UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS ECONÓMICO ADMINISTRATIVAS

DOCTORADO EN CIENCIAS ECONÓMICO ADMINISTRATIVAS

MATERIA: TEMAS SELECTOS 1 “ESTADÍSTICA PARA LAS CIENCIAS ECONÓMICO ADMINISTRATIVAS”

TERCER SEMESTRE

CATEDRÁTICO: DRA. CARLA CAROLINA PÉREZ HERNÁNDEZ

ALUMNA: ALEJANDRA DE JESÚS GUZMÁN DIMAS

LAB 45

ECONGEO (CUADERNO R)

Lab 45 (Markdown)

Alejandra Guzmán

2024-04-04

EJEMPLO 4 PLOT A HOOVER CURVE

```
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

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

Check de ind vector

```
ind
```

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

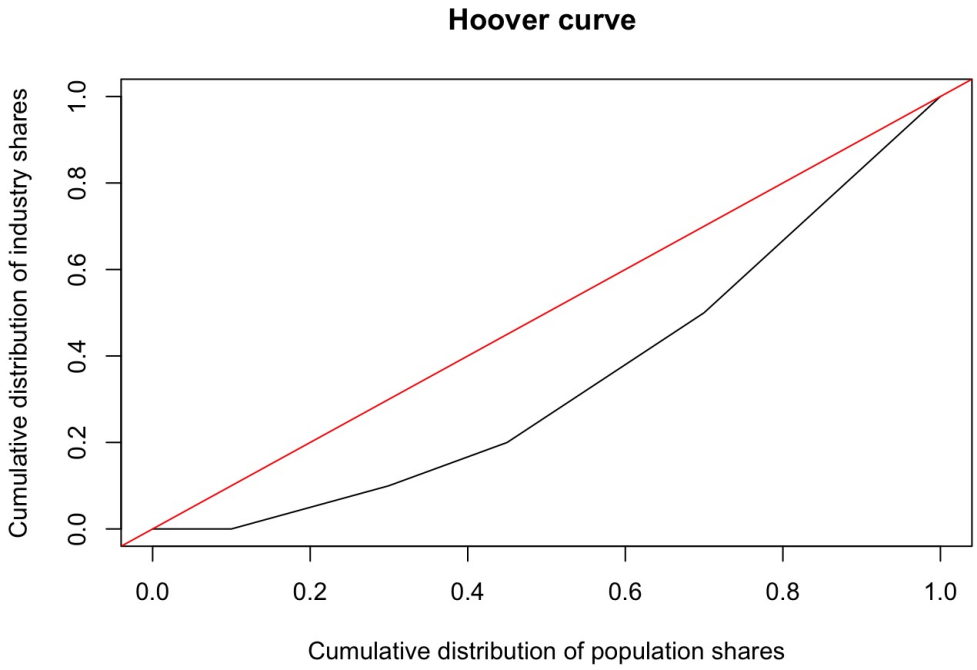
Check de pop vector

```
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
```

EJEMPLO GINI 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
```

EJEMPLO HOOVER.GINI 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
```

EJEMPLO LOCATIONAL.GINI 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)
```

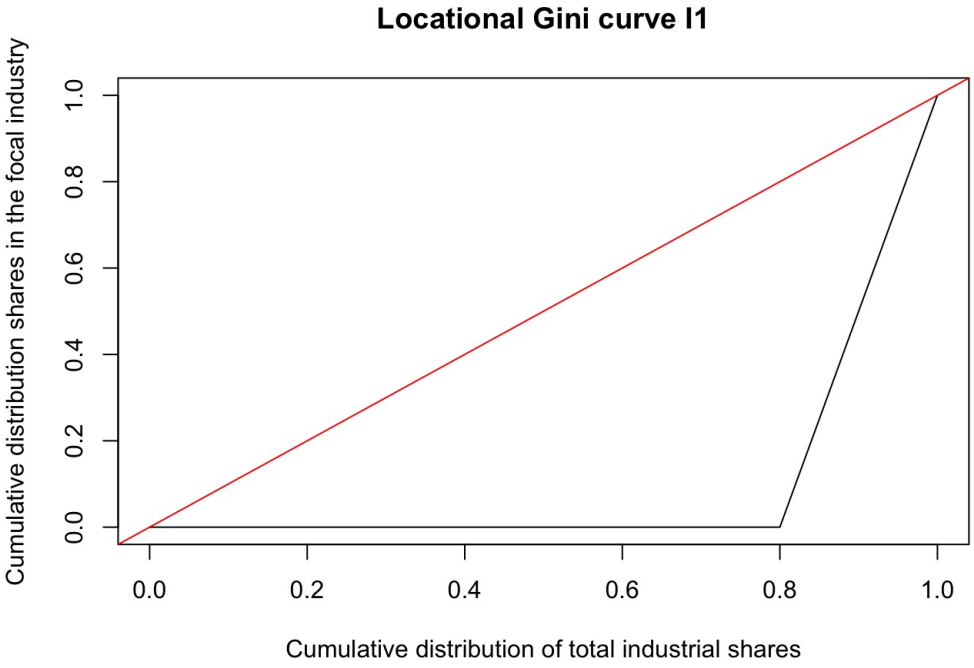
```
##   Industry Loc.Gini
## 1       I1     0.40
## 2       I2     0.18
## 3       I3     0.27
## 4       I4     0.31
## 5       I5     0.28
```

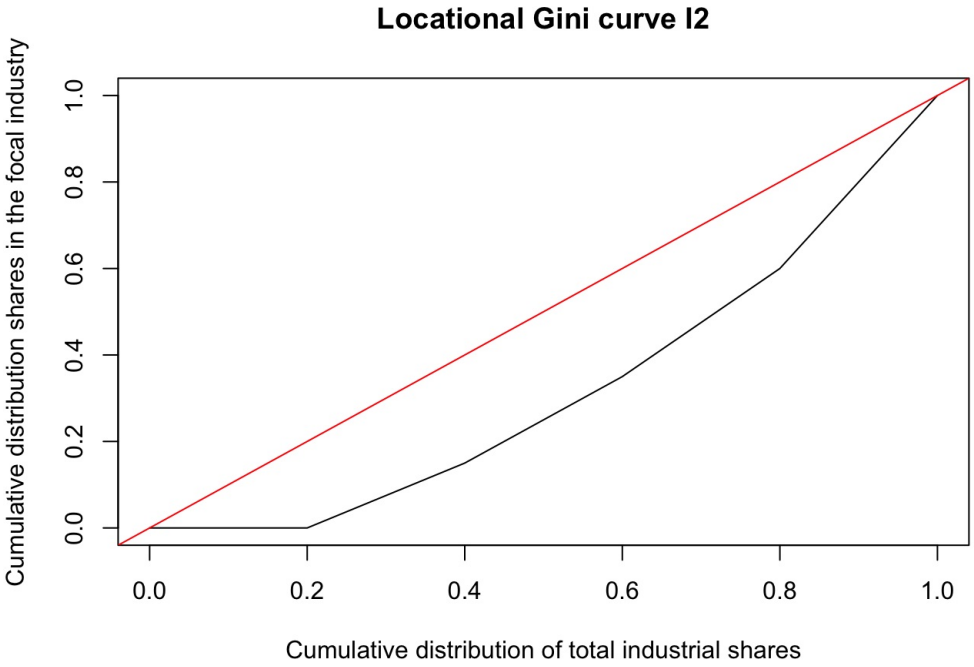
EJEMPLO LOCATIONAL.GINI.CURVE Generate a region -industry matrix

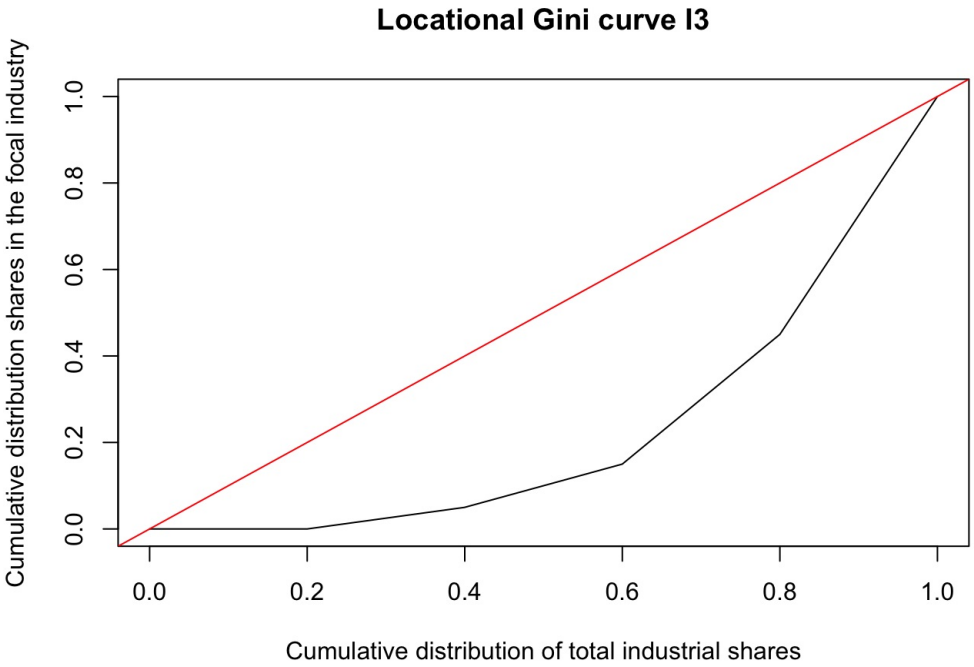
```
mat=matrix(
  c(100,0,0,0,0,
    0,15,5,70,10,
    0,20,10,20,50,
    0,25,30,5,40,
    0,40,55,5,0), ncol=5, byrow = T)
rownames(mat)= c("R1", "R2", "R3", "R4", "R5")
colnames(mat)=c("I1", "I2", "I3", "I4", "I5")
```

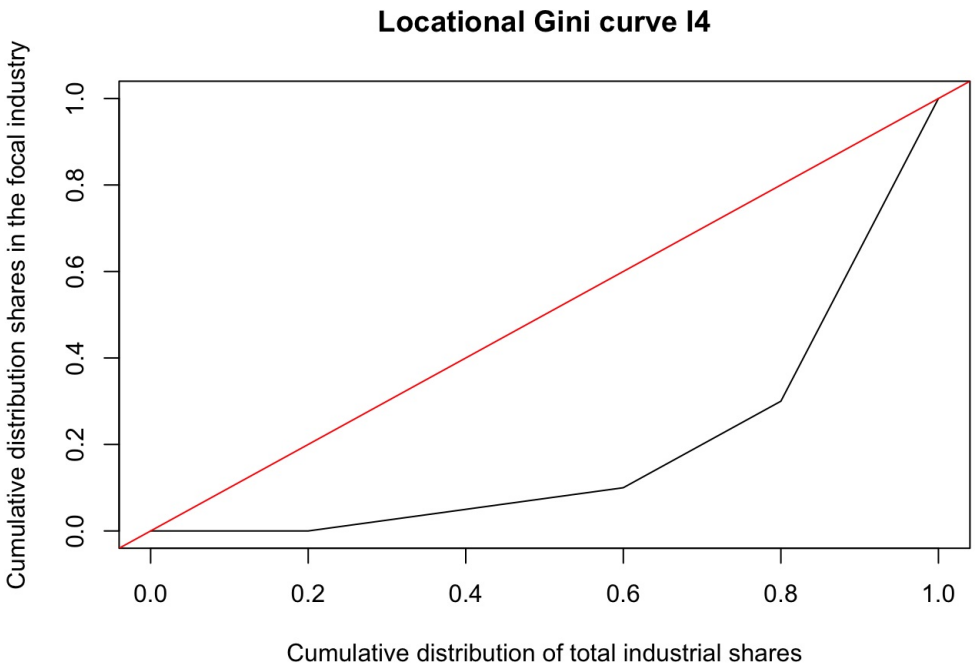
Run the function (shows industry 5)

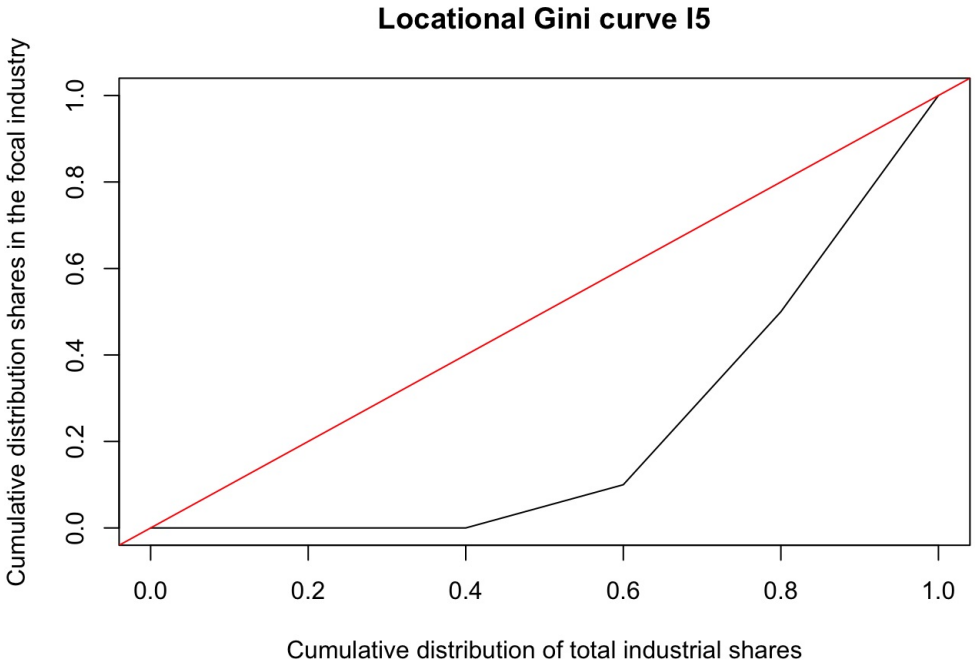
```
locational.Gini.curve(mat)
```


Locational Gini curve 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(mat, pdf=TRUE)
```

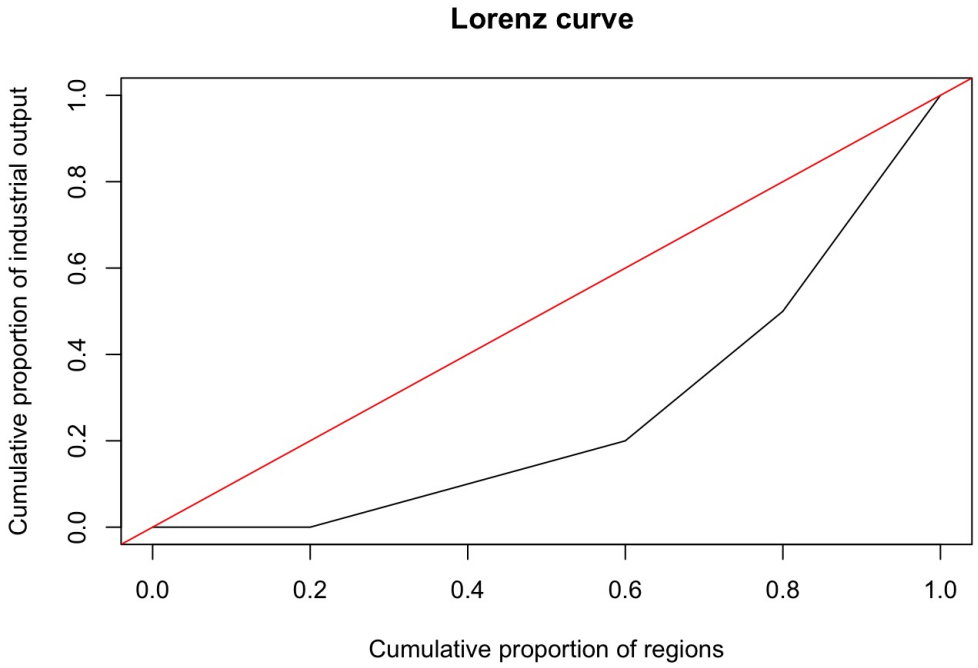
```
## [1] "locational.Gini.curve.pdf has been saved to your current working directory"
```

EJEMPLO LORENZ.CURVE Generate vector 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)
```

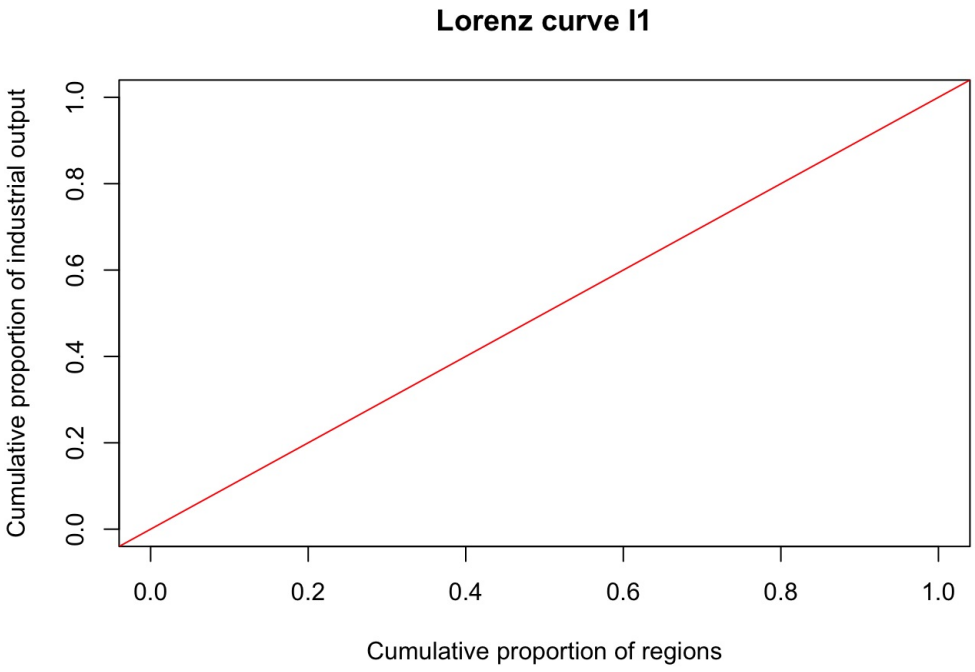
```
## $cum.reg
## [1] 0.0 0.2 0.4 0.6 0.8 1.0
##
## $cum.out
## [1] 0.0 0.0 0.1 0.2 0.5 1.0
```

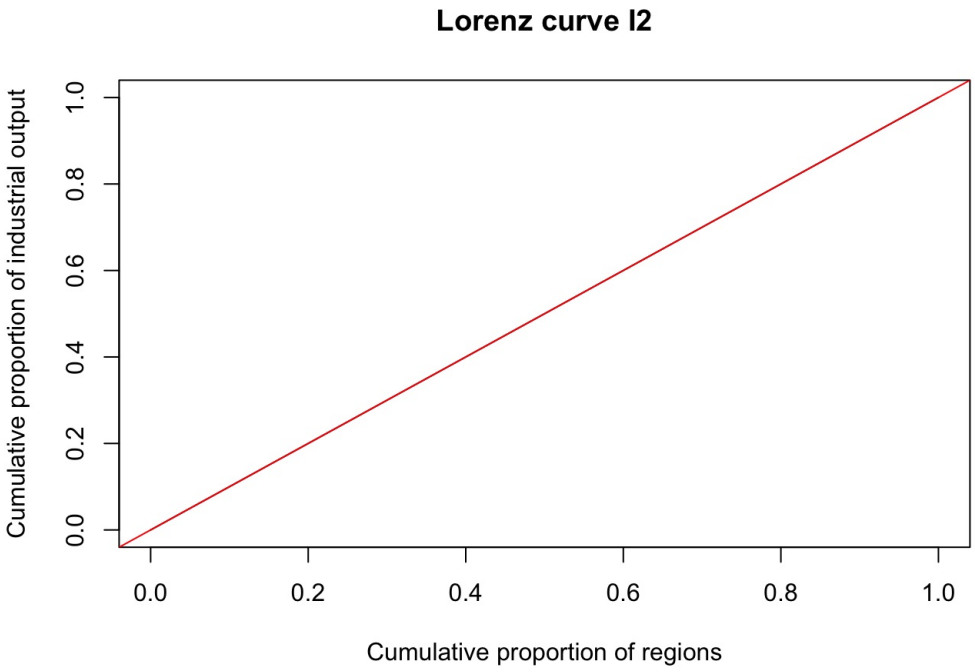
Generate a region - industry matrix

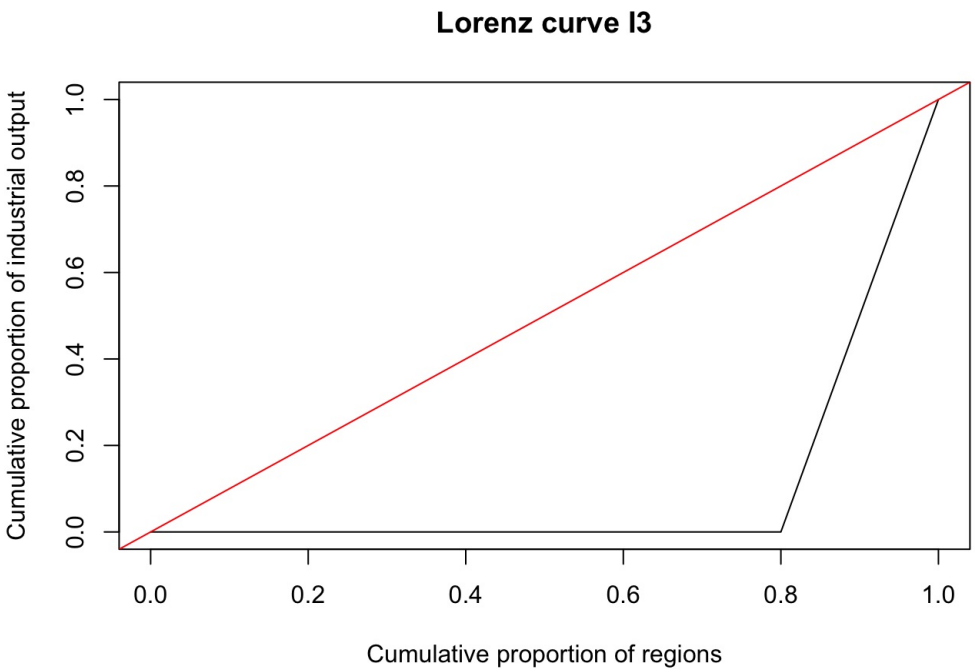
```
mat=matrix(
  c(0,1,0,0,
    0,1,0,0,
    0,1,0,0,
    0,1,0,1,
    0,1,1,1), ncol=4, byrow = T)
rownames(mat)= c("R1", "R2", "R3", "R4", "R5")
colnames(mat)= c("I1", "I2", "I3", "I4")
```

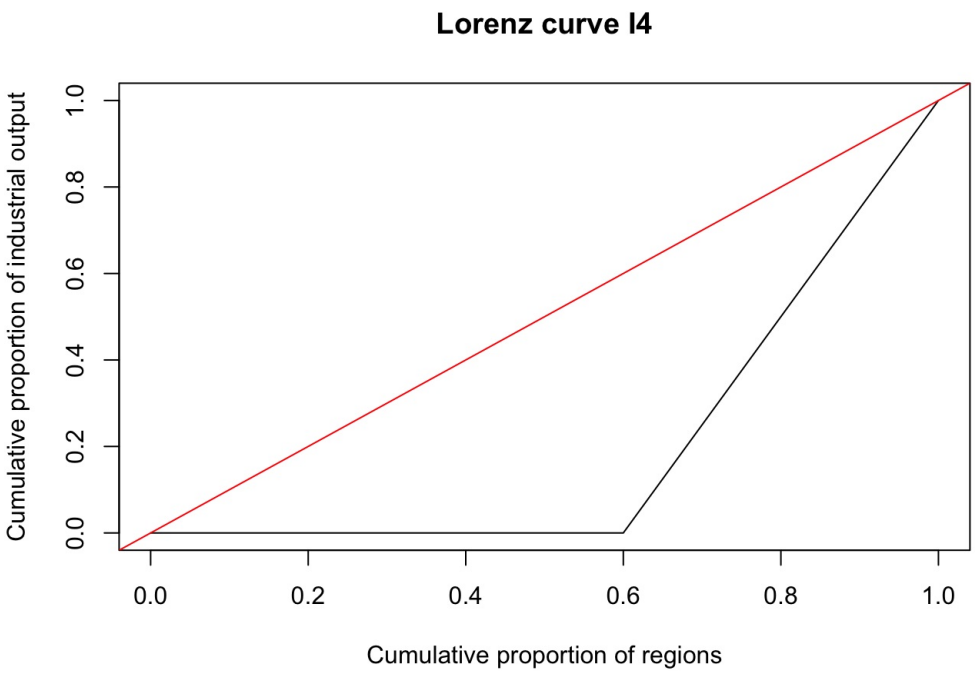
Run the function

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


Lorenz curve l1
Cumulative proportion of industrial output
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative proportion of regions


Lorenz curve l2
Cumulative proportion of industrial output
0.0
0.2
0.4
0.6
0.8
1.0
Cumulative proportion of regions


Lorenz curve l3
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)
```

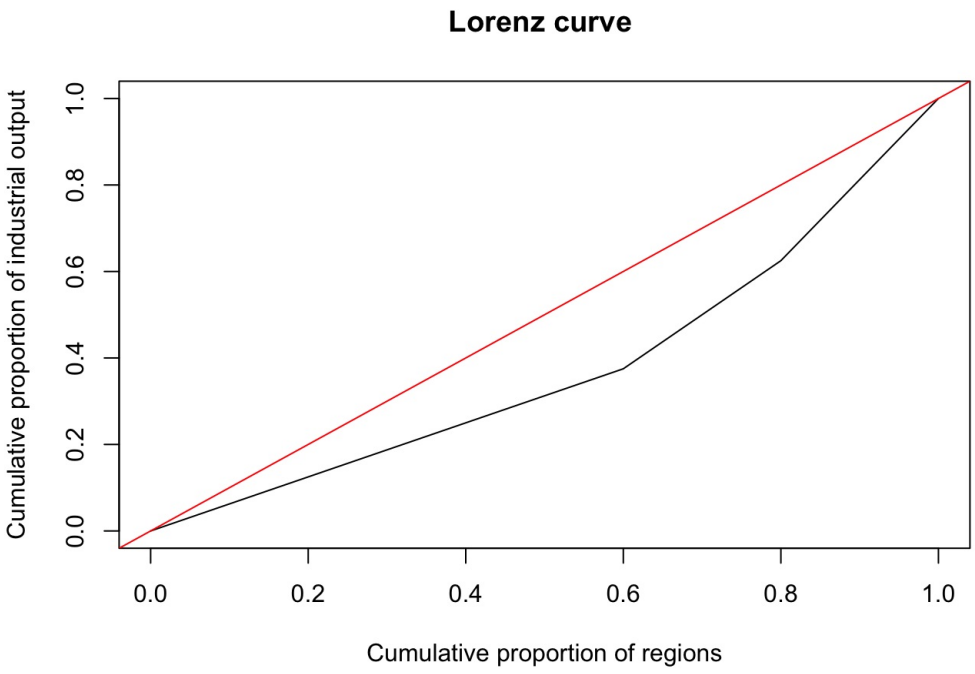
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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)
```

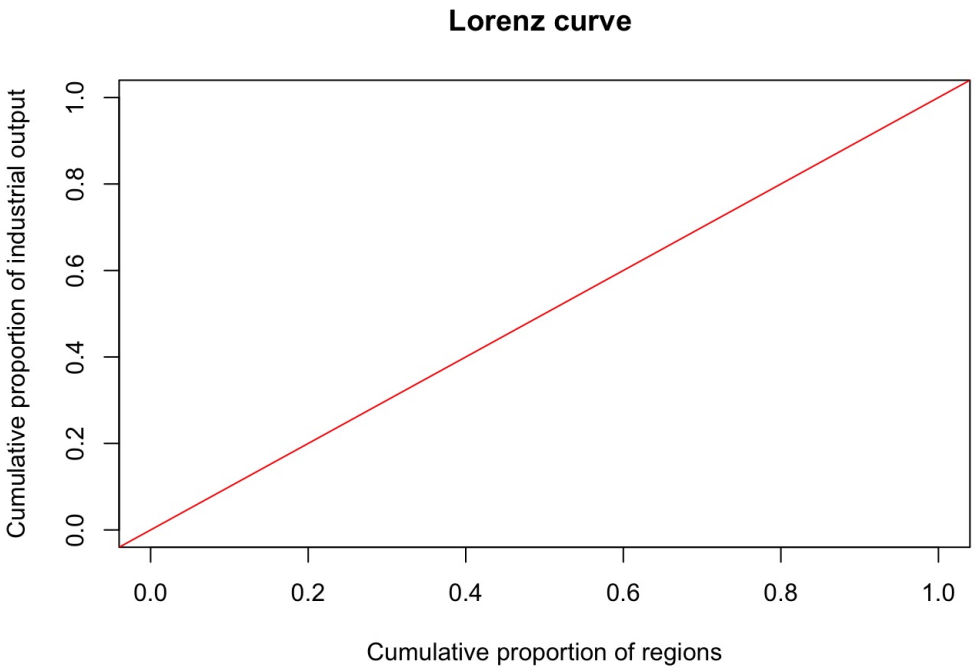
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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)
```

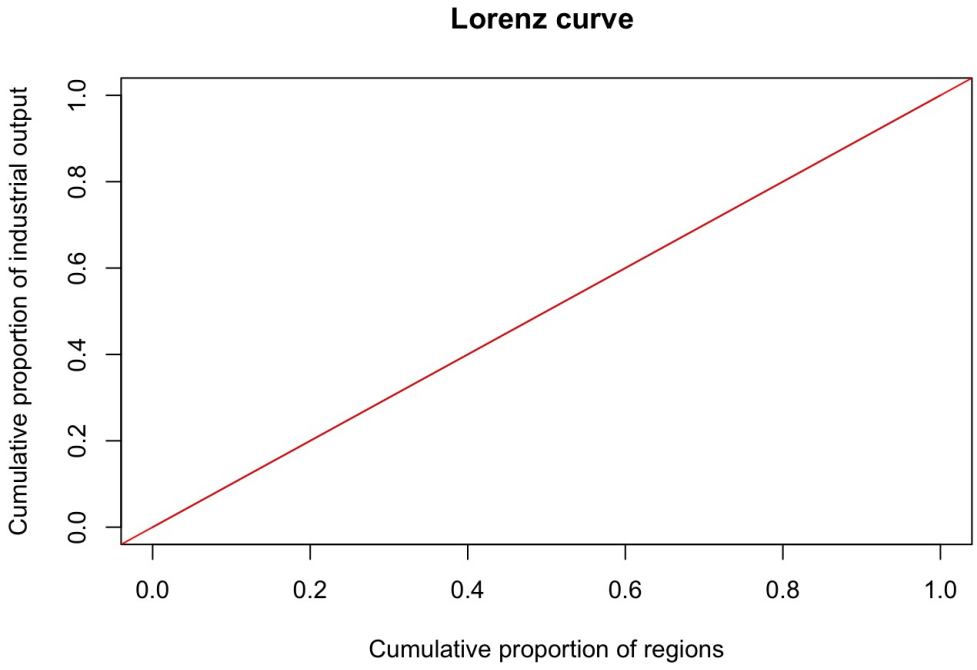
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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)
```

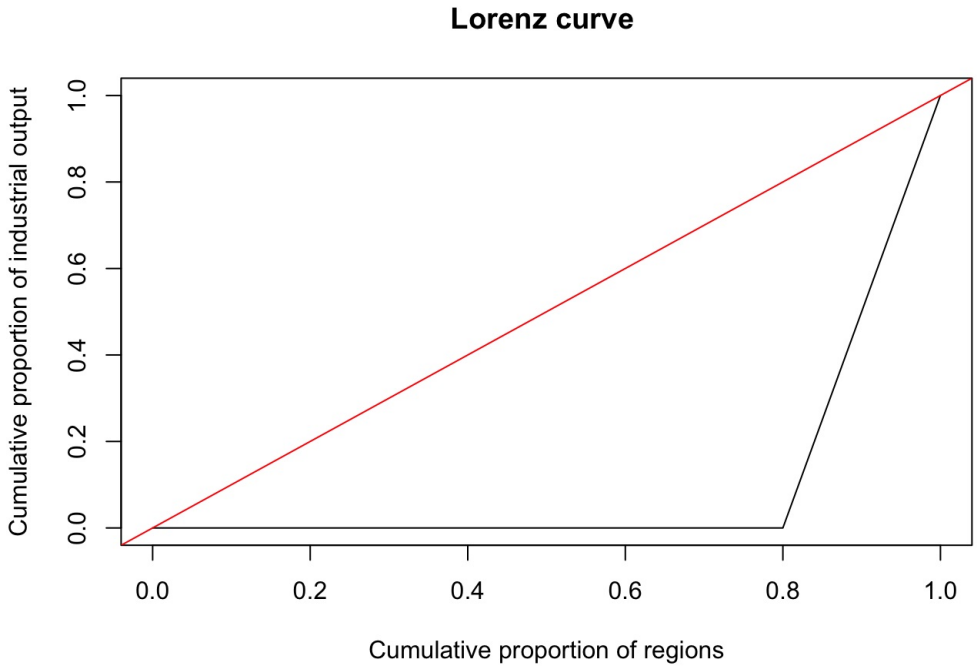
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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)
```

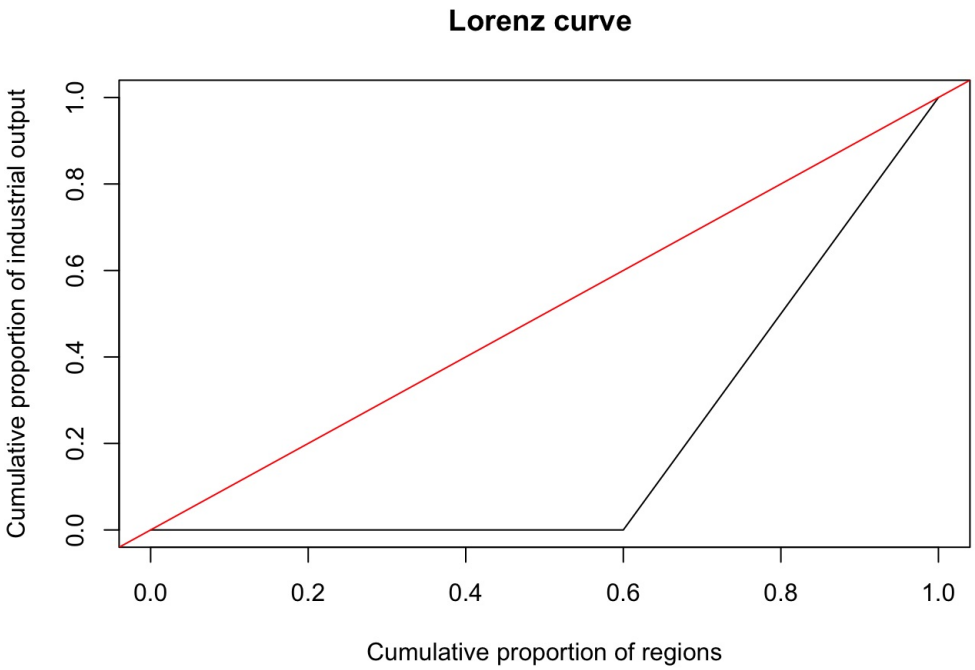
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

```
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)
```

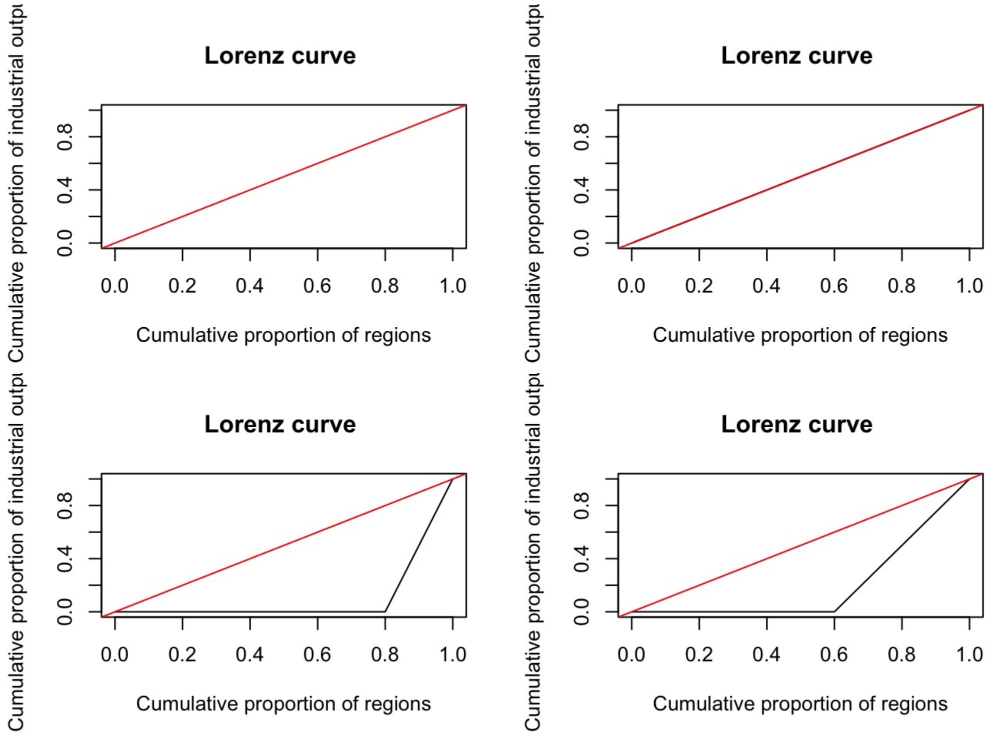
```
## [1] "Lorenz.curve.pdf has been saved to your current working directory"
```

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