0.0 0.5 1.0
```

#Compare the distribution of the #industries Comparamos la distribución de las industriaas

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



#Ejemplo 4. locational Gini curve

generate a region - industry matrix

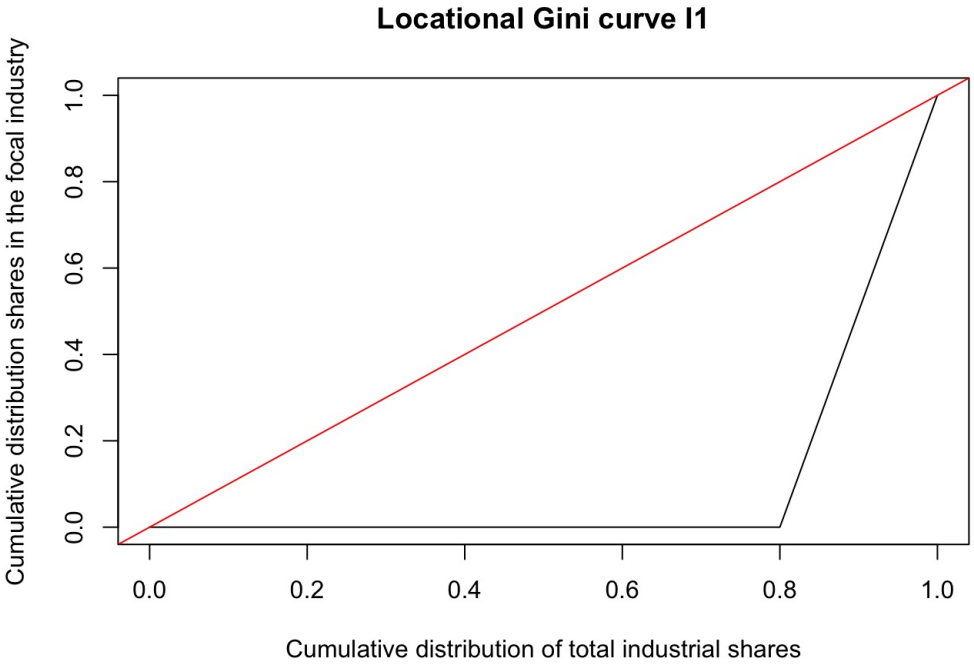
#generamos una otra matriz de industria región y la nombramos mat2

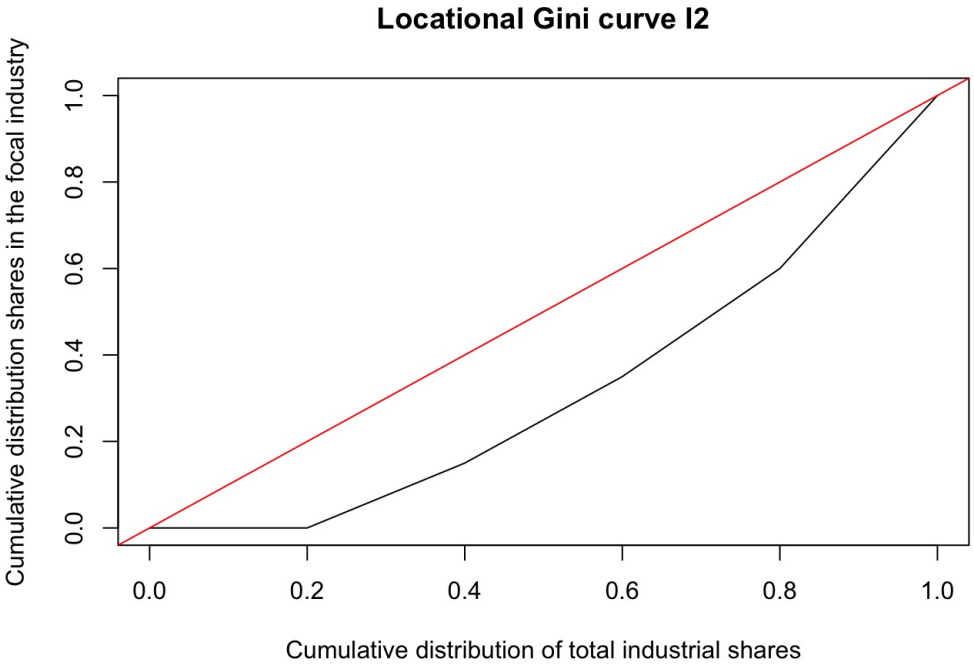
```
mat2 = matrix (
  c (100, 0, 0, 0, 0,
     0, 15, 5, 70, 10,
     0, 20, 10, 20, 50,
     0, 25, 30, 5, 40,
     0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat2) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat2) <- c ("I1", "I2", "I3", "I4", "I5")
```

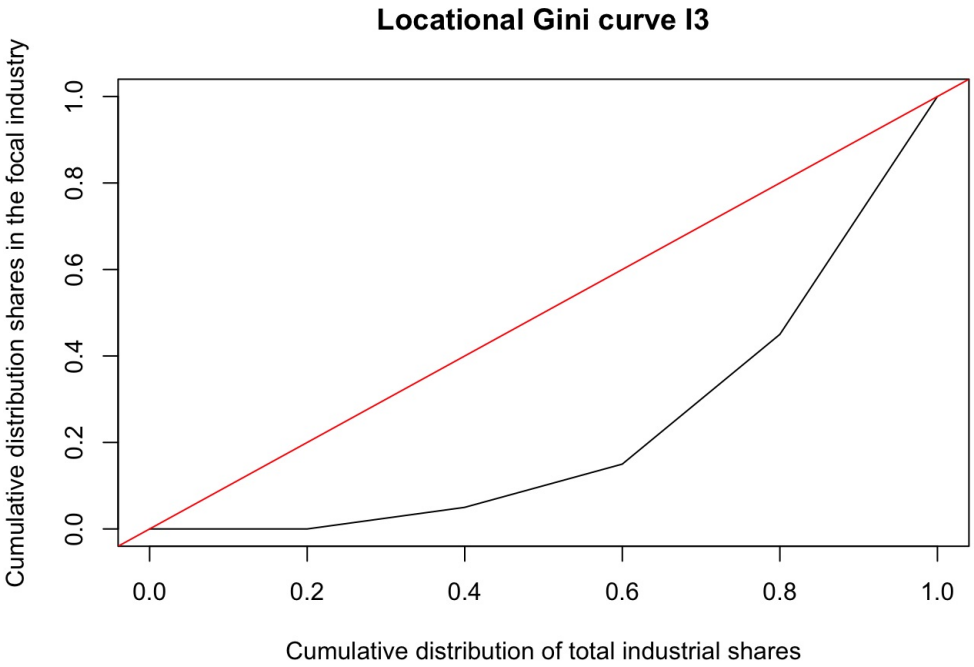
run the function (shows industry #5)

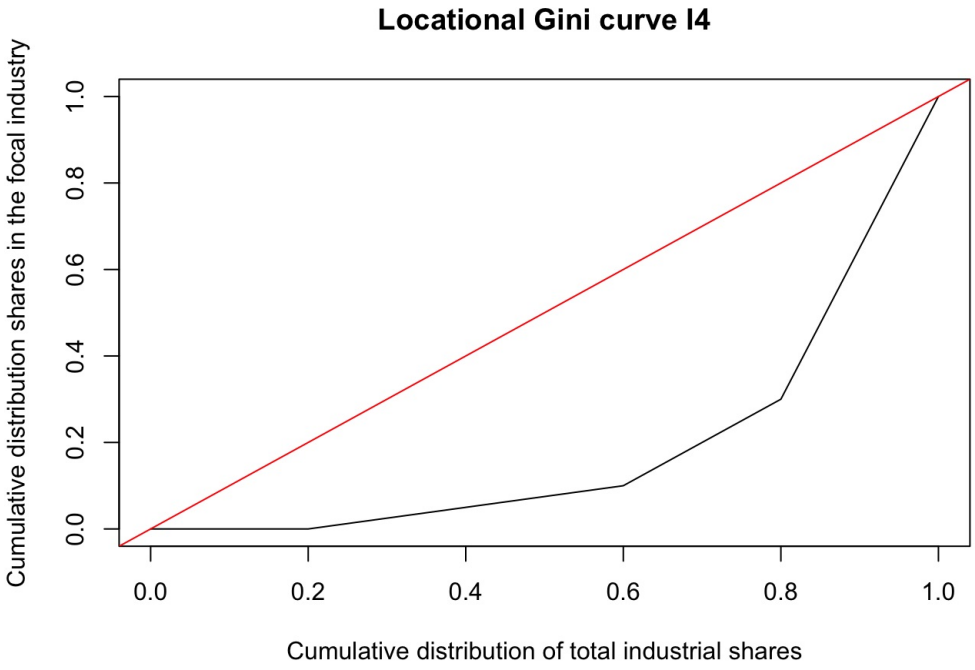
##utilizamos la función locational.gini para hacer el gráfico

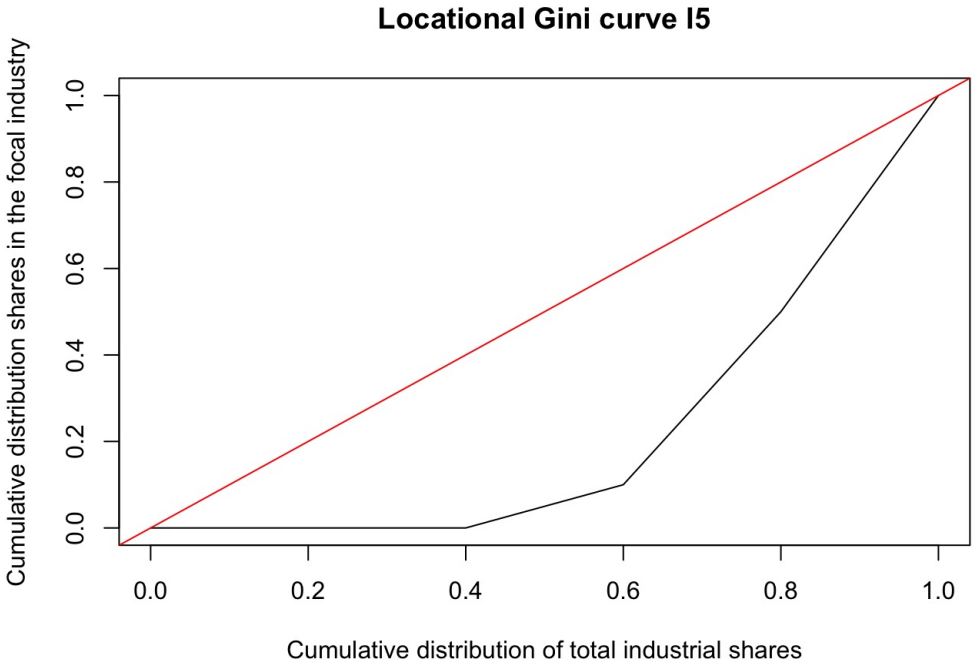
```
locational.Gini.curve (mat2)
```


Locational Gini curve l1
Cumulative distribution shares in the focal industry
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of total industrial shares


Locational Gini curve l2
Cumulative distribution shares in the focal industry
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of total industrial shares


Locational Gini curve l3
Cumulative distribution shares in the focal industry
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative distribution of total industrial shares





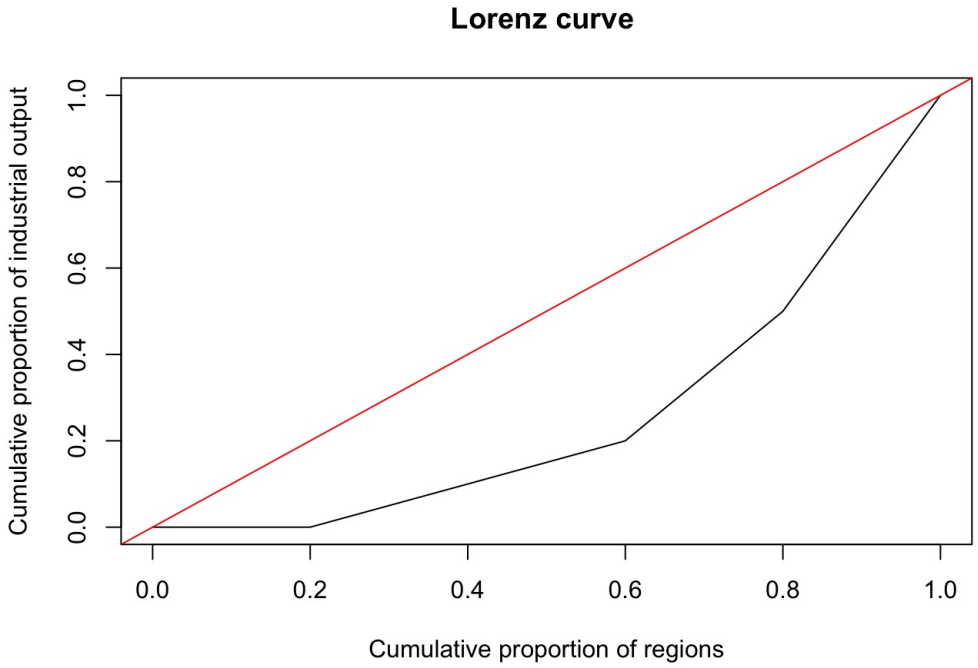

```
#locational.Gini.curve (mat2, pdf = TRUE)sólos si queremos generar el pdf
```

#Ejemplo 5. Lorenze curve

generate vectors of industrial count

##utilizamos el vector que ya habíamos creado #ind <- c(0, 10, 10, 30, 50) ## run the function #corremos la función y generamos la curva del vector

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




```
#Lorenz.curve (ind, pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (ind, plot = FALSE)
```

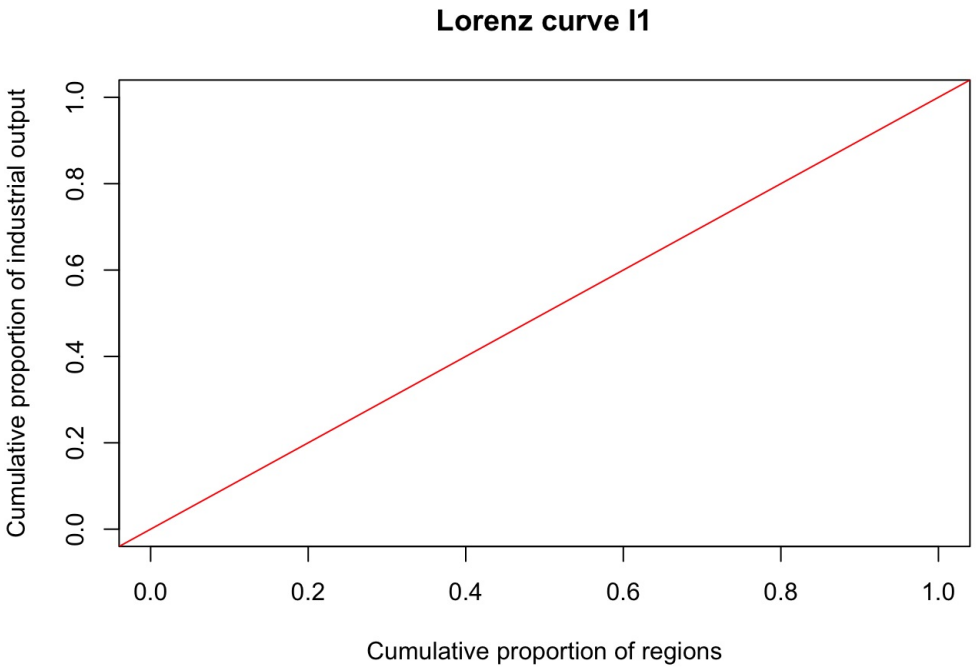
```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
## [1] 0.0 0.0 0.1 0.2 0.5 1.0
```

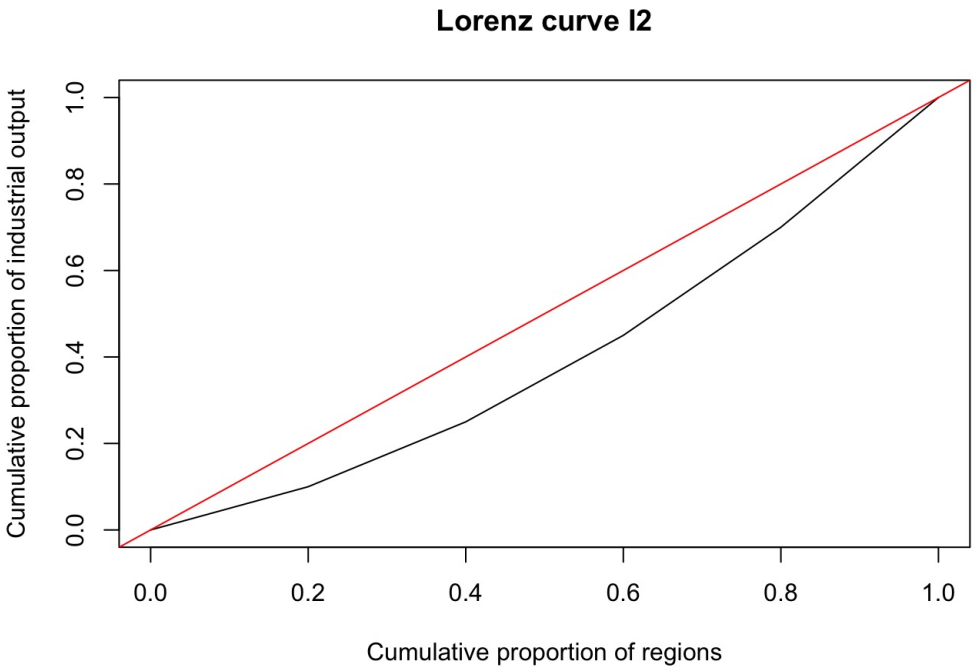
generate a region - industry matrix

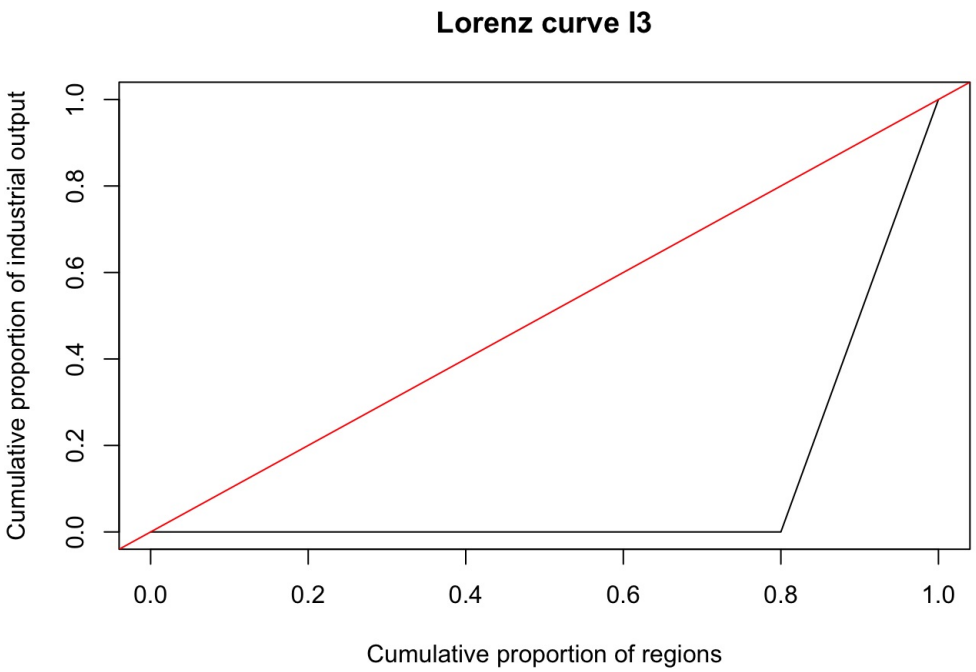
ocupamos la matriz mat que ya habíamos generado

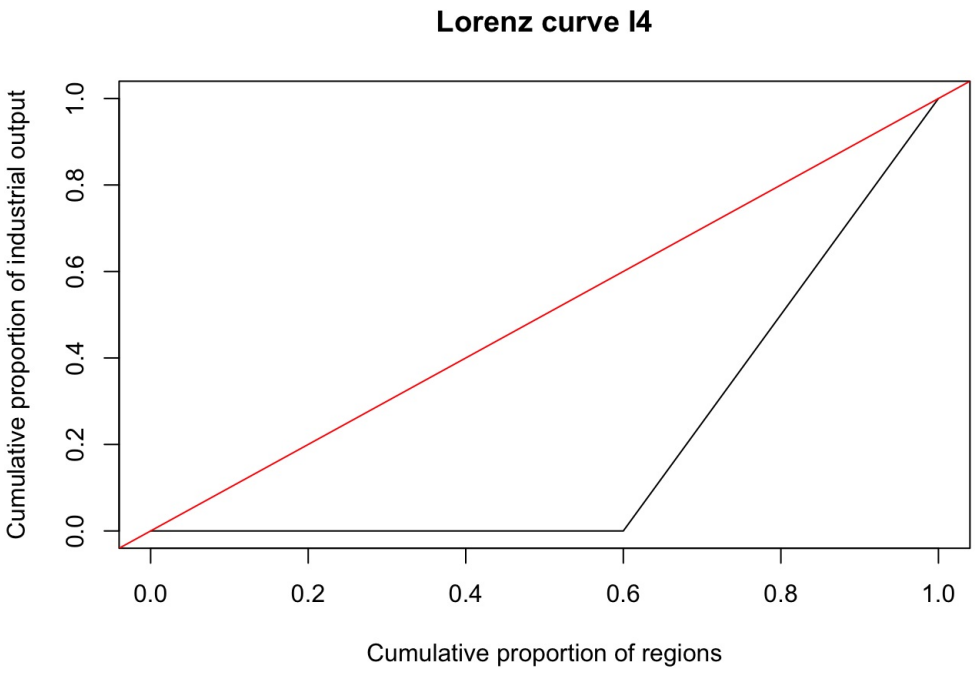
#mat = matrix (c (0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,0, 1, 0, 1, 0, 1, 1, 1), ncol = 4, byrow = T) rownames(mat) <- c (“R1”, “R2”, “R3”, “R4”, “R5”) colnames(mat) <- c (“I1”, “I2”, “I3”, “I4”) ## run the function ## corremos la función para la matriz

```
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


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


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



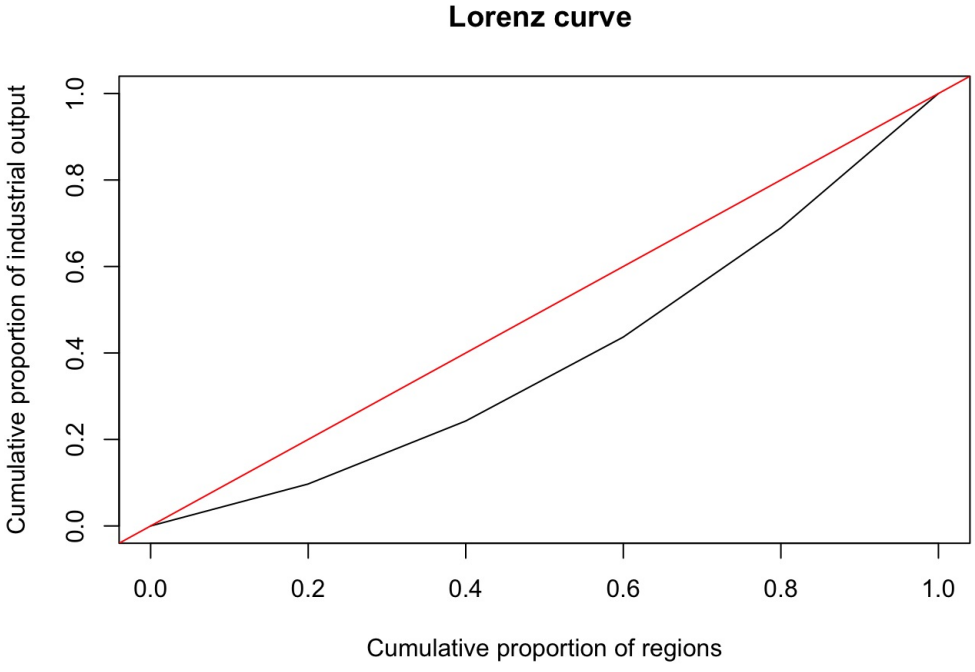

```
#Lorenz.curve (mat, pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (mat, plot = FALSE)
```

```
## $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
```

run the function by aggregating all industries

#corremos la función y le agragamos las industrias que esta en los renglones

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



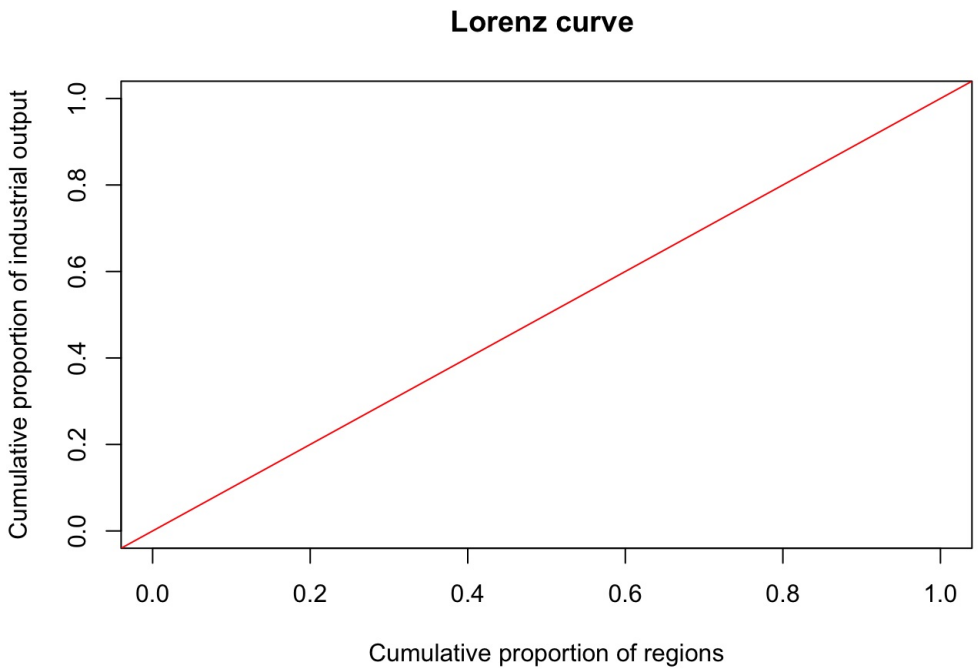

```
#Lorenz.curve (rowSums(mat), pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (rowSums(mat), plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##                    R1         R2         R3         R4         R5
## 0.00000000 0.09708738 0.24271845 0.43689320 0.68932039 1.00000000
```

run the function for industry #1 only (perfect equality)

#graficamos y hacemos los cálculos sólo para la industria 1

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



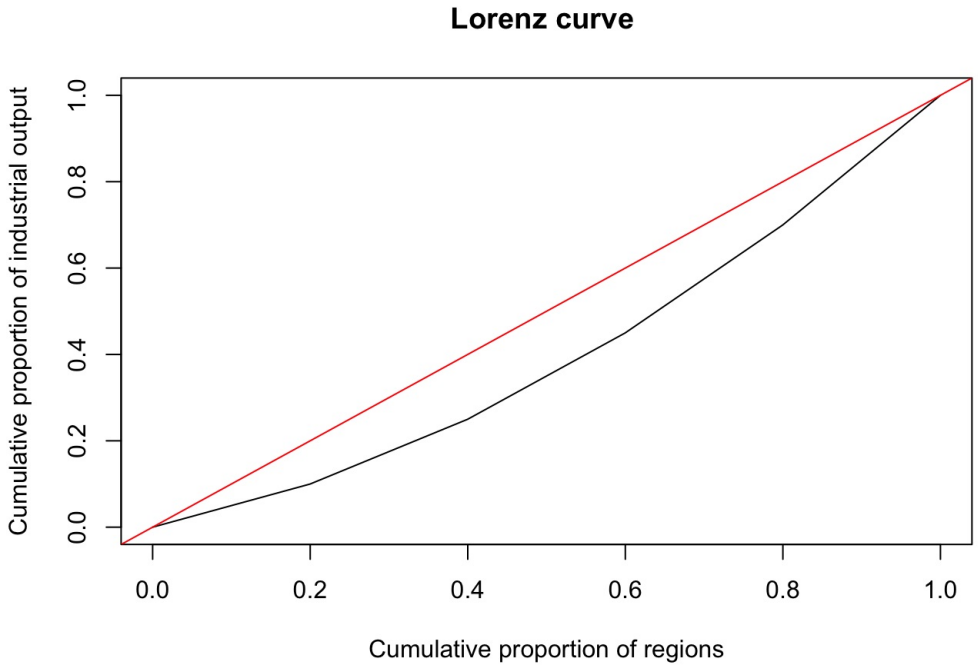

```
#Lorenz.curve (mat[,1], pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (mat[,1], plot = FALSE)
```

```
## $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
```

run the function for industry #2 only (perfect equality)

#graficamos y hacemos los cálculos sólo para la industria 2

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



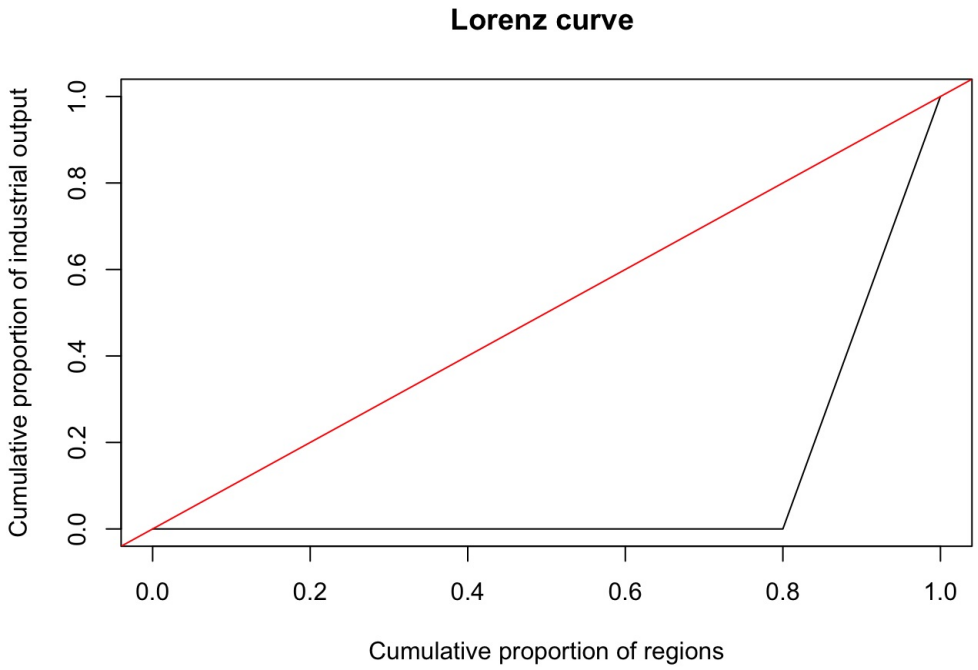

```
#Lorenz.curve (mat[,2], pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (mat[,2], plot = FALSE)
```

```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
##        R1   R2   R3   R4   R5
## 0.00 0.10 0.25 0.45 0.70 1.00
```

run the function for industry #3 only (perfect unequality)

#graficamos y hacemos los cálculos sólo para la industria 3

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



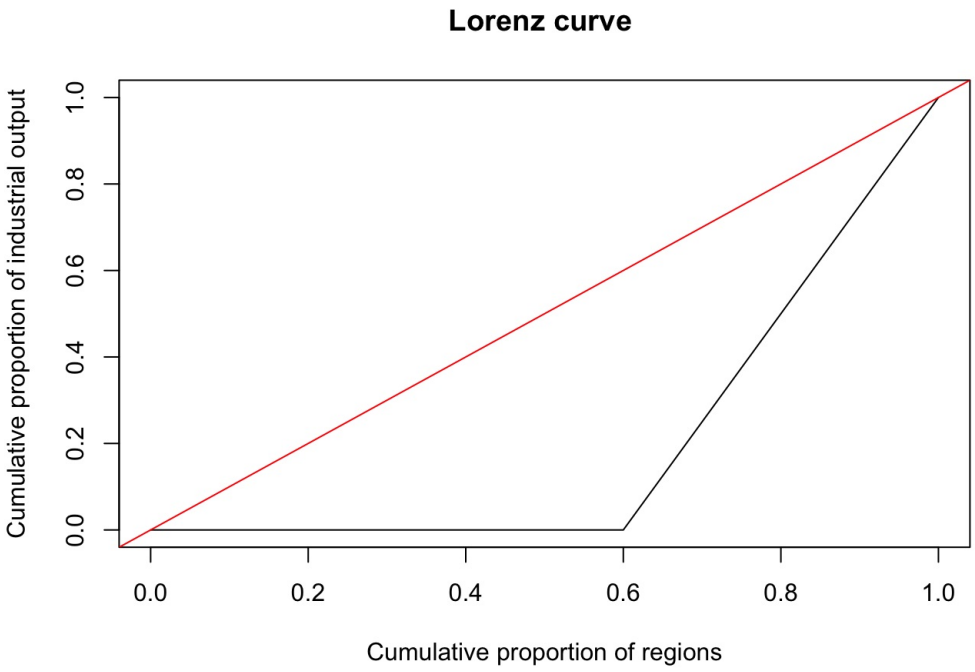

```
#Lorenz.curve (mat[,3], pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (mat[,3], plot = FALSE)
```

```
## $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
```

run the function for industry #4 only (top 40% produces 100% of the output)

#graficamos y hacemos los cálculos sólo para la industria 4

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

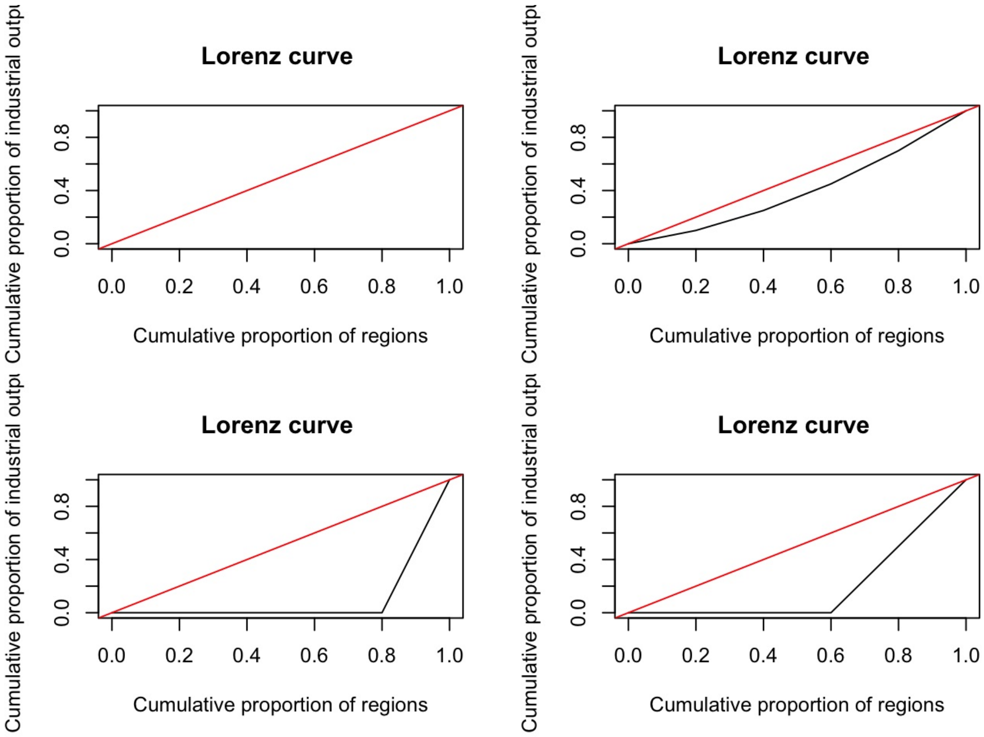



```
#Lorenz.curve (mat[,4], pdf = TRUE) sólo si queremos generar un pdf de la curva
Lorenz.curve (mat[,4], plot = FALSE)
```

```
## $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
```

#Compare the distribution of the #industries #comparamos todas la industrias en una sola cuadrícula

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


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