R_Intro_iBio2019

T.Moyano & J.Maldonado

8 de agosto de 2019

INTRODUCCIÓN a R

[iBio] http://www.ibio.cl/2019/08/06/workshop-ii-ibio-bioinformatica iBio

Para insertar un "comentario" en el código se utiliza el símbolo gato (#).

Cuando está presente, todo lo que está a la derecha del (#) no es interpretado por R.

De esta forma podemos evitar que se ejecute un código y también podemos dejar comentarios de ayuda.

```
# 2+2
```

R puede realizar funcionas matemáticas básicas:

```
2+2
```

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

```
2*3
```

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

```
8/2
```

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

```
10-3
```

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

Hay diferentes formas de asignar variables:

```
a <- 2+2

a
```

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

```r
b = 2+3

b
```

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

```r
a*b
```

```
## [1] 20
```

```r
a*b -> c
```

Con ls() listamos los objetos o variables creadas:

```r
ls()
```

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

Hay diferentes tipos de estructuras de datos: Vectores, Matrices, Dataframe, etc

Creación de vectores:

```r
vectorletras<-c("plantas","hongos","milenio","iBio","cocacola")

vectorletras
```

```
## [1] "plantas"  "hongos"   "milenio"  "iBio"     "cocacola"
```

```r
vectornumeros<-c(0,1,2,3,4,5)

vectornumeros
```

```
## [1] 0 1 2 3 4 5
```

Para ver elementos específicos de un vector:

```r
vectornumeros[3]
```

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

Operaciones matemáticas con vectores:

```r
vectornumeros2<-vectornumeros*(a+b+1)

vectornumeros2
```

```
## [1]  0 10 20 30 40 50
```

Creación de matrices

```
matriz1<-cbind(vectornumeros,vectornumeros2)

matriz2<-cbind(matriz1,vectorletras)
```

```
## Warning in cbind(matriz1, vectorletras): number of rows of result is not a
## multiple of vector length (arg 2)
```

Cuidado con las dimensiones

```
vectorletras<-c(vectorletras,"queso")

matriz2<-cbind(matriz1,vectorletras)

matriz2
```

```
##      vectornumeros vectornumeros2 vectorletras
## [1,] "0"           "0"            "plantas"
## [2,] "1"           "10"           "hongos"
## [3,] "2"           "20"           "milenio"
## [4,] "3"           "30"           "iBio"
## [5,] "4"           "40"           "cocacola"
## [6,] "5"           "50"           "queso"
```

Fíjese en las comillas, ahora todo es texto.

Creación de dataframes

```
dataframe<-data.frame(matriz1,vectorletras)

dataframe[,1]
```

```
## [1] 0 1 2 3 4 5
```

Funciones básicas

```
sum(dataframe[,1])
```

```
## [1] 15
```

```
dataframe[c(1:2,5:6),]
```

```
##   vectornumeros vectornumeros2 vectorletras
## 1             0              0      plantas
## 2             1             10       hongos
## 5             4             40     cocacola
## 6             5             50        queso
```

```r
colSums(dataframe[c(1:2,5:6),1:2])
```

```
##  vectornumeros vectornumeros2
##             10            100
```

```r
colMeans(dataframe[c(1:2,5:6),1:2])
```

```
##  vectornumeros vectornumeros2
##            2.5           25.0
```

Para obtener ayuda acerca de funciones se puede usar el signo de pregunta seguido del comando:

```r
?colMeans
```

```
## starting httpd help server ... done
```

Cuando trabajamos en R, podemos elegir el directorio de trabajo.

Configurar el directorio de trabajo (origen y destino de los archivos de la sesión)

- Modo gráfico (Session -> Set Working Directory -> Choose Directory)
- Modo comando

```r
setwd("D:/Lab/curso_iBio")
```

Usando datos tabulados

Para esta parte usaremos el set de datos "iris", el cual es un ejemplo clásico de R que es utilizado generalemente por los desarrolladores de packetes para mostrar la forma de usar las funciones de R.

> This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

Ejemplo1 https://rpubs.com/moeransm/intro-iris

Ejemplo2 https://www.kaggle.com/antoniolopez/iris-data-visualization-with-r

El siguiente comando les muestra los datos almacenados en la variable "iris" que está disponible siempre en R

```
iris
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
## 33           5.2         4.1          1.5         0.1     setosa
## 34           5.5         4.2          1.4         0.2     setosa
## 35           4.9         3.1          1.5         0.2     setosa
## 36           5.0         3.2          1.2         0.2     setosa
## 37           5.5         3.5          1.3         0.2     setosa
## 38           4.9         3.6          1.4         0.1     setosa
## 39           4.4         3.0          1.3         0.2     setosa
## 40           5.1         3.4          1.5         0.2     setosa
## 41           5.0         3.5          1.3         0.3     setosa
## 42           4.5         2.3          1.3         0.3     setosa
## 43           4.4         3.2          1.3         0.2     setosa
## 44           5.0         3.5          1.6         0.6     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 46           4.8         3.0          1.4         0.3     setosa
## 47           5.1         3.8          1.6         0.2     setosa
## 48           4.6         3.2          1.4         0.2     setosa
```

```
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
```

```
## 103          7.1         3.0          5.9         2.1  virginica
## 104          6.3         2.9          5.6         1.8  virginica
## 105          6.5         3.0          5.8         2.2  virginica
## 106          7.6         3.0          6.6         2.1  virginica
## 107          4.9         2.5          4.5         1.7  virginica
## 108          7.3         2.9          6.3         1.8  virginica
## 109          6.7         2.5          5.8         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 111          6.5         3.2          5.1         2.0  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

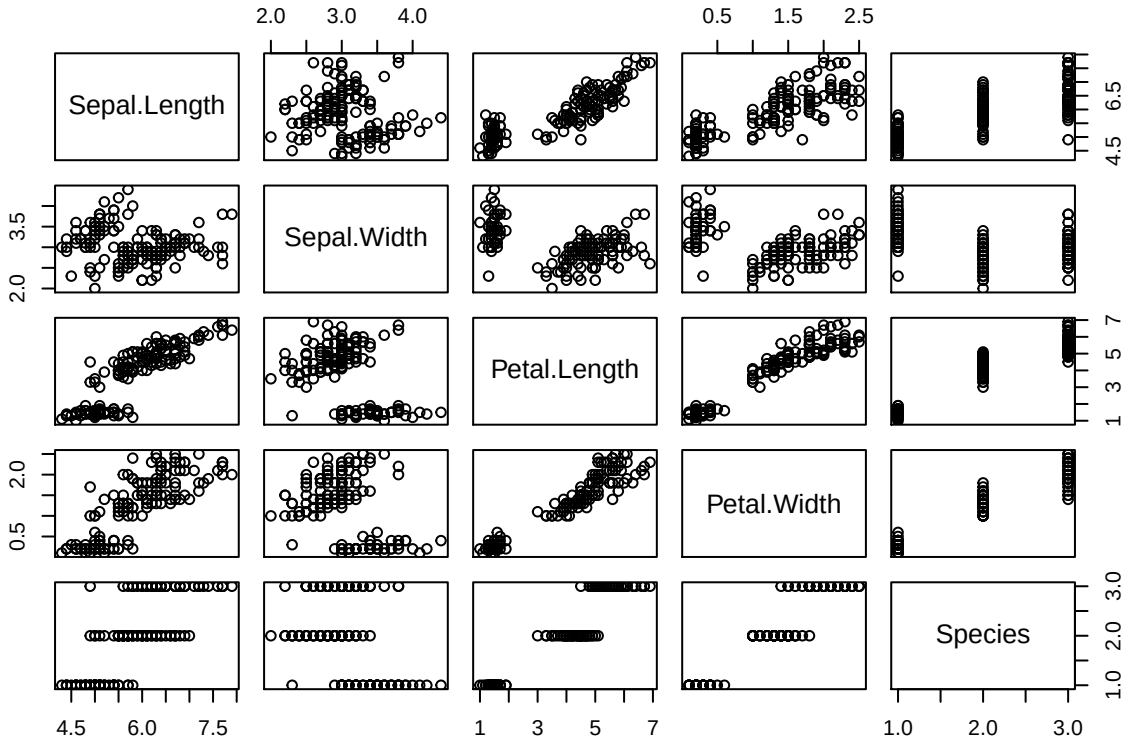
Podemos copiar objetos

En este ejemplo vamos a pasar los datos de la variable de sistema "iris" a una nueva variable:

```
tabla_iris<-iris
```

Ahora vamos a visualizar información del set de datos iris:

```
plot(tabla_iris) # grafica lo que puede graficar
```




```
summary(tabla_iris) # resumen estadístico de las columnas
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```r
str(tabla_iris) # "representación textual" del objeto
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
dim(tabla_iris) # dimensiones del objeto
```

```
## [1] 150   5
```

Vamos a escribirla variable "tabla_iris"" en un archivo de texto, separando las columnas por tabulaciones y sin comillas

```r
write.table(tabla_iris,"tabla_iris.txt",sep="\t",quote=F,col.names = NA)
```

Vamos a leer la tabla de texto que hemos creado, con los nombres en la primera columna y un encabezado.

Este ejemplo demuestra que pueden importar cualquier archivo de texto tabulado (ej csv) conociendo el separador de columnas y el diseño de la tabla.

```r
tabla_texto<-read.table("tabla_iris.txt",sep="\t",row.names = 1,header=T)
```

Tarea:

Usando Excel, importe el contenido del archivo tabla_iris.txt (a Excel) asegurándose que el separador de decimales usado por Excel sea el correcto para interpretar los datos como numéricos.

Hay que tener cuidado con los . y , de acuerdo al idioma.

Importando a R archivos de tipo Excel

Existen paquetes o conjuntos de funciones que se descargan de repositorios oficiales para tareas específicas o más completas que las ofrecidas por los packetes base de R:

Una de ellas es "readxl" la cual permite leer archivos ".xls" y ".xlsx".

El siguiente comando les permitirá instalar dicho paquete/función para que quede disponible en su equipo:

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

Para cargar en memoria el paquete (tenerlo disponible en la sesión actual) se usa la función "library"

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.6.1
```

Al leer el archivo Excel debe asegurarse que el nombre de la hoja (sheet) es el correcto. En este caso es "Hoja1".

```
misdatos<-read_excel("ibio.xlsx",sheet = "Hoja1")
```

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

```
summary(as.data.frame(misdatos))
```

```
##       ...1         Sepal.Length    Sepal.Width     Petal.Length
##  Min.   :  1.00   Min.   :4.300   Min.   :2.000   Min.   :1.000
##  1st Qu.: 38.25   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
##  Median : 75.50   Median :5.800   Median :3.000   Median :4.350
##  Mean   : 75.50   Mean   :5.843   Mean   :3.057   Mean   :3.758
##  3rd Qu.:112.75   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
##  Max.   :150.00   Max.   :7.900   Max.   :4.400   Max.   :6.900
##   Petal.Width      Species
##  Min.   :0.100   Length:150
##  1st Qu.:0.300   Class :character
##  Median :1.300   Mode  :character
##  Mean   :1.199
##  3rd Qu.:1.800
##  Max.   :2.500
```

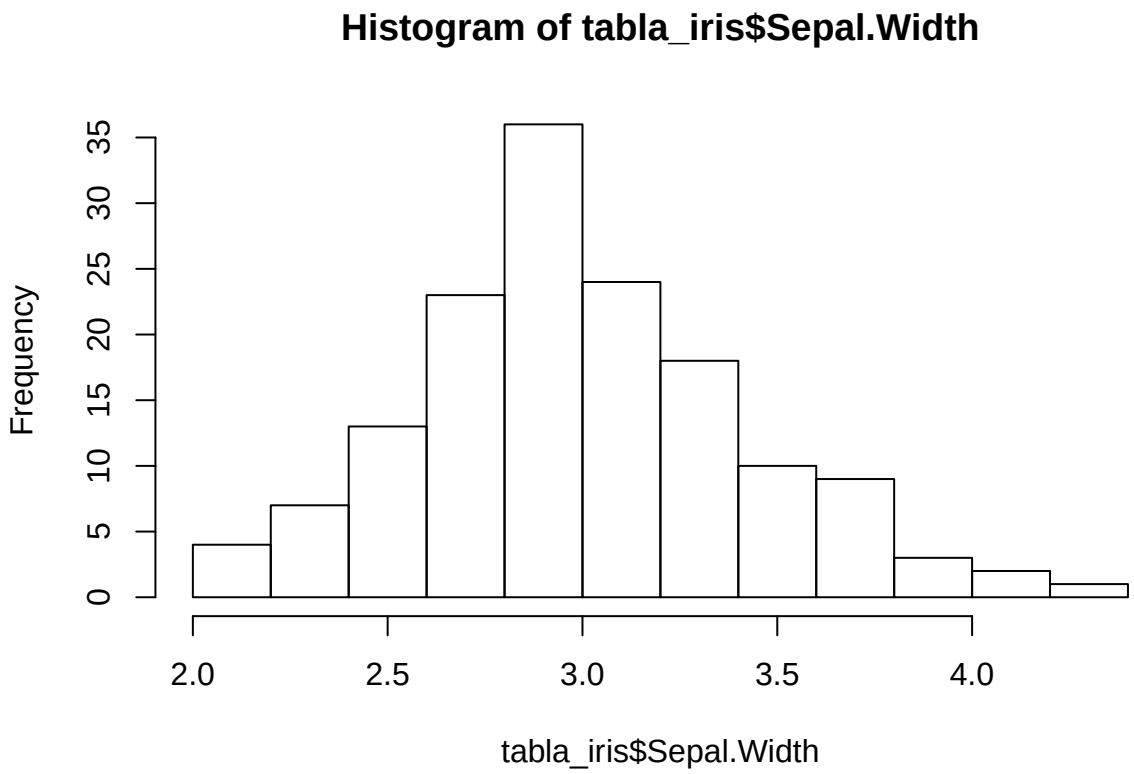
Visualizar los datos

Existen muchas funciones para visualizar datos cuya aplicación depende del tipo de datos y de lo que deseamos observar.

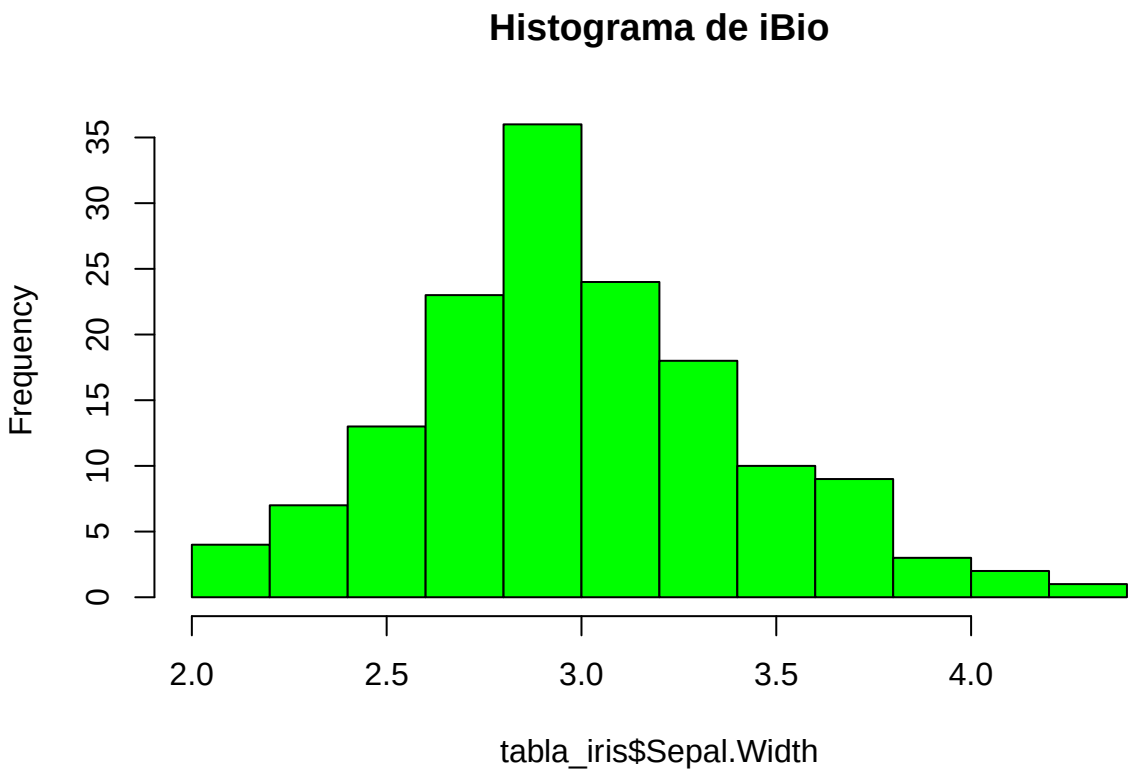
Revise el siguiente sitio para tener una idea de opciones disponibles: https://www.r-graph-gallery.com

Una forma habitual de revisar los datos es usar histogramas:

```
hist(tabla_iris$Sepal.Width,nclass = 10)
```




```
?hist # ayuda de la función hist

hist(tabla_iris$Sepal.Width,nclass = 10, main="Histograma de iBio",col = "green")
```




```
#hist(tabla_iris$Species) # no se puede hace un historama de una columna que contiene sólo palabras.
```

```
mean(tabla_iris$Sepal.Length)
```

```
## [1] 5.843333
```

Para filtrar la tabla seleccionando objetos que cumplan con alguna "cualidad"" pueden utilizar el siguiente comando.

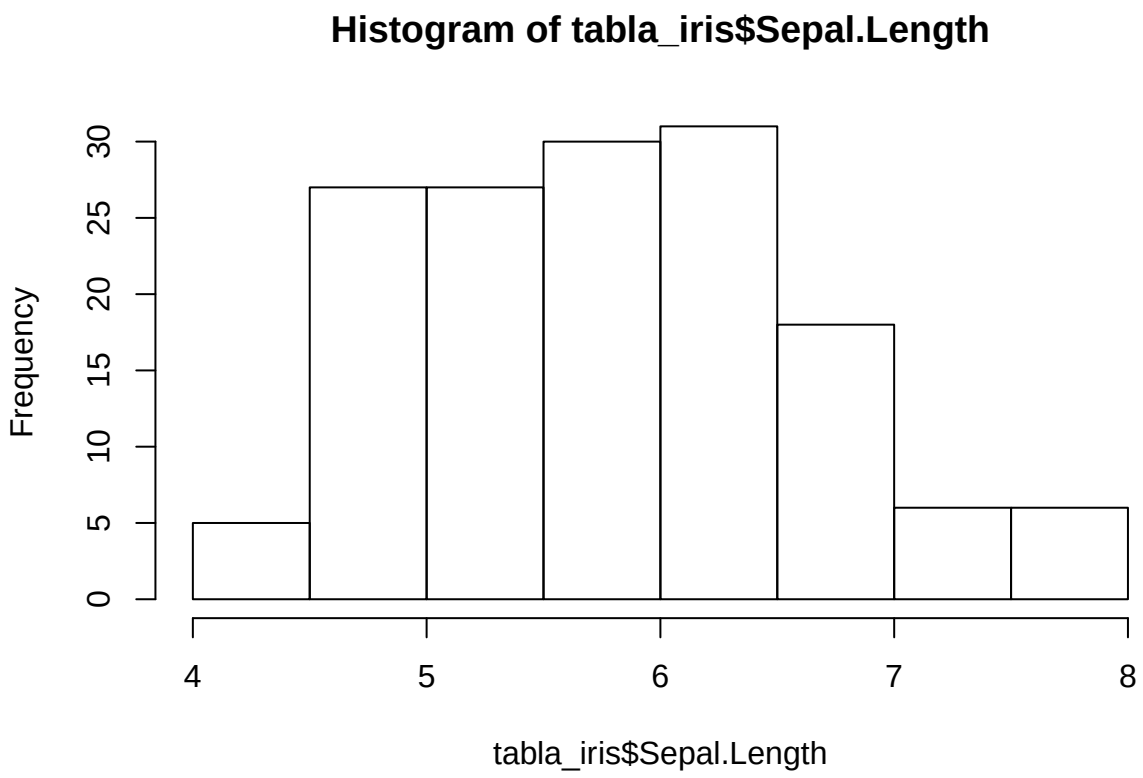
En este caso vamos ver el resultado sólo de los datos que pertenecen a la especie "virginica":

```
mean(tabla_iris$Sepal.Length[tabla_iris$Species=="virginica"])
```

```
## [1] 6.588
```

Prueba de Shapiro Wilk para evaluar normalidad (muestras pequeñas)

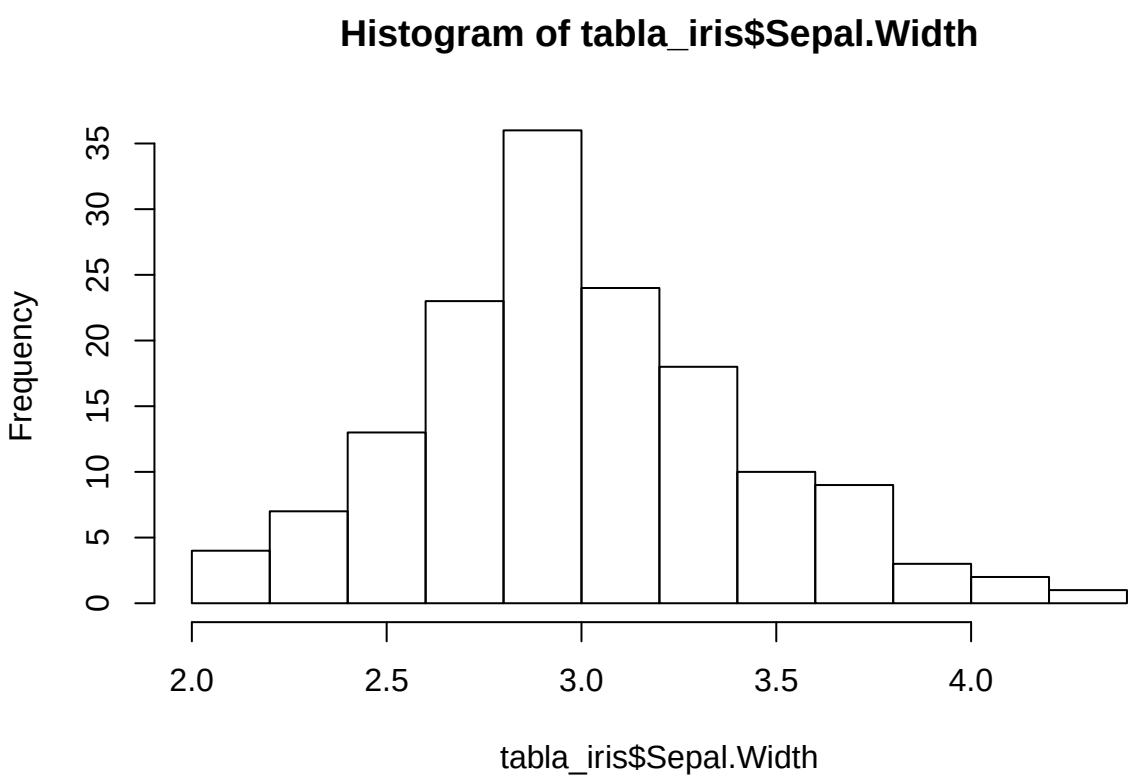
```
hist(tabla_iris$Sepal.Length,nclass = 12)
```




```
shapiro.test(tabla_iris$Sepal.Length)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tabla_iris$Sepal.Length
## W = 0.97609, p-value = 0.01018
```

```
hist(tabla_iris$Sepal.Width,nclass = 12)
```




```
shapiro.test(tabla_iris$Sepal.Width)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tabla_iris$Sepal.Width
## W = 0.98492, p-value = 0.1012
```

Para ver correlación entre los datos (todas las comparaciones posibles)

```
#cor(tabla_iris) # esto no funciona pues la matriz tiene una columna con texto
```

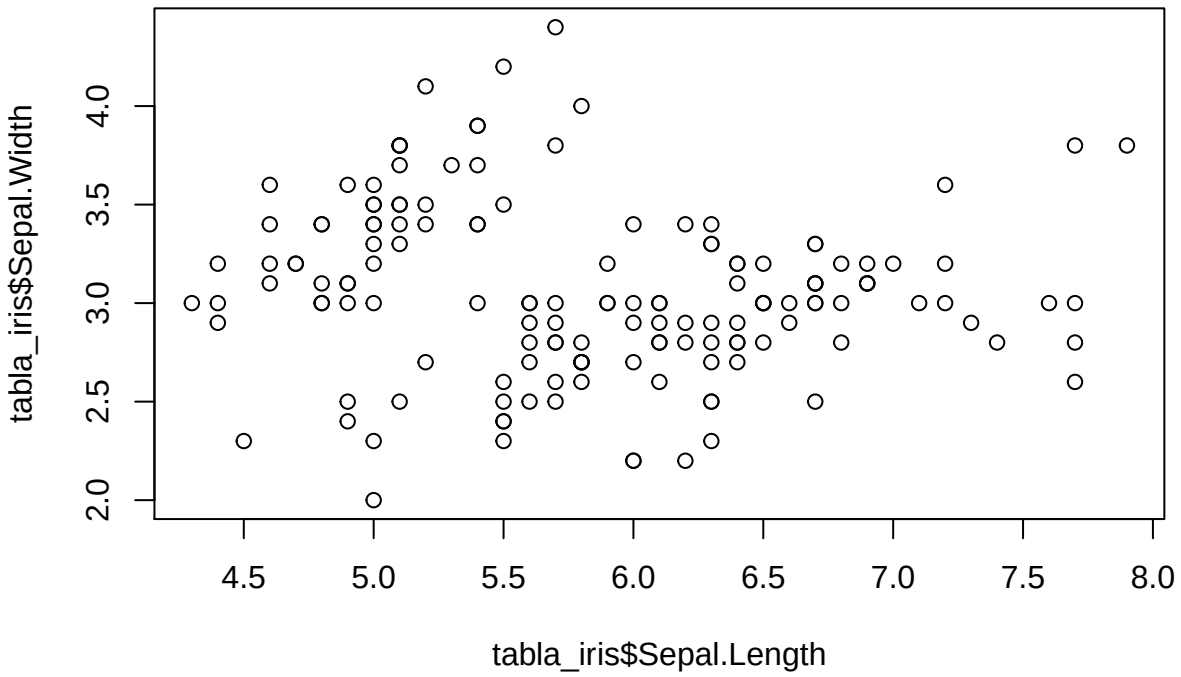
Cuidado cuando hay texto, vamos a eliminar la columna que tiene texto

```
tabla_iris_sinespecies<-tabla_iris[,-5]

cor(tabla_iris_sinespecies)
```

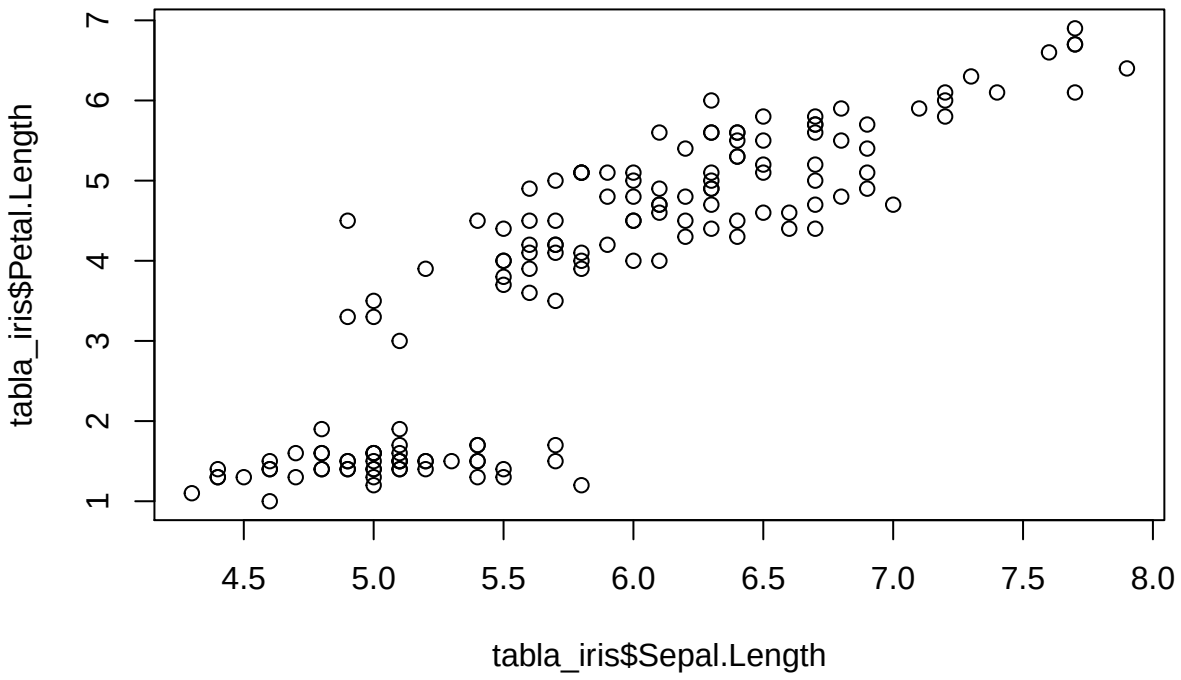
```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

Grafiquemos los datos pero seleccionando algunas columnas

```
plot(tabla_iris$Sepal.Length,tabla_iris$Sepal.Width)
```

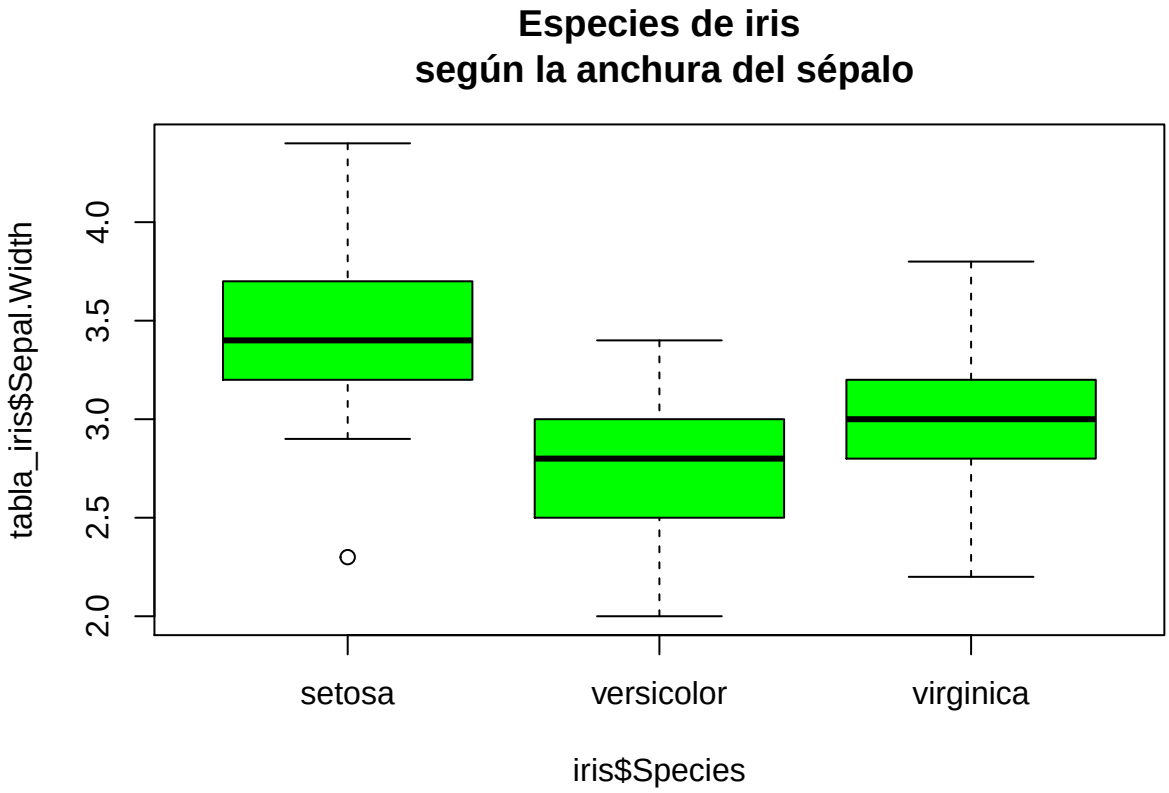



```
plot(tabla_iris$Sepal.Length,tabla_iris$Petal.Length)
```



Grafiquemos los datos usando boxplot y agrupando por especie y camviando el color a verde

```
boxplot(tabla_iris$Sepal.Width~iris$Species,col ="green",main ="Especies de iris\n según la anchura del
```



Guardar los gráficos

Para guardar los datos hay varias opciones... la más común es usar las opciones de la herramienta de RStudio que aparece en la parte superior de la ventana de los gráficos.

Si queremos guardar el gráfico como PDF usamos los siguientes comandos:

```
pdf("grafico.pdf")

boxplot(tabla_iris$Sepal.Width~iris$Species,col ="green",main ="Especies de iris\nsegún la anchura del s

dev.off()
```

```
## pdf
##   2
```

Fitro de datos

Por ejemplo, queremos los datos que no sean "virginica""

```
#2especies<-tabla_iris[tabla_iris$Species!="virginica",] ## recordatorio de que los nombres de variable
```

ojooooo, nombres de variables en R no pueden partir con un número

```
especies_2<-tabla_iris[tabla_iris$Species!="virginica",]
```

Haremos un "test de t"" para ver si las diferencias son significativas

```
t.test(especies_2$Sepal.Width~especies_2$Species)
```

```
##
##  Welch Two Sample t-test
##
## data:  especies_2$Sepal.Width by especies_2$Species
## t = 9.455, df = 94.698, p-value = 2.484e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5198348 0.7961652
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    3.428                    2.770
```

Guardamos este resultado en una variable para su posterior uso

```
resultado_test_de_t<-t.test(especies_2$Sepal.Width~especies_2$Species)
```

Este resultado se guardó en un objeto tipo "lista""

Para visualizarlos las opciones o variables contenidas en la "lista" usamos "ls"

De esta forma podemos conocer las variables y seleccionar la que deseamos ver (en este caso, p.value)

```
ls(resultado_test_de_t)
```

```
##  [1] "alternative" "conf.int"    "data.name"   "estimate"    "method"
##  [6] "null.value"  "p.value"     "parameter"   "statistic"   "stderr"
```

```
resultado_test_de_t$p.value
```

```
## [1] 2.484228e-15
```

"Cruzar tablas #### Para el ejemplo, generaremos una tabla con"IDs" aleatorios.

```
aleatorio<-sample(1:150,25)
```

Lo dejaremos como texto y usaremos estos "IDs" para hacer una "sub-selección" desde la tabla iris.

Con esto obtendremos una nueva tabla que es un "sub-conjunto" de la tabla iris:

```
aleatorio<-as.character(aleatorio)

tabla_nueva<-tabla_iris[aleatorio,]

tabla_nueva
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 102          5.8         2.7          5.1         1.9  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 150          5.9         3.0          5.1         1.8  virginica
## 72           6.1         2.8          4.0         1.3 versicolor
## 79           6.0         2.9          4.5         1.5 versicolor
## 39           4.4         3.0          1.3         0.2     setosa
## 139          6.0         3.0          4.8         1.8  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 82           5.5         2.4          3.7         1.0 versicolor
## 21           5.4         3.4          1.7         0.2     setosa
## 144          6.8         3.2          5.9         2.3  virginica
## 17           5.4         3.9          1.3         0.4     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 86           6.0         3.4          4.5         1.6 versicolor
## 131          7.4         2.8          6.1         1.9  virginica
## 98           6.2         2.9          4.3         1.3 versicolor
## 33           5.2         4.1          1.5         0.1     setosa
## 145          6.7         3.3          5.7         2.5  virginica
## 103          7.1         3.0          5.9         2.1  virginica
## 88           6.3         2.3          4.4         1.3 versicolor
## 13           4.8         3.0          1.4         0.1     setosa
## 117          6.5         3.0          5.5         1.8  virginica
## 51           7.0         3.2          4.7         1.4 versicolor
```

Vamos a mezclar ambas tablas. Esto producirá una tabla con las dimensiones de la tabla más pequeña (tabla_nueva), sumando las columnas de la tabla_iris. Quedarán duplicadas las columnas pero es un ejemplo.

```
tablamezclada<-merge(tabla_nueva,tabla_iris,by.x=0,by.y=0)

dim(tablamezclada)
```

```
## [1] 25 11
```

```
head(tablamezclada)
```

```
##   Row.names Sepal.Length.x Sepal.Width.x Petal.Length.x Petal.Width.x
## 1       102            5.8           2.7            5.1           1.9
## 2       103            7.1           3.0            5.9           2.1
## 3       112            6.4           2.7            5.3           1.9
## 4       117            6.5           3.0            5.5           1.8
```

```
## 5        13           4.8          3.0           1.4          0.1
## 6       131           7.4          2.8           6.1          1.9
##   Species.x Sepal.Length.y Sepal.Width.y Petal.Length.y Petal.Width.y
## 1 virginica           5.8          2.7           5.1          1.9
## 2 virginica           7.1          3.0           5.9          2.1
## 3 virginica           6.4          2.7           5.3          1.9
## 4 virginica           6.5          3.0           5.5          1.8
## 5    setosa           4.8          3.0           1.4          0.1
## 6 virginica           7.4          2.8           6.1          1.9
##   Species.y
## 1 virginica
## 2 virginica
## 3 virginica
## 4 virginica
## 5    setosa
## 6 virginica
```

Usando funciones que recorren las tablas

Funciones por *fila* usando la "desviación estándar" como ejemplo

```
apply(X = tabla_iris_sinespecies,1,sd)
```

```
##   [1] 2.179449 2.036950 1.997498 1.912241 2.156386 2.230844 1.936276
##   [8] 2.109305 1.822773 2.068816 2.307957 2.016598 2.032035 1.883923
##  [15] 2.566450 2.467117 2.307235 2.143789 2.369775 2.173131 2.238117
##  [22] 2.120338 2.087263 1.995829 1.977161 2.048577 2.019901 2.201515
##  [29] 2.205297 1.950000 1.977161 2.199053 2.338625 2.447277 2.032035
##  [36] 2.135416 2.357082 2.155613 1.851801 2.148643 2.123480 1.796292
##  [43] 1.882153 1.961929 2.070427 1.960230 2.192981 1.941649 2.269178
##  [50] 2.112463 2.371181 2.070427 2.326657 1.855398 2.176388 1.927650
##  [57] 2.002290 1.635033 2.275229 1.626858 1.750000 1.862794 2.181742
##  [64] 2.054872 1.782321 2.234577 1.786057 2.041241 2.155613 1.925920
##  [71] 1.798842 2.027313 2.193931 2.146315 2.163909 2.205297 2.357258
##  [78] 2.201515 1.950000 1.961292 1.888562 1.915724 1.944222 2.046949
##  [85] 1.714643 1.853600 2.224110 2.229163 1.812917 1.822773 1.905037
##  [92] 2.027108 1.966384 1.687207 1.859211 1.903287 1.873277 2.082266
##  [99] 1.658061 1.873277 1.908533 1.869715 2.361320 2.145538 2.095034
## [106] 2.683747 1.544884 2.639918 2.412468 2.173323 1.994994 2.123480
## [113] 2.176388 1.823915 1.678044 1.881489 2.174090 2.544275 2.821790
## [120] 2.165448 2.139120 1.701715 2.820165 2.050000 2.118962 2.483948
## [127] 1.976529 1.919201 2.095034 2.556039 2.621068 2.633597 2.061553
## [134] 2.173131 2.285279 2.555223 1.830073 2.121320 1.865476 2.177728
## [141] 2.033880 2.068010 1.869715 2.142429 1.975686 2.021551 2.075853
## [148] 2.046745 1.791415 1.884144
```

Funciones por *columna* usando "desviación estándar" como ejemplo

```
apply(X = tabla_iris_sinespecies,2,sd)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.8280661    0.4358663    1.7652982    0.7622377
```

Funciones por *columna* usando el *promedio*

```
apply(X = tabla_iris_sinespecies,2,mean)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

Funciones por grupo (agrupando según la información contenida en la columna "Species")

```
tapply(iris$Petal.Length, iris$Species, sd)
```

```
##     setosa versicolor  virginica
##  0.1736640  0.4699110  0.5518947
```

Creación de funciones "personales"

Una función recibe un valor y devuelve una respuesta según el valor que se ingresó

En este ejemplo, la función recibirá un valor al cual llamaremos "lo_que_entra" (puede tener cualquier nombre), y a ese valor le calculará el log2(1+X). El resultado será devuelto al usuario.

```
funcion_ibio<-function(lo_que_entra)
{ salida= log2(lo_que_entra+1)
  return(salida)
}
```

Usando funciones "personales"

```
funcion_ibio(7)
```

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

Análisis complejos

Análisis de componentes Principales (PCA)

En R la función del paquete base para calcular PCA es "prcomp". Hay más paquetes que permiten hacer PCA con mas o menos funciones.

Una vez realizado el cálculo nos gustaría graficar los resultados. Existen muchos paquetes de R que proveen formas de realizar gráficos, uno de ellos es autoplot del paquete ggfortify.

```
#install.packages("ggfortify") #instalación en caso que no lo tengan

library(ggfortify)
```

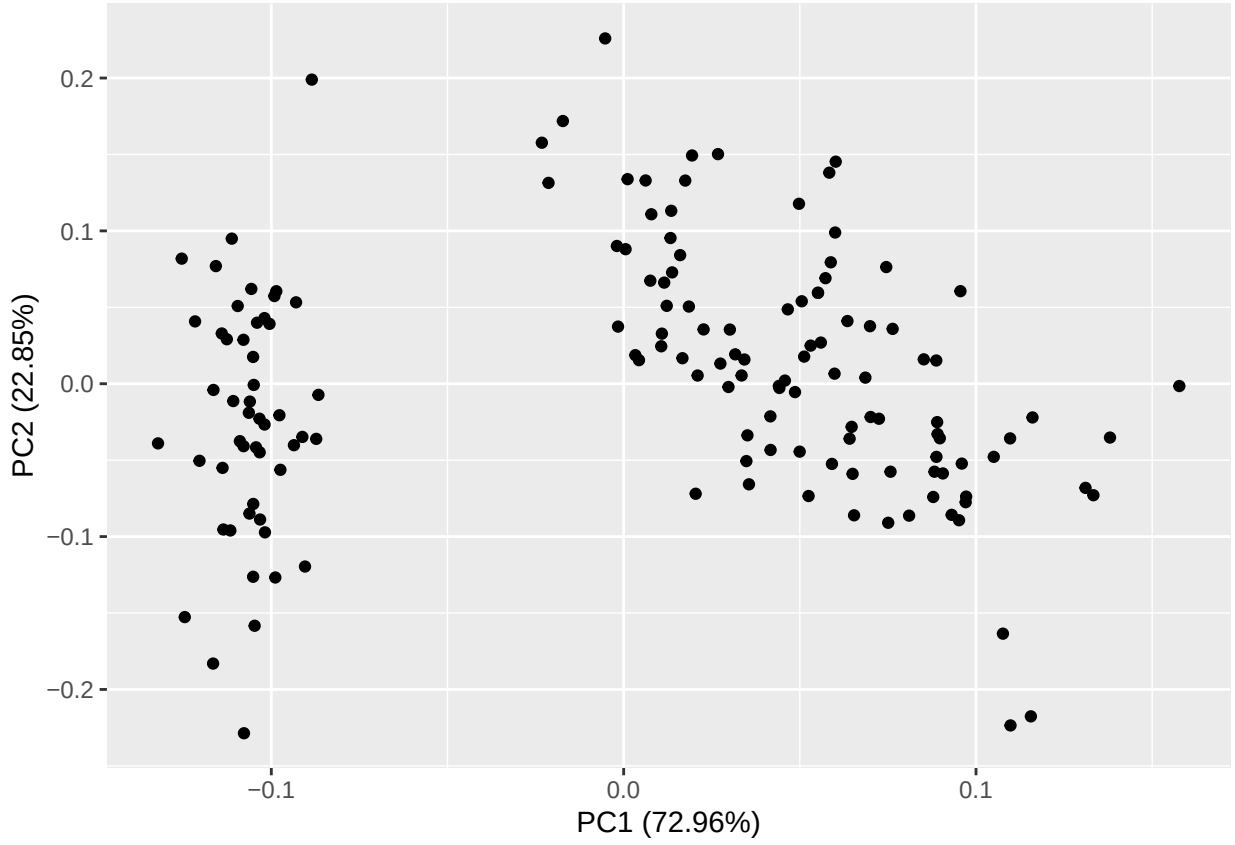
```
## Warning: package 'ggfortify' was built under R version 3.6.1
```

```
## Loading required package: ggplot2
```

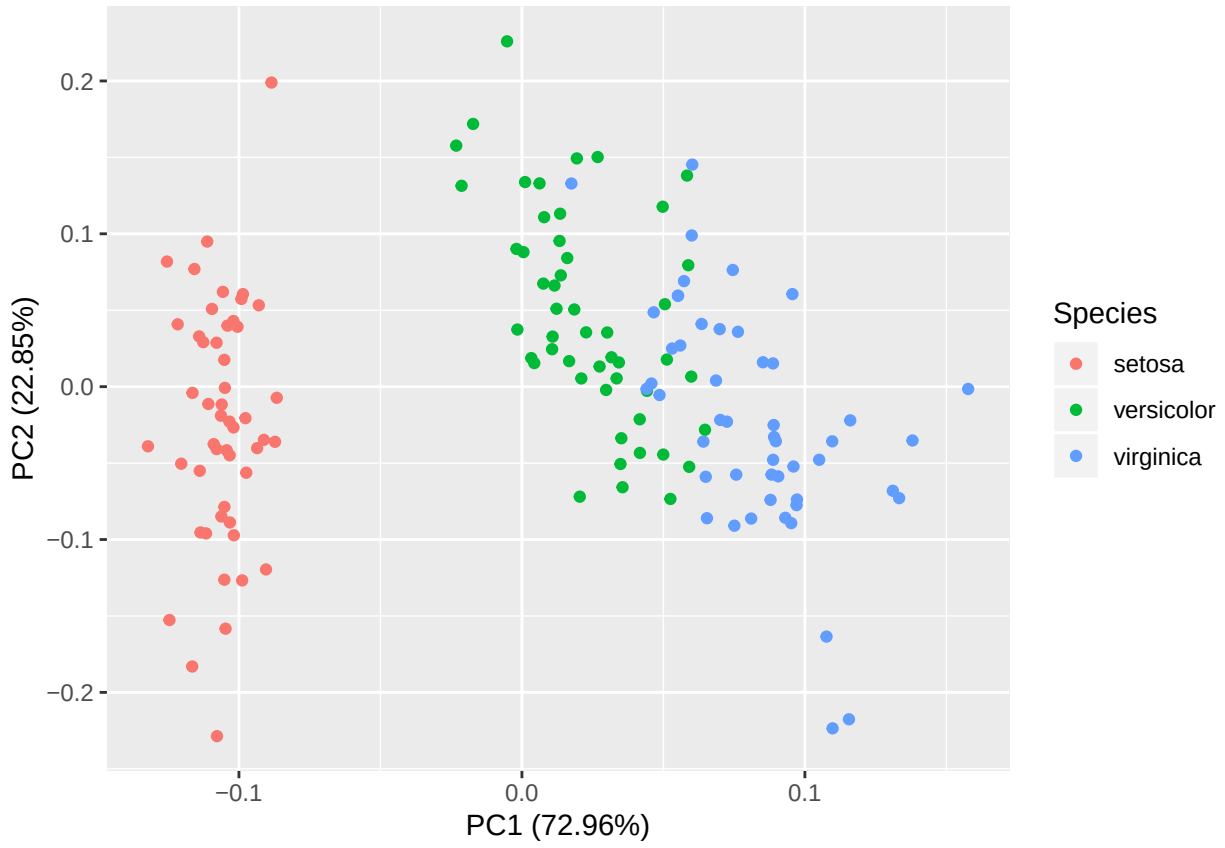
```
pca = prcomp(tabla_iris_sinespecies, scale. = T)

summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

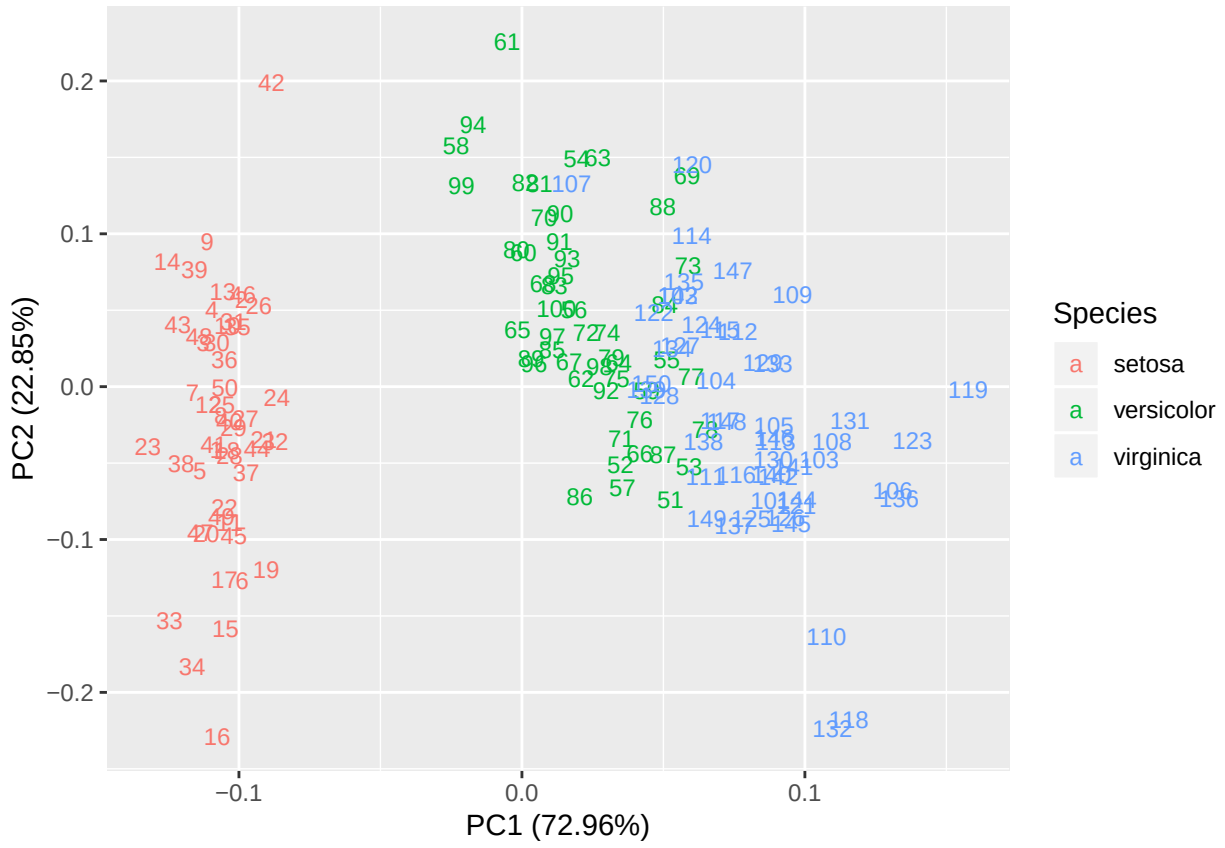
```
autoplot(pca)
```



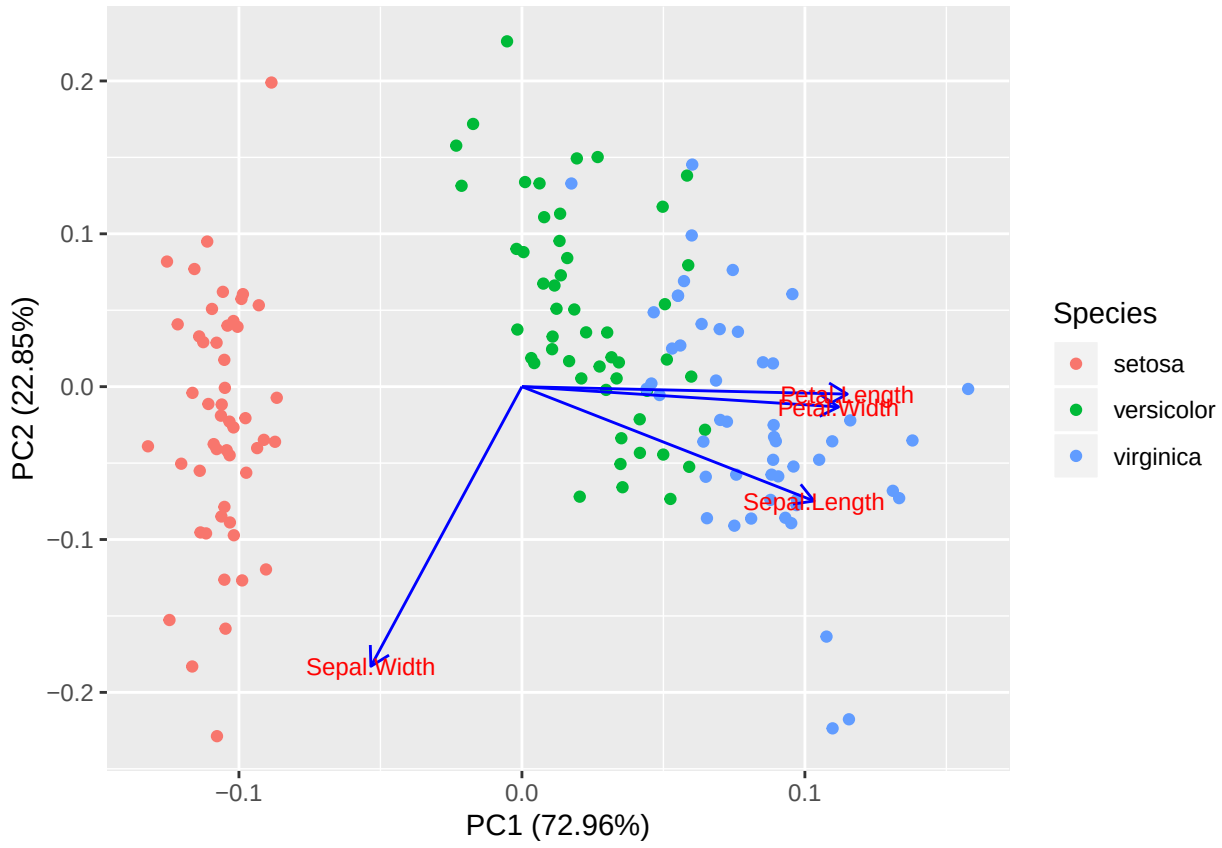

```
autoplot(pca, data = tabla_iris, colour = 'Species')
```




```
autoplot(pca, data = tabla_iris, colour = 'Species', shape = FALSE, label.size = 3)
```




```
autoplot(pca, data = tabla_iris, colour = 'Species',
         loadings = TRUE, loadings.colour = 'blue',
         loadings.label = TRUE, loadings.label.size = 3)
```



Heatmap

Al igual que en el caso de PCA, hay muchas formas de hacer un heatmap.

ACá algunos ejemplos: https://www.r-graph-gallery.com/heatmap.html

En este caso vamos a usar el paquete gplots

```
#install.packages("gplots") #instalación en caso que no lo tengan

library(gplots)
```

```
## Warning: package 'gplots' was built under R version 3.6.1
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

Este comando es para cerrar los graficos en caso que exista una sesión "dev"" abierta

```
dev.off()
```

```
## null device
##           1
```

Con este comando podrán abrir una ventana para la salida gráfica que es independiente del panel principal de RStudio

```
#x11()
```

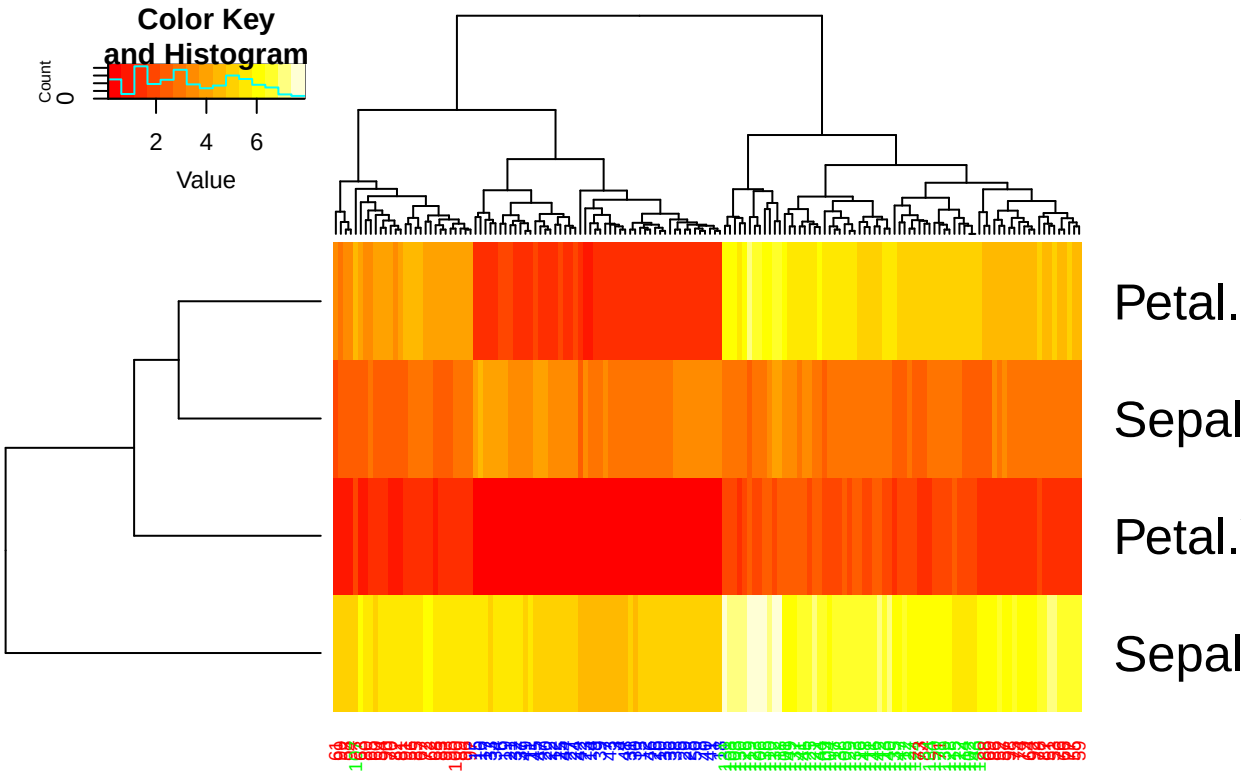
Asignamos los colores según especie para poder distinguirlas en el resultado del heatmap

Como los datos están ordenados, sabemos que el nombre de la especie cambia cada 50 filas por lo cual creamos un vector de colores que cambia cada 50 elementos.

```
colores<-c(rep("blue",50),rep("red",50),rep("green",50))
```

Creamos el heatmap usando heatmap.2 que es una función de gplots

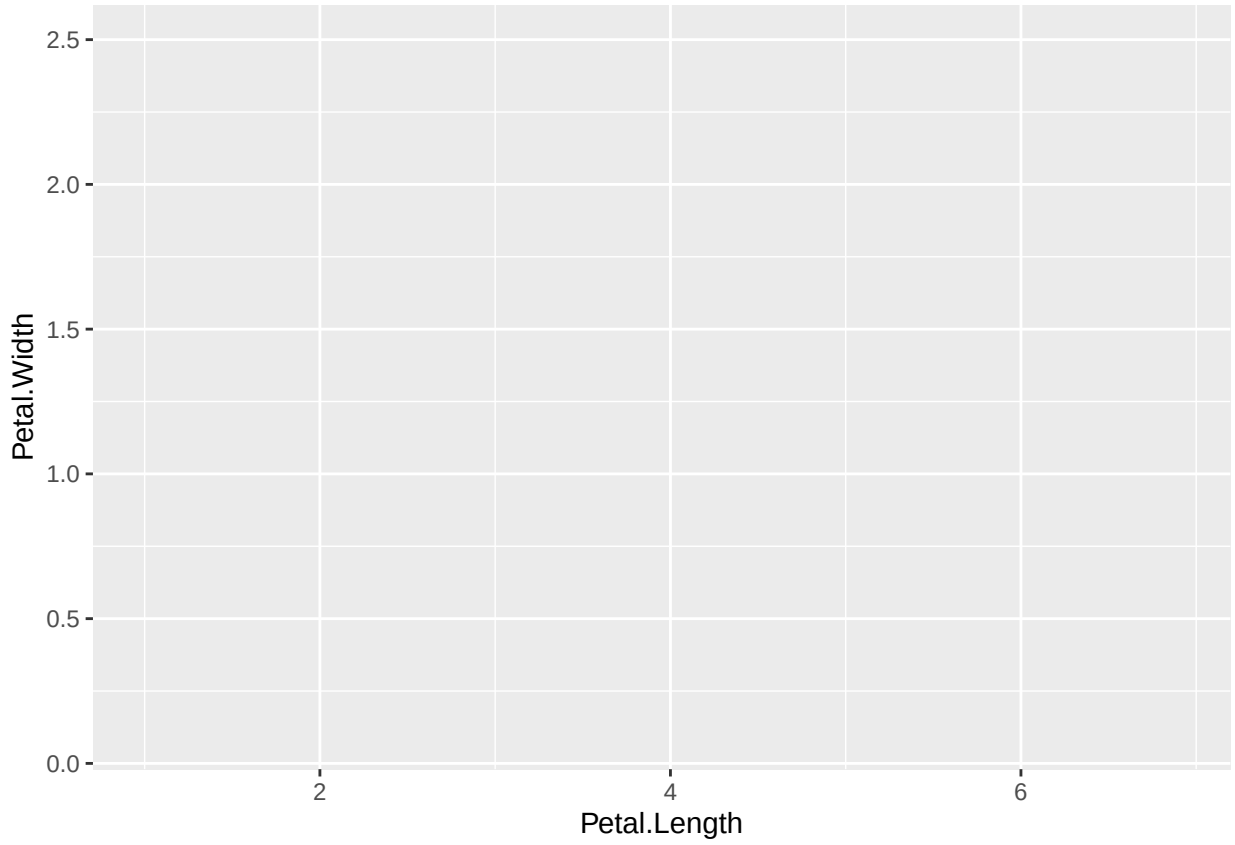
```
heatmap.2(as.matrix(t(tabla_iris_sinespecies)),trace="none",colCol = colores,cex.lab=100)
```



Uso del paquete ggplot2

Este paquete es uno de los más utilizados para prodcir gráficos personalizados gracias a que es posible ir aplicando capas de información sobre el gráfico.

```
#install.packages("ggplot2")
library(ggplot2)

p <-ggplot(iris,aes(x =Petal.Length,y =Petal.Width,colour =Species)) #construye el objeto que será la b
print(p)
```




```
p <-p+geom_point() # se agregan los puntos
print(p)
```


```
p<-p+geom_smooth() # se agregan los límites de confianza
print(p)
```

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


```
dev.off() #resetear la ventana de salida
```

```
## null device
##           1
```

```
p <-ggplot(iris,aes(x =Species,y =Petal.Width,colour=Species)) # gráfico de cajas
p <-p+ geom_boxplot()
print(p)
```