Correlacion

Mendoza Domínguez Diana Paola

2023-12-11

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

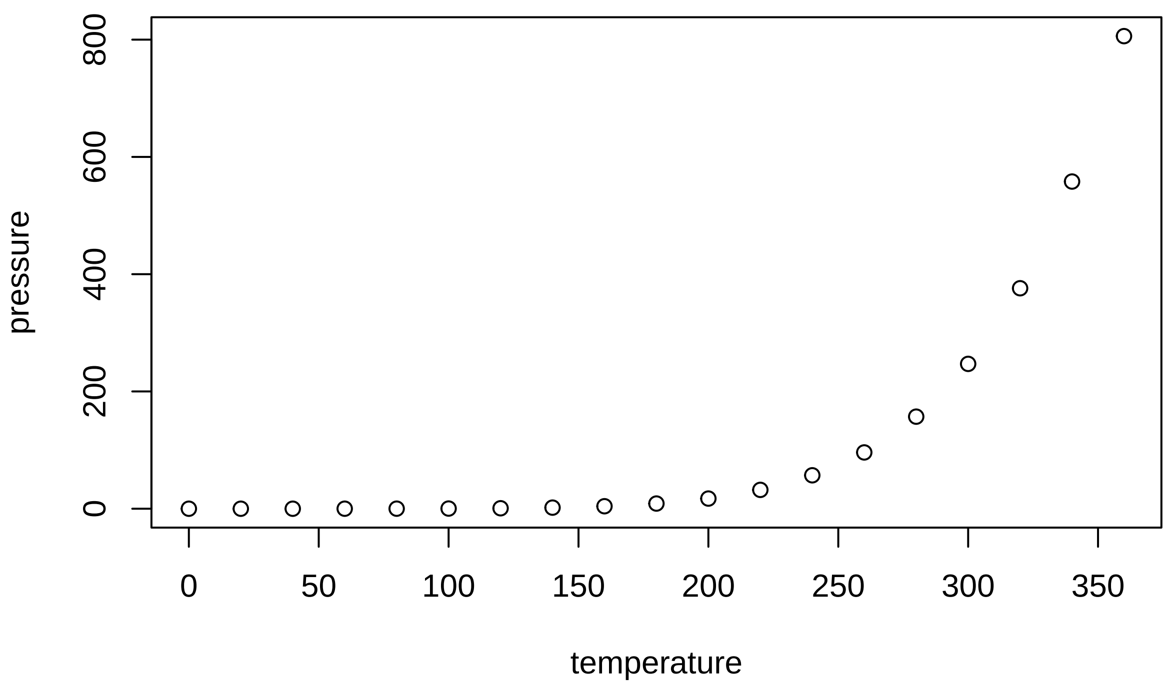
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Coeficiente de correlación de Pearson

#Para datos con distribuciones normal

#Área: Estadística Paramétrica

#Utilizamos la matriz "penguins.xlsx"

#1.- Instalación de paquetería

```
install.packages("readxl")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

#Abrimos libreria

```
library("readxl")
```

#2.- Exportación de la matriz de datos

```
penguins<-read_excel("penguins.xlsx")
```

#2.1.-Nombre de las columnas

#Para conocer el nombre de las columnas de nuestra base #de datos, se ocupa: # **colnames(BD)**

```
colnames(penguins)
```

```
## [1] "ID"              "especie"         "isla"            "largo_pico_mm"
## [5] "grosor_pico_mm"  "largo_aleta_mm"  "masa_corporal_g" "genero"
## [9] "año"
```

#3.- Exploración de la matriz

#3.1.- Dimensión de la matriz

#Se utiliza el siguiente comando para saber #la dimensión de la matriz: **dim(BD)**

```
dim(penguins)
```

```
## [1] 344   9
```

#4.- Tipo de variables

#Para observar las variables y el tipo, que tenemos ocupamos # **str(penguins)**

```
str(penguins)
```

```
## tibble [344 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ID             : chr [1:344] "i1" "i2" "i3" "i4" ...
##  $ especie        : chr [1:344] "Adelie" "Adelie" "Adelie" "Adelie" ...
##  $ isla           : chr [1:344] "Torgersen" "Torgersen" "Torgersen" "Torgersen" ...
##  $ largo_pico_mm  : num [1:344] 39.1 39.5 40.3 37.8 36.7 39.3 38.9 39.2 34.1 42 ...
##  $ grosor_pico_mm : num [1:344] 18.7 17.4 18 18.1 19.3 20.6 17.8 19.6 18.1 20.2 ...
##  $ largo_aleta_mm : num [1:344] 181 186 195 190 193 190 181 195 193 190 ...
##  $ masa_corporal_g: num [1:344] 3750 3800 3250 3700 3450 ...
##  $ genero         : chr [1:344] "male" "female" "female" "female" ...
##  $ año            : num [1:344] 2007 2007 2007 2007 2007 ...
```

#5.- En busca de datos perdidos

#Buscamos si tenemos datos perdidos o no con # **anyNA(penguins)**

```r
anyNA(penguins)
```

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

Para sacar el coeficiente de correlación de Pearson

#1.- Seleccionamos las variables que vayamos a correlacionar, #ocupamos: # **str(penguins)** # **penguins$especie**

```r
str(penguins)
```

```
## tibble [344 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ID             : chr [1:344] "i1" "i2" "i3" "i4" ...
##  $ especie        : chr [1:344] "Adelie" "Adelie" "Adelie" "Adelie" ...
##  $ isla           : chr [1:344] "Torgersen" "Torgersen" "Torgersen" "Torgersen" ...
##  $ largo_pico_mm  : num [1:344] 39.1 39.5 40.3 37.8 36.7 39.3 38.9 39.2 34.1 42 ...
##  $ grosor_pico_mm : num [1:344] 18.7 17.4 18 18.1 19.3 20.6 17.8 19.6 18.1 20.2 ...
##  $ largo_aleta_mm : num [1:344] 181 186 195 190 193 190 181 195 193 190 ...
##  $ masa_corporal_g: num [1:344] 3750 3800 3250 3700 3450 ...
##  $ genero         : chr [1:344] "male" "female" "female" "female" ...
##  $ año            : num [1:344] 2007 2007 2007 2007 2007 ...
```

```r
penguins$especie
```

```
##   [1] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##   [7] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [13] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [19] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [25] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [31] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [37] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [43] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [49] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [55] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [61] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [67] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [73] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [79] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [85] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [91] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
##  [97] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [103] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [109] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [115] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [121] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [127] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [133] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [139] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [145] "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"    "Adelie"
## [151] "Adelie"    "Adelie"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [157] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [163] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
```

```
## [169] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [175] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [181] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [187] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [193] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [199] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [205] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [211] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [217] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [223] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [229] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [235] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [241] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [247] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [253] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [259] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [265] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [271] "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"    "Gentoo"
## [277] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [283] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [289] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [295] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [301] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [307] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [313] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [319] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [325] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [331] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [337] "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap" "Chinstrap"
## [343] "Chinstrap" "Chinstrap"
```

#2.- Se seleccionan las filas 1 a la 61, que corresponden a la #especie Adeli y las variables cuantitativas. # **adeli<-penguins[1:61,4:7]**

```
adeli<-penguins[1:61,4:7]
```

#3.- Visualización de la matriz

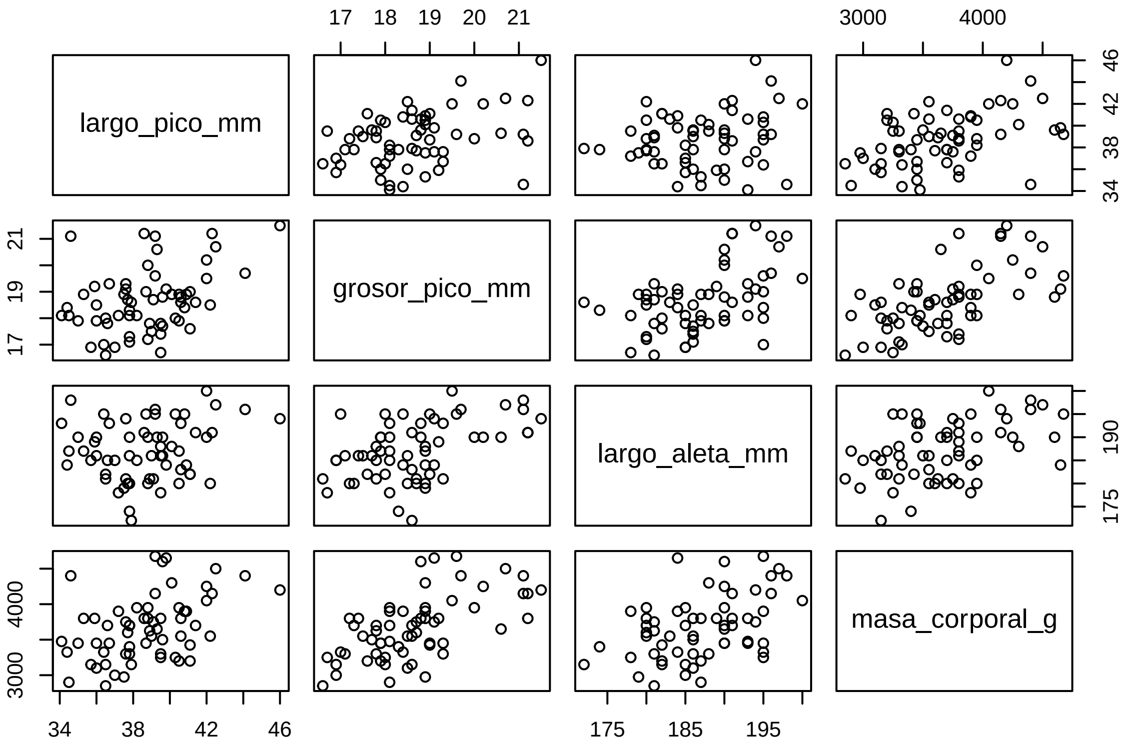
#Para poder vizualizar nuestro objeto

```
adeli
```

```
## # A tibble: 61 x 4
##    largo_pico_mm grosor_pico_mm largo_aleta_mm masa_corporal_g
##            <dbl>          <dbl>          <dbl>           <dbl>
##  1          39.1           18.7            181            3750
##  2          39.5           17.4            186            3800
##  3          40.3           18              195            3250
##  4          37.8           18.1            190            3700
##  5          36.7           19.3            193            3450
##  6          39.3           20.6            190            3650
##  7          38.9           17.8            181            3625
##  8          39.2           19.6            195            4675
##  9          34.1           18.1            193            3475
## 10          42             20.2            190            4250
## # i 51 more rows
```

#4.- Generación del gráfico de correlación # **plot(adeli)**

```
plot(adeli)
```



#5.- Cálculo de la correlación de Pearson # **cor(adeli)**

```
cor(adeli)
```

```
##                 largo_pico_mm grosor_pico_mm largo_aleta_mm masa_corporal_g
## largo_pico_mm       1.0000000      0.3778875      0.1766987       0.4535845
## grosor_pico_mm      0.3778875      1.0000000      0.4760336       0.6144894
## largo_aleta_mm      0.1766987      0.4760336      1.0000000       0.4458517
## masa_corporal_g     0.4535845      0.6144894      0.4458517       1.0000000
```

#6.- Organización visual de la tabla de correlaciones

6.1.- Se genera un nuevo objeto con el nombre de pearson,

#es decir,

```
pearson<-cor(adeli)
```

#6.2.- Se abre la librería knitr

```
library(knitr)
```

#6.3.- Se utiliza la funcion kable

```
kable(pearson)
```

	largo_pico_mm	grosor_pico_mm	largo_aleta_mm	masa_corporal_g
largo_pico_mm	1.0000000	0.3778875	0.1766987	0.4535845
grosor_pico_mm	0.3778875	1.0000000	0.4760336	0.6144894

	largo_pico_mm	grosor_pico_mm	largo_aleta_mm	masa_corporal_g
largo_aleta_mm	0.1766987	0.4760336	1.0000000	0.4458517
masa_corporal_g	0.4535845	0.6144894	0.4458517	1.0000000

Coeficiente de correlación de Spearman

#Para datos con distribucion NO Normal

#Área: Estadistica NO Paramétrica

#Se utiliza la matriz marvel_dc.csv

#1.- Exportación de la matriz de datos

```
marvel_dc<-read_excel("marvel_dc.xlsx")
```

```
## New names:
## * `` -> `...1`
```

#2.- Exploración de la matriz

#2.1.- Dimensión de la matriz

#Se utiliza el siguiente comando para saber #la dimensión de la matriz: # **dim(BD)**

```
dim(marvel_dc)
```

```
## [1] 39 11
```

#2.2.- En busca de datos perdidos

#Buscamos si tenemos datos perdidos o no con # **anyNA(BD)**

```
anyNA(marvel_dc)
```

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

#3.- Tipo de variables

#Para identificar las variables cuantitativas # **str(BD)**

```
str(marvel_dc)
```

```
## tibble [39 x 11] (S3: tbl_df/tbl/data.frame)
##  $ ...1               : num [1:39] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Original Title     : chr [1:39] "Iron Man" "The Incredible Hulk" "Iron Man 2" "Thor" ...
##  $ Company            : chr [1:39] "Marvel" "Marvel" "Marvel" "Marvel" ...
##  $ Rate               : num [1:39] 7.9 6.7 7 7 6.9 8 7.2 6.9 7.7 8 ...
##  $ Metascore          : num [1:39] 79 61 57 57 66 69 62 54 70 76 ...
##  $ Minutes            : num [1:39] 126 112 124 115 124 143 130 112 136 121 ...
##  $ Release            : num [1:39] 2008 2008 2010 2011 2011 ...
##  $ Budget             : num [1:39] 1.4e+08 1.5e+08 2.0e+08 1.5e+08 1.4e+08 2.2e+08 2.0e+08 1.7e+08 1
##  $ Opening Weekend USA: num [1:39] 9.86e+07 5.54e+07 1.28e+08 6.57e+07 6.51e+07 ...
##  $ Gross USA          : num [1:39] 3.19e+08 1.35e+08 3.12e+08 1.81e+08 1.77e+08 ...
##  $ Gross Worldwide    : num [1:39] 5.85e+08 2.63e+08 6.24e+08 4.49e+08 3.71e+08 ...
```

#4.- Para saber el nombre y posición de la variable ocupamos: # **colnames(BD)**

```
colnames(marvel_dc)
```

```
##  [1] "...1"                "Original Title"      "Company"
##  [4] "Rate"                "Metascore"           "Minutes"
##  [7] "Release"             "Budget"              "Opening Weekend USA"
## [10] "Gross USA"           "Gross Worldwide"
```

#5.- Seleccionamos las variables: # rate, minutos, budget y gross.worldwide, con: # **marvel<-marvel_dc[,c(4,6,8,11)]** # *Nota: elegimos columnas nuevas, debido a que la número 4 #y la 6 son carácteres y necesitamos utilizar numéricas

#Ocuparemos las variables: # rate, metascore, gross USA y gross Worldwide

```r
marvel<-marvel_dc[,c(4,5,10,11)]
```

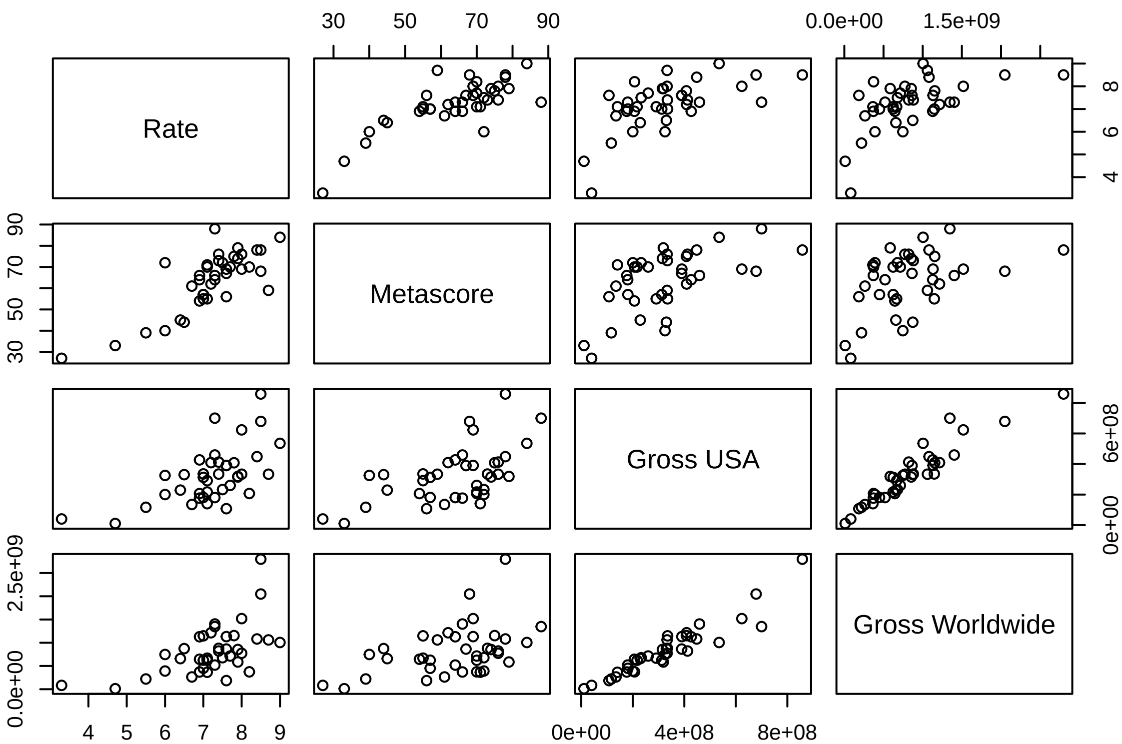
#6.- Verificar que el nombre de las variables #esten correctas utilizando: # **colnames(marvel)**

```r
colnames(marvel)
```

```
## [1] "Rate"             "Metascore"        "Gross USA"        "Gross Worldwide"
```

#7.- Realizar un plot de exploración con: # **plot(marvel)**

```r
plot(marvel)
```



#8.- Realizar la correlación de spearman con: # **spearman<-cor(marvel, method = "spearman")**

```r
spearman<-cor(marvel,method="spearman")
```

#9.- Vizualizar el objeto

```r
spearman
```

```
##                      Rate Metascore Gross USA Gross Worldwide
## Rate            1.0000000 0.6938601 0.5830256       0.5289085
## Metascore       0.6938601 1.0000000 0.5201540       0.3926474
## Gross USA       0.5830256 0.5201540 1.0000000       0.9536437
## Gross Worldwide 0.5289085 0.3926474 0.9536437       1.0000000
```

#9.2.- Se abre la librería knitr

```r
library(knitr)
```

#10.- Se utiliza la funcion kable para tabla en #formato markdown. # **kable(spearman)**

```r
kable(spearman)
```

	Rate	Metascore	Gross USA	Gross Worldwide
Rate	1.0000000	0.6938601	0.5830256	0.5289085
Metascore	0.6938601	1.0000000	0.5201540	0.3926474
Gross USA	0.5830256	0.5201540	1.0000000	0.9536437
Gross Worldwide	0.5289085	0.3926474	0.9536437	1.0000000