

DOCTORADO EN CIENCIAS ECONÓMICO-ADMINISTRATIVAS

TS1- Complejidad Económica

Cuaderno en R: Notebook.

Catedrático: Dra. Carla Carolina Pérez Hernández.

Estudiante: Leislíe Rocío Manjarrez Olmos.



R Notebook

CODIGO MODIFICADO EN DIA MARTES

Laboratorio 0- Analisis de redes en R - Algebra matricial

Objetivo: Explorar las funciones de algebra lineal con un caso hipotetico

En este ejercicio vamos a: 1. Cargar nuestra matriz hipotetica de datos 2. Hacer diversas operaciones matriciales 3. Comprender el algoritmo base de las recomendaciones de AMAZON

Cargar la matriz

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

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

Ver solo el vector de clientes que compran el producto "corbata" (producto de la columna 1)

Ver solo un fragmento de la base original, en relacion a los dos productos que me interesan. Puede ser util para explorar clientes y productos

Computar grado de centralidad

Funcion util para enfocarnos en productos y no solo en clientes

Suma de matrices

Crear nuevo un objeto llamado x

Multiplicacion de matrices checar tamanos. Checar tamano de la matriz -dim-

El algoritmo de recomendacion de AMAZON, hace basicamente lo anterior. Para tener el numero de veces que 2 productos han sido comprados por el mismo cliente. Identificar productos que van frecuentemente juntos (co ocurrencias par)

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

No es una matriz identidad (como en la correlacion)

Similaridad de clientes - matriz de co ocurrencia de clienes

No es una matriz identidad (como en la correlacion)

R Notebook

Code ▾

Laboratorio 1 - Analisis de redes en R - Algebra matricial

Objetivo: Explorar las funciones de algebra lineal con un caso hipotetico

En este ejercicio vamos a: 1. Cargar nuestra matriz hipotetica de datos 2. Hacer diversas operaciones matriciales 3. Comprender el algoritmo base de las recomendaciones de AMAZON

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

Hide

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

M enter para poder ver la matriz y dim(M) para dimensionar la matriz

Hide

```
M
dim(M)
```

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

Hide

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

Ver solo un fragmento de la base original, en relacion a los dos productos que me interesan. Puede ser util para explorar clientes y productos

Hide

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

Computar grado de centralidad, productos relacionados con las personas.

Hide

rowSums (M)

Funcion util para enfocarnos en productos y no solo en clientes, matriz transpuesta

Hide

t(M)

Suma de matrices

Hide

M + M

Crear nuevo un objeto llamado x

Hide

X = M + M

Matriz al cuadrado

Hide

M ^ 2

Multiplicacion de matrices

Hide

M * M

Multiplicacion de matrices checar tamanos, solo pueda hacerla si el numero de columnas coincide con el numero de filas por eso primero reviso con dim. Checar tamaño de la matriz -dim-

Hide

M %*% t(M)

El algoritmo de recomendacion de AMAZON, hace basicamente lo anterior. Para tener el numero de veces que 2 productos han sido comprados por el mismo cliente. Identificar productos que van frecuentemente juntos (co ocurrencias par)

Hide

$t(M) \% \% M$

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

Hide

$P = t(M) \% \% M$

No es una matriz identidad (como en la correlacion)

Hide

$\text{diag}(P) = 0$

Similaridad de clientes - matriz de co ocurrencia de clienes

Hide

$C = M \% \% t(M)$

No es una matriz identidad (como en la correlacion)

Hide

$\text{diag}(C) = 0$

R Notebook

Code ▾

Laboratorio 5- Pivotar a lo largo

Prerrequisitos: instalar paquete tidyverse y readr `install.packages("tidyverse")` `install.packages("readr")`

Cargar paquete tidyverse y readr

Hide

```
library(tidyverse)
library(readr)
```

Cargar datos en environment seleccionar Import Dataset y seleccionar archivo en csv. Una vez verificado que la tabla cargo correctamente copiar codigo esquina inferior izquierda e importar datos. Pegar el codigo copiado y sustituir ICE_2014 por Tabla1

Hide

```
library(readr)
Tabla1 <- read_delim("ICE_2014.csv",
                    delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

Permite ver la tabla

Hide

```
head(Tabla1)
```

Permite ver todas las columnas

Hide

```
names(Tabla1)
```

Pivotar tabla "A LO LARGO"

Hide

```
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", values_to = "ranking")
```

Exportar resultado: tabla ordenada

Hide

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

Permite ver la ruta donde guardo el archivo

Hide

```
getwd()
```


R Notebook

Code ▾

Laboratorio 7 - Gestion de datos de red - curaduria de datos

Objetivo: Explorar la organizacion de los datos En este ejercicio vamos a: 1. Cargar nuestra matriz hipotetica de datos 2. Transformar la matriz en un dataframe 3. Conventir un dataframe en matriz

Primero hay que instalar EconGeo `install.packages("visNetwork")` `install.packages("htmlwidgets")` `install.packages("igraph")`
`install.packages("reshape")` `install.packages("Matrix")` `install.packages("RSiena")` `install.packages("networkD3")` `install.packages("curl")`
`install.packages("devtools")` `library(devtools)` `devtools::install_github("PABalland/EconGeo", force = T)`

Despues hay que llamar la libreria de EconGeo

Hide

```
library (EconGeo)
```

Se llama a la matriz M con la que se ha trabajado

Hide

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

El comando `?RCA` muestra a que se refiere la Ventaja Comparativa Revelada (VCR o RCA). La matriz de RCA es un ejemplo de matriz con personas y productos/ paises y productos-industrias

Hide

```
?RCA
```

Este código cambia la estructura de la matriz a un listado (dataframe) conforme a lo solicitado: customer, product y count

Hide

```
EL = get.list (M)
colnames (EL) = c ("Customer", "Product", "Count")
```

Comando para convertir un dataframe (lista) en matriz, es decir regresamos al estado original de la matriz M

Hide

```
MM = get.matrix (EL)
```

Al final EL es una edge list o lista de aristas o lista de adyacentes (source-target—adyacentes-mis 3 columnas)

Hide

```
EL
```

Matriz adyacencia- Comando utilizado para pasar una edge list a una matriz de adyacencia. Buscar Andrea- Para observar que compro cada cliente ir a panel de datos y en el listado (dataframe) en busqueda colocar el nombre del cliente para observar que los productos y la Ventaja Comparativa Reveleada

Hide

```
get.matrix (EL)
```

Genera el objeto x que crea una tabla (subset) que contiene el dataframe (listado) del producto denominado corbata (Tie)

Hide

```
x = subset (EL, Product == "Tie")
```

R Notebook

[Code ▼](#)

LABORATORIO 8 - RELACIONAMIENTO “CASO HIPOTETICO”

Objetivo: Estimar el relacionamiento (relatedness, proximidad, metrica de distancia entre productos) En este ejercicio vamos a: 1. Cargar nuestra matriz hipotetica de datos 2. Calcular co ocurrencias entre personas y productos 3. Estimar el relacionamiento, teniendo como input las co-ocurrencias 4. Graficar

1.Como se mide la proximidad (relatedness) para crear el espacio producto Material: Balland, P.A. (2017) Economic Geography in R: Introduction to the EconGeo Package, Papers in Evolutionary Economic Geography, 17 (09): 1-75 Para instalar: <https://www.paballand.com/install-r> (<https://www.paballand.com/install-r>)

Una vez instalado EconGeo llamar su libreria

[Hide](#)

```
library (EconGeo)
```

Llamamos a la matriz M que se encuentra en el link escrito y al ser un archivo cvs decimos que esta separado por comas, la fila inicial va a tener los titulos (header, T) y en la primer columna voy a estrablecer el nombre de mis variables

[Hide](#)

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

Llamo a la matriz

[Hide](#)

```
M
```

Co-ocurrencia entre personas/paises/estados

[Hide](#)

```
co.occurrence (M)
```

Co-ocurrencia entre productos t es transpuesta

Hide

```
c = co.occurrence (t(M))
```

Llamo a la matriz

Hide

```
c
```

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 s puede que no esten tan relacionados, abajo de 1 no es relacionado mientras que mayor a 1 si es relacionado

Hide

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

Llamo a la matriz

Hide

```
r
```

Para ver el algoritmo solo ejecutar relatedness. Ejecutar la regla: si el relacionamiento es menor a 1 = 0 (no habra arista que ligue producto) y si es mayor a 1 = 1 (si habra arista que ligue producto)

Hide

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

Cargar libreria para graficar el espacio-producto

Hide

```
library (igraph)
```

Comando para realizar el grafico 1

Hide

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

Para ver el gráfico

Hide

```
plot(g1)
```

R Notebook

[Code ▼](#)

Laboratorio 9- T1_03_ESPACIO PRODUCTO(2) - RELACIONAMIENTO “CASO REAL”: EXPORTACIONES HIDALGUENSES

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

Primero llamar la libreria de EconGeo

[Hide](#)

```
library (EconGeo)
```

Seleccionar el archivo desde la ruta del ordenador, utilizando el panel Data y el icono Import Dataset. Considerar el archivo como From Text (base). Seleccionar en Heading la opcion Yes en la pantalla que aparece y dar clic en Importar. Copiar la ruta que aparece en la consola una vez cargados los datos y pegar en read.csv (“RUTA”). Al correr el comando aparece la matriz en el panel Data

[Hide](#)

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

Para visualizar los primeros 10 datos de la matriz y que es una matriz de 83X429

[Hide](#)

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

Ejecuta la co-ocurrencias en en este caso de los municipios (lugares)

[Hide](#)

```
co.occurrence (M)
```

Para sacar la co-ocurrencia entre productos t debe ser transpuesta y crear un objeto denominado c

Hide

```
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)

Hide

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

Teniendo el numero de co-ocurrencia hay que transformar a matriz binaria con el comando siguiente

Hide

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

Graficar en un primer momento aunque sera muy sucia

Hide

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

Exportar resultados del relacionamiento no binario

Hide

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

R Notebook

[Code ▼](#)

Laboratorio 10- T1_04_REDES COMPLEJAS(1) - Vision de redes complejas - Parte 1

Objetivo: Estimar el Maximum Spanning Tree -arbol de expansion maxima- (asegurar una vision clara del espacio-producto) Red troncal:

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

Regla 1: mantener n-1 conexiones como maximo Regla 2: Quitar las conexiones con el peso mas bajo, nos vamos quedar con las del peso maximo (menos conexiones) Regla 3: No crear nodos aislados

En este ejercicio vamos a: 1. Usar un matriz hipotetica de datos 2. Graficar sus proximos adyacentes 1)La visualizacion 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 obstruiran las conexiones mas relevantes.

Calculamos el arbol de expansion maxima (MST) de la matriz de proximidad. MST es el conjunto de enlaces que conecta todos los nodos de la red utilizando un numero minimo de conexiones y la suma maxima posible de proximidades.

Calculamos el MST usando el algoritmo de Kruskal: Basicamente, 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 definicion, el MST incluye todos los productos, pero el numero de enlaces es el minimo posible.

Despues de seleccionar los enlaces utilizando los criterios mencionados anteriormente, construimos una visualizacion utilizando un algoritmo de dise?o dirigido por la fuerza.

Primero crear una matriz aleatoria de 200*200

[Hide](#)

```
M <- matrix(runif(200*200, min=0, max=100), ncol=200)
diag(M)<-0
head (M[,1:6])
dim (M)
```

Llamar la libreria para graficar la matriz anterior

[Hide](#)

```
library (igraph)
```

Comando para graficar la matriz anterior, obteniendo un grafico saturado por lo cual se usa el MST (igraph calcula el arbol de expansion minima)

[Hide](#)

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

Transformamos la matriz en NEGATIVA para identificar los maximos

[Hide](#)

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

Grafica nueva red con la matriz negativa, ver el espacio-producto, tecnologico

[Hide](#)

```
g <- graph.adjacency(M, mode="undirected", weighted =TRUE)
MST <- minimum.spanning.tree(g)
plot(MST, vertex.shape="none", vertex.label.cex=.7)
```

Matriz de adyacencias (nuevo relacionamiento).Ver las aristas que hay entre los nodos

[Hide](#)

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

Ver las aristas (pesos)

[Hide](#)

```
head(A)
```

Exportar el grafico en gml

[Hide](#)

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

Exportar resultados para cytoscape o gephi

[Hide](#)

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

R Notebook

Code ▼

Laboratorio 11: Vision clara del espacio-producto: CASO REAL_exportaciones hidalguenses

Objetivo: Estimar el Maximum Spanning Tree -arbol de expansion maxima- (asegurar una vision clara del espacio-producto) Red troncal:

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

Regla 1: Mantener n-1 conexiones como maximo Regla 2: Quitar las conexiones con el peso mas bajo, nos vamos quedar con las del peso maximo (menos conexiones) Regla 3: No crear nodos aislados

1.Como crear una vision clara del espacio-producto: arbol de expansion maxima (MST) Paquete: Balland, P.A. (2017) Economic Geography in R: Introduction to the EconGeo Package, Papers in Evolutionary Economic Geography, 17 (09): 1-75 Para instalar: <https://www.paballand.com/install-r> (<https://www.paballand.com/install-r>)

Cargar libreria de EconGeo

Hide

```
library(EconGeo)
```

Ubicar archivo a utilizar, al ejecutar señala la ruta la cual debe ser copiada y pegada despues de read.csv

Hide

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

Comandos para visualizar una parte de la matriz compleja cargada con anterioridad

Hide

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

Realizar un primer grafico de la matriz, cargar libreria para ello

Hide

```
library (igraph)
```

Comandos para generar grafico denominado red_hidalgo1

Hide

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

Modificamos la matriz original a una matriz negativa

Hide

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

Graficamos la matriz negativa, grafica de adyacencias con un input la matriz invertida a la que se le calcula el MST

Hide

```
red_hidalgo2 <-graph.adjacency(M, mode="undirected", weighted=TRUE)  
MST <- minimum.spanning.tree(red_hidalgo2)  
plot (MST, vertex.shape= "none", vertex.label.cex=.7)
```

Exportar los datos del grafico limpio

Hide

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

Generar matriz de adyacencias

Hide

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

Exportar la matriz de adyacencias creada

Hide

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

R Notebook

[Code ▼](#)

Laboratorio 12- Video 05 - DENSIDAD DEL RELACIONAMIENTO: CASO HIPOTETICO

Objetivo: Estimar la densidad del relacionamiento (rd). La densidad varia entre 0 y 1, los valores mas grandes indican que la region ha avanzado a una VCR en muchos bienes proximos al bien i y tiene mayor probabilidad de exportarlo en el futuro mayor densidad, mayor cercania. Menor densidad, mayor lejania.

En este laboratorio vamos a: 1. Cargar una matriz hipotetica de datos 2. Estimar la VCR de la matriz hipotetica 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 cercania 5. Predecir la entrada de una nueva industria

Econometria usar count(densidad del relacionamiento) como regresora (predictora) de la entrada de una nueva industria

Primero cargar la libreria de EconGeo

[Hide](#)

```
library (EconGeo)
```

Como se mide la densidad del relacionamiento (rd)? Paquete: Balland, P.A. (2017) Economic Geography in R: Introduction to the EconGeo Package, Papers in Evolutionary Economic Geography, 17 (09): 1-75 Para instalar: <https://www.paballand.com/install-r> (<https://www.paballand.com/install-r>)

Preguntar que es la Ventaja Comparativa Revelada (VCR)

[Hide](#)

```
?RCA
```

En la consola Help despues de Examples copiar y pegar el ejemplo para generate a region - industry matrix

[Hide](#)

```
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")
```

Con los comandos anteriores se genera una matriz llamada mat. Visualizar matriz creada que tiene regiones e industrias con una dimensión de 5X4

Hide

```
mat
dim(mat)
```

Para sacar la VCR utilizar el siguiente comando

Hide

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

Visualizar matriz binarizada

Hide

```
mat
```

Recordar que la VCR es basicamente es un indicador de especializacion, produce mas de lo esperado basado en el tamaño de la region y de la industria. A partir de la matriz calcular las co-ocurrencias de la matriz transpuesta y visualizarla

Hide

```
c = co.occurrence(t(mat))
c
```

Calcular el relacionamiento con base en las co-ocurrencias detectadas con anterioridad y visualizar resultado

Hide

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

Con base al resultado aplicar la regla binaria y visualizar resultado

Hide

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

Una vez calculado el relacionamiento calcular la densidad del relacionamiento (en relacion a distancia- mas cercano o mas lejano).

Asegurate que hayas computado la matriz binaria de VCR antes de computar el Relacionamiento-densidad. Usar la matriz de VCR (mat) y la de espacio-producto (r) para calcular la densidad de relacionamiento. Visualizar ambas previamente.

Hide

```
mat
r
```

Con las dos matrices previas, calcular la densidad del relacionamiento y visualizar resultado

Hide

```
rd = relatedness.density(mat,r)
rd
```

Rango 0-100 La Industria 1 (I1) esta relacionada con la Industria 2 y 3 (I2, I3) (I1)R = 2 (I2;I3) Y la Region 1 (R1) ya tiene la Industria 2 (I2), pero no tiene la Industria 3 (I3) Hay 2 industrias relacionadas con la Industria 1, por eso 2 Pero la Region 1 solo tiene 1 de esas industrias relacionadas Entonces $1/2=0.5$

Convertir la matriz a una lista y visualizar resultado

Hide

```
rd = get.list(rd)
rd
```

En CE se tiene la hipotesis que la densidad del relacionamiento es una variable explicativa para la entrada o no hacia una industria La entrada de una industria depende de la densidad del relacionamiento Por último vamos a predecir una entrada para ello ejecutar siguiente comando

Hide

```
?entry.list
```


Mira la seccion de ejemplos y copia-pegas los datos a la consola del primer grupo de comandos El ejemplo genera una matriz 1. Visualizar resultado generate a first region - industry matrix in which cells represent the presence/absence of a RCA (period 1)

Hide

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

mat1
```

Para predecir entrada se requieren al menos 2 momentos o tiempos por lo cual es necesario otra matriz

Copiar y pegar de los ejemplos el segundo grupo de comando para crear segunda matriz. Visualizar resultado generate a second region - industry matrix in which cells represent the presence/absence of a RCA (period 2)

Hide

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

Evidenciar las entradas con el siguiente comando y visualizar resultado

Hide

```
d = entry.list(mat1, mat2)
d
```

Lo anterior permite ver las posibilidades de entrada 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 entro 0 puede tener potencial de entrada, pero no ha entrado

Hacer una combinacion de bases: hacer un match entre Region e Industria de las bases densidad del relacionamiento y la tabla region, industria, entrada y periodo

Visualizar resultado

Hide

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

Fijarse muy bien antes de hacer el merge de que ambas matrices coincidan en cuanto a la estructura

Ambas matrices deben estar en formato de lista para que coincidan las columnas a unir

Con estos resultado podemos hacer econometria incipiente como una regresion con el siguiente comando lm para referir a un modelo lineal con variable dependiente Entry e independiente Count

La entrada dependera de la densidad del relacionamiento

Hide

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

R Notebook

Code ▼

Laboratorio 13- Gráficos en R con ggplot2

Instalar primero las paqueterías necesarias `install.packages("tidyverse")` `install.packages("readr")` `install.packages("ggplot2")`

Llamar a la libreria de ggplot2

Hide

```
library(ggplot2)
```

Cargar las BD desde Files / Upload Leer los datos en la nube

Hide

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

Para visualizar los datos ejecutar el siguiente comando

Hide

```
names(green_data)
```

Generar primer grafico

Hide

```
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 = TRUE)
```

Visualizar p1

Hide

p1

Generar un segundo grafico reciclando el codigo anterior

Hide

```
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 = TRUE)
```

Visualizar p2

Hide

p2

Generar un tercer grafico reciclando el codigo anterior

Hide

```
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 = TRUE)
```

Visualizar p3

Hide

p3

Para conocer la correlacion entre las variables primero instalar el siguiente paquete `install.packages("psych")`

Cargar posteriormente la libreria

Hide

```
library(psych)
```

Luego cargar la libreria de readr y posteriormente crear un dataframe denominado correl

Hide

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

Visualizar la tabla anterior

Hide

```
head(correl)
```

Para correlacionar de forma parida las variables de la tabla generada con anterioridad ejecutar el siguiente comando

Hide

```
attach(correl)
names(correl)
pairs(correl)
pairs.panels(correl)
```

Generar otra correlacion ahora con el metodo pearson con las medidas de complejidad

Hide

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

Visualizar resultado

Hide

```
complex_corr
```

Redondear el coeficiente de correlacion y visualizar resultado

Hide

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

Para generar un mapa de calor instalar primero paquete ggcorrplot `install.packages("ggcorrplot")`

Llamar las siguientes librerias

Hide

```
library(ggcorrplot)
library(ggplot2)
```

Generar un grafico mas

Hide

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

Visualizar resultado

Hide

p4

Para generar un conjunto de graficos en una sola cuadrícula instalar el siguiente paquete `install.packages("ggpubr")`

Mandar llamar al paquete

Hide

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

Para arreglar la visualización en un grid de 2X2 instalar el siguiente paquete y llamarlo

`install.packages("gridExtra")`

Hide

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

R Notebook

Code ▾

Laboratorio 16 -Análisis de redes en R- Indicadores básicos globales

Cargar librería de EconGeo

Hide

```
library(EconGeo)
```

Cargar data de las interacciones de los personajes de los miserables

Hide

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

Ver el encabezado

Hide

```
head(EL)
```

Transformar dataframe en matriz

Hide

```
MM <- get.matrix(EL)
```

Visualizar matriz (la red es no dirigida porque si la doblara sería lo mismo) El ejemplo para distinguir entre matrices dirigidas y no dirigidas Twitter sigue a alguien pero ese alguien no me sigue (unidireccional) Facebook manda solicitud amistad y al aceptarme nos hacemos amigos (bidireccional)

Hide

```
head(MM)
```

Llamar librería para graficar

Hide


```
library(igraph)
```

Crear objeto g para crear grafico proveniente del dataframe

Hide

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

Visualizar objeto grafico

Hide

```
plot(g)
```

Eliminar el peso de la relacion

Hide

```
EL$Weight = NULL
```

Visualizar resultado

Hide

```
head(EL)
```

Cargar libreria para visualizaciones

Hide

```
library(networkD3)
```

Generar grafico con networkD3. Este tipo de graficos pueden guardar como una pagina web

Hide

```
simpleNetwork(EL)
```

Code ▼

Hide

Hide

Hide

Hide

Hide

Manipulando número de columnas

Hide

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

Manipulando número de filas

Hide

```
p + facet_wrap(~ Region, nrow = 3)
```

R Notebook

Code ▼

Laboratorio 25- Tidy data- Datos ordenados- Parte 1

Objetivo: Introducción práctica a los datos ordenados (tidy data) y a las herramientas que provee el paquete tidyr

En este ejercicio vamos a: 1. Cargar datos (tibbles) 2. Convertir nuestros tibbles en dataframe (para su exportacion) 3. Exportar dataframes originales 4. Pivotar tabla 4a 5. Exportar resultados (tabla pivotante)

Pre-requisitos: instalar paqueteria `install.packages("tidyverse")`

Instalar paquete de datos `install.packages("remotes") remotes::install_github("cienciadedatos/datos") install.packages("datos")`

Llamar la libreria de tidyverse

Hide

```
library("tidyverse")
```

Llamar libreria de datos

Hide

```
library("datos")
```

Visualizar las tablas a utilizar (tabla1 a tabla4b). Ver datos como tibble

Hide

```
datos::tabla1  
datos::tabla2  
datos::tabla3  
datos::tabla4a  
datos::tabla4b
```

Ver datos como dataframe

Hide

```
df1 <- data_frame(tabla1)
df2 <- data_frame(tabla2)
df3 <- data_frame(tabla3)
df4a <- data_frame(tabla4a)
df4b <- data_frame(tabla4b)
```

Visualizar encabezados dataframe

Hide

```
head(df1)
head(df2)
head(df3)
head(df4a)
head(df4b)
```

Exportar los dataframe originales

Hide

```
write.csv(df1, file = "df1.csv")
write.csv(df1, file = "df2.csv")
write.csv(df1, file = "df3.csv")
write.csv(df1, file = "df4a.csv")
write.csv(df1, file = "df4b.csv")
```

Explicación de que es tibble

Hide

```
vignette("tibble")
```

La mayoría de las funciones que usaras en este libro producen tibbles, ya que son una de las características transversales de tidyverse. Si ya estas familiarizado con `data.frame()`, es importante que tomes en cuenta que `tibble()` hace menos cosas como nunca cambia el tipo de los inputs (por ejemplo nunca convierte caracteres en factores), nunca cambia el nombre de las variables y nunca asigna nombres a las filas

Empezar a ordenar datos con la tabla4a (pivotar)

Hide

```
t4a_PIVOTANTE = tabla4a %>%  
  pivot_longer(cols = c ("1999", "2000"), names_to = "anio", values_to = "casos")
```

Exportar resultado: tablas ordenadas

Hide

```
write.csv(t4a_PIVOTANTE, file = "t4a_PIVOTANTE.csv")
```