

# Capítulo 2. Introducción al R

Javier Rodríguez Barrios

## 2. Generalidades de R y RStudio

### Introducción

#### 2.1 Programa R y sus ventajas en el análisis de datos.

R es un lenguaje y un entorno estadístico, desarrollado en 1995 por Ross Ihaka y Robert Gentleman - Universidad de Auckland, Nueva Zelanda (Team, n.d.). R se puede ejecutar en diferentes sistemas operativos como Windows, Mac y Linux. Una de las principales ventajas del trabajo en el programa R, es su versatilidad en el desarrollo de figuras básicas y avanzadas, para el análisis de una o más variables, de igual forma permite desarrollar procedimientos univariados y multivariados de una forma eficiente. R se puede descargar de una nube compuesta de servidores en diferentes lugares del mundo: <https://cloud.r-project.org>. Una vez al año se genera una versión principal de R y dos o tres versiones menores, por lo que es recomendable realizar las actualizaciones.

Las ventajas que ofrece R se resumen de la siguiente manera: Es un programa gratuito, de código abierto, compatible con diferentes plataformas (Windows, Linus o MAC OSX). Presenta una gran variedad de pruebas estadísticas, permite reproducir procedimientos de forma sencilla, mediante la generación de archivos que contienen las rutinas, procedimientos o secuencias de comandos. Cuenta con una comunidad a nivel mundial, que constantemente realiza aportes para ofrecer más opciones de procedimientos. De esta manera, los usuarios tienen la cultura de compartir sus códigos, sus experiencias y apoyarse entre sí. Cuenta con miles de aplicaciones, llamadas paquetes o librerías, que cuentan con documentación y permiten realizar múltiples procedimientos estadísticos, entre otros.

La instalación de R para Windows se realiza con el siguiente procedimiento:

1. Descargar la versión más actual de R en <https://cran.r-project.org/bin/windows/base/>
2. Haga doble clic en el archivo que acaba de descargar para instalar R
3. Hacer clic en Aceptar y dar “siguiente” en todos los procedimientos requeridos.

Para la versión de R, para MAC OSX, se realiza el siguiente procedimiento:

1. Ingresar a <https://cran.r-project.org/bin/macosx/>
2. Hacer doble clic en el archivo que acaba de descargar para instalar R
3. Hacer clic en Aceptar y dar “siguiente” en todos los procedimientos requeridos.

Este procedimiento se realiza para ir renovando de forma las versiones que periódicamente son publicadas, debido a que en algunos casos necesitaremos de las mejoras que estas presentan, o porque algunos complementos o paquetes que requerimos instalar no son compatibles con versiones antiguas. El procedimiento no se puede realizar en RStudio, para ello se requiere abrir la consola de R (normalmente hay un acceso directo en el escritorio del computador). El procedimiento se describe a continuación:

1. Es necesario haber instalado el paquete `installr` y cargarlo con `library(installr)`.
2. Posteriormente aplicamos el comando `updateR ()`. Se realizarán una serie de preguntas orientadas a la actualización del R sin necesidad de desinstalar la versión actual. Recomendando aceptar las opciones que se presenten.

El comando `updateR ()` permite realizar lo siguiente: encontrar la última versión R, descargarla, ejecutar su instalador, eliminar el archivo de instalación de la versión anterior, copiar y actualizar paquetes antiguos a la nueva instalación R.

Para más información, se puede ingresar al tutorial del paquete `installr` desde: <https://cran.r-project.org/web/packages/installr/index.html>

## 2.2 Programa RStudio.

RStudio es un programa de entorno integrado de desarrollo (IDE por sus siglas en inglés). En términos generales, este programa genera un entorno amigable, para desarrollar y ejecutar comandos de R, de una forma eficiente. Cuenta con herramientas para revisar los códigos realizados, ejecutar figuras, entre otros procedimientos requeridos por los analistas de datos. Las ventajas que ofrece RStudio, se pueden resumir en: Es un IDE de código abierto y por ello se actualiza con relativa frecuencia, su entorno es más amigable y completo que el de la plataforma de R. También permite compartir los códigos realizados entre comunidades relacionadas. Esta plataforma se actualiza un par de veces al año.

La versión que se instala en el computador, para sistemas de Windows, MAC y Linux, se consigue en: <https://www.rstudio.com/products/rstudio/download/>

Adicional a la versión instalada, se cuenta con otra opción en línea, a la que se puede acceder por este enlace: <https://rstudio.cloud/content/>

En caso de requerir actualizar la versión de R, es necesario actualizar también a la versión de RStudio, para que haya compatibilidad, de lo contrario, no se podrá realizar la interfaz entre estos programas.

## 2.3 Tutoriales de R Y RStudio

### 2.3.1 Tutorial rápido de R.

Esta es una guía que permite obtener información sobre las diferentes utilidades del R, tanto en su instalación, componentes y el procesamiento numérico y gráfico de los datos (Figura 1). <https://www.statmethods.net/>



Figure 1: Imagen de la plataforma gráfica con comandos de R

### 2.3.2 Motor de búsqueda de R (Rseek).

Corresponde a una plataforma similar a la de google, en la que la comunidad que hace parte de la plataforma R, dispone de su información, para que sea de acceso gratuito (Figura 2). Se puede acceder a ella en el siguiente enlace: <https://rseek.org/>

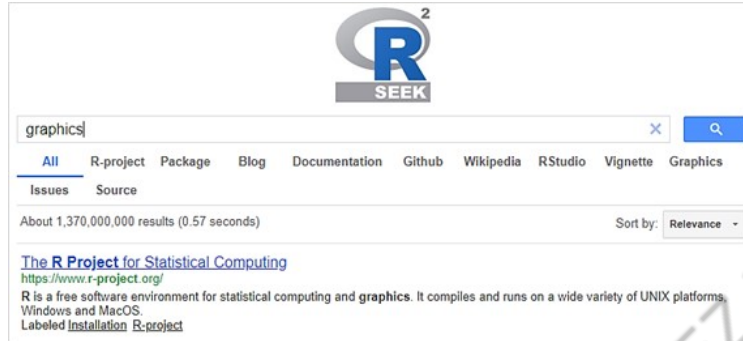


Figure 2: Imagen de la plataforma gráfica con comandos de R

**2.3.3 Otros sitios de utilidad** Sitio web de R, en el cual pueden descargarse las versiones más recientes de R. <http://www.r-project.org>

<http://www.cran.r-project.org>

Enlace sobre opciones gráficas, manuales de soporte y paquetes disponibles en R <http://search.r-project.org/>

Sitio web de R, en el cual se visualizan los tópicos generales de análisis que pueden realizarse. <http://www.cran.r-project.org/web/views>

Se cuenta con un espacio para realizar consultas sobre diferentes temas de análisis en R. <https://es.stackoverflow.com/questions/tagged/r> <http://stats.stackexchange.com/questions/tagged/r>

Se enumeran paquetes relacionados con análisis y modelos gráficos, son alrededor de 30 paquetes. <http://cran.r-project.org/web/views/gR.html>

Enlace sobre respuestas a preguntas sobre temas de programación en R <http://stackoverflow.com/questions/tagged/r>

R-bloggers: espacio informativo con más de 500 bloggers que proporcionan noticias y tutoriales sobre R. <https://www.r-bloggers.com/about/>

Hojas de trucos de RStudio, Cuenta con diferentes guías rápidas para ejecutar procedimientos de ejecución y de programación en R. <https://www.rstudio.com/resources/cheatsheets/>

En estos enlaces se puede visualizar a algunos de los trucos de R, como: Introducción al R, Introducción al RStudio, Visualización de datos en ggplot2, entre otros.

Actualmente se cuenta, con un conjunto de paquetes de R, denominados “Tidyverse”, diseñados para apoyar el proceso de importar, transformar, visualizar, modelar y comunicar la información en procesos de “ciencia de datos”. Han tomado mucho auge y son los que dominan a los procedimientos de análisis de datos en diferentes ámbitos. <https://soka.gitlab.io/blog/post/2019-10-06-r-tidyverse-intro/>

**2.3.4 Opciones gráficas** El siguiente demo se ejecuta en R, permite visualizar opciones gráficas (Figura 3). `demo(image)`

La siguiente demostración, permite visualizar diferentes opciones de figuras tridimensionales, incluidos los modelos de elevación digital de terreno en un volcán (izquierda) (Figura 4). `demo(persp)`

A continuación, se relacionan otras demostraciones gráficas ofrecidas por R. `example(contour)`

`demo(graphics)`

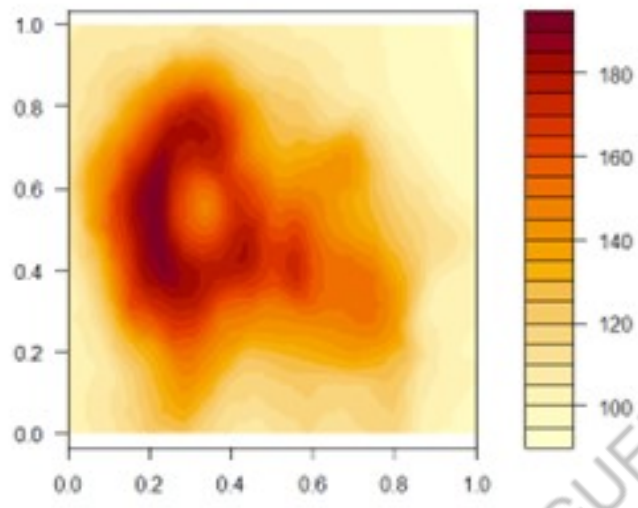


Figure 3: Representación de curvas de nivel en un mapa realizado con R.

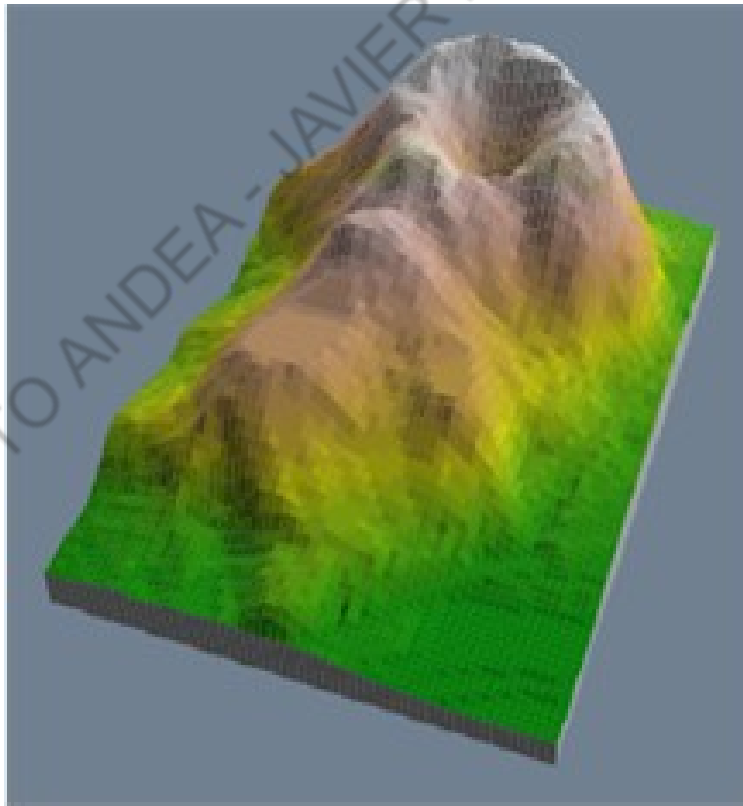


Figure 4: Representación tridimensional en R.

```
demo(plotmath)
```

```
demo(Hershey)
```

Las siguientes demostraciones gráficas, requieren del paquete gráfico “lattice”. `require("lattice")`

```
demo(lattice)
```

```
example(wireframe)
```

Las siguientes demostraciones gráficas, requieren del paquete gráfico “rgl”. `require("rgl")`

```
demo(rgl)
```

```
example(persp3d)
```

PROYECTO ANDEA - JAVIER RODRÍGUEZ BARRIOS

## Ejemplo 1. NOCIONES DE RSTUDIO

A continuación, se describen algunos conceptos generales, relacionados al entorno de comandos de R, los cuales consisten en operaciones y funciones aritméticas básicas, así como en la creación de bases de datos (dataframes) y subconjuntos de datos.

```
#-----  
# 1) Operaciones aritméticas básicas  
  
# Suma  
5 + 7
```

```
## [1] 12
```

```
# Resta  
5 - 3
```

```
## [1] 2
```

```
# Multiplicación  
5 * 7
```

```
## [1] 35
```

```
# División  
5/3
```

```
## [1] 1.666667
```

```
# Exponentes  
2^3
```

```
## [1] 8
```

```
# Devolver el resto de la división de 5/3  
5 %% 3
```

```
## [1] 2
```

```
# Logaritmos y exponenciales :  
x = 5/3  
log2(x)      # Logaritmo en base 2 de x
```

```
## [1] 0.7369656
```

```
log10(x)     # Logaritmo en base 10 de x
```

```
## [1] 0.2218487
```

```
exp(x)      # Exponencial de x
```

```
## [1] 5.29449
```

```
# Funciones trigonométricas :  
cos(x)      # Coseno de x
```

```
## [1] -0.09572355
```

```
sin(x)      # Seno de x
```

```
## [1] 0.995408
```

```
tan(x)      # Tangente de x
```

```
## [1] -10.39878
```

```
acos(x/100) # Arcoseno de x
```

```
## [1] 1.554129
```

```
asin(x/100) # Arcseno de x
```

```
## [1] 0.01666744
```

```
atan(x/100) # Arcotangente de x
```

```
## [1] 0.01666512
```

```
abs(x)      # Valor absoluto de x
```

```
## [1] 1.666667
```

```
sqrt(x)     # Raíz cuadrada de x
```

```
## [1] 1.290994
```

```
#-----  
# 2) Asignación de valores a objetos o a variables  
sitios <- 2      # Número de sitios = 2  
sitios = 2      # Otra forma  
n.sitios <- "dos" # Número de sitios como un caracter  
dos.sitios <- TRUE # Objeto lógico  
  
# Imprimir los valores asignados  
sitios          # Imprime el número de sitios
```

```
## [1] 2
```

```

print(sitios)  # Otra forma

## [1] 2

n.sitios      # Sitios asignados como caracter

## [1] "dos"

dos.sitios    # Sentencia lógica

## [1] TRUE

# Identificar el tipo de asignación
class(sitios)  # Tipología de los sitios

## [1] "numeric"

class(n.sitios)  # Tipología del número de sitios

## [1] "character"

class(dos.sitios)  # Tipología de la sentencia lógica

## [1] "logical"

is.numeric(sitios)  # Identificador: is.numeric(), is.character(), is.logical()

## [1] TRUE

as.character(sitios)  # Cambiar: as.numeric(), as.character(), as.logical()

## [1] "2"

# Operaciones con las asignaciones
3 * sitios  # Multiplicar 3 por el número de sitios

## [1] 6

# Área de un rectángulo
altura <- 20      # Altura del rectángulo
ancho <- 10       # Ancho
área <- altura*ancho  # Cálculo de área
área             # Imprimir el área

## [1] 200

```



```
print(área)      # Otra forma de imprimir el área
```

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

```
ls()             # Permite visualizar a los objetos creados
```

```
## [1] "altura"      "ancho"      "área"      "dos.sitios" "n.sitios"
## [6] "sitios"       "x"
```

```
#-----
```

```
# 3) Vectores
```

```
sitios <- c(2, 3, 2, 3)  # Vector sitios
sitios      # Imprimir el vector
```

```
## [1] 2 3 2 3
```

```
sitios <- c("dos", "tres", "dos", "dos")  # Vector como caracter
sitios
```

```
## [1] "dos" "tres" "dos" "dos"
```

```
abundancia <- c(TRUE, FALSE, TRUE, TRUE)  # vector con elementos lógicos
abundancia
```

```
## [1] TRUE FALSE TRUE TRUE
```

```
sitios <- c(2, 3, 2, 3)  # Vector sitios
names(sitios) <- c("dos", "tres", "dos", "dos")  # Nombres de los elementos del vector sitios
sitios <- c(dos= 2, tres= 3, dos= 2, dos= 2)  # Otra forma
sitios
```

```
## dos tres dos dos
##    2    3    2    2
```

```
length(sitios)  # Elementos del vector sitios
```

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

```
sitios[2]        # Segundo elemento del vector sitios
```

```
## tres
##    3
```

```
sitios[c(1,4)]   # Primer y cuarto elemento del vector
```

```
## dos dos
##    2    2
```

```
sitios [1:3]      # Tres primeros elementos del vector sitios
```

```
## dos tres dos  
## 2 3 2
```

```
sitios [-1]      # Eliminar el primer elemento del vector
```

```
## tres dos dos  
## 3 2 2
```

```
sitios[abundancia == TRUE] # Abundancias con valor verdadero
```

```
## dos dos dos  
## 2 2 2
```

```
sitios[sitios >= 2] # Sitios con abundancias con valores mayor o igual a 2
```

```
## dos tres dos dos  
## 2 3 2 2
```

```
sitios[sitios != 3] # Sitios con abundancias diferentes a 3
```

```
## dos dos dos  
## 2 2 2
```

```
#-----
```

```
# 4) Operaciones con escalares y vectores
```

```
sitios.1 <- c(A = 2, B = 3, C = 2, D = 3) # Vector sitios en muestreo 1  
sitios.1
```

```
## A B C D  
## 2 3 2 3
```

```
sitios.2 <- c(A = 4, B = 6, C = 4, D = 5) # Vector sitios en muestreo 2  
sitios.2
```

```
## A B C D  
## 4 6 4 5
```

```
sitios * 3      # Producto de los sitios x 2
```

```
## dos tres dos dos  
## 6 9 6 6
```

```
sitios.1 * sitios.2 # Producto de los sitios 1 y 2
```

```
## A B C D  
## 8 18 8 15
```

```

sitios.1 <- c(2, 3, 2, 3)  # Vector sitios en muestreo 1
sitios.1

## [1] 2 3 2 3

max(sitios.1)             # Valor máximo

## [1] 3

min(sitios.1)             # Valor mínimo

## [1] 2

range(sitios.1)           # Rango de los datos del vector

## [1] 2 3

sum(sitios.1)             # Suma de los datos del vector

## [1] 10

prod(sitios.1)            # Producto de los datos del vector

## [1] 36

mean(sitios.1)            # Promedio de los datos del vector

## [1] 2.5

sd(sitios.1)              # Desviación estándar

## [1] 0.5773503

var(sitios.1)             # Varianza de los datos del vector

## [1] 0.3333333

sort(sitios.1)            # Organizar elementos de forma ascendente

## [1] 2 2 3 3

#-----
# 5) Matrices
Matriz <- matrix(1:15, nrow = 3, ncol = 5)  # nrow y nrow: Número de filas y columnas
Matriz

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    4    7   10   13
## [2,]    2    5    8   11   14
## [3,]    3    6    9   12   15

```

```
Matriz <- matrix(1:15, nrow = 5, ncol = 3)    # 1:15: Secuencia de valores de 1 a 15
Matriz
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

```
Matriz <- matrix(c(1:15),5,3, byrow= FALSE)  # 5,3: Número de filas y columnas
Matriz
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

```
# Primera opción para crear una Matriz
Matriz <- matrix(c(5, 6, 7, 8, 9,      # Elementos de la Matriz
                  2, 4, 5, 9, 8,
                  7, 3, 4, 8, 7),      # 15 elementos
                5,3, byrow=F)         # 5 columnas y 3 filas
Matriz
```

```
##      [,1] [,2] [,3]
## [1,]    5    2    7
## [2,]    6    4    3
## [3,]    7    5    4
## [4,]    8    9    8
## [5,]    9    8    7
```

```
# Otra forma con las funciones cbind() o rbind()
col1 <- c(5, 6, 7, 8, 9)      # Col 1 a 3, son las tres columnas de la Matriz
col2 <- c(2, 4, 5, 9, 8)
col3 <- c(7, 3, 4, 8, 7)
Matriz <- cbind(col1, col2, col3)      # Nombres de las columnas
rownames(Matriz) <- c("A","B","C","D","E") # Nombres de las filas
Matriz
```

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
# Otra forma con comandos "colnames" y "dimnames"
Matriz <- matrix(
  data = c(5, 6, 7, 8, 9,      # Elementos de la Matriz
           2, 4, 5, 9, 8,
           7, 3, 4, 8, 7),
  nrow = 5, byrow = FALSE,      # nrow= columnas.
  dimnames = list(c("A","B","C","D","E"), # dimnames= lista con vector fila y columna
                  c("col1", "col2", "col3"))
Matriz
```

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
# Otra forma con comandos "colnames" y "dimnames"
nombres <- LETTERS[1:nrow(Matriz)]      # Letras para las filas
colnames(Matriz) <- c("col1","col2","col3") # Nombres de las columnas
dimnames(Matriz)=(list(nombres,colnames(Matriz))) # Elaboración de la Matriz
Matriz
```

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
# Algunos comandos para las matrices
t(Matriz)      # Transpuesta de la Matriz
```

```
##      A B C D E
## col1 5 6 7 8 9
## col2 2 4 5 9 8
## col3 7 3 4 8 7
```

```
ncol(Matriz)      # Número de columnas
```

```
## [1] 3
```

```
nrow(Matriz)      # Número de filas
```

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

```
dim(Matriz)      # Dimensión: Filas y columnas
```

```
## [1] 5 3
```

```
Matriz[2,]          # Ver la segunda fila (coma a la derecha del dato)
```

```
## col1 col2 col3  
##    6    4    3
```

```
Matriz[,2]          # Ver la segunda columna (coma a la izquierda del dato)
```

```
## A B C D E  
## 2 4 5 9 8
```

```
Matriz[2:4,]        # Filas 2 a la 4
```

```
##    col1 col2 col3  
## B     6    4    3  
## C     7    5    4  
## D     8    9    8
```

```
Matriz[c(2,4),]     # Filas 2 y 4
```

```
##    col1 col2 col3  
## B     6    4    3  
## D     8    9    8
```

```
Matriz[, "col1"]     # Ver la columna 1
```

```
## A B C D E  
## 5 6 7 8 9
```

```
Matriz[3, "col1"]    # Seleccionar el valor de la 3a fila y columna 1
```

```
## [1] 7
```

```
Matriz[3, 1]         # Otra forma para hacer lo anterior
```

```
## [1] 7
```

```
Matriz[3, 1:3]       # Valores de la fila 3 y de las columnas 1:3
```

```
## col1 col2 col3  
##    7    5    4
```

```
Matriz[-3,]          # Excluye a la 3a fila
```

```
##    col1 col2 col3  
## A     5    2    7  
## B     6    4    3  
## D     8    9    8  
## E     9    8    7
```

```
#-----
```

```
# 6) Operaciones matriciales
```

```
Matriz * 3 # Multiplicar por 3
```

```
##   col1 col2 col3
## A   15   6  21
## B   18  12   9
## C   21  15  12
## D   24  27  24
## E   27  24  21
```

```
log10(Matriz) # Logaritmo en base 10
```

```
##           col1           col2           col3
## A 0.6989700 0.3010300 0.8450980
## B 0.7781513 0.6020600 0.4771213
## C 0.8450980 0.6989700 0.6020600
## D 0.9030900 0.9542425 0.9030900
## E 0.9542425 0.9030900 0.8450980
```

```
round(log10(Matriz),2) # Redondear valores a 2 decimales
```

```
##   col1 col2 col3
## A 0.70 0.30 0.85
## B 0.78 0.60 0.48
## C 0.85 0.70 0.60
## D 0.90 0.95 0.90
## E 0.95 0.90 0.85
```

```
sqrt(Matriz) # Raíz cuadrada
```

```
##           col1           col2           col3
## A 2.236068 1.414214 2.645751
## B 2.449490 2.000000 1.732051
## C 2.645751 2.236068 2.000000
## D 2.828427 3.000000 2.828427
## E 3.000000 2.828427 2.645751
```

```
rowSums(Matriz) # Sumas de filas
```

```
##   A  B  C  D  E
## 14 13 16 25 24
```

```
colSums(Matriz) # Sumas de columnas
```

```
## col1 col2 col3
##   35  28  29
```

```
colMeans(Matriz)      # Promedios de columnas
```

```
## col1 col2 col3  
## 7.0  5.6  5.8
```

```
rowMeans(Matriz)      # Promedios de filas
```

```
##      A      B      C      D      E  
## 4.666667 4.333333 5.333333 8.333333 8.000000
```

```
apply(Matriz, 1, mean) # Promedios de filas
```

```
##      A      B      C      D      E  
## 4.666667 4.333333 5.333333 8.333333 8.000000
```

```
apply(Matriz, 2, mean) # Promedios de columnas
```

```
## col1 col2 col3  
## 7.0  5.6  5.8
```

```
summary(Matriz)      # Resumen estadístico de las columnas
```

```
##      col1      col2      col3  
## Min.   :5   Min.   :2.0   Min.   :3.0  
## 1st Qu.:6   1st Qu.:4.0   1st Qu.:4.0  
## Median :7   Median :5.0   Median :7.0  
## Mean   :7   Mean   :5.6   Mean   :5.8  
## 3rd Qu.:8   3rd Qu.:8.0   3rd Qu.:7.0  
## Max.   :9   Max.   :9.0   Max.   :8.0
```

```
var(Matriz)          # Matriz de varianza covarianza
```

```
##      col1 col2 col3  
## col1 2.50 4.25 1.25  
## col2 4.25 8.30 2.90  
## col3 1.25 2.90 4.70
```

```
cor(Matriz)          # Matriz de correlación
```

```
##      col1      col2      col3  
## col1 1.0000000 0.9329962 0.3646625  
## col2 0.9329962 1.0000000 0.4643121  
## col3 0.3646625 0.4643121 1.0000000
```

```
solve(var(Matriz))    # Inversa de la Matriz de varianza covarianza
```

```
##      col1      col2      col3  
## col1 3.2380952 -1.7301587 0.2063492  
## col2 -1.7301587 1.0780423 -0.2050265  
## col3 0.2063492 -0.2050265 0.2843915
```



```
eigen(var(Matriz))      # Valores y vectores propios
```

```
## eigen() decomposition
## $values
## [1] 11.9189751  3.3439221  0.2371028
##
## $vectors
##      [,1]      [,2]      [,3]
## [1,] 0.4205324 -0.2518028  0.87163514
## [2,] 0.8143607 -0.3187584 -0.48498429
## [3,] 0.3999614  0.9137770  0.07101004
```

```
#-----
# 7) Crear base de datos - data frames
datos <- data.frame(
  "n" = 1:4,                      # Consecutivo
  "indiv." = c("a", "b", "c", "d"), # Individuos
  "sexo" = as.character(c("f", "f", "m", "m")), # Sexo
  "variable" = c(1.2, 3.4, 4.5, 5.6) # Valor de la variable
)
datos      # Impresión de los datos
```

```
##   n indiv. sexo variable
## 1 1      a    f      1.2
## 2 2      b    f      3.4
## 3 3      c    m      4.5
## 4 4      d    m      5.6
```

```
head(datos)      # Muestra los primeros datos
```

```
##   n indiv. sexo variable
## 1 1      a    f      1.2
## 2 2      b    f      3.4
## 3 3      c    m      4.5
## 4 4      d    m      5.6
```

```
dim(datos)      # Número de filas y de columnas
```

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

```
length(datos)   # Número de columnas
```

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

```
names(datos)    # Nombres de las columnas
```

```
## [1] "n"      "indiv." "sexo"   "variable"
```

```
str(datos)      # Estructura de la base de datos
```

```
## 'data.frame':  4 obs. of  4 variables:
## $ n      : int  1 2 3 4
## $ indiv.  : chr  "a" "b" "c" "d"
## $ sexo    : chr  "f" "f" "m" "m"
## $ variable: num  1.2 3.4 4.5 5.6
```

```
class(datos)    # Tipo de estructura
```

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

```
t(datos)        # Transpuesta de la base de datos
```

```
##      [,1] [,2] [,3] [,4]
## n      "1"  "2"  "3"  "4"
## indiv.  "a"  "b"  "c"  "d"
## sexo    "f"  "f"  "m"  "m"
## variable "1.2" "3.4" "4.5" "5.6"
```

```
is.data.frame(datos)      # Verifica si datos es un data frame
```

```
## [1] TRUE
```

```
is.data.frame(Matriz)     # Verifica si Matriz es un data frame
```

```
## [1] FALSE
```

```
Matriz
```

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
Matriz1 <- as.data.frame(Matriz) # Convertir la Matriz a un data frame
Matriz1
```

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
#-----
# 8) Subconjuntos del data frame (datos)
```

```
datos
```

```
##   n indiv. sexo variable
## 1 1      a    f      1.2
## 2 2      b    f      3.4
## 3 3      c    m      4.5
## 4 4      d    m      5.6
```

```
str(datos)      # Estructura de la base de datos
```

```
## 'data.frame':   4 obs. of  4 variables:
## $ n          : int  1 2 3 4
## $ indiv.     : chr  "a" "b" "c" "d"
## $ sexo       : chr  "f" "f" "m" "m"
## $ variable: num  1.2 3.4 4.5 5.6
```

```
names(datos)    # Nombres de las columnas
```

```
## [1] "n"          "indiv."      "sexo"        "variable"
```

```
datos$indiv.    # Elementos de la columna "indiv."
```

```
## [1] "a" "b" "c" "d"
```

```
datos[, 'indiv.'] # Otra forma de hacer lo anterior
```

```
## [1] "a" "b" "c" "d"
```

```
datos[, c(1, 3)] # Columnas 1 y 3
```

```
##   n sexo
## 1 1    f
## 2 2    f
## 3 3    m
## 4 4    m
```

```
datos[, -1]     # Excluir columna 1
```

```
##   indiv. sexo variable
## 1      a    f      1.2
## 2      b    f      3.4
## 3      c    m      4.5
## 4      d    m      5.6
```

```
datos$variable >= 4      # Datos con esta condición
```

```
## [1] FALSE FALSE TRUE TRUE
```

```
datos[datos$variable >= 4, ]      # Filas con esta condición
```

```
##  n indiv. sexo variable
## 3 3      c      m      4.5
## 4 4      d      m      5.6
```

```
datos[datos$variable >= 4, c("indiv.", "variable")]      # Variables con esa condición
```

```
##  indiv. variable
## 3      c      4.5
## 4      d      5.6
```

```
subset(datos, variable >= 4)      # Individuos con esa condición
```

```
##  n indiv. sexo variable
## 3 3      c      m      4.5
## 4 4      d      m      5.6
```

```
attach(datos)      # attach, permite llamar a las columnas solo dando sus nombres
datos[variable >= 4, ]      # Comando resumido
```

```
##  n indiv. sexo variable
## 3 3      c      m      4.5
## 4 4      d      m      5.6
```

```
detach(datos)      # Para finalizar lo anterior
```

```
#-----
```

```
# 9) Agregar columnas o filas al data frame (datos)
```

```
nivel <- factor(c("bajo", "bajo", "alto", "alto"))      # Factor generado
nivel
```

```
## [1] bajo bajo alto alto
## Levels: alto bajo
```

```
datos
```

```
##  n indiv. sexo variable
## 1 1      a      f      1.2
## 2 2      b      f      3.4
## 3 3      c      m      4.5
## 4 4      d      m      5.6
```

```
datos$grupo <- nivel          # nueva columna llamada "grupo"
datos
```

```
##  n indiv. sexo variable grupo
## 1 1      a    f      1.2 bajo
## 2 2      b    f      3.4 bajo
## 3 3      c    m      4.5 alto
## 4 4      d    m      5.6 alto
```

```
datos1 <- cbind(datos, group = nivel) # otra forma de incluir la nueva columna
datos1
```

```
##  n indiv. sexo variable grupo group
## 1 1      a    f      1.2 bajo bajo
## 2 2      b    f      3.4 bajo bajo
## 3 3      c    m      4.5 alto alto
## 4 4      d    m      5.6 alto alto
```

```
#-----
# 10) Data frame con valores aleatorios
Matriz          # Data frame para calcular sus promedios y desviaciones
```

```
##  col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

```
media = colMeans(Matriz)      # Promedios de las columnas
media
```

```
## col1 col2 col3
## 7.0  5.6  5.8
```

```
de= sqrt(var(Matriz))        # Desviaciones estándar de las columnas
de
```

```
##          col1      col2      col3
## col1 1.581139 2.061553 1.118034
## col2 2.061553 2.880972 1.702939
## col3 1.118034 1.702939 2.167948
```

```
# Generador de números aleatorios: rnorm(n, media, de)
# *Cada corrida es independiente de la anterior
col1 <- abs(round(rnorm(10, 7.0, 1.6),3)) # Números aleatorios de la columna 1
col1
```

```
## [1] 6.572 5.865 6.233 8.351 7.820 6.756 5.679 6.775 8.331 7.097
```

```
col2 <- abs(round(rnorm(10, 5.6, 2.8),3)) # Números aleatorios de la columna 2
col2
```

```
## [1] 6.437 4.556 3.655 2.576 0.716 2.347 6.181 9.337 10.136 7.120
```

```
col3 <- abs(round(rnorm(10, 5.8, 2.2),3)) # Números aleatorios de la columna 3
col3
```

```
## [1] 6.244 5.126 4.819 5.823 6.052 3.875 4.598 1.166 9.327 10.022
```

```
Matriz2 <- cbind(col1, col2, col3) # Fusión de las columnas
Matriz2
```

```
##      col1  col2  col3
## [1,] 6.572  6.437  6.244
## [2,] 5.865  4.556  5.126
## [3,] 6.233  3.655  4.819
## [4,] 8.351  2.576  5.823
## [5,] 7.820  0.716  6.052
## [6,] 6.756  2.347  3.875
## [7,] 5.679  6.181  4.598
## [8,] 6.775  9.337  1.166
## [9,] 8.331 10.136  9.327
## [10,] 7.097  7.120 10.022
```

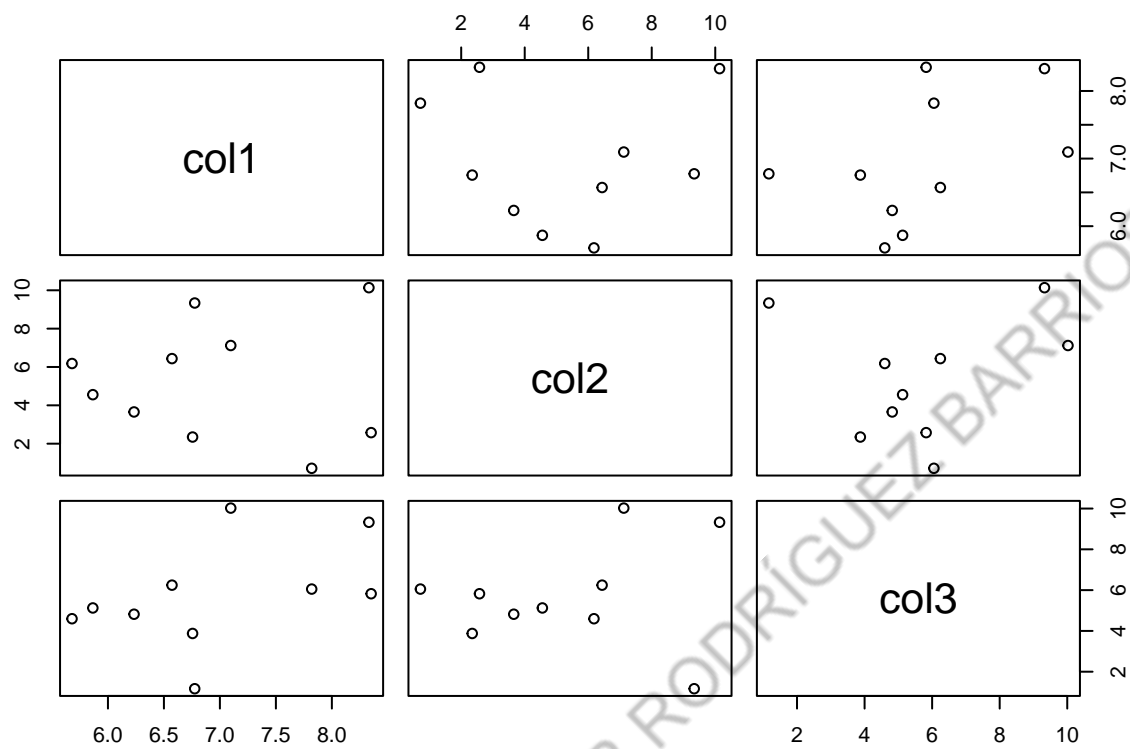
```
Matriz2 <- data.frame(Matriz2) # Data Frame - Matriz2
Matriz2
```

```
##      col1  col2  col3
## 1  6.572  6.437  6.244
## 2  5.865  4.556  5.126
## 3  6.233  3.655  4.819
## 4  8.351  2.576  5.823
## 5  7.820  0.716  6.052
## 6  6.756  2.347  3.875
## 7  5.679  6.181  4.598
## 8  6.775  9.337  1.166
## 9  8.331 10.136  9.327
## 10 7.097  7.120 10.022
```

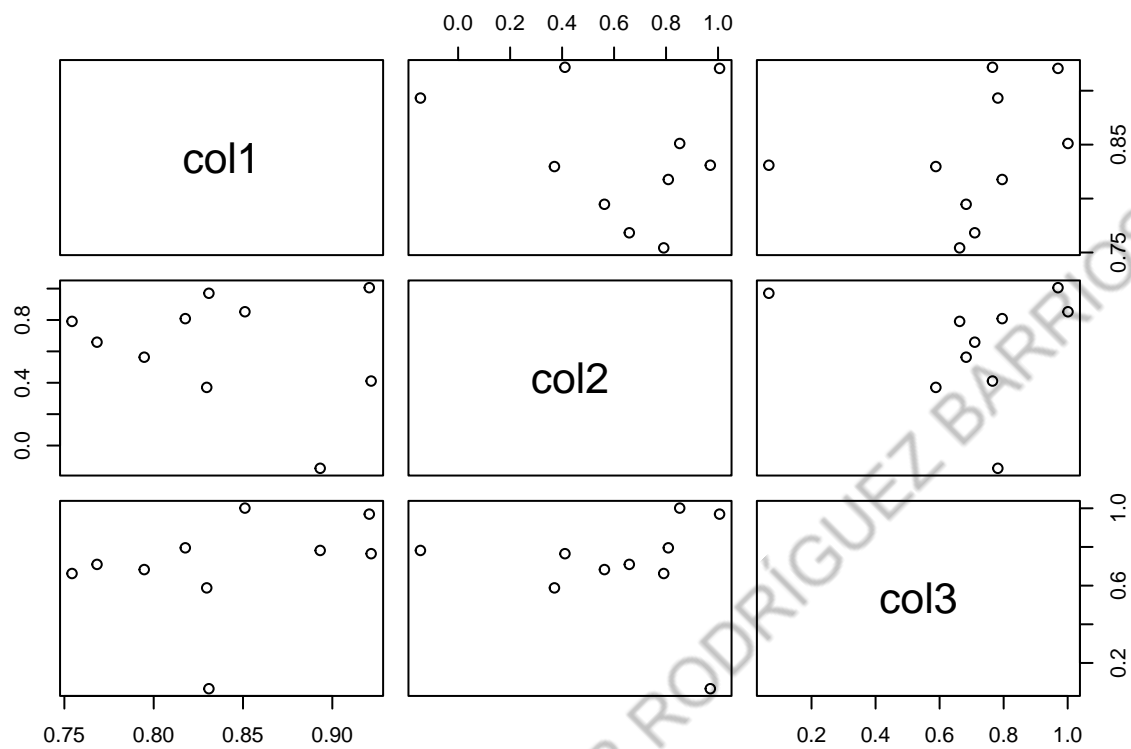
```
str(Matriz2) # Estructura del data frame
```

```
## 'data.frame':  10 obs. of  3 variables:
## $ col1: num  6.57 5.87 6.23 8.35 7.82 ...
## $ col2: num  6.437 4.556 3.655 2.576 0.716 ...
## $ col3: num  6.24 5.13 4.82 5.82 6.05 ...
```

```
x11() # Panel gráfico adicional
plot(Matriz2) # Figura de pares
```

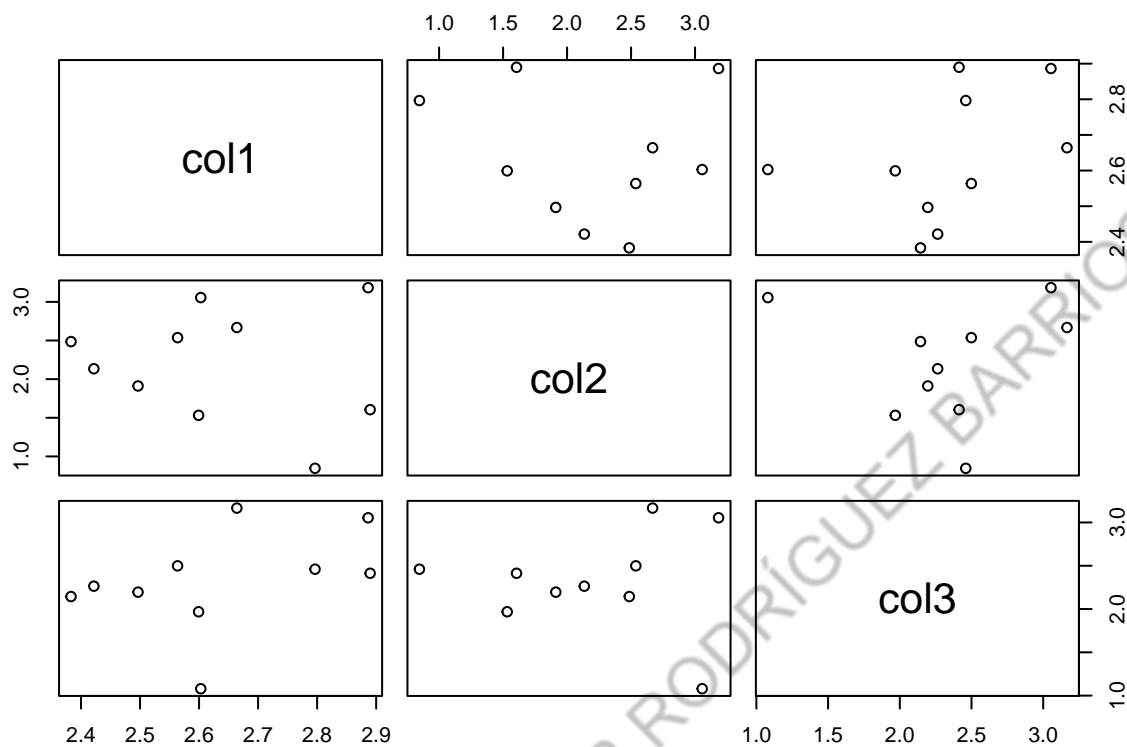


```
plot(log10(Matriz2)) # Figura con transformación logarítmica
```



```
plot(sqrt(Matriz2)) # Figura con transformación cuadrática
```





#-----

# 11) Otra opción de realizar data frame con valores aleatorios

Matriz # Dataframe para relacionar la numeración de cada fila

```
##   col1 col2 col3
## A    5    2    7
## B    6    4    3
## C    7    5    4
## D    8    9    8
## E    9    8    7
```

# Generar 10 números aleatorios

```
Matriz3= sample( # Muestra a realizar (sample)
  1:nrow(Matriz[1:5,]), # Filas de la base de datos (1:5)
  size=10, # Valores aleatorios a generar
  replace=TRUE) # Muestreo por reemplazamiento
```

Matriz3 # Los 10 números, representan el orden en que aparece cada observación

```
## [1] 2 5 3 4 1 1 2 4 4 2
```

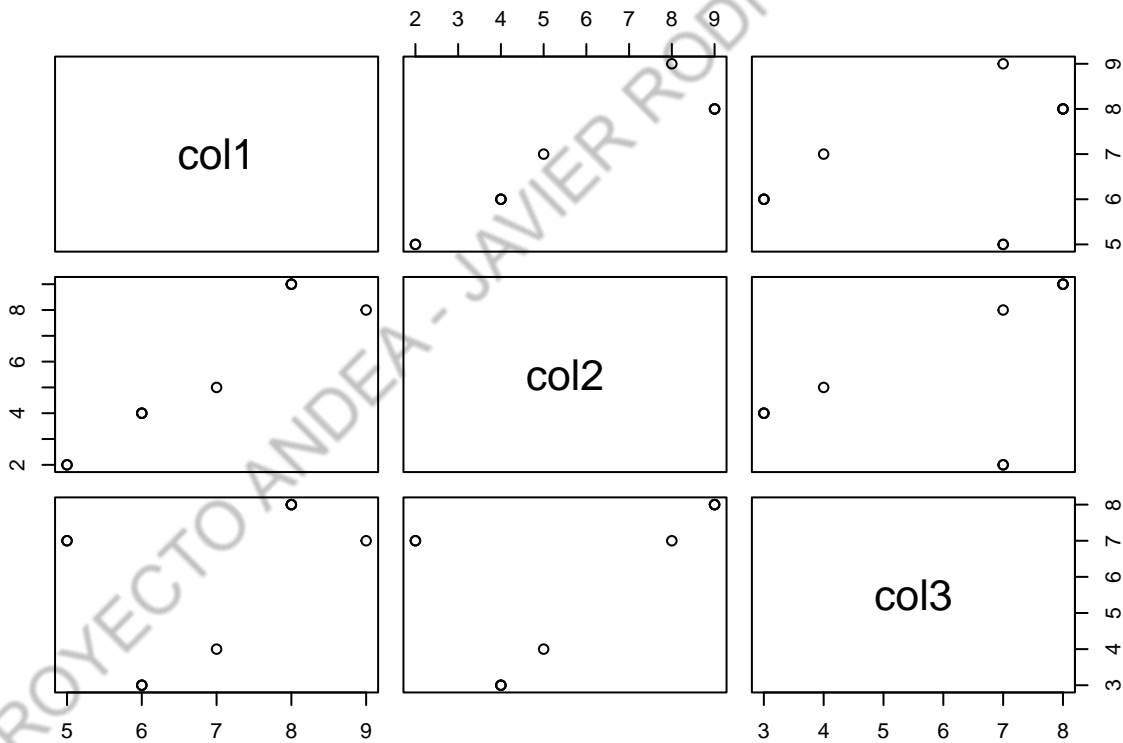
```
Matriz3<- Matriz[Matriz3, ] # Data frame generado con 10 obs. aleatorias
Matriz3 <- data.frame(Matriz3) # Data Frame - Matriz3
head(Matriz3)
```

```
##      col1 col2 col3
## B      6    4    3
## E      9    8    7
## C      7    5    4
## D      8    9    8
## A      5    2    7
## A.1    5    2    7
```

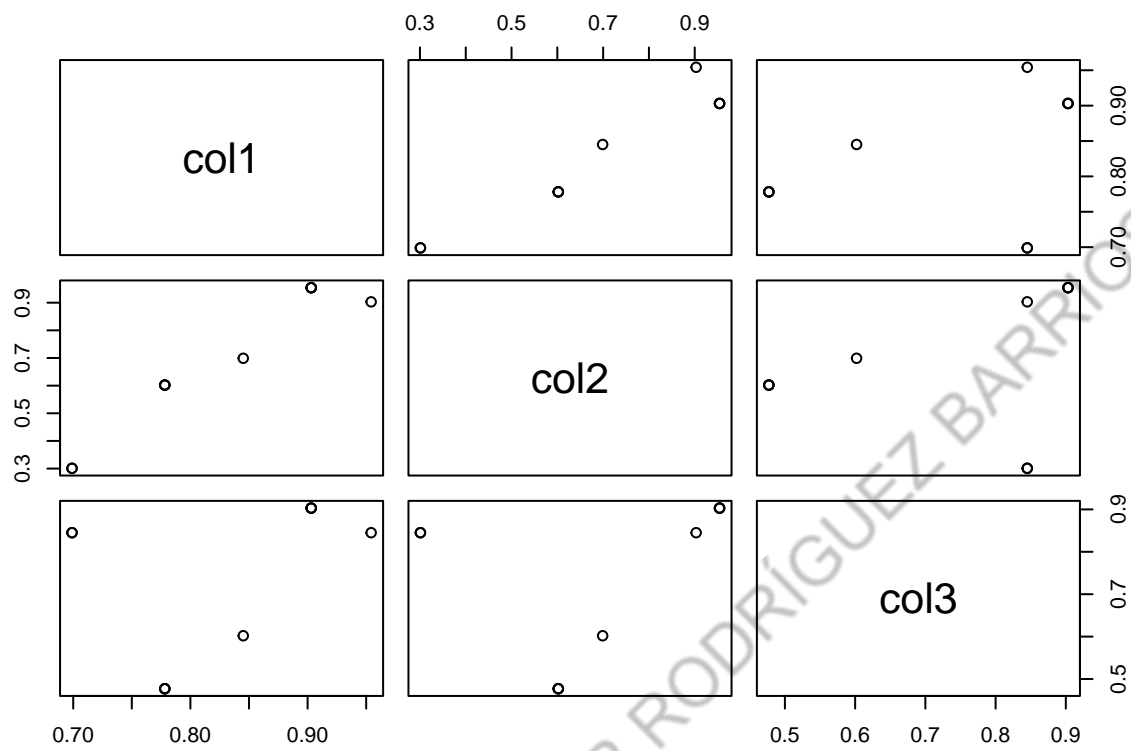
```
str(Matriz3)           # Estructura de Matriz3
```

```
## 'data.frame':      10 obs. of  3 variables:
## $ col1: num  6 9 7 8 5 5 6 8 8 6
## $ col2: num  4 8 5 9 2 2 4 9 9 4
## $ col3: num  3 7 4 8 7 7 3 8 8 3
```

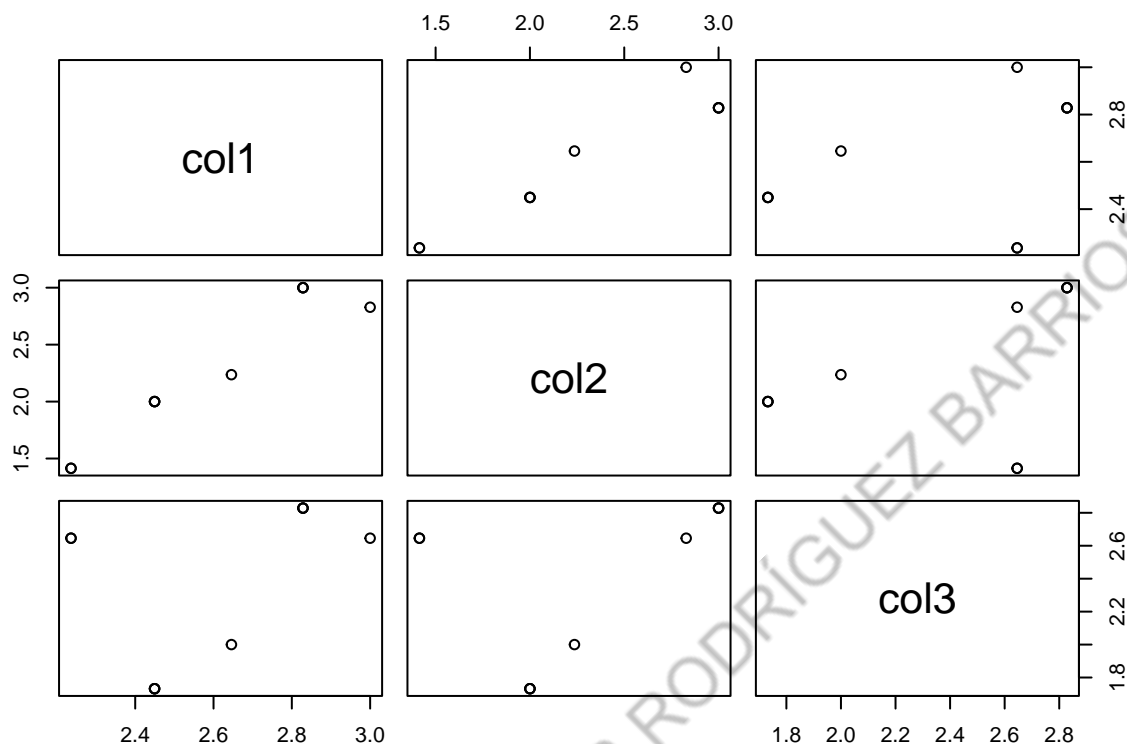
```
x11()                  # Panel gráfico adicional
plot(Matriz3)           # Figura de pares
```



```
plot(log10(Matriz3))   # Figura con transformación logarítmica
```



```
plot(sqrt(Matriz3)) # Figura con transformación cuadrática
```



“

### 3. Ejercicios propuestos.

- Basado en el documento Apuntes\_R.pdf **Enlace**. Dar respuesta a los siguientes requerimientos:
  - Correr en RStudio la asignación de valores (páginas 3 a 5).
  - Realizar los ejercicios vectoriales (páginas 5 a 9).
  - Realizar las figuras en RStudio (páginas 10 a 12).
  - Generar muestras aleatorias (páginas 13 a 14).
- Basado en el capítulo 4 del libro “R para la ciencia de datos” de Wickham y Golemund, cual se encuentra en el siguiente enlace **Enlace**. Dar respuesta a los siguientes requerimientos:
  - Ejecutar y analizar los comandos de los ítems 4.1 a 4.5.
- Basado en los capítulos 4 y 5 del libro “Manual. RStudio en Ciencias Sociales” de Boccardo y Ruiz (2018), el cual se encuentra en el siguiente enlace **Enlace**. Dar respuesta a los siguientes requerimientos:
  - Identificar los elementos fundamentales del uso de sintaxis en RStudio y a continuación, realizar los ejercicios: 4.3, 4.5, 4.6, 4.7.
  - Realizar los procedimientos para abrir bases de datos desde diferentes formatos de Microsoft Excel (ejercicios 6.3, 6.4), descargando los archivos de extensión xlsx y csv del siguiente **Enlace**. Nota: para instalar el paquete “readxl” Se recomienda realizar actualizaciones de R y de RStudio (la versión más actualizada). Nota: todos los procedimientos deben ser analizados y documentados. Las respuestas a los ejercicios no deben superar 5 páginas.