Graficos

Agustín Muñoz González

22/4/2020

Preparamos el entorno.

Limpiamos los registros y attachamos. (setear el directorio de trabajo!)

```
rm(list=ls())
tit=read.csv('titanic.csv', header=T)
attach(tit)
```

¿Cuándo usamos cada gráfico?

Antes que nada mencionar que no todos los tipos de gráficos sirven para lo mismo.

Por ejemplo si queremos ver relaciones entre variables de tipo numérico, un grafico de barras no va a aportar información muy clara, y en cambio un gráfico de dispersión es mas adecuado. Pero si nos interesa ver como se distribuyen los datos de cierta variable numerica en otra variabla categórica entonces el adecuado sería un gráfico de caja, ya que un diagrama de dispersión, de barras o de torta serían simplemente manchas negras (si los datos numéricos son muchos).

Como para tener en la cabeza, los distintos gráficos son adecuados para las siguientes situaciones (a grandes rasgos)

- Histograma: Para estudiar densidad de una variable numérica.
- Gráfico de caja: Para estudiar var categorica vs var numérica. Es decir, la disposición de la numérica en las categorías o grupos de la categórica.
- Diagrama de dispersión: Para estudiar var numérica vs var numérica.
- Gráfico de barras o de torta: Para estudiar la distribución de una (o más) var categórica en el total de datos.

Comando table()

El comando *table()* aplicado a una variable categórica (class=Factor) cuenta la frecuencia (apariciones) de cada categoría.

```
freq.sex=table(sex)
```

Es importante que la clase de las variables categóricas sean factor. Sino deberíamos cambiarlo. Por ejemplo la variable 'survived' es categórica pero no esta cargada como una clase Factor

```
class(survived)
```

```
## [1] "integer"
```

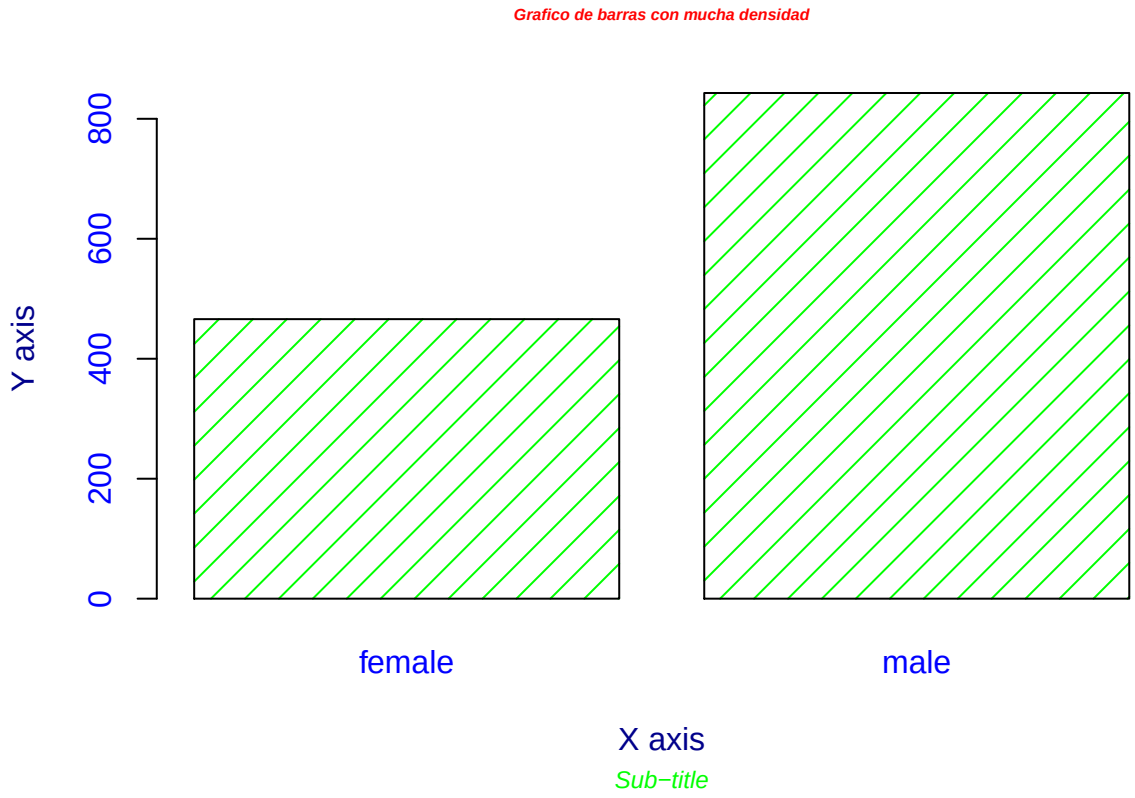
entonces hay que transformarla

```
survived=as.factor(survived)
```

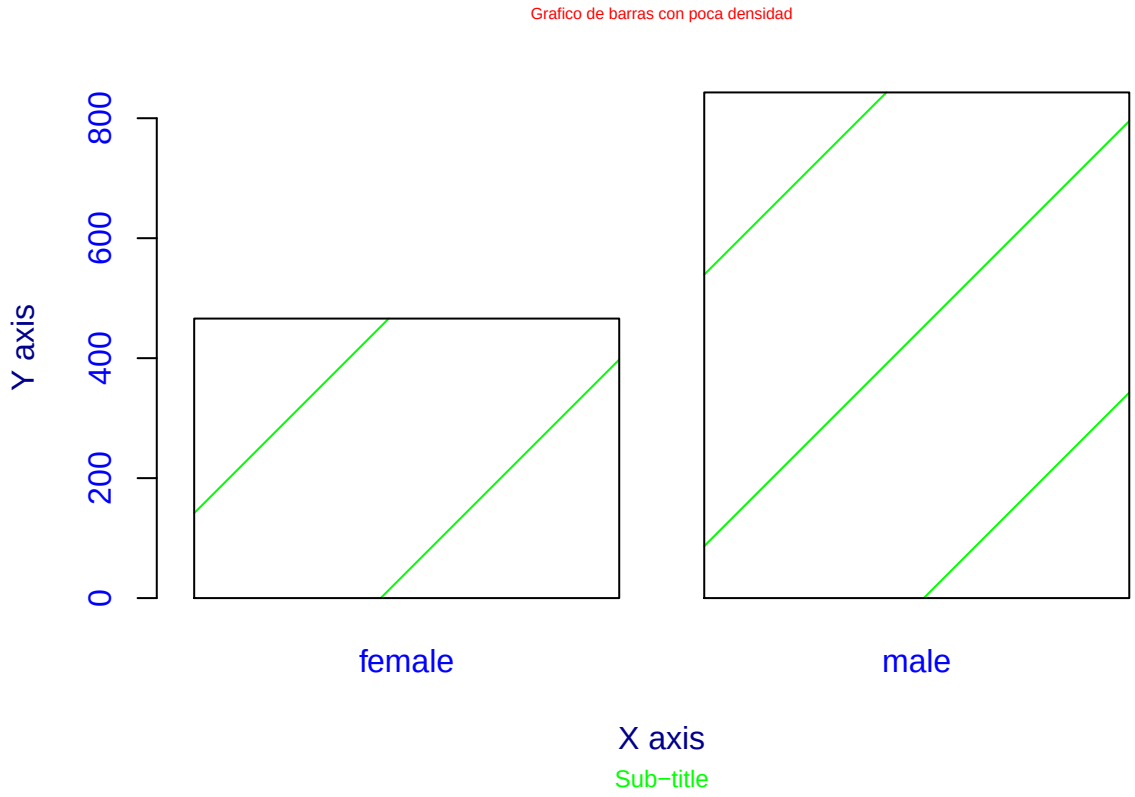
Gráfico de barras - Comando barplot()

El comando *barplot()* realiza un diagrama de barras de una variable categórica. Juguemos un poco con las opciones de graficos.

```
barplot(freq.sex, col='green', density=8, main = "", xlab="", ylab="",
     col.axis="blue")
title(main = "Grafico de barras con mucha densidad", sub = "Sub-title",
      xlab = "X axis", ylab = "Y axis",
      cex.main = 0.5,   font.main= 4, col.main= "red",
      cex.sub = 0.75, font.sub = 3, col.sub = "green",
      col.lab ="darkblue"
      )
```

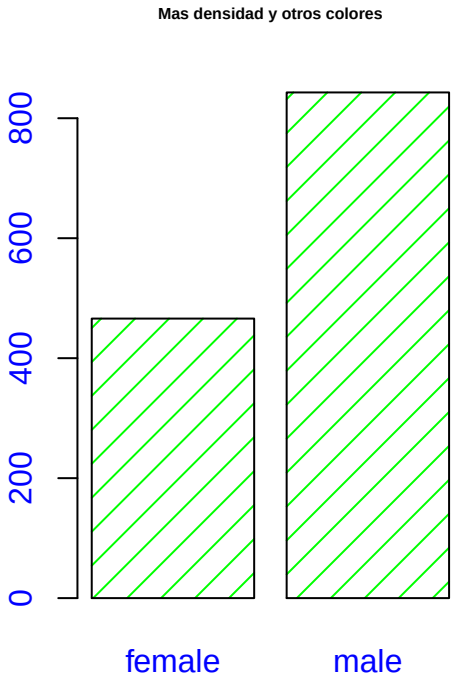


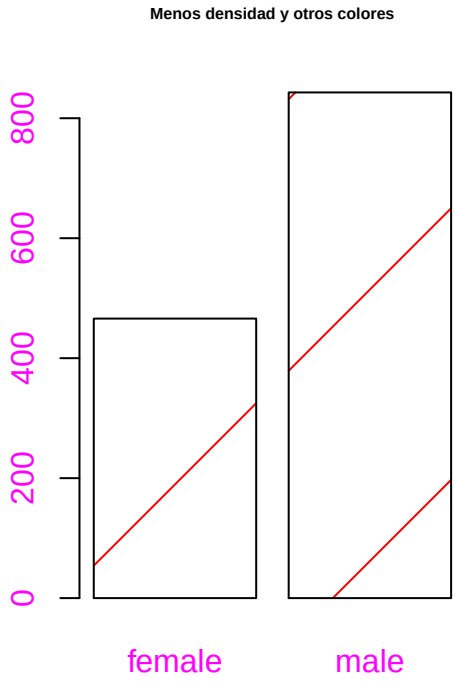

```
# density define la cantidad de lineas diagonales azules.
# cex = character expansion ratio, cambia el tamaño de los textos.
barplot(freq.sex, col='green', density=1, main = "", xlab="", ylab="",
     col.axis="blue")
title(main = "Grafico de barras con poca densidad", sub = "Sub-title",
      xlab = "X axis", ylab = "Y axis",
      cex.main = 0.5,   font.main= 1, col.main= "red",
      cex.sub = 0.75, font.sub = 1, col.sub = "green",
      col.lab ="darkblue"
      )
```



Podemos mostrar varios graficos en la misma ventana.

```
par(mfrow=c(1,2))
# par() permite mostrar varios graficos en la misma ventana, donde c(n,m)
# significa que queremos tener n filas y m columnas (para mostrar nxm graficos).
# En nuestro caso queremos mostrar 2 graficos uno al lado del otro,
# entonces 1 fila 2 columnas.
barplot(freq.sex, col='green', density=8,
        main = "Mas densidad y otros colores", cex.main=0.5, col.axis="blue")
barplot(freq.sex, col='red', density=1,
        main = "Menos densidad y otros colores", cex.main=0.5, col.axis="magenta")
```






```
par(mfrow=c(1,1))
# lo volvemos a dejar como estaba
```

Gráfico de torta - Comando pie()

Vamos a trabajar ahora con la variable 'pclass'. Como no tiene clase Factor, la convertimos.

```
pclass<-as.factor(pclass)
counts.clase<- table(pclass)
counts.clase
```

```
## pclass
##   1   2   3
## 323 277 709
```

Hagamos un gráfico de torta.

```
pie(counts.clase, col=c("blue","green","red"),
main="Grafico de Torta de Clases")
legend("bottomright",
c("1ra clase","2da clase", "3ra clase"),
fill=c("blue","green","red")
)
```

Más de una variable.

Podemos considerar tablas formada por varias variables.

```
counts<- table(sex,pclass)
counts
```

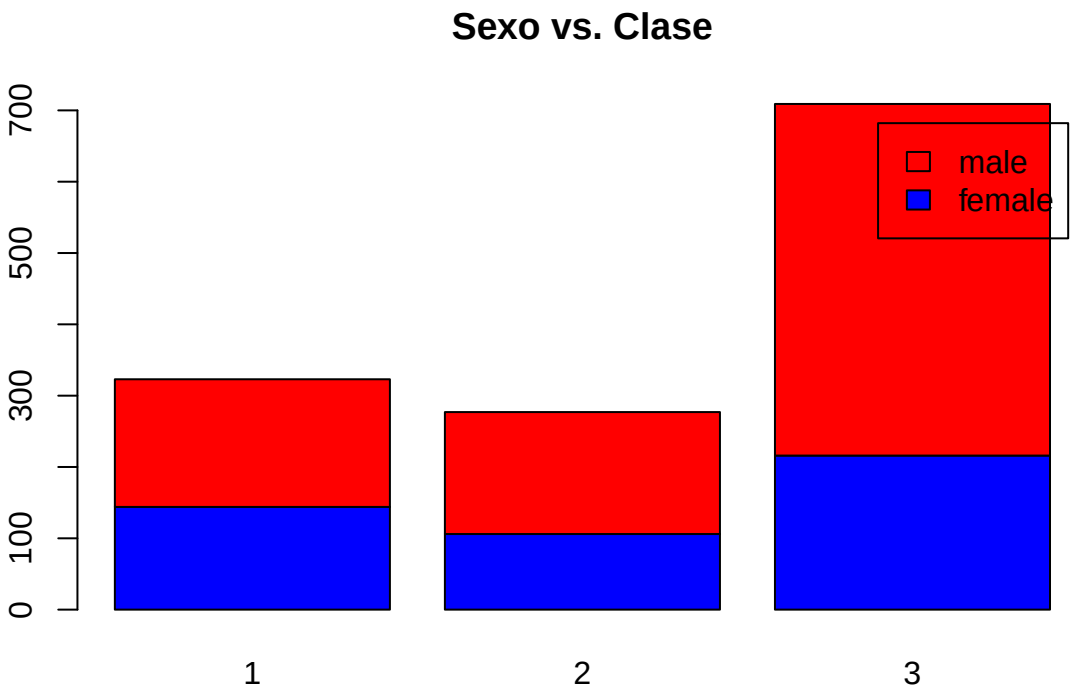
```
##         pclass
## sex        1   2   3
##   female 144 106 216
##   male   179 171 493
```

Y obtenemos una tabla con la informacion las personas dividida por sexo y clase en la que estaban.

Y podemos hacer un gráfico de barras que especifique estas divisiones

```
barplot(counts,col= c( " blue " , "red "),
        main="Sexo vs. Clase",
```

```
legend = rownames( counts ) )
```



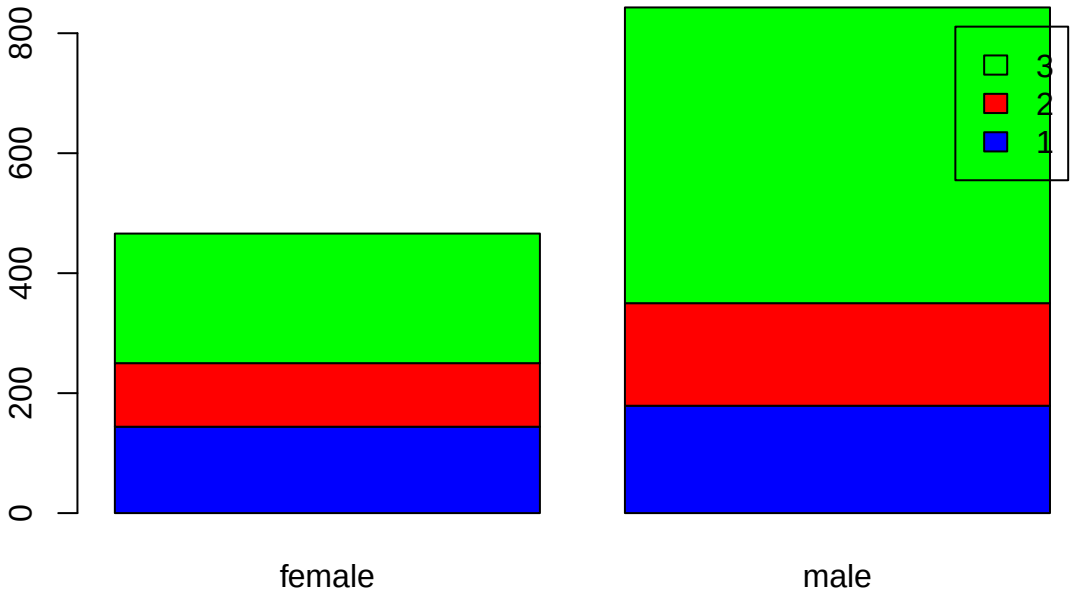
Podemos organizarla al revés (o sea la traspuesta de la matriz anterior)

```
counts_transposed<- table(pclass,sex)
counts_transposed
```

```
##       sex
## pclass female male
##      1    144  179
##      2    106  171
##      3    216  493
```

Y graficamos

```
barplot(counts_transposed,col= c( " blue " , "red ", "green"),
        main="Sexo vs. Clase",
legend = rownames( counts_transposed ) )
```


Sexo vs. Clase
0
200
400
600
800
3
2
1
female
male

Histograma - Comando hist()

Primero un poco de información sobre histogramas.

El histograma es el más conocido de los gráficos para resumir un conjunto de datos cuantitativos o numéricos.

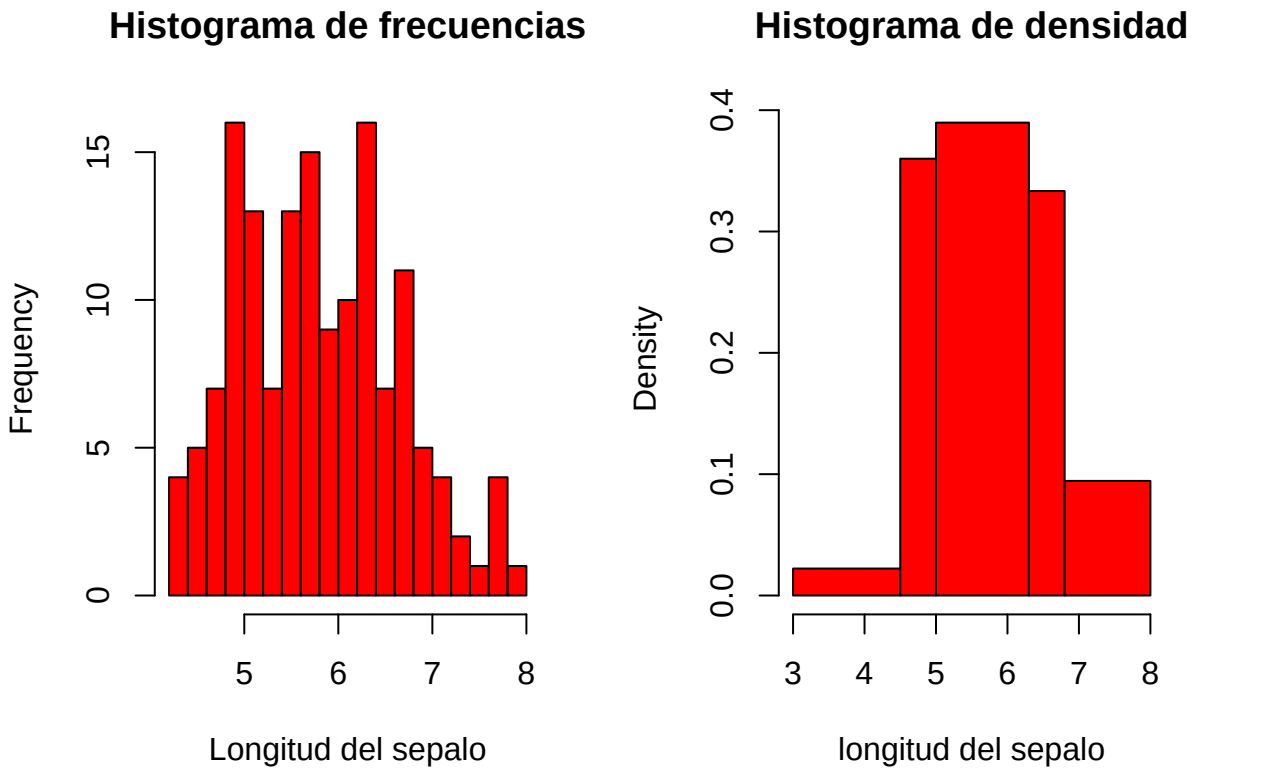
- Para construir un histograma es necesario previamente construir una tabla de frecuencias: dividimos el rango de los n datos en intervalos o clases, que son excluyentes (disjuntos) y exhaustivas (la union da todo el intervalo).
- Contamos la cantidad de datos en cada intervalo o clase i, es decir la frecuencia, f_i, y calculamos la frecuencia relativa: $fr_i = f_i/n$.
- Graficamos el histograma en un par de ejes coordenados representando en las abscisas los intervalos y sobre cada uno de ellos **un rectángulo cuya área es la frecuencia (o frecuencia relativa) de dicho intervalo**.

Vamos a trabajar sobre los datos 'iris'.

```
data("iris")
attach(iris)
```

Veamos dos ejemplos

```
par(mfrow=c(1,2))
hist(Sepal.Length,main="Histograma de frecuencias",col="2",nclass=15,
     xlab="Longitud del sepalo")
# nclass especifica la cantidad de intervalos queremos (i.e. la cantidad de clases.)
# col='2' es col='red'. Cada color tiene un numero asociado.
hist(Sepal.Length,freq=F,main="Histograma de densidad",col="2",
     breaks = c(3,4.5,5,6.3,6.8,8),xlab="longitud del sepalo")
```




```
# En el segundo caso estoy considerando la frecuencia relativa. Es decir, la altura del
# histograma no es el área de la frecuencia sino de la frecuencia relativa.
```

```
# freq: if TRUE, the histogram graphic is a representation of frequencies, the counts
# component of the result; if FALSE, probability densities, component density, are plotted.
# (ver ?hist).
# breaks especifica dónde van a ser los cortes del intervalo.
par(mfrow=c(1,1))
```

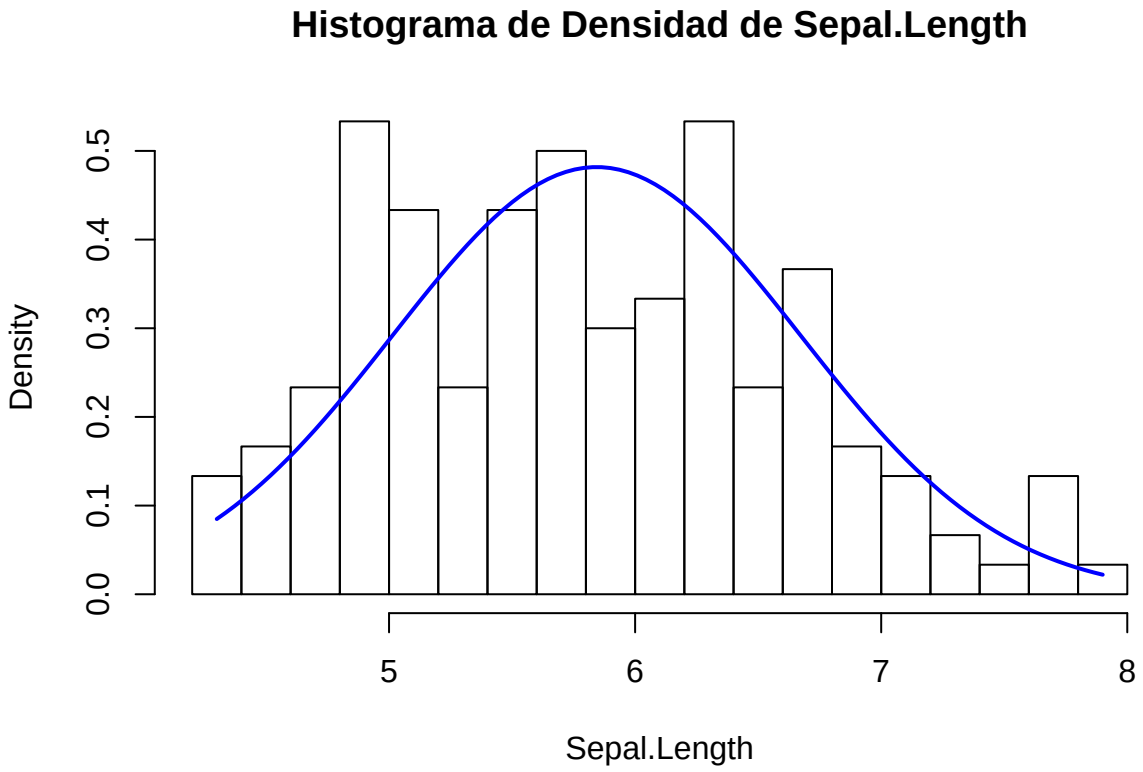
Ahora guardamos los gráficos en un archivo pdf poniendo el códido entre *pdf()* y *graphics.off()*.

```
pdf ("histogramas.pdf ")
par(mfrow=c(1,2))
hist(Sepal.Length,main="Histograma de frecuencias",col="2",nclass=15,
     xlab="Longitud del sepalo")
# nclass especifica la cantidad de intervalos queremos (i.e. la cantidad de clases
# o columnas del histograma.)
# col='2' es col='red'. Cada color tiene un numero asociado.
hist(Sepal.Length,freq=F,main="Histograma de densidad",col="2",
     breaks = c(3,4.5,5,6.3,6.8,8),xlab="longitud del sepalo")
# En el segundo caso estoy considerando la frecuencia relativa. Es decir, la altura del
# histograma no es el área de la frecuencia sino de la frecuencia relativa.
# freq: if TRUE, the histogram graphic is a representation of frequencies, the counts
# component of the result; if FALSE, probability densities, component density, are plotted.
# (ver ?hist).
# breaks especifica dónde van a ser los cortes del intervalo si es una lista de valores
# y especifica la cantidad de intervalos (# col del hist) si es un entero.
par(mfrow=c(1,1))
graphics.off()
```

En ocasiones es útil superponer el histograma y la curva de densidad.

```
media.es<-mean(Sepal.Length)
desvio.es<-sd(Sepal.Length)
# R entiende como desvio estandar (standard deviation) como la varianza matematica.
# Y como varianza entiende a la varianza matematica sin la raiz cuadrada.
# O sea: sq()=varianza_matematica, var()=varianza_matematica^2 --> var()=sq()^2.
grilla<-seq(range(Sepal.Length)[1],
range(Sepal.Length)[2],length=100)
# Armamos los puntos por los que va a pasar la curva densidad.
funn<-dnorm(grilla,media.es,desvio.es)
# Calculamos la densidad normal sobre grilla.

#Y graficamos ambas.
hist(Sepal.Length,nclass=15,freq=F,
main="Histograma de Densidad de Sepal.Length")
lines(grilla,funn,col="blue",lwd=2)
```



Por último agrego un ejemplo que puede ser útil cuando queremos especificar las divisiones de algún eje en el gráfico (en este ejemplo, el eje x).

```
dat <- sample(100, 1000, replace=TRUE)
hist(dat, xaxt='n')
axis(side=1, at=seq(0,100, 10), labels=seq(0,1000,100))
```

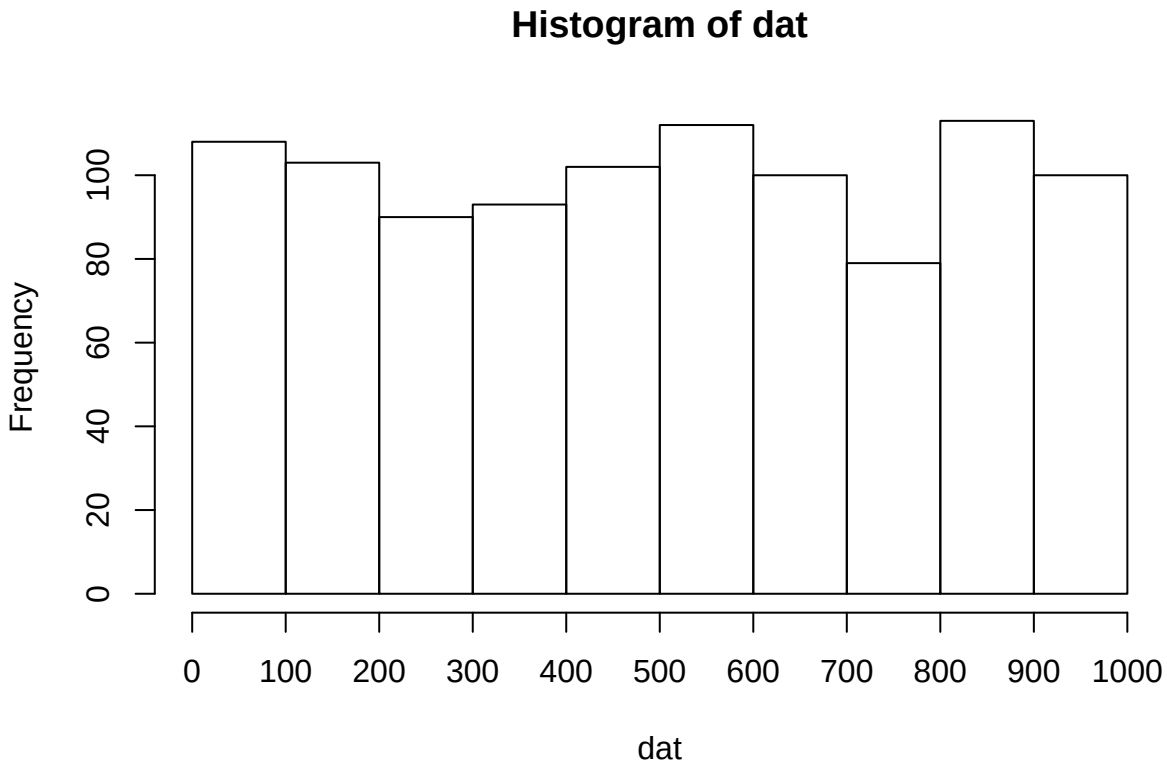


Diagrama de Caja - Comando boxplot()

Los cuartiles y la mediana dividen a la muestra en cuatro partes igualmente pobladas: 25% de la muestra en cada una de ellas.

Entre Q1 y Q2 se halla el 50% central de los datos y el rango de estos es el intervalo intercuartil IQR=Q3-Q1.

Observemos qué porcentaje de datos hay a la izquierda de Q1 (25%), a la derecha de Q3 (25%), entre Q1 y Q3 (50%), entre Q1 y el máximo (75%), y entre el mínimo y Q3 (75%).

Resultan muy útiles para describir la muestra las siguientes medidas

- Mínimo
- Q1 cuartil inferior
- Q2 mediana
- Q3 cuartil superior
- Máximo

En general:

- El comando *quantile(data, 0.25)* devuelve el cuartil Q1.
- El comando *quantile(data, 0.75)* devuelve el cuartil Q3.
- Uno puede calcular el cuartil *quantile(data, n)* para cualquier $0 \leq n \leq 1$ (notar que quantile(data, 0)=min(data), quantile(data, 1)=max(data)).
- Y IQR(data) es el Rango intercuartil Q3-Q1.

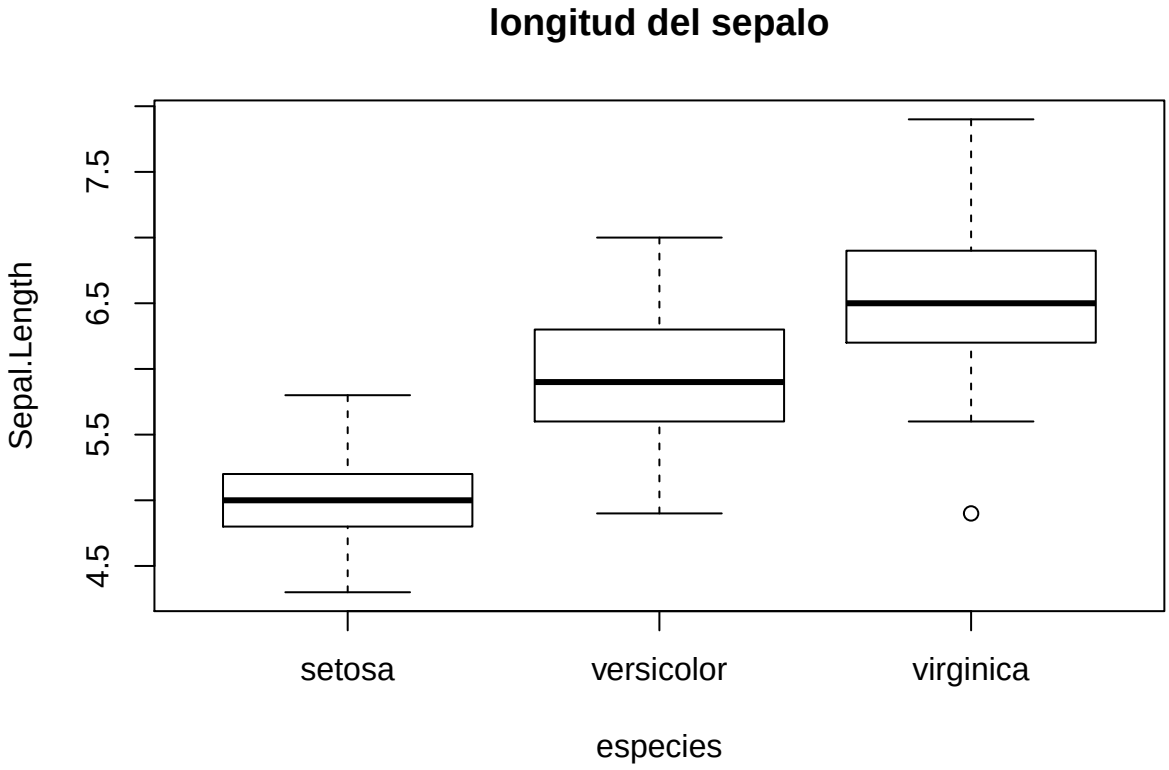
La media es el promedio, intuitivamente es el valor al que 'tiende' la muestra. La mediana es un valor respecto del cual los datos quedan divididos a la mitad en tanto que vamos a tener un 50% de la muestra en el intervalo [minimo, mediana] y la otra mitad en el intervalo [mediana, maximo]. Entonces si por ejemplo la muestra se trata de datos concentrados cerca del mínimo y el máximo no es alcanzado muchas veces pero es un número considerablemente superior al mínimo (tan superior que nos cambie mucho la media) entonces media y mediana serán muy distintas, con mediana << media. Es decir, la media no 've' la densidad, la mediana sí.

El **gráfico de caja** o **boxplot** es un gráfico que representado por un rectángulo (una 'caja') y unos 'bigotes'. Los extremos inferior y superior de la caja son los cuartiles inferior y superior respectivamente, por lo que la longitud de la caja es IQR. La altura de la caja es arbitraria, no dice nada. Dentro de la caja, entre Q1 y Q2 va a estar la mediana (ojo no tiene por qué estar justo a la mitad, porque la distribución en IQR podría concentrarse por ej en Q1 y entonces la mediana estaría cerca de Q1). Ahora bien, tomando como referencia c=1.5*IQR (es decir, 1 vez y media el rango entre Q1 y Q3), marcamos el bigote superior desde el extremo superior de la caja hasta el punto máximo de los datos dentro del intervalo [Q3,Q3+c], y análogamente marcamos el bigote inferior desde el extremo inferior hasta el dato mínimo en el intervalo [Q1-c,Q1]. Notar que estos 'maximos' y 'minimos' relativos en los bigotes no tienen por qué ser los máximos y mínimos totales (digamos si max o min total están a mas de 1 vez y media IQR, max rel y min rel van a ser distintos a los totales). Los puntos por fuera de los bigotes (i.e. por fuera de una vez y media IQR) se llaman 'outliers' y se marcan con puntos.

Si hacemos el boxplot de una muestra de datos con distribución normal, la caja representa al centro de la campana y los bigotes representan la dispersión de la campana (donde los puntos outliers son los puntos de la campana donde ya no queda masa, i.e. hay muy poca probabilidad de obtener esos puntos).

Ejemplo.

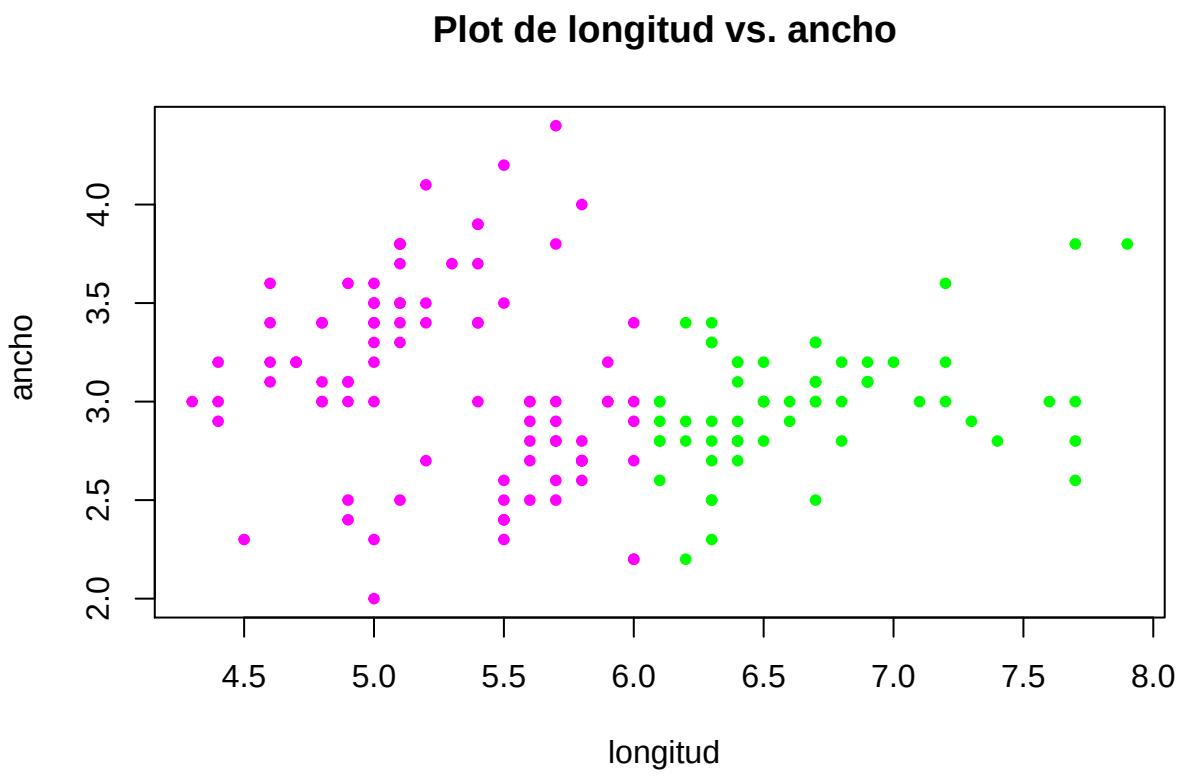
```
boxplot(Sepal.Length~Species, xlab="especies",
main="longitud del sepalo")
```


longitud del sepalo
Sepal.Length
4.5
5.5
6.5
7.5
setosa
versicolor
virginica
especies

Gráficos de dispersión o scatterplot.

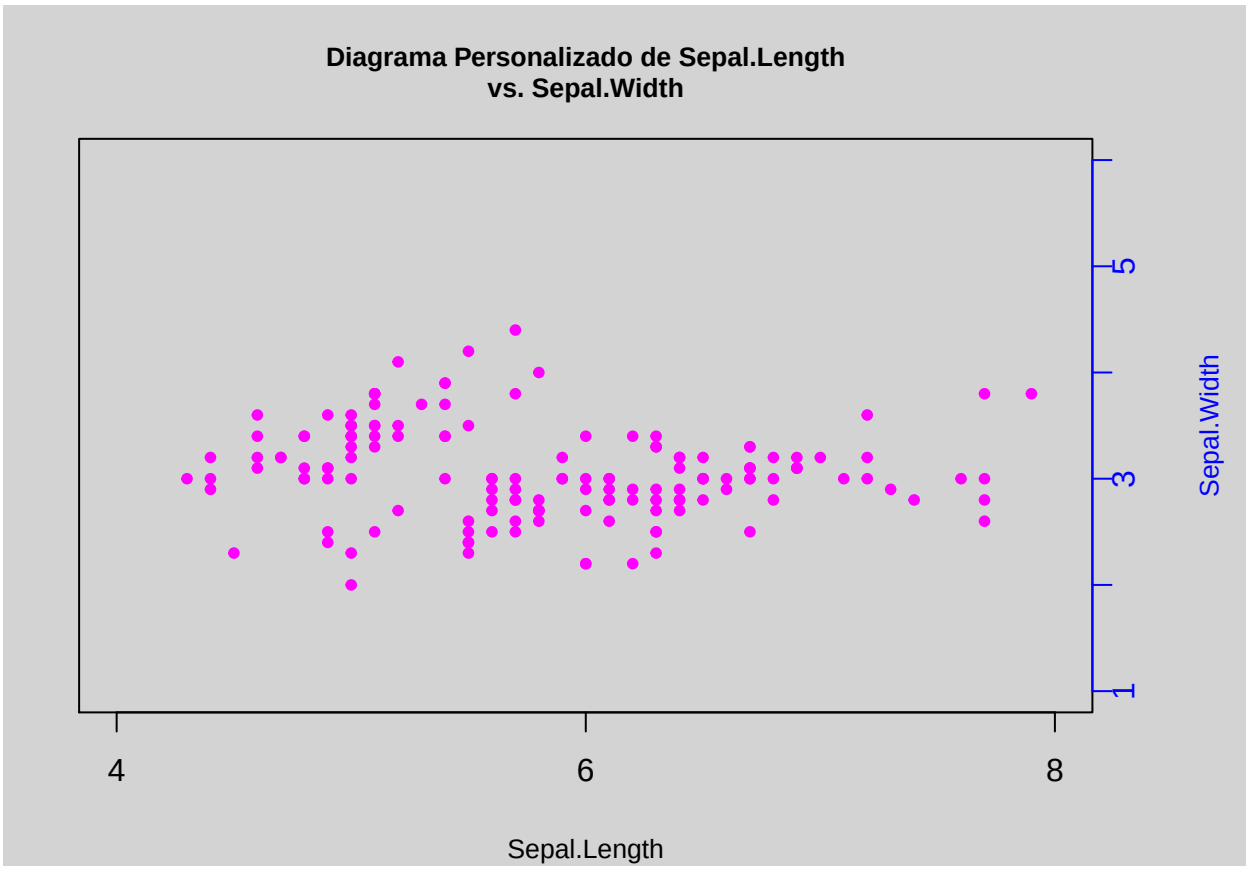
Estos gráficos son útiles para estudiar como se relacionan distintas variables de los datos. Diagrama de dispersión de Sepal.Length vs. Sepal.Width

```
plot(Sepal.Length,Sepal.Width,xlab =" longitud", ylab =" ancho",
main =" Plot de longitud vs. ancho",pch=16,type="n")
#solo graficamos la caja
points(Sepal.Length[Sepal.Length<=6],Sepal.Width[Sepal.Length<=6],
pch=20,col="magenta")
points(Sepal.Length[Sepal.Length>6],Sepal.Width[Sepal.Length>6],
pch=20,col="green")
```



O un poco mas personalizado

```
par(bg="lightgray",mar=c(4,2,3.5, 4))
#c(bottom, left, top, right) defalut es c(5, 4, 4, 2) + 0.1.
plot(Sepal.Length,Sepal.Width,type="n",xlim=c(4,8),
ylim=c(1,6),xlab="", ylab="",xaxt="n", yaxt="n")
#solo graficamos la caja
points(Sepal.Length,Sepal.Width,pch=20,col="magenta")
#solo graficamos los puntos con el simbolo deseado
#Ahora nos encargamos de los ejes
axis(1,c(4,6,8),cex=2)
mtext("Sepal.Length",side=1,cex=0.8,line=3)
axis(4,cex=0.8,col="blue",labels=FALSE)
mtext(c(1,3,5),side=4,at=c(1,3,5),col="blue",line=0.3)
mtext("Sepal.Width",side=4,cex=0.8,line=2.5,col="blue")
#titulo
title("Diagrama Personalizado de Sepal.Length
vs. Sepal.Width",cex.main=0.8)
```



Veamos otro ejemplo. Primero repasemos un poco dos formas de construir matrices por filas o columnas.

```
x<- 1:4 ; y<- 11:14
A<- cbind(x,y) # pego por columnas
A
```

```
##      x  y
## [1,] 1 11
## [2,] 2 12
## [3,] 3 13
## [4,] 4 14
```

```
B<-rbind(x,y) #pego por filas
B
```

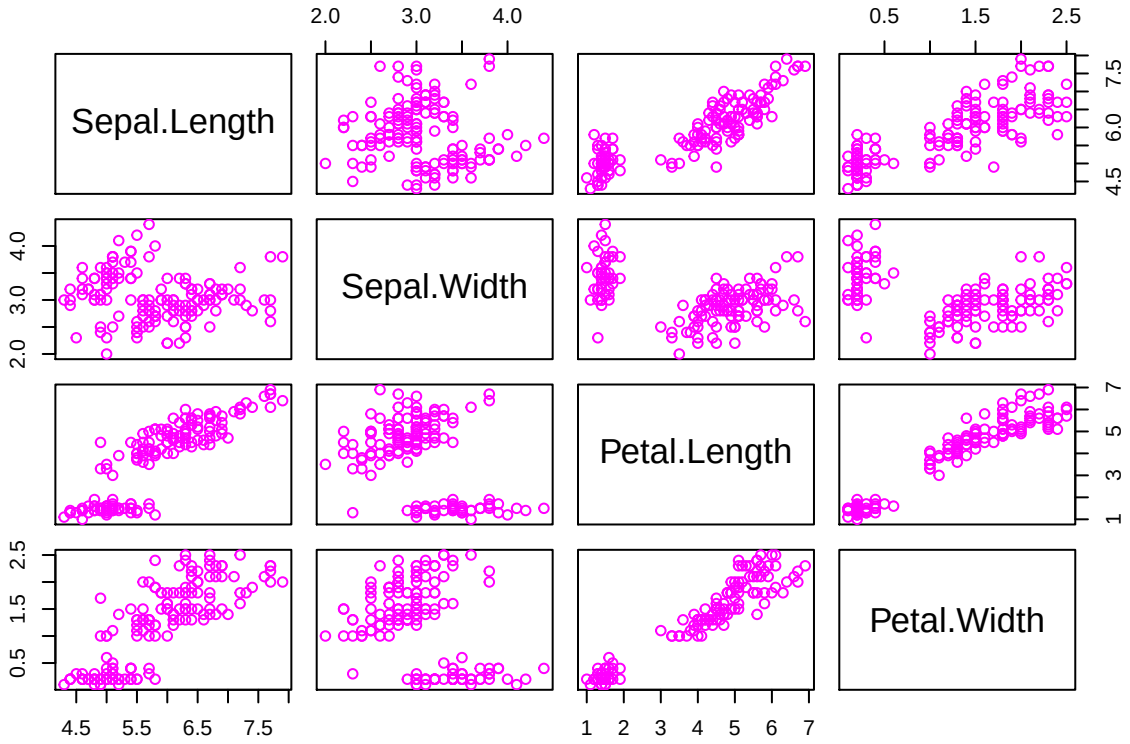
```
##   [,1] [,2] [,3] [,4]
## x    1    2    3    4
## y   11   12   13   14
```

Ahora consideremos la matriz formada por las siguientes columnas de datos de iris.

```
SUB<-cbind( Sepal.Length, Sepal.Width, Petal.Length, Petal.Width )
```

Tenemos el siguiente diagrama de dispersión para todas las variables de SUB.

```
pairs(SUB,col="magenta")
```



Limpiamos registros y desattachamos.

```
rm(list=ls())
detach(iris)
```