LAB 59 (MD)

Yunuén Hernández Álvarez

24/10/2021

ECONGEO

Instalar paqueterias

install.packages("devtools")

```
library(devtools)
```

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

devtools::install_github("PABalland/EconGeo", force = T)

```
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

This function plots a Hoover curve from regions - industries matrices following Hoover (1936).

generate vectors of industrial and population count

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

check the ind vector

```
ind
```

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

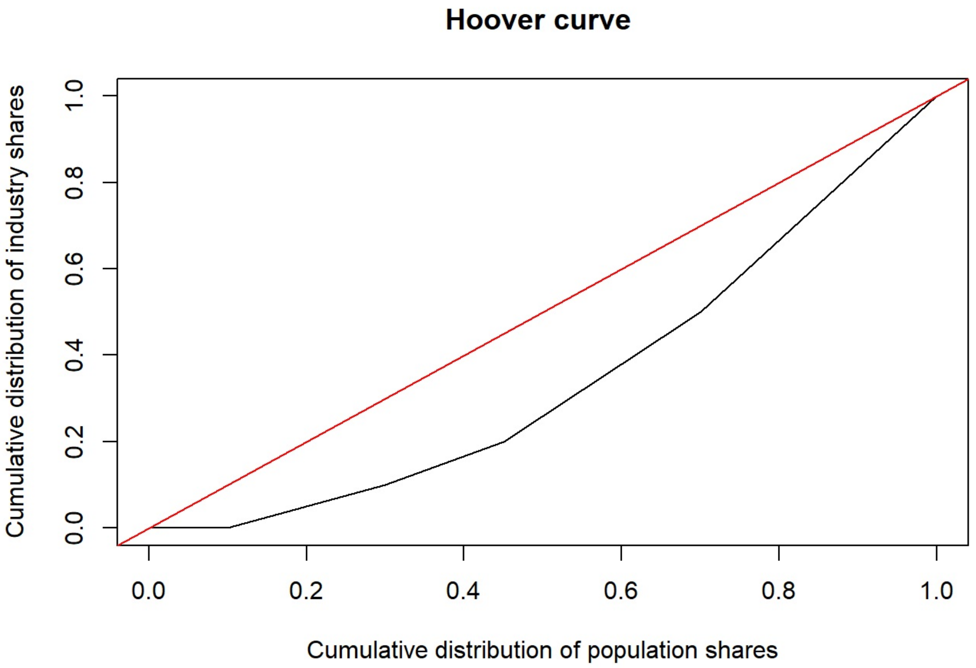
check the pop vector

```
pop
```

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

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

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



compute the corresponding Hoover Gini

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

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

COMPUTE THE GINI COEFFICIENT

This function computes the Gini coefficient. The Gini index measures spatial inequality. It ranges from 0 (perfect income equality) to 1 (perfect income inequality) and is derived from the Lorenz curve. The Gini coefficient is defined as a ratio of two surfaces derived from the Lorenz curve. The numerator is given by the area between the Lorenz curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle). This index gives an indication of the unequal distribution of an industry accross n regions. Maximum inequality in the sample occurs when n-1 regions have a score of zero and one region has a positive score. The maximum value of the Gini coefficient is (n-1)/n and approaches 1 (theoretical maximum limit) as the number of observations (regions) increases.

Examples

generate vectors of industrial count

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

run the function

```
Gini (ind)
```

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

generate a region - industry matrix

```
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

```
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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COMPUTE THE HOOVER GINI

This function computes the Hoover Gini, named after Hedgar Hoover. The Hoover index is a measure of spatial inequality. It ranges from 0 (perfect equality) to 1 (perfect inequality) and is calculated from the Hoover curve associated with a given distribution of population, industries or technologies and a reference category. In this sense, it is closely related to the Gini coefficient and the Hoover index. The numerator is given by the area between the Hoover curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle).

Examples

generate vectors of industrial and population count

```
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)

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

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

generate a region - industry matrix

```
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")
```

run the function

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

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

run the function by aggregating all industries

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

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

run the function for industry #1 only

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

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

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

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

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

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

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

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

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

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

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

COMPUTE THE LOCATIONAL GINI COEFFICIENT FROM REGIONS - INDUSTRIES MATRICES

This function computes the locational Gini coefficient as proposed by Krugman from regions - industries matrices. The higher the coefficient (theoretical limit = 0.5), the greater the industrial concentration. The locational Gini of an industry that is not localized at all (perfectly spread out) in proportion to overall employment would be 0.

Examples

generate a region - industry matrix

```
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")
```

run the function

```
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
```

PLOT A LOCATIONAL GINI CURVE FROM REGIONS - INDUSTRIES MATRICES

This function plots a locational Gini curve following Krugman from regions - industries matrices.

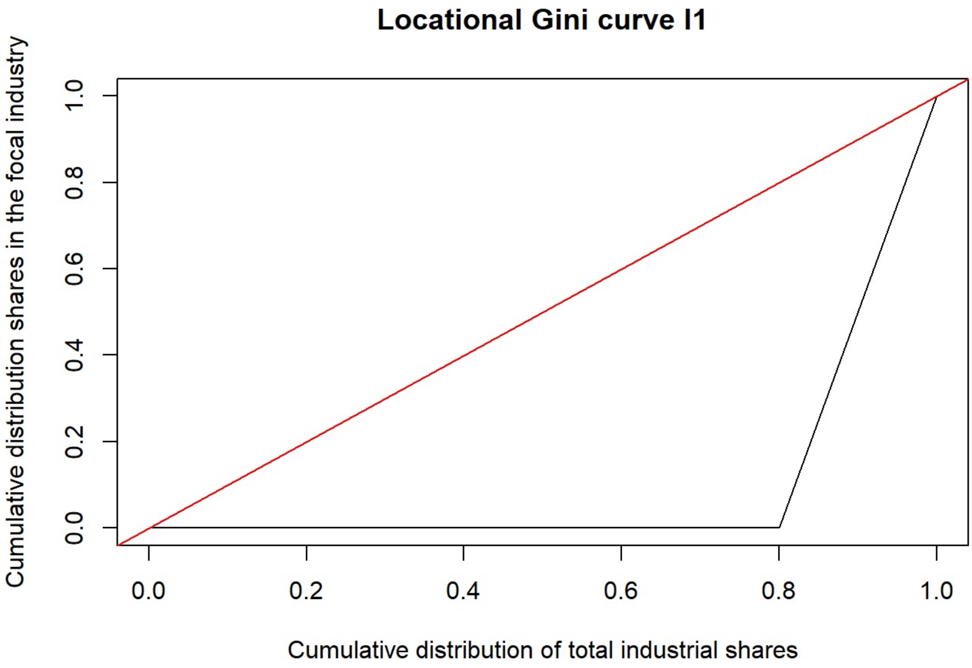
Examples

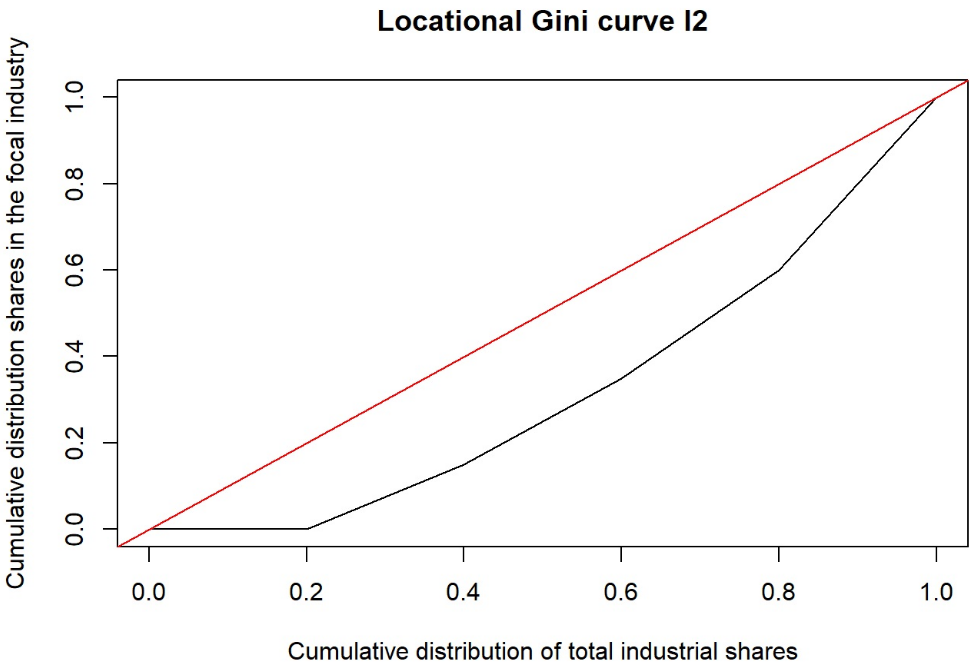
generate a region - industry matrix

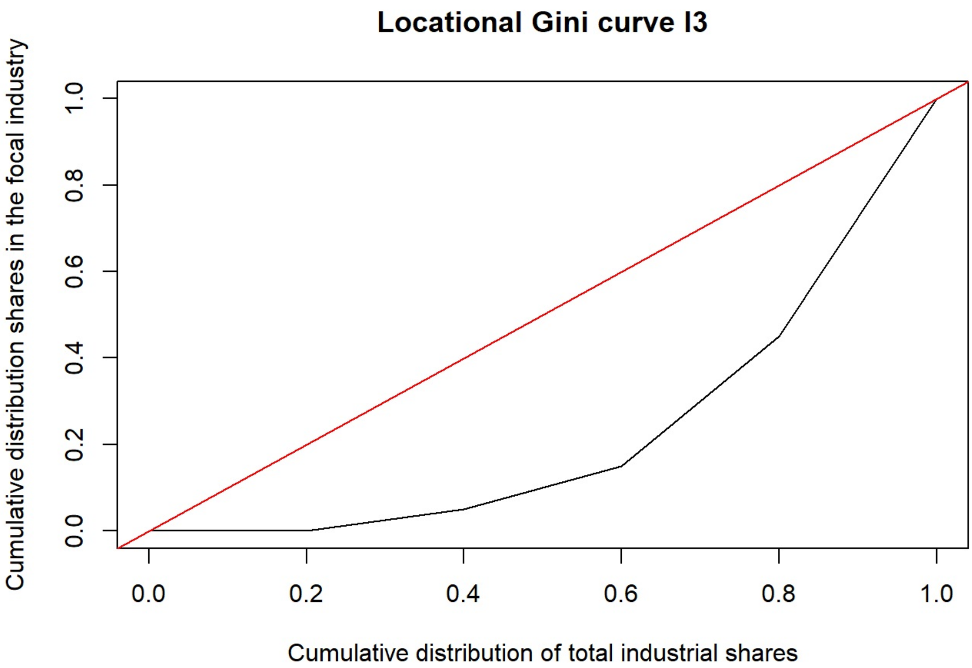
```
mat = matrix (
  c (100, 0, 0, 0, 0,
     0, 15, 5, 70, 10,
     0, 20, 10, 20, 50,
     0, 25, 30, 5, 40,
     0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5")
```

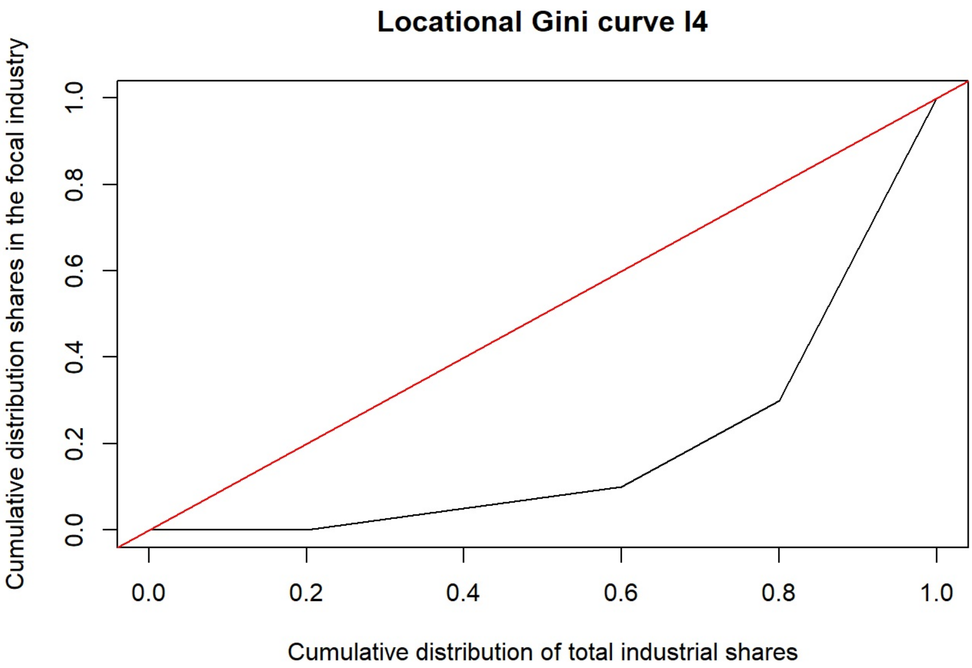
run the function (shows industry #5)

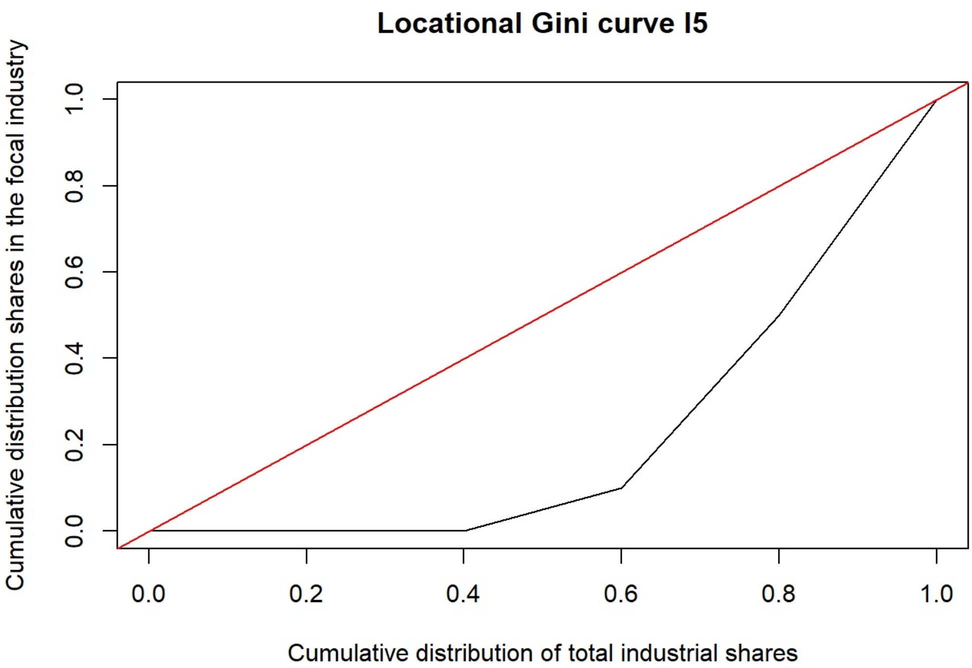
```
locational.Gini.curve (mat)
```


Locational Gini curve I1
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 I2
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 I3
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 (mat, pdf = TRUE)
```

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

PLOT A LORENZ CURVE FROM REGIONAL INDUSTRIAL COUNTS

This function plots a Lorenz curve from regional industrial counts. This curve gives an indication of the unequal distribution of an industry accross regions.

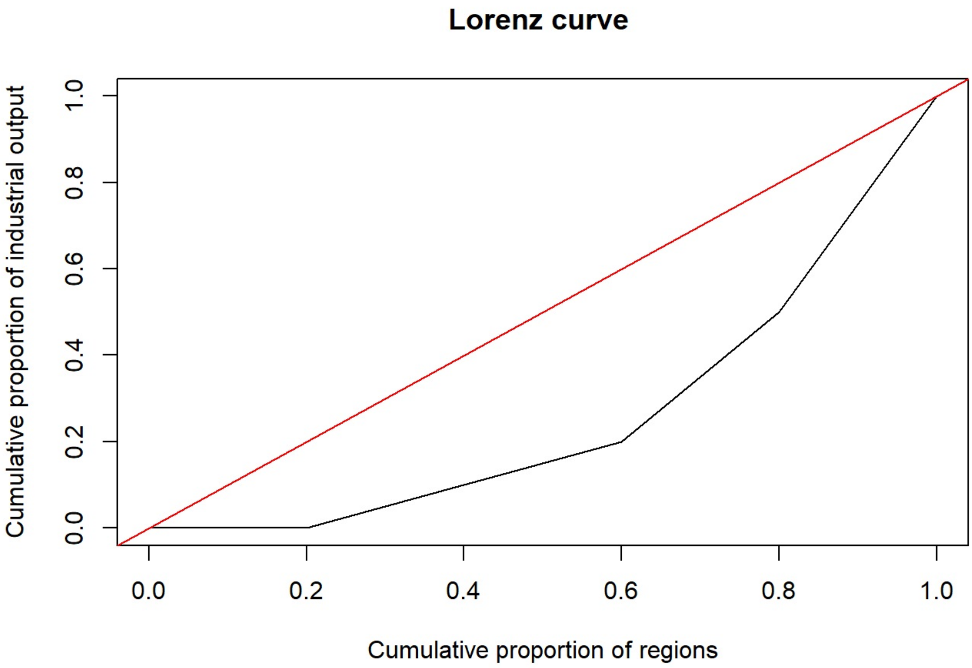
Examples

generate vectors of industrial count

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

run the function

```
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
```

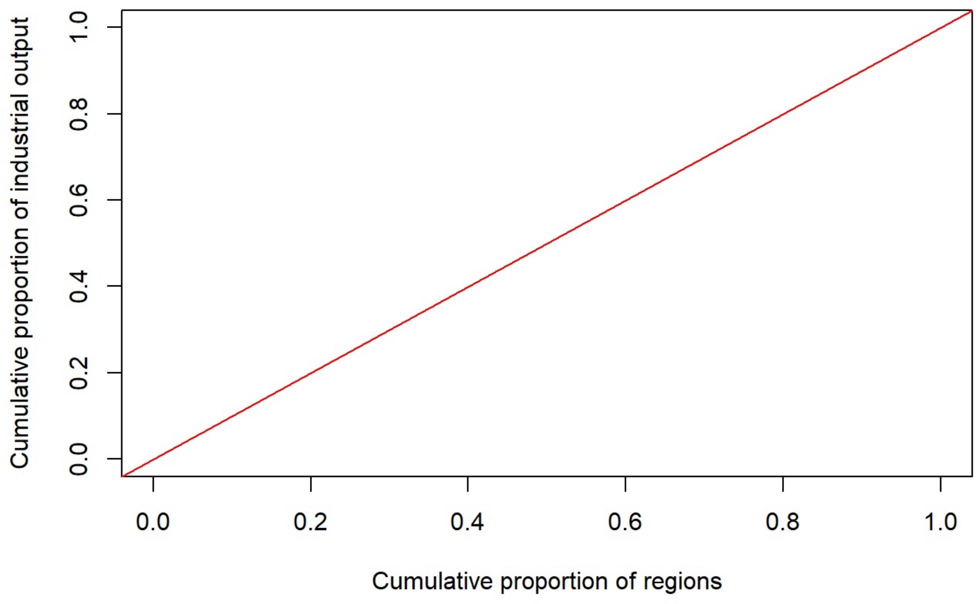
generate a region - industry matrix

```
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")
```

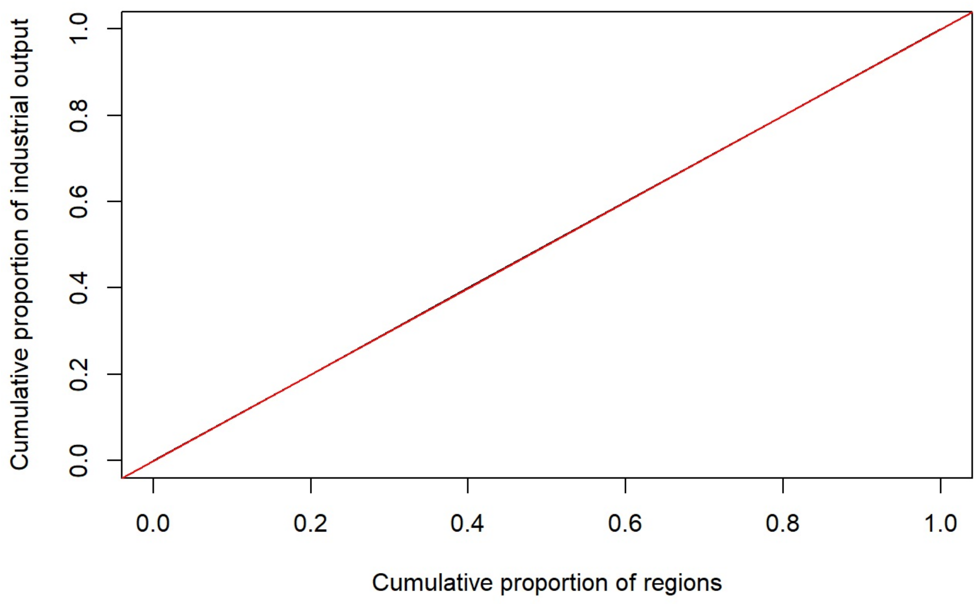
run the function

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

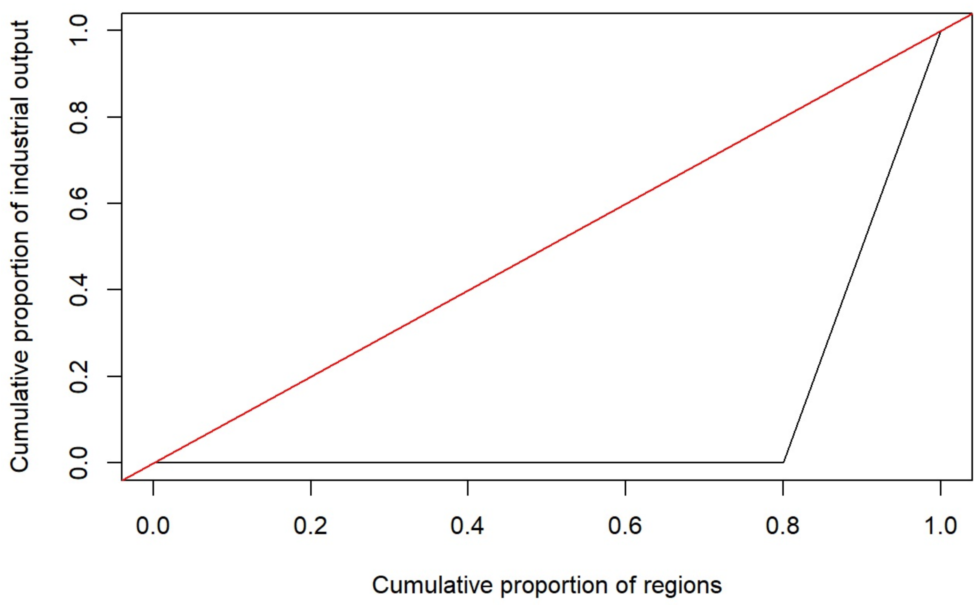
Lorenz curve l1

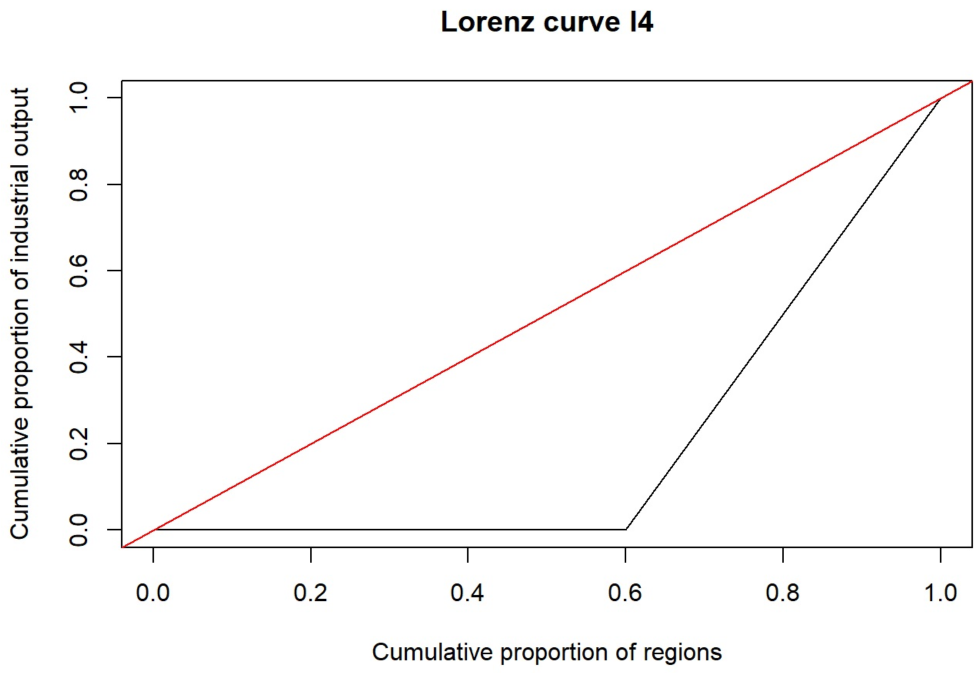


Lorenz curve l2



Lorenz curve l3






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

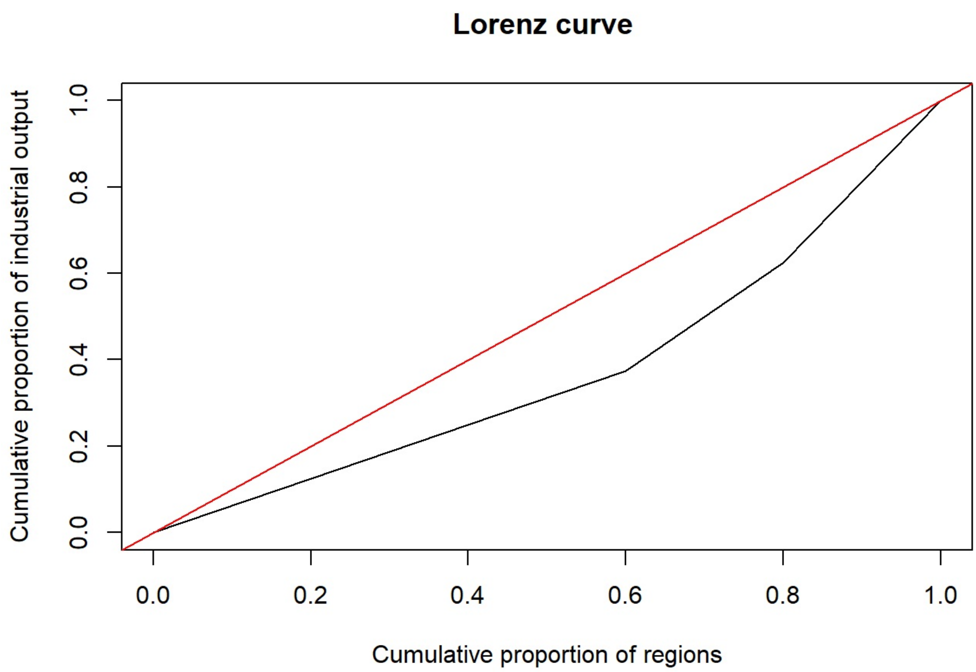
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

run the function by aggregating all industries

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




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

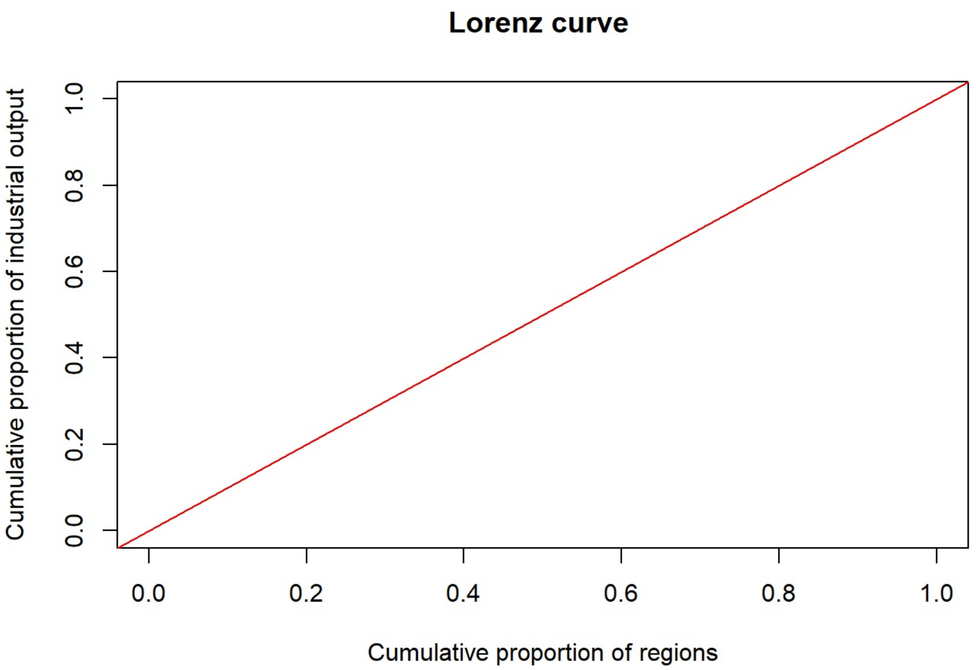
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

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

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




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

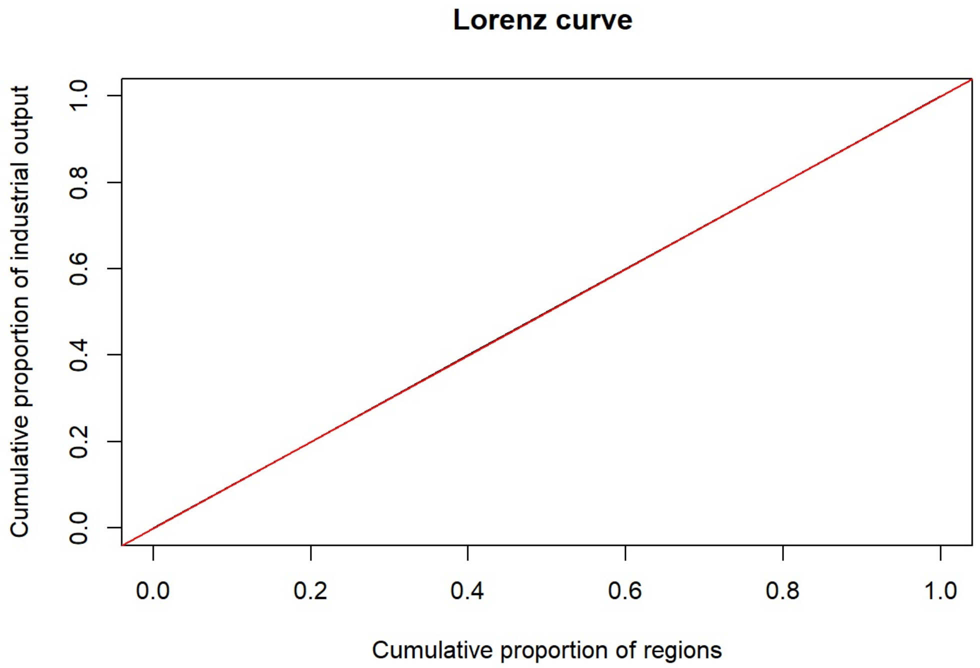
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

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

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




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

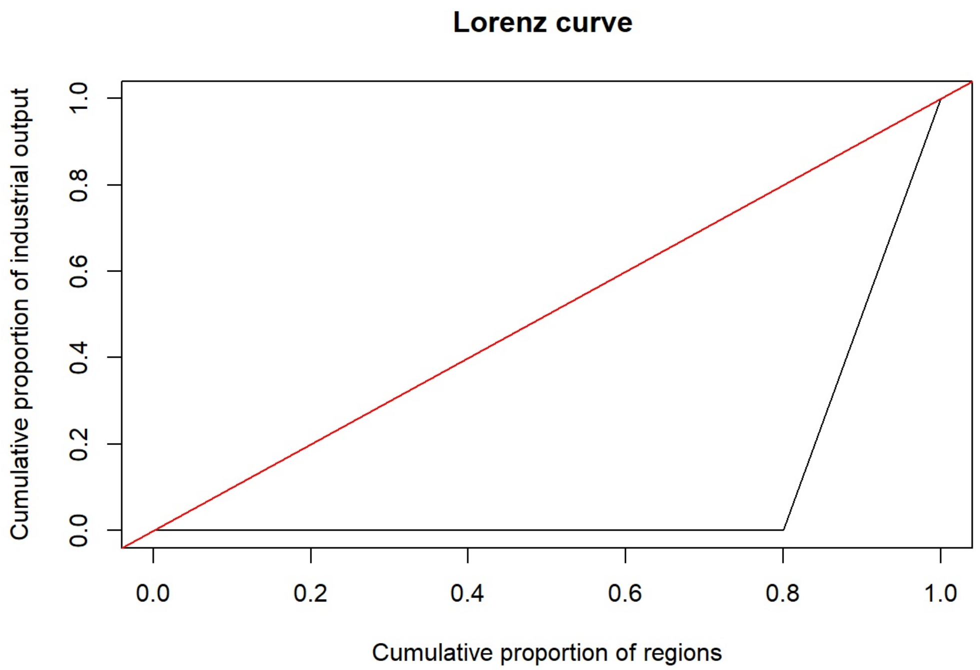
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

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

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




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

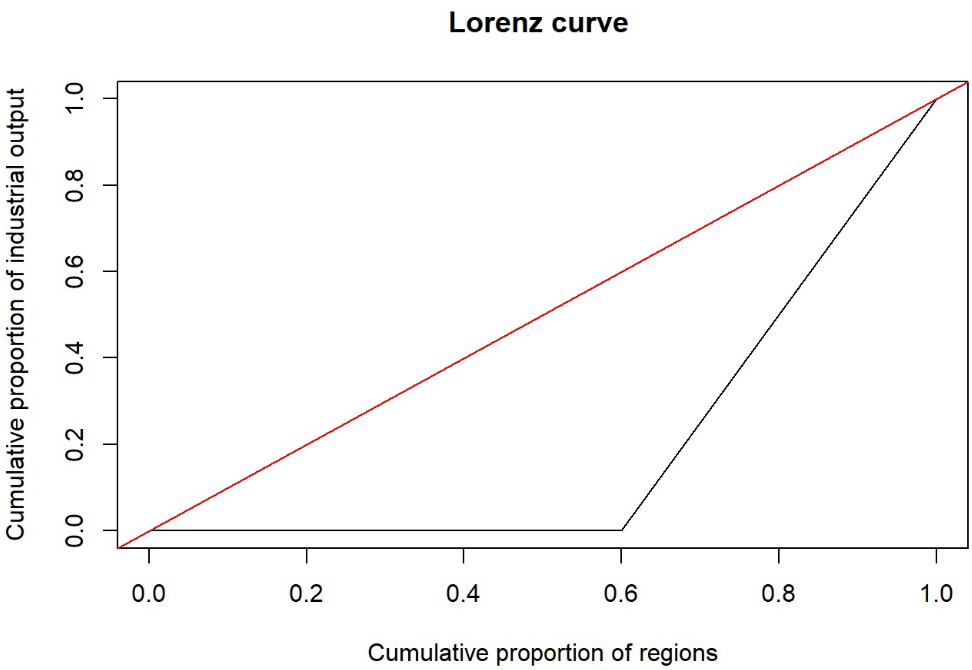
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

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

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




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

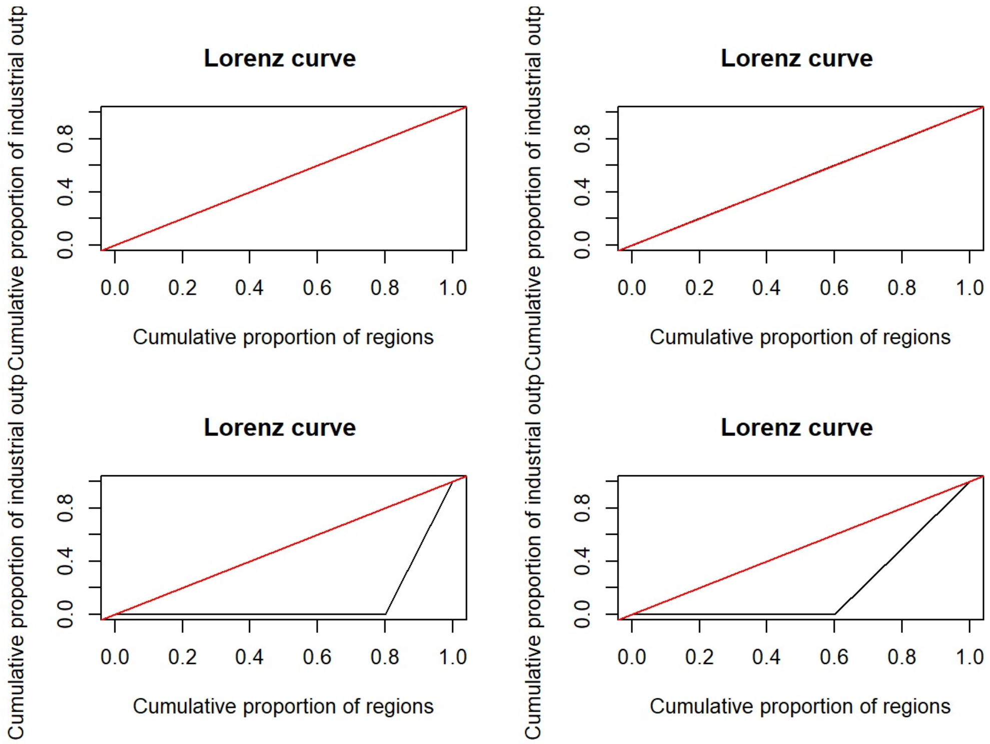
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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
```

Compare the distribution of the #industries

```
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