UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS ECONÓMICO ADMINISTRATIVAS

DOCTORADO EN CIENCIAS ECONÓMICO ADMINISTRATIVAS

TEMAS SELECTOS III: COMPLEJIDAD ECONÓMICA

ACTIVIDAD: LAB 24

Cuadernos de Rstudio Markdown

Docente: Dra. Carla Carolina Pérez Hernández

Estudiante: Luis Armando González Arellano

lab 1 MD

2025-10-01

Es la estructura de red que AMAZON ocupa para hacer recomendaciones en base a tus compras actuales

```
M = as.matrix(
  read.csv("https://raw.githubusercontent.com/PABalland/ON/master/amz.csv" ,
           sep = ",",
           header = T,
           row.names = 1))
M
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         1     1     1
## Ron      1    1         0     0     1
## Andrea   1    1         0     0     1
## David    0    0         1     1     1
## Cesar    1    1         0     0     1
## Paula    1    1         1     1     1
```

```
dim(M)
```

```
## [1] 6 5
```

Ver sólo el vector de clientes que compran el producto “corbata” (producto de la columna 1)

```
M[1:6,1]
```

```
## Pierre    Ron Andrea  David  Cesar  Paula
##      0      1      1      0      1      1
```

Ver sólo un fragmento de la base original, en relación a los dos productos que me interesan

```
M[1:6,1:2]
```

```
##        Tie Book
## Pierre   0    0
## Ron      1    1
## Andrea   1    1
## David    0    0
## Cesar    1    1
## Paula    1    1
```

Computar grado de centralidad

```
rowSums(M)
```

```
## Pierre    Ron Andrea  David  Cesar  Paula
##      3      3      3      3      3      5
```

Suma de matrices

```
t(M)
```

```
##           Pierre Ron Andrea David Cesar Paula
## Tie            0   1      1     0     1     1
## Book           0   1      1     0     1     1
## Surfboard      1   0      0     1     0     1
## Short          1   0      0     1     0     1
## Water          1   1      1     1     1     1
```

```
M+M
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         2     2     2
## Ron      2    2         0     0     2
## Andrea   2    2         0     0     2
## David    0    0         2     2     2
## Cesar    2    2         0     0     2
## Paula    2    2         2     2     2
```

Crear nuevo un objeto llamado x

```
X=M+M
```

multiplicación de matrices

```
M^2
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         1     1     1
## Ron      1    1         0     0     1
## Andrea   1    1         0     0     1
## David    0    0         1     1     1
## Cesar    1    1         0     0     1
## Paula    1    1         1     1     1
```

```
M*M
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         1     1     1
## Ron      1    1         0     0     1
## Andrea   1    1         0     0     1
## David    0    0         1     1     1
## Cesar    1    1         0     0     1
## Paula    1    1         1     1     1
```

```
M%*%t(M)
```

```
##        Pierre Ron Andrea David Cesar Paula
## Pierre      3   1      1     3     1     3
## Ron         1   3      3     1     3     3
## Andrea      1   3      3     1     3     3
## David       3   1      1     3     1     3
## Cesar       1   3      3     1     3     3
## Paula       3   3      3     3     3     5
```

identificar productos que van frecuentemente juntos (co ocurrencias par)

```
t(M)%*%M
```

```
##           Tie Book Surfboard Short Water
## Tie         4    4         1     1     4
## Book        4    4         1     1     4
## Surfboard   1    1         3     3     3
## Short       1    1         3     3     3
## Water       4    4         3     3     6
```

similaridad de productos (de gustos) matriz de co ocurrencia de productos

```
P=t(M)%*%M
diag(P)=0
P
```

```
##           Tie Book Surfboard Short Water
## Tie         0    4         1     1     4
## Book        4    0         1     1     4
## Surfboard   1    1         0     3     3
## Short       1    1         3     0     3
## Water       4    4         3     3     0
```

similaridad de clientes - matriz de co ocurrencia de clienes

```
C=M%*%t(M)
diag(C)=0
C
```

```
##        Pierre Ron Andrea David Cesar Paula
## Pierre      0   1      1     3     1     3
## Ron         1   0      3     1     3     3
## Andrea      1   3      0     1     3     3
## David       3   1      1     0     1     3
## Cesar       1   3      3     1     0     3
## Paula       3   3      3     3     3     0
```

lab5_24md

2025-10-05

Cargar datos

```
knitr::opts_chunk$set(echo = TRUE)

library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(tidyr)
```

```
tabla1 <- read_csv("ice 2024.csv")
```

```
## Rows: 83 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): MUNICIPIO
## dbl (14): Ks,0, Ks,2, Ks,4, Ks,6, Ks,8, Ks,10, Ks,12, Ks,14, Ks,16, Ks,18, K...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(tabla1)
```

```
## # A tibble: 6 × 15
##   MUNICIPIO   `Ks,0` `Ks,2` `Ks,4` `Ks,6` `Ks,8` `Ks,10` `Ks,12` `Ks,14` `Ks,16`
##   <chr>        <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Tepeji Del…      3      3      2      1      1       1       1       1       1
## 2 Atitalaquia      5      5      5      4      3       2       2       2       2
## 3 Tepeapulco       7      8     11     11      8       7       6       6       6
## 4 Tizayuca         2      1      1      2      2       3       3       3       3
## 5 Mineral De…      4      4      4      5      4       4       4       4       4
## 6 Pachuca          1      2      3      3      5       5       5       5       5
## # ℹ 5 more variables: `Ks,18` <dbl>, `Ks,20` <dbl>, `Ks,22` <dbl>,
## #   `Ks,24` <dbl>, `Ks,26` <dbl>
```

```
names(tabla1)
```

```
##  [1] "MUNICIPIO" "Ks,0"      "Ks,2"      "Ks,4"      "Ks,6"      "Ks,8"
##  [7] "Ks,10"     "Ks,12"     "Ks,14"     "Ks,16"     "Ks,18"     "Ks,20"
## [13] "Ks,22"     "Ks,24"     "Ks,26"
```

1. Pivotar tabla “A LO LARGO”

```
t1_PIVOTANTE = tabla1 %>%
  pivot_longer(cols = c("Ks,0"   ,   "Ks,2"   ,   "Ks,4"   ,   "Ks,6"   ,   "Ks,8"    ,  "Ks,10"  ,   "Ks,12"  ,
"Ks,14"  ,   "Ks,16"  ,   "Ks,18"   ,  "Ks,20"  ,   "Ks,22" , "Ks,24"  ,   "Ks,26" ), names_to = "iteracion", val
ues_to = "ranking")
```

Exportar resultado: tabla ordenada

```
write.csv(t1_PIVOTANTE, file = "ice_CHARTICULATOR.csv")
```

lab_7_24

2025-10-05

Alumno: Luis Armando González Arellano

Laboratorio - Gestión de datos de red - curaduría de datos

Objetivo: Explorar la organización de los datos

En este ejercicio vamos a: 1. Cargar nuestra matriz hipotética de datos 2. Transformar la matriz en un dataframe 3. Conventir un dataframe en matriz

LABORATORIO: Gestión de datos de red

Gestión de datos de red 1. Transformar la matriz en un dataframe

```
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
```

```
M = as.matrix(
  read.csv("https://raw.githubusercontent.com/PABalland/ON/master/amz.csv" ,
           sep = ",",
           header = T,
           row.names = 1))
```

ejemplo de una matriz de RCA con personas y productos/ paises y productos-industrias

```
EL = get_list(M)
colnames(EL) = c("customer","product","count")
```

convertir un dataframe(lista) en matriz

```
MM = get_matrix(EL)
```

edge list (lista de aristas(source-target—-adyacentes-mis 3 columnas)

```
EL
```

```
##    customer   product count
## 1    Pierre       Tie     0
## 2       Ron       Tie     1
## 3    Andrea       Tie     1
## 4     David       Tie     0
## 5     Cesar       Tie     1
## 6     Paula       Tie     1
## 7    Pierre      Book     0
## 8       Ron      Book     1
## 9    Andrea      Book     1
## 10    David      Book     0
## 11    Cesar      Book     1
## 12    Paula      Book     1
## 13   Pierre Surfboard     1
## 14      Ron Surfboard     0
## 15   Andrea Surfboard     0
## 16    David Surfboard     1
## 17    Cesar Surfboard     0
## 18    Paula Surfboard     1
## 19   Pierre     Short     1
## 20      Ron     Short     0
## 21   Andrea     Short     0
## 22    David     Short     1
## 23    Cesar     Short     0
## 24    Paula     Short     1
## 25   Pierre     Water     1
## 26      Ron     Water     1
## 27   Andrea     Water     1
## 28    David     Water     1
## 29    Cesar     Water     1
## 30    Paula     Water     1
```

Matriz de adyancencia

```
get_matrix(EL)
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         1     1     1
## Ron      1    1         0     0     1
## Andrea   1    1         0     0     1
## David    0    0         1     1     1
## Cesar    1    1         0     0     1
## Paula    1    1         1     1     1
```

Buscar solo un tipo de producto

```
X= subset(EL,product=="Tie")
X
```

```
##   customer product count
## 1   Pierre     Tie     0
## 2      Ron     Tie     1
## 3   Andrea     Tie     1
## 4    David     Tie     0
## 5    Cesar     Tie     1
## 6    Paula     Tie     1
```

LAB_8_24

2025-10-05

Hecho con gusto por Luis Armando González Arellano LABORATORIO - RELACIONAMIENTO “CASO HIPOTÉTICO”

Objetivo: Estimar el relacionamiento (relatedness, proximidad, métrica de distancia entre productos)

En este ejercicio vamos a:

. Cargar nuestra matriz hipotética de datos

. Calcular co ocurrencias entre personas y productos

. Estimar el relacionamiento, teniendo como input las co ocurrencias

. Graficar

```
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
```

1.Cómo se mide la proximidad (relatedness) para crear el espacio producto

```
M= as.matrix(read.csv("https://raw.githubusercontent.com/PABalland/ON/master/amz.csv", sep= ",", header= T, row.n
ames=1))

M = as.matrix(
  read.csv("https://raw.githubusercontent.com/PABalland/ON/master/amz.csv" ,
           sep = ",",
           header = T,
           row.names = 1))
M
```

```
##        Tie Book Surfboard Short Water
## Pierre   0    0         1     1     1
## Ron      1    1         0     0     1
## Andrea   1    1         0     0     1
## David    0    0         1     1     1
## Cesar    1    1         0     0     1
## Paula    1    1         1     1     1
```

co-ocurrencia entre personas/paises/estados

```
co_occurrence(M)
```

```
##        Pierre Ron Andrea David Cesar Paula
## Pierre      0   1      1     3     1     3
## Ron         1   0      3     1     3     3
## Andrea      1   3      0     1     3     3
## David       3   1      1     0     1     3
## Cesar       1   3      3     1     0     3
## Paula       3   3      3     3     3     0
```

co-ocurrencia entre productos t es transpuesta

```
c= co_occurrence(t(M))
c
```

```
##           Tie Book Surfboard Short Water
## Tie         0    4         1     1     4
## Book        4    0         1     1     4
## Surfboard   1    1         0     3     3
## Short       1    1         3     0     3
## Water       4    4         3     3     0
```

estima el relacionamiento o proximidad pero normalizado, para asegurar que el numero de co ocurrencias que observamos es mayor al numero de co ocurrencias probables (probailidad condicional)

```
r=relatedness(c)
r
```

```
##                 Tie      Book Surfboard     Short    Water
## Tie       0.0000000 1.6000000 0.5121951 0.5121951 1.082707
## Book      1.6000000 0.0000000 0.5121951 0.5121951 1.082707
## Surfboard 0.5121951 0.5121951 0.0000000 1.9687500 1.038462
## Short     0.5121951 0.5121951 1.9687500 0.0000000 1.038462
## Water     1.0827068 1.0827068 1.0384615 1.0384615 0.000000
```

#tie and Surfboard puede que no esten tan relacionados, abajo de 1 no es relacionado mientras que mayor a 1 si es relacionado

```
r[r<1] =0
r[r>1] =1
r
```

```
##           Tie Book Surfboard Short Water
## Tie         0    1         0     0     1
## Book        1    0         0     0     1
## Surfboard   0    0         0     1     1
## Short       0    0         1     0     1
## Water       1    1         1     1     0
```

```
library(igraph)
```

```
##
## Adjuntando el paquete: 'igraph'
```

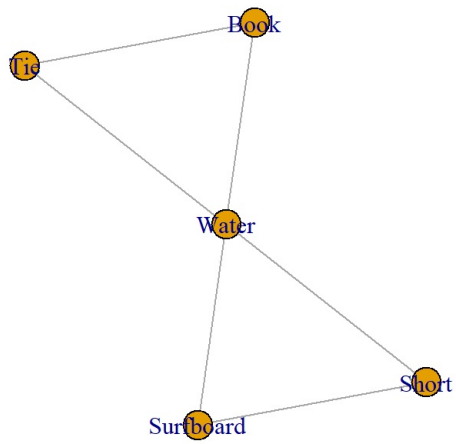
```
## The following object is masked from 'package:EconGeo':
##
##     diversity
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

graficar matriz de adyacientes

```
g1= graph_from_adjacency_matrix(r, mode= "undirected")
plot(g1)
```



lab_9_24

2025-10-05

Alumno_ Luis Armando González Arellano T1_03_ESPACIO PRODUCTO(2) - RELACIONAMIENTO “CASO REAL”: EXPORTACIONES HIDALGUENSES

Objetivo: Estimar el relacionamiento (relatedness, proximidad, métrica de distancia entre productos)

n este ejercicio vamos a: 1. Cargar nuestra matriz de datos 2. Calcular co ocurrencias entre lugares y productos 3. Estimar el relacionamiento, teniendo como input las co ocurrencias 4. Graficar 5. Exportar los resultados para trabajarlos con cytoscape o gephi ## practica 2: exportaciones hidalguenses #

```
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
```

#seleccionar archivo

```
M = as.matrix(
  read.csv("rca_2014.csv" ,
           sep = ",",
           header = T,
           row.names = 1))
```

Ver datos de la matriz en el rango 1:10

```
head (M[,1:10])
```

```
##                         X102 X210 X401 X403 X406 X510 X602 X603 X702 X703
## Acatlán                    0    0    0    0    0    0    0    0    1    0
## Acaxochitlán               0    0    0    0    0    0    0    0    1    0
## Actopan                    0    0    0    0    0    0    0    0    1    0
## Agua Blanca De Iturbide    0    0    0    0    0    0    0    0    1    0
## Ajacuba                    0    0    0    0    0    0    1    0    0    0
## Alfajayucan                0    0    0    0    0    0    0    0    0    0
```

Ver las dimensiones de la matriz

```
dim (M)
```

```
## [1]  83 429
```

co-ocurrencia entre personas/paises/estados

co_occurrence(M) (lo dejo sin chunk ya que este calculo no es el que se usa para graficar y hace que el cuadernos se extienda demasiado)

co-ocurrencia entre productos t es transpuesta

```
c=co_occurrence(t(M))
```

estima el relacionamiento o proximidad pero normalizado, para asegurar que el numero de co ocurrencias que observamos es mayor al numero de co ocurrencias probables (probailidad condicional) tie and Surfboard puede que no esten tan relacionados, abajo de 1 no es relacionado mientras que mayor a 1 si es relacionado.

```
r=relatedness(c)
```

opcional (sólo para trabajar con datos binarios)

```
r[r<1] = 0
r[r>1] = 1
```

```
library (igraph)
```

```
##
## Adjuntando el paquete: 'igraph'
```

```
## The following object is masked from 'package:EconGeo':
##
##     diversity
```

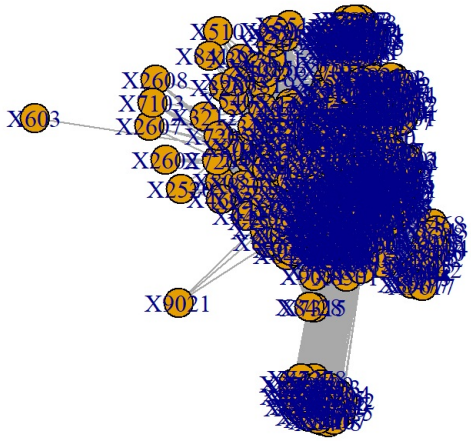
```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

Graficar

```
g1= graph_from_adjacency_matrix(r,mode="undirected")

plot(g1)
```



#exportar resultados del

relacionamiento binario

```
write.csv (r,file= "relatedness.csv")
```

#exportar resultados de co ocurrencias

```
write.csv(c, file="coocurrence.csv")
```

LAB_10_24

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH) Alumno: Luis Armando González Arellano

T1_04_REDES COMPLEJAS(1) - Visión de redes complejas - parte 1

Objetivo: Estimar el Maximum Spanning Tree -árbol de expansión máxima- (asegurar una visión clara del espacio-producto) Red troncal: Estructura general de la red: vamos a poder v. redes complejas

Regla 1: mantener n-1 conexiones como máximo Regla 2: Quitar las conexiones con el peso más bajo, nos vamos quedar con las del peso máximo (menos conexiones) Regla 3: No crear nodos aislados ——————————————————————————————— En este ejercicio vamos a: 1. Usar un matriz hipotetica de datos 2. Graficar sus próximos adyacentes #####

1)la visualización del espacio - producto sea una red conectada: evitar islas de productos aislados. 2)PROBLEMA: tratar de visualizar demasiados enlaces puede crear una complejidad visual innecesaria donde se obstruirán las conexiones más relevantes. calculamos el árbol de expansión máxima (MST) de la matriz de proximidad. MST es el conjunto de enlaces que conecta todos los nodos de la red utilizando un número mínimo de conexiones y la suma máxima posible de proximidades. Calculamos el MST usando el algoritmo de Kruskal: Básicamente, el algoritmo clasifica los valores de la matriz de proximidad en orden descendente y luego incluye enlaces en el MST si y solo si conectan un producto aislado. Por definición, el MST incluye todos los productos, pero el número de enlaces es el mínimo posible. Después de seleccionar los enlaces utilizando los criterios mencionados anteriormente,construimos una visualización utilizando un algoritmo de diseño dirigido por la fuerza.

crear una matriz aleatoria de 200*200

```
M <- matrix(runif(200*200, min=0, max=200), ncol=200)
diag (M) <-0
head (M [,1:6])
```

```
##            [,1]      [,2]       [,3]       [,4]      [,5]      [,6]
## [1,]   0.000000 173.90556 144.881559 148.163792  95.11526  56.10137
## [2,] 135.582070   0.00000   5.014418 165.697056 189.24522 198.85030
## [3,]   3.177223  91.43372   0.000000  93.243359 104.00381 156.61178
## [4,]  96.276586  58.19822 134.880999   0.000000  30.00684  41.27132
## [5,] 133.466029 126.37603 183.666164   6.942432   0.00000  68.91695
## [6,]  32.606857  80.68070 106.880874 163.192889 121.39477   0.00000
```

Para visualizar las dimensiones

```
dim (M)
```

```
## [1] 200 200
```

grafica la matriz

```
library(igraph)
```

```
##
## Adjuntando el paquete: 'igraph'
```

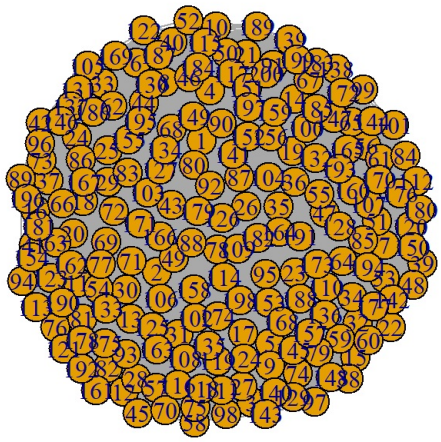
```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
g<- graph.adjacency(M, mode="undirected", weighted="true")
```

```
## Warning: `graph.adjacency()` was deprecated in igraph 2.0.0.
## ℹ Please use `graph_from_adjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plot(g)
```



Convertir la matriz en negativa

```
M<- -M
```

Calcular la matriz de adyacencias

```
g<- graph.adjacency(M, mode="undirected", weighted="true")
```

Calulcar el MST

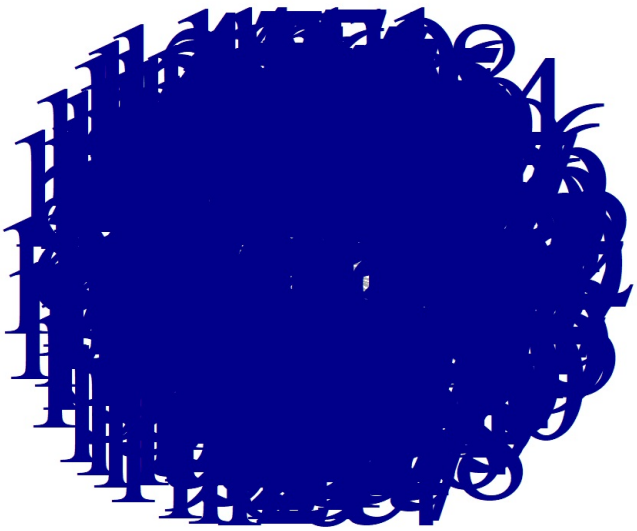
```
MST <- minimum.spanning.tree(g)
```

```
## Warning: `minimum.spanning.tree()` was deprecated in igraph 2.0.0.
## ℹ Please use `mst()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
MST <-mst(g)
```

grafica nueva con matriz negativa, ver el espacio-producto, tecnológico

```
plot (MST, vertex.shape="none", vertex.label.cex=7)
```



matriz de adyacencias (nuevo

relacionamiento)

```
A<- get.adjacency(MST, sparse= F)
```

```
## Warning: `get.adjacency()` was deprecated in igraph 2.0.0.
## ℹ Please use `as_adjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Exportar resultados

```
write.graph(MST,file= "g.gml", format="gml")
```

```
## Warning: `write.graph()` was deprecated in igraph 2.0.0.
## ℹ Please use `write_graph()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
write.csv(A,file="adyacentes.csv")
```

LAB_11_24

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH)

Alumnos: Luis Armando González

LABORATORIO 4: Visión clara del espacio-producto: CASO REAL_exportaciones hidalguenses

Objetivo: Estimar el Maximum Spanning Tree -Árbol de expansión máxima- (asegurar una visión clara del espacio-producto)

Red troncal: Estructura general de la red: vamos a poder ver redes complejas

Regla 1: mantener n-1 conexiones como máximo

Regla 2: Quitar las conexiones con el peso más bajo, nos vamos quedar con las del peso máximo (menos conexiones)

Regla 3: No crear nodos aislados

1.Cómo crear una Visión clara del espacio-producto: árbol de expansión máxima (MST)

```
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
```

Abrir datos

```
M = as.matrix(
  read.csv("relatednessbinario.csv",
           sep = ",",
           header = T,
           row.names = 1))
```

ver la matriz M

```
head (M[,1:10])
```

```
##      X102 X210 X401 X403 X406 X510 X602 X603 X702 X703
## X102    0    0    1    1    0    0    0    0    0    0
## X210    0    0    0    1    1    0    0    0    1    0
## X401    1    0    0    0    0    0    0    0    0    0
## X403    1    1    0    0    1    0    0    0    0    0
## X406    0    1    0    1    0    0    0    0    1    0
## X510    0    0    0    0    0    0    0    0    0    0
```

Ver las dimensiones

```
dim (M)
```

```
## [1] 429 429
```

cargar paquetria igraph

```
library(igraph)
```

```
##
## Adjuntando el paquete: 'igraph'
```

```
## The following object is masked from 'package:EconGeo':
##
##     diversity
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

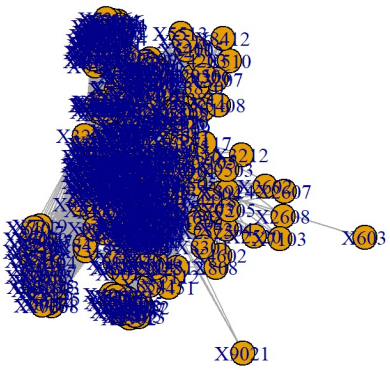
```
## The following object is masked from 'package:base':
##
##     union
```

grafica la matriz de proximidades natural

```
red_hidalgo1<- graph.adjacency(M, mode="undirected", weighted = "true")
```

```
## Warning: `graph.adjacency()` was deprecated in igraph 2.0.0.
## ℹ Please use `graph_from_adjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plot (red_hidalgo1)
```



Transforma la matriz en en

NEGATIVA y para identificar los máximos

```
M <- -M
head(M[,1:6])
```

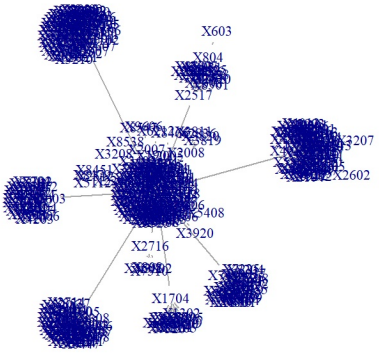
```
##      X102 X210 X401 X403 X406 X510
## X102    0    0   -1   -1    0    0
## X210    0    0    0   -1   -1    0
## X401   -1    0    0    0    0    0
## X403   -1   -1    0    0   -1    0
## X406    0   -1    0   -1    0    0
## X510    0    0    0    0    0    0
```

grafica la nueva matriz negativa con MST

```
red_hidalgo2 <- graph_from_adjacency_matrix(M, mode = "undirected", weighted = "true")
MST<- minimum.spanning.tree(red_hidalgo2)
```

```
## Warning: `minimum.spanning.tree()` was deprecated in igraph 2.0.0.
## ℹ Please use `mst()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plot (MST, vertex.shape= "none", vertex.label.cex=.7)
```



Exportar grafico con formato para

cyto

```
write.graph(MST,file="redhidalgo2.gml", format= "gml")
```

```
## Warning: `write.graph()` was deprecated in igraph 2.0.0.
## ℹ Please use `write_graph()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

matriz de proximos adyaentes (nuevo relacionamiento)

```
A <- get.adjacency(MST, sparse = F)
```

```
## Warning: `get.adjacency()` was deprecated in igraph 2.0.0.
## ℹ Please use `as_adjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

exportar matriz

```
write.csv(A, file = "AdyacentesConMST.csv")
```

LAB_12_24

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH) Alumno: Luis Armando González Arellano video 05 - DENSIDAD DEL RELACIONAMIENTO: CASO HIPOTÉTICO

Objetivo: Estimar la densidad del relacionamiento (rd) La densidad varia entre 0 y 1, los valores más grandes indican que la región ha avanzado a una VCR en muchos bienes proximos al bien i y tiene mayor probabilidad de exportarlo en el futuro mayor densidad, mayor cercanía. Menor densidad, mayor lejanía. ——————————————————————————————— En este laboratorio vamos a: 1. Cargar una matriz hipotética de datos 2. Estimar la VCR de la matriz hipotética 3. Calcular las co-ocurrencias y el relacionamiento (espacio-producto) 4. Estimar la densidad del relacionamiento (rd) Buscamos saber si las industrias que estan relacionadas estan presentes en las regiones o no. Nivel de cercanía 5. Predecir la entrada de una nueva industria Econometría usar count(densidad del relacionamiento) como regresora (predictora) de la entrada de una nueva industria

LABORATORIO 5: densidad del relacionamiento (rd)

```
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 a region- industry matrix

```
set.seed (31)
mat <- matrix(sample(0:100,20,replace=T),ncol=4)
rownames(mat) <- c ("R1","R2","R3","R4","R5")
colnames(mat) <- c ("I1","I2","I3","I4")
dim(mat)
```

```
## [1] 5 4
```

Transformar la matriz en binaria (seria similar a obtener una RCA)

```
mat= rca(mat,binary=T)
```

calcula la matriz de co-ocurrencia de la mat transpuesta

```
c= co_occurrence(t(mat))
c
```

```
##    I1 I2 I3 I4
## I1  0  1  1  1
## I2  1  0  0  1
## I3  1  0  0  0
## I4  1  1  0  0
```

calcula el relacionamiento con base en las co-ocurrencias

```
r=relatedness(c)
r
```

```
##           I1        I2       I3        I4
## I1 0.0000000 0.9090909 1.944444 0.9090909
## I2 0.9090909 0.0000000 0.000000 1.5000000
## I3 1.9444444 0.0000000 0.000000 0.0000000
## I4 0.9090909 1.5000000 0.000000 0.0000000
```

Convertir el relacionamiento en binario

```
r[r<1]=0
r[r>1]=1
```

Calcular la Densidad del relacionamiento

```
rd= relatedness_density(mat,r)
rd
```

```
##     I1  I2  I3  I4
## R1   0 100   0 100
## R2   0   0 100 100
## R3 100   0   0   0
## R4 100   0 100   0
## R5   0 100 100   0
```

para econometría

```
rd=get_list(rd)
rd
```

```
##    Region Industry Count
## 1      R1       I1     0
## 2      R2       I1     0
## 3      R3       I1   100
## 4      R4       I1   100
## 5      R5       I1     0
## 6      R1       I2   100
## 7      R2       I2     0
## 8      R3       I2     0
## 9      R4       I2     0
## 10     R5       I2   100
## 11     R1       I3     0
## 12     R2       I3   100
## 13     R3       I3     0
## 14     R4       I3   100
## 15     R5       I3   100
## 16     R1       I4   100
## 17     R2       I4   100
## 18     R3       I4     0
## 19     R4       I4     0
## 20     R5       I4     0
```

6. Predecir entrada

generate a first region - industry matrix in which cells represent the presence/absence

of a RCA (period 1)

```
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")
```

generate a second region - industry matrix in which cells represent the presence/absence

of a RCA (period 2)

```
mat2 <- mat1
mat2[3, 1] <- 1

mat2
```

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

```
mat1
```

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

#ver que las posibilidades de entrada es cuando tienes un 0 de VCR #cuando tienes 1 puedes: permanecer o salir, pero no entrar # NA, no puede entrar # 1 tiene potencial de entrada y entró # 0 puede tener potencial de entrada, pero no ha entrado

```
d= entry_list(mat1, mat2)
d
```

```
##    region industry entry period
## 1      R1       I1     0      2
## 2      R2       I1     0      2
## 3      R3       I1     1      2
## 4      R4       I1    NA      2
## 5      R5       I1    NA      2
## 6      R1       I2    NA      2
## 7      R2       I2     0      2
## 8      R3       I2     0      2
## 9      R4       I2     0      2
## 10     R5       I2    NA      2
## 11     R1       I3     0      2
## 12     R2       I3     0      2
## 13     R3       I3     0      2
## 14     R4       I3     0      2
## 15     R5       I3    NA      2
## 16     R1       I4    NA      2
## 17     R2       I4    NA      2
## 18     R3       I4    NA      2
## 19     R4       I4    NA      2
## 20     R5       I4     0      2
```

```
colnames(d)= c("Region","Industry","Entry","Period")
d= merge(d, rd, by=c ("Region", "Industry"))

d
```

```
##    Region Industry Entry Period Count
## 1      R1       I1     0      2     0
## 2      R1       I2    NA      2   100
## 3      R1       I3     0      2     0
## 4      R1       I4    NA      2   100
## 5      R2       I1     0      2     0
## 6      R2       I2     0      2     0
## 7      R2       I3     0      2   100
## 8      R2       I4    NA      2   100
## 9      R3       I1     1      2   100
## 10     R3       I2     0      2     0
## 11     R3       I3     0      2     0
## 12     R3       I4    NA      2     0
## 13     R4       I1    NA      2   100
## 14     R4       I2     0      2     0
## 15     R4       I3     0      2   100
## 16     R4       I4    NA      2     0
## 17     R5       I1    NA      2     0
## 18     R5       I2    NA      2   100
## 19     R5       I3    NA      2   100
## 20     R5       I4     0      2     0
```

lm(dependiente~independiente1+independiente2, data=datos)

que tanto la “densidad del relacionamiento” impacta la entrada

```
summary (lm(d$Entry~ d$Count))
```

```
##
## Call:
## lm(formula = d$Entry ~ d$Count)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3333  0.0000  0.0000  0.0000  0.6667
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.347e-17  9.623e-02   0.000    1.000
## d$Count      3.333e-03  1.843e-03   1.809    0.104
##
## Residual standard error: 0.2722 on 9 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.2667, Adjusted R-squared:  0.1852
## F-statistic: 3.273 on 1 and 9 DF,  p-value: 0.1039
```

LAB_13_24

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH) Alumno: Luis Armando González Arellano

LABORATORIO - Gráficos en R con ggplot2

cargar libreria ggplot2

```
library(ggplot2)
```

leer data frame

```
green_data <- read.csv("REGESIONES FINALES.csv")
```

echando un ojo a los datos

```
names(green_data)
```

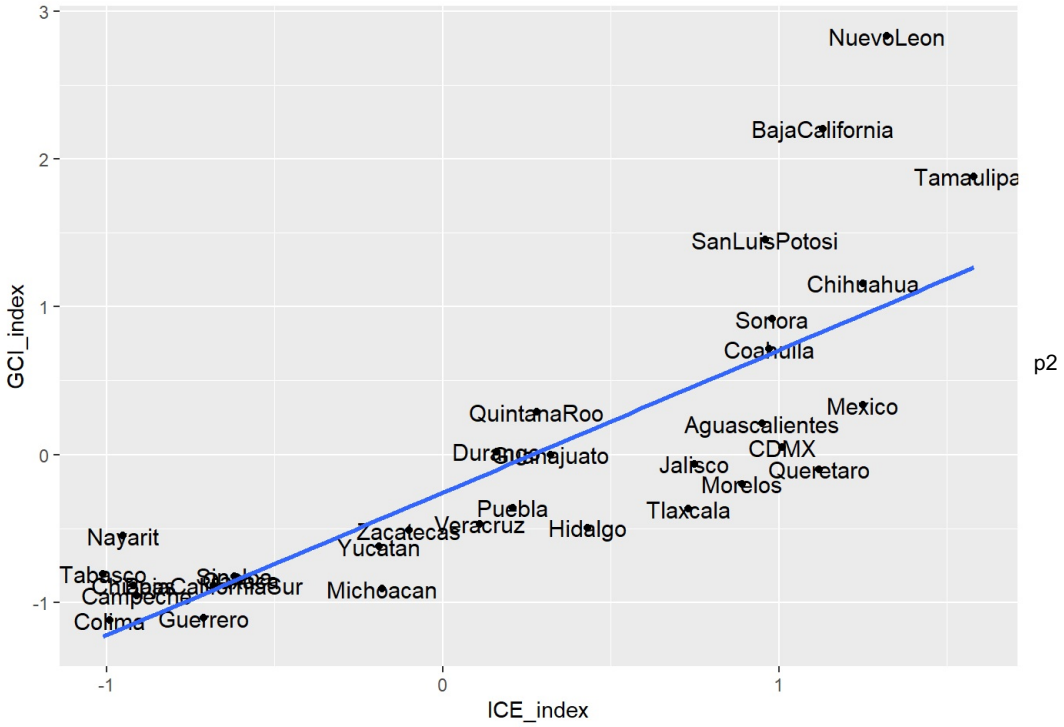
```
## [1] "STATE"     "GCI_rank"  "ICE_rank"  "GCI_index" "ICE_index" "PIBE"
## [7] "LPIBE"
```

colocando objetos

```
p1 <-  ggplot(data = green_data,
              mapping=aes(x=ICE_index,
                          y=GCI_index,))+
  geom_point()+
  geom_text(label= green_data$STATE,
            color= "black",
            size = 4)+
  geom_point()+
  geom_smooth(method = lm,
              se = FALSE,
              fullrange = T)

p1
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

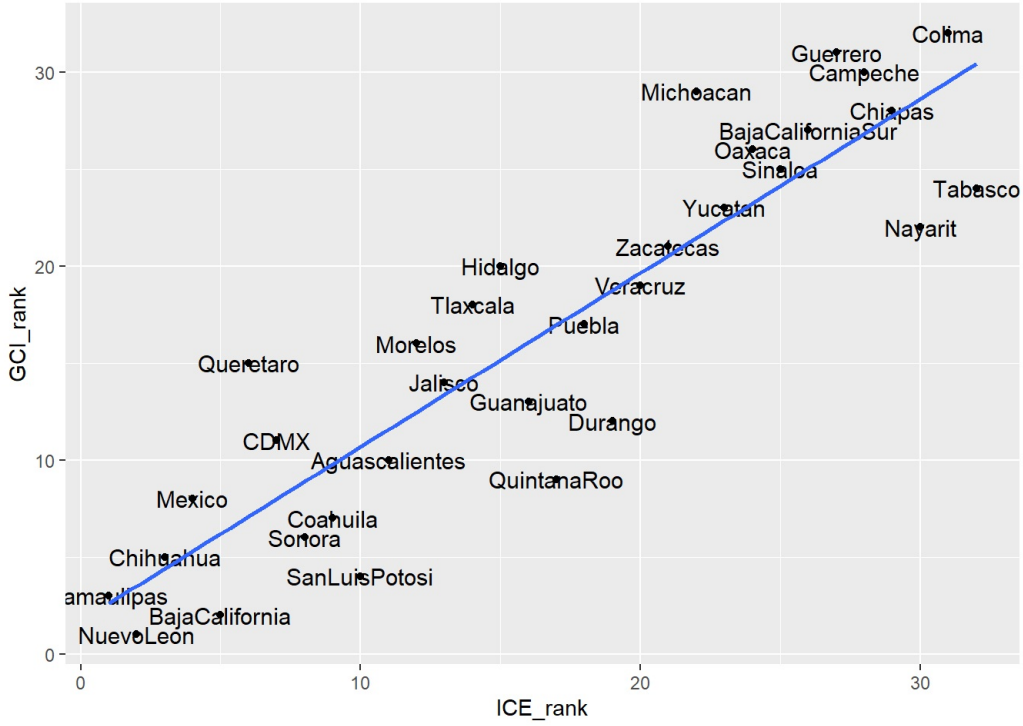


p2

```
p2 <- ggplot(data = green_data,
             mapping=aes(x=ICE_rank,
                         y=GCI_rank,))+
  geom_point()+
  geom_text(label= green_data$STATE,
            color= "black",
            size = 4)+
  geom_point()+
  geom_smooth(method = lm,
              se = FALSE,
              fullrange = T)

p2
```

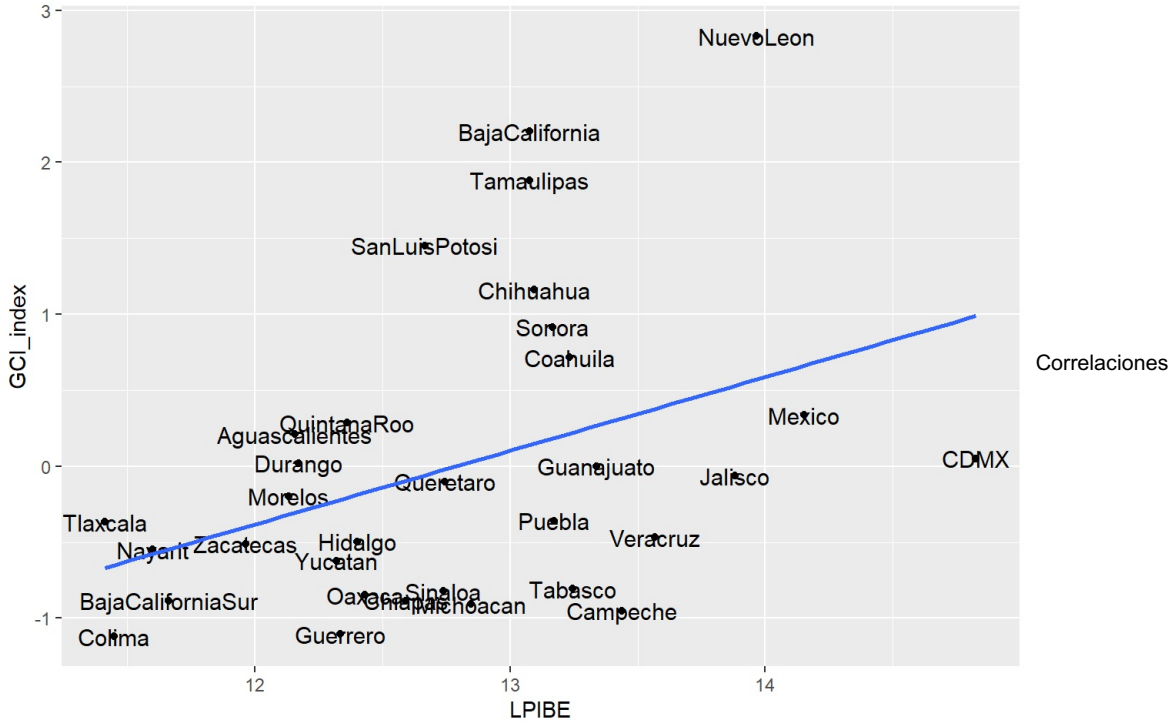
```
## `geom_smooth()` using formula = 'y ~ x'
```



p3

```
p3 <- ggplot(data = green_data,
             mapping=aes(x=LPIBE,
                         y=GCI_index,))+
  geom_point()+
  geom_text(label= green_data$STATE,
            color= "black",
            size = 4)+
  geom_point()+
  geom_smooth(method = lm,
              se = FALSE,
              fullrange = T)
p3
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
library(psych)
```

```
##
## Adjuntando el paquete: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(readr)
correl <- read_csv("correl.csv")
```

```
## Rows: 32 Columns: 5
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (5): GCI_rank, ICE_rank, GCI_index, ICE_index, LPIBE
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

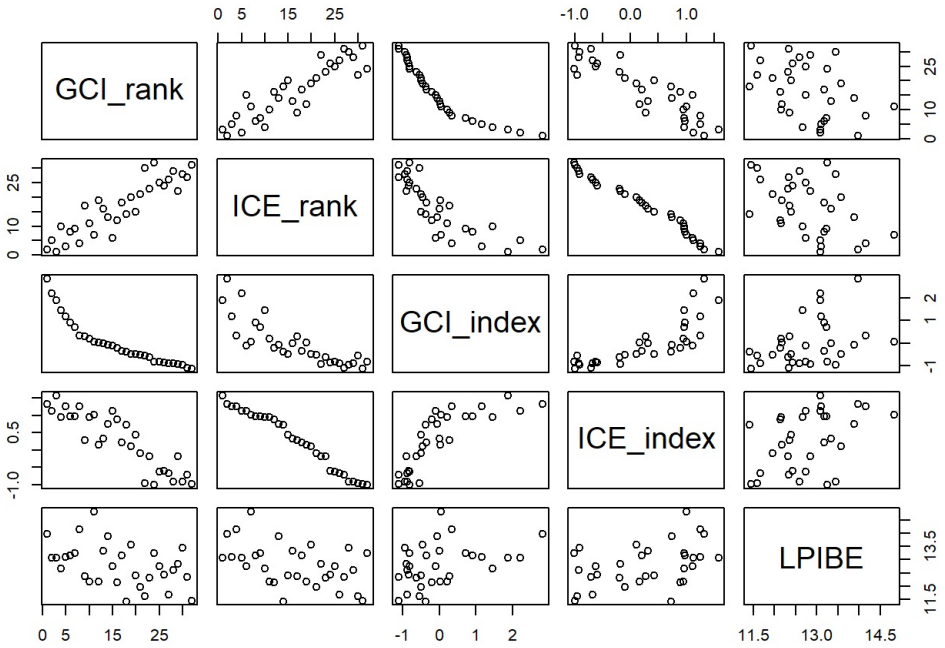
```
head(correl)
```

```
## # A tibble: 6 × 5
##   GCI_rank ICE_rank GCI_index ICE_index LPIBE
##      <dbl>    <dbl>     <dbl>     <dbl> <dbl>
## 1       10       11     0.210      0.95  12.2
## 2        2        5     2.20       1.13  13.1
## 3       27       26    -0.885     -0.68  11.7
## 4       30       28    -0.951     -0.91  13.4
## 5       28       29    -0.885     -0.92  12.6
## 6        5        3     1.16       1.25  13.1
```

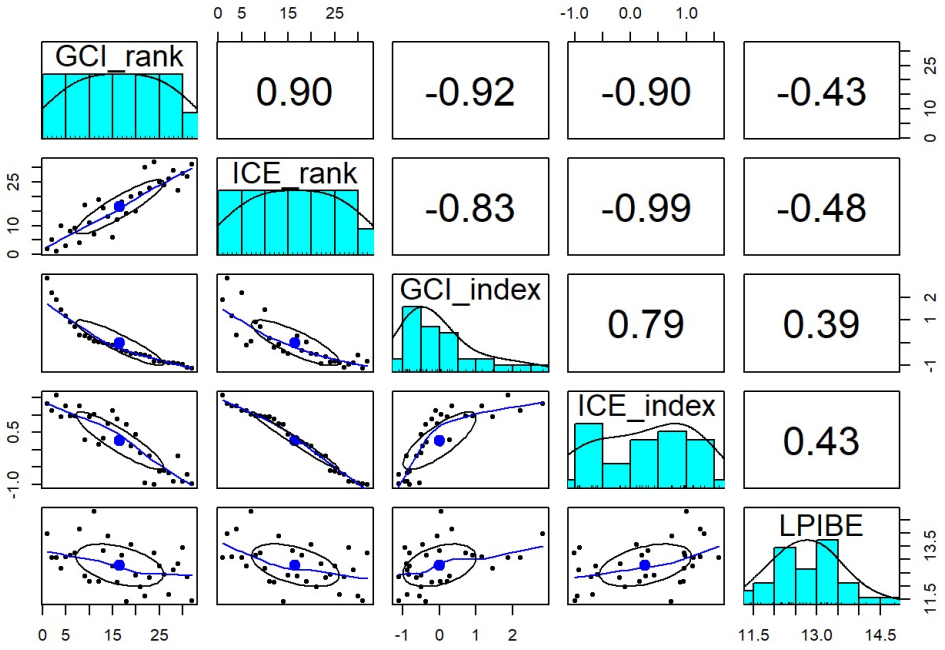
```
attach(correl)
names(correl)
```

```
## [1] "GCI_rank"  "ICE_rank"  "GCI_index" "ICE_index" "LPIBE"
```

```
pairs(correl)
```




```
pairs.panels(correl)
```



correlación de pearson

```
complex_corr <- cor(correl,method = "pearson")
complex_corr
```

```
##             GCI_rank   ICE_rank  GCI_index  ICE_index      LPIBE
## GCI_rank   1.0000000  0.8969941 -0.9166564 -0.8957551 -0.4288321
## ICE_rank   0.8969941  1.0000000 -0.8274473 -0.9875750 -0.4750548
## GCI_index -0.9166564 -0.8274473  1.0000000  0.7946666  0.3905008
## ICE_index -0.8957551 -0.9875750  0.7946666  1.0000000  0.4272882
## LPIBE     -0.4288321 -0.4750548  0.3905008  0.4272882  1.0000000
```

Calculamos la correlación redondeada

```
complex_corr = round(complex_corr,digits = 2)
complex_corr
```

```
##           GCI_rank ICE_rank GCI_index ICE_index LPIBE
## GCI_rank      1.00     0.90     -0.92     -0.90 -0.43
## ICE_rank      0.90     1.00     -0.83     -0.99 -0.48
## GCI_index    -0.92    -0.83      1.00      0.79  0.39
## ICE_index    -0.90    -0.99      0.79      1.00  0.43
## LPIBE        -0.43    -0.48      0.39      0.43  1.00
```

#matriz de correlación

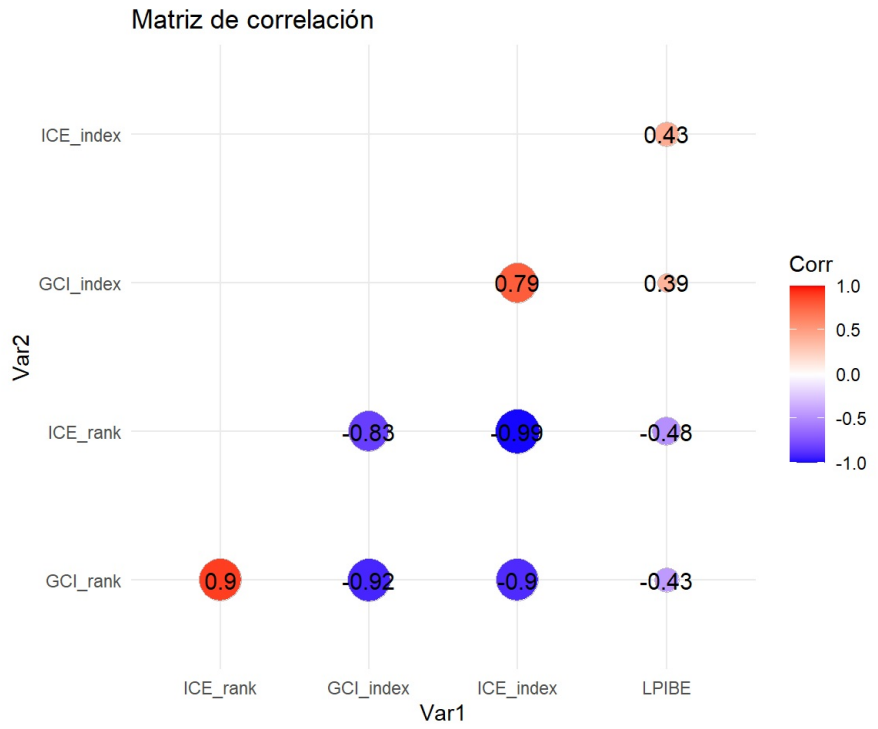
```
library(ggcorrplot)
library(ggplot2)



p4 <- ggcorrplot(complex_corr, method = "circle", type = "lower", lab = TRUE)+
  ggtitle("Matriz de correlación")+
  theme_minimal()
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## ℹ The deprecated feature was likely used in the ggcorrplot package.
##   Please report the issue at <https://github.com/kassambara/ggcorrplot/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p4
```




```
require(ggpubr)
```

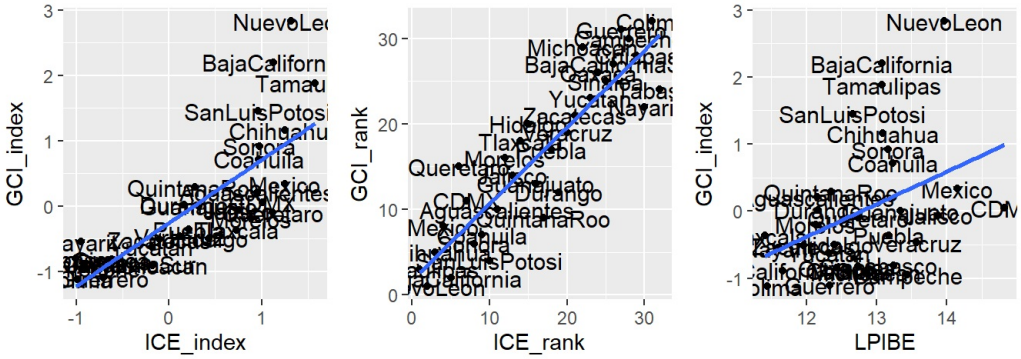
```
## Cargando paquete requerido: ggpubr
```

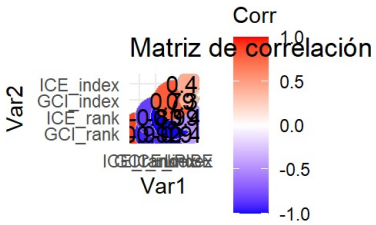
```
ggpubr :: ggarrange (p1, p2, p3, p4, etiquetas = c ("A", "B", "C" , "D"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning in as_grob.default(plot): Cannot convert object of class character into
## a grob.
```





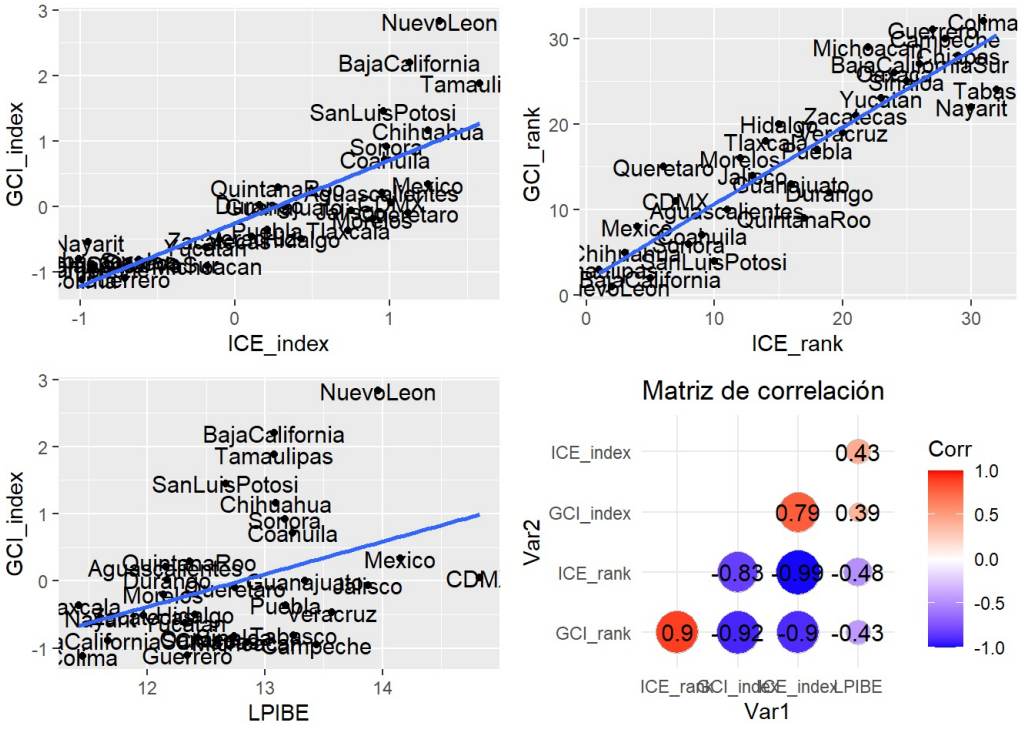

```
library(ggplot2)
require (gridExtra)
```

```
## Cargando paquete requerido: gridExtra
```

Crear un grid con las visualizaciones

```
F1 <- grid.arrange (p1, p2, p3, p4, nrow = 2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Lab_16_24md

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH) Alumno: Luis Armando Gonzalez Arellano

##Laboratorio -Analisis de redes en R- indicadores básicos globales

```
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
```

#Cargar data

```
EL = read.csv("https://raw.githubusercontent.com/PABalland/ON/master/lesmis-el.csv")
```

ver encabezado

```
head(EL)
```

```
##     Character1  Character2 Weight
## 1 Gillenormand JeanValjean      2
## 2      Zephine   Listolier      3
## 3         Joly     Feuilly      5
## 4       Brevet       Judge      2
## 5   Bamatabois JeanValjean      2
## 6     Gavroche JeanValjean      1
```

Transformar a matriz

```
MM <- get_matrix(EL)
```

Graficar red

```
library(igraph)
```

```
##
## Adjuntando el paquete: 'igraph'
```

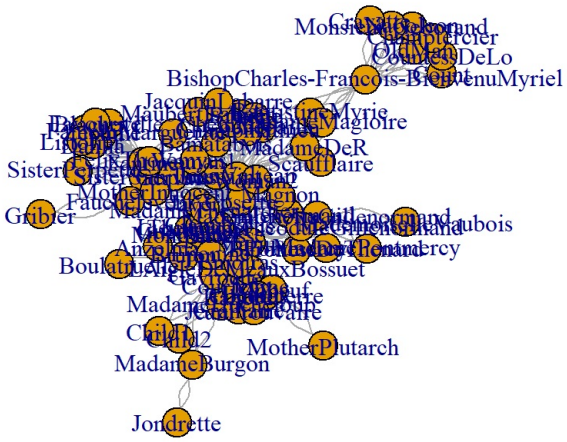
```
## The following object is masked from 'package:EconGeo':
##
##     diversity
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

se crea grafico a partir del “EL”

```
g <- graph_from_data_frame(d=EL, directed= F)
plot(g)
```



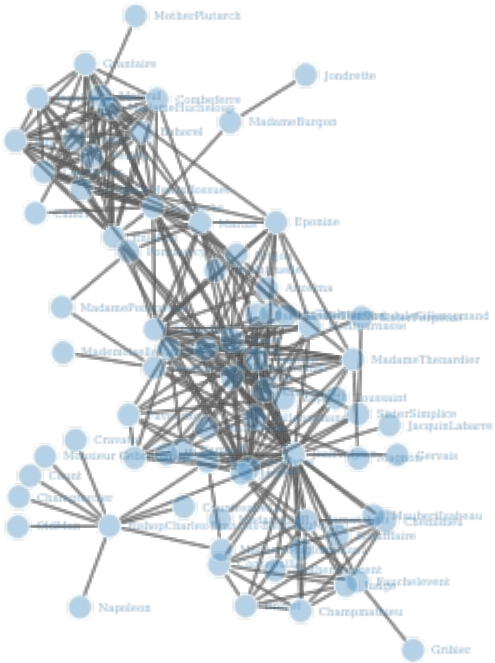
quitar los “pesos”

```
EL$Weight= NULL
head(EL)
```

```
##     Character1  Character2
## 1 Gillenormand JeanValjean
## 2      Zephine   Listolier
## 3         Joly     Feuilly
## 4       Brevet       Judge
## 5   Bamatabois JeanValjean
## 6     Gavroche JeanValjean
```

graficar red simple

```
library(networkD3)
simpleNetwork(EL)
```



LAB_18C_24

2025-10-05

PGI - 2019 ——————————————————————————————————————-

```
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 a region - industry matrix

```
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")
```

a vector of GDP of regions

```
vec <- c(5, 10, 15, 25, 50)
```

run the function

```
prody(mat, vec)
```

```
##            I1       I2       I3       I4
## [1,] 21.20755 18.56203 22.77426 22.57471
```

INSTALAR paquetrias y llamarlas

```
library(haven)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(EconGeo)
```

cargar datos

```
datos1<-read_dta("country_hsproduct6digit_year-001.dta")
head(datos1)
```

```
## # A tibble: 6 × 9
##   location_id product_id  year export_value import_value hs_eci hs_coi
##         <dbl>      <dbl> <dbl>        <dbl>        <dbl>  <dbl>  <dbl>
## 1           0       5000  1994            0         4007  0.581 -0.604
## 2           0       5000  1995        18008         7199 -0.503 -0.713
## 3           0       5000  1996            0         3020 -0.681 -0.710
## 4           0       5000  1997           NA           NA -1.29  -0.825
## 5           0       5000  1998           NA           NA  0.198 -0.726
## 6           0       5000  1999        14472        46668 -0.116 -0.830
## # i 2 more variables: location_code <chr>, hs_product_code <chr>
```

filtrando Año 2019

```
datos2019 = datos1 [datos1$year == "2019" ,]
```

exportar los datos

```
write.csv(datos2019, file = "datos2019_6DIG.csv")
```

match entre exportaciones y gini

```
green_products = read.csv("GINI_2019_F.csv")
all_products = read.csv("datos2019_6DIG.csv")
merged.full.1 <-  merge(green_products, all_products, by="location_code", all.x= T)
write.csv(merged.full.1, file = "exportaciones2019_match.csv")
```

Similarly to Hartmann et al. (2017), countries with an average export value under 1 billion #dollars were excluded from the analysis to avoid taking into account small countries. T

```
ExportByCountry = merged.full.1 %>%
  group_by(location_code) %>%
  dplyr::summarise(suma_exports  = sum(export_value, na.rm = TRUE),count= n())
write.csv(ExportByCountry, file = "ExportByCountry.csv")
```

#importar vectores

```
EL <- read.csv("exportadores.csv")
MM = get_matrix(EL)
class(MM)
```

```
## [1] "matrix" "array"
```

```
head (MM[,1:10])
```

```
##          100110     100190  100200    100300   100400    100510     100590
##     NA        0          0       0         0        0         0          0
## ALB  0        0       8578       0         0        0     57504       3992
## ARG  0 73613259 2152739990   76133 624105675   176030 218047241 5742306962
## ARM  0        0       7988       0         0        0         0          0
## AUT  0 34607398  176404668 5130845  14804700  4937560 141719122   90438897
## BEL  0 43915684  109239286  313402  47358263 10255346   4878943   35068525
##       100610   100620
##            0        0
## ALB       NA        0
## ARG 24208185 32139787
## ARM        0        0
## AUT    92371   225233
## BEL  1456927 22535848
```

```
dim (MM)
```

```
## [1]   70 5046
```

```
write.csv(MM, file = "MatrizExportadores.csv")
```

match entre paises ordenados

```
green_products = read.csv("GINI_2019_F.csv")
all_products = read.csv("id_orden.csv")
merged.full.1 <-  merge(all_products, green_products, by="location_code", all.x= T)
write.csv(merged.full.1, file = "ordendelgini.csv")
```

cargar vector de gini

```
vectoriza <- c(30.1, 42.9, 30, 30.2, 27.2, 40.3, 25.3, 41.6, 53.5, 31.7, 38.2, 51.3, 48.2, 31.2, 25.3, 31.7, 27.7
, 41.9, 45.7, 31.9, 34.3, 30.8, 27.7, 30.7, 31.2, 32.8, 35.9, 33.1, 48.2, 28.9, 30, 37.6, 35, 30.8, 40.9, 34.6, 2
9.7, 37.7, 35.3, 34.2, 34.5, 26, 29.3, 35.5, 33.5, 31, 50.5, 38.5, 29.2, 27.7, 49.8, 41.6, 28.8, 32.8, 45.7, 34.8
, 37.7, 38.8, 34.5, 23.2, 24.4, 29.3, 34.9, 41.9, 42.7, 26.6, 39.7, 41.5, 50.3)
vec <- as.data.frame(vectoriza)
class(vec)
```

```
## [1] "data.frame"
```

```
dim(vec)
```

```
## [1] 69  1
```

```
vectorfinal = (t(vec))
dim(vectorfinal)
```

```
## [1]  1 69
```

```
matriz = as.matrix(
  read.csv("MatrizExportadores2.csv",
           sep = ",",
           header = T,
           row.names = 1))


dim(matriz)
```

```
## [1]   69 5045
```

#Step 3

```
p1 <- prody(matriz, vectorfinal)
p2 <- as.data.frame(t(p1))
write.csv(p2, "PGI_2019calculada.csv")
```

LAB_20_24

2025-10-05

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH) Alumno: Luis Armando González Arellano

LABORATORIO - Gráficos en R con ggplot2 faceting (romper un gráfica en varias dentro de una cuadrícula)

instalando paquete install.packages("ggplot2")

cargar libreria ggplot2

```
library(ggplot2)
```

leer base

```
densidad_verde <- read.csv("PARA FACETING.csv")
```

VER DATOS

```
names(densidad_verde)
```

```
##  [1] "ID"           "datamex_code" "Product"      "PCI.2018"     "N_PCI.2018"
##  [6] "Region"       "rca_bin"      "DENSITY"      "GCI_2018"     "GP_SINVCR"
## [11] "GCP_2018"
```

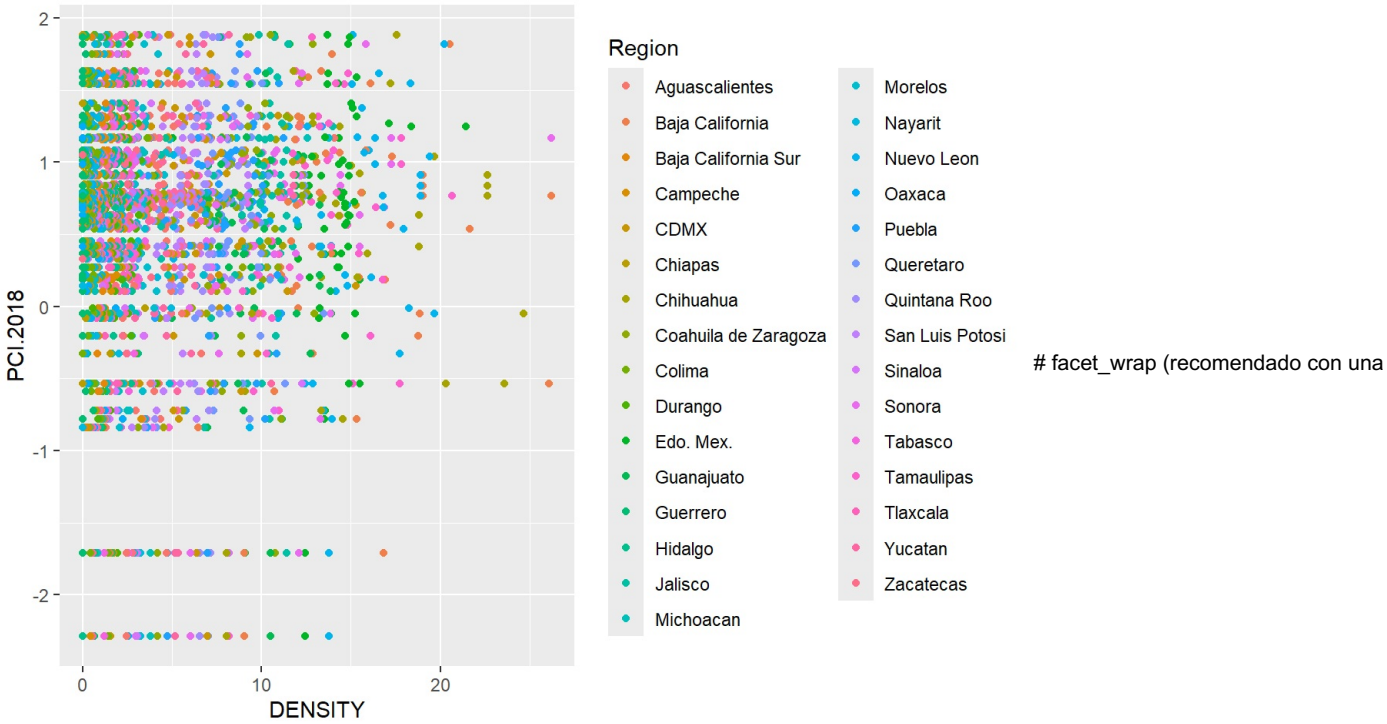
colocando grafica base

```
p <- ggplot(data=densidad_verde,
            mapping = aes(x=DENSITY,
                          y=PCI.2018,
                          color = Region))+
  geom_point()
```

abrir objeto

```
p
```

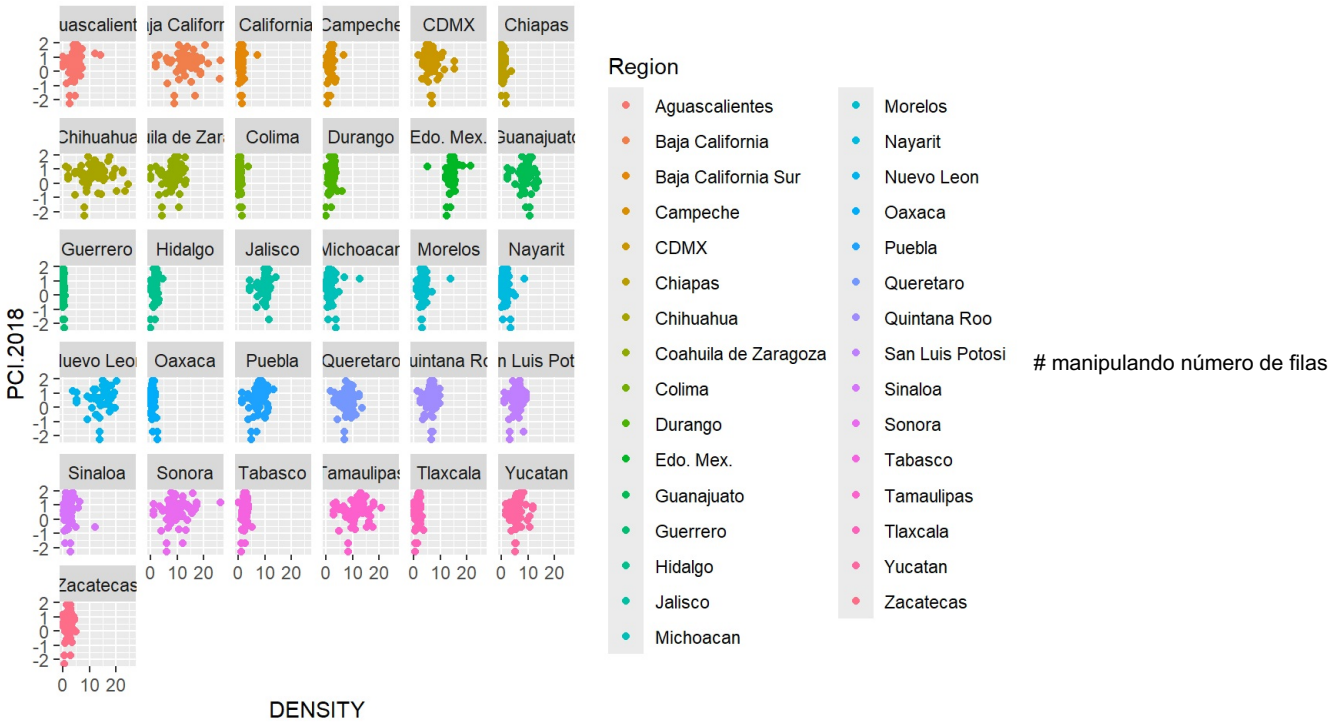
```
## Warning: Removed 496 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



facet_wrap (recomendado con una variable)

```
p + facet_wrap(~Region)
```

```
## Warning: Removed 496 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



manipulando número de filas

```
p + facet_wrap(~Region, nrow=5)
```

```
## Warning: Removed 496 rows containing missing values or values outside the scale range
## (`geom_point()`).
```