Graficos basicos en R

Jose Manuel Pinzon / Laboratorio Fintrade

20/9/2021

Índice

Una de las grandes ventajas del Software R es la gran capacidad que tiene para la creación de gráficos, es un programa diseñado y empleado primordialmente para realizar análisis estadístico y construir gráficos tanto para datos cuantitativos como cualitativos. El objetivo de esta guía es dar una introducción a la creación de gráficos basicos y sencillos con Rbase.

Se explicará como crear y personalizar los siguientes tipos de gráficos:

1. Histogramas (Datos cuantitativos)
2. Boxplot (Datos cuantitativos)
3. Diagramas de barras (Datos cuantitativos)
4. Diagramas de torta (Datos cuantitativos)
5. Graficos de dispersión (Datos cuantitativos, mirar la relación de variables cuantitativas)

Base de datos.

Para el desarrollo de esta guía se va a trabajar con la base de datos *wage2.xlsx*. Esta base de datos se encuentra en el repositorio de Github Fintrade2020. Para acceder ingrese al siguiente link https://github.com/Fintrade2020/R_Basico. Una vez haya ingresado a este link dirígase a la carpeta *clases*, allí encontrará el archivo, por favor descárguelo.

Este archivo de excel cuenta con 935 observaciones y 12 variables, de las cuales 8 son cuantitativas y 4 son cualitativas.

- **wage:** Salario. (Variable cuantitativa)
- **hours:** Horas de trabajo a la semana. (Variable cuantitativa)
- **IQ:** Puntaje IQ. (Variable cuantitativa)
- **KWW:** Puntaje de conocimiento del area. (Variable cuantitativa)
- **educ:** Años de estudio. (Variable cuantitativa)
- **exper:** Años de experiencia. (Variable cuantitativa)
- **tenure:** Años en el mismo trabajo. (Variable cuantitativa)
- **age:** Edad. (Variable cuantitativa)
- **married:** Estado civil. 1 - Casado. 0 - Soltero. (Variable cualitativa)
- **black_:** Color de piel. 1 - Negro. 0 - Blanco.
- **south:** Lugar de residencia. 1 - Sur. 0 - Norte
- **urban:** Lugar de residencia. 1 - Urbano. 0 - Rural.

Cargar base de datos

```
library(readxl)
Wage2 <- read_excel("Wage2.xlsx") ##Importar archivo.
str(Wage2) #Caracteristicas del archivo.
```

```
## tibble [935 x 12] (S3: tbl_df/tbl/data.frame)
##  $ wage   : num [1:935] 769 808 825 650 562 ...
##  $ hours  : num [1:935] 40 50 40 40 40 40 40 40 45 40 ...
##  $ IQ     : num [1:935] 93 119 108 96 74 116 91 114 111 95 ...
##  $ KWW    : num [1:935] 35 41 46 32 27 43 24 50 37 44 ...
##  $ educ   : num [1:935] 12 18 14 12 11 16 10 18 15 12 ...
##  $ exper  : num [1:935] 11 11 11 13 14 14 13 8 13 16 ...
##  $ tenure : num [1:935] 2 16 9 7 5 2 0 14 1 16 ...
##  $ age    : num [1:935] 31 37 33 32 34 35 30 38 36 36 ...
##  $ married: num [1:935] 1 1 1 1 1 1 0 1 1 1 ...
##  $ black  : num [1:935] 0 0 0 0 0 1 0 0 0 0 ...
##  $ south  : num [1:935] 0 0 0 0 0 0 0 0 0 0 ...
##  $ urban  : num [1:935] 1 1 1 1 1 1 1 1 0 1 ...
```

Al cargar la base de datos, se puede evidenciar que las variables categóricas también están definidas como tipo numerico. A continuación se va a ajustar el tipo de dato a **Factor**

```
Wage2$married <- as.factor(Wage2$married)
Wage2$black <- as.factor(Wage2$black)
Wage2$south <- as.factor(Wage2$south)
Wage2$urban <- as.factor(Wage2$urban)
str(Wage2) #Caracteristicas DF
```

```
## tibble [935 x 12] (S3: tbl_df/tbl/data.frame)
##  $ wage   : num [1:935] 769 808 825 650 562 ...
##  $ hours  : num [1:935] 40 50 40 40 40 40 40 40 45 40 ...
##  $ IQ     : num [1:935] 93 119 108 96 74 116 91 114 111 95 ...
##  $ KWW    : num [1:935] 35 41 46 32 27 43 24 50 37 44 ...
##  $ educ   : num [1:935] 12 18 14 12 11 16 10 18 15 12 ...
##  $ exper  : num [1:935] 11 11 11 13 14 14 13 8 13 16 ...
##  $ tenure : num [1:935] 2 16 9 7 5 2 0 14 1 16 ...
##  $ age    : num [1:935] 31 37 33 32 34 35 30 38 36 36 ...
##  $ married: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 1 2 2 2 ...
##  $ black  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ south  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ urban  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 2 ...
```

El comando **summary** le va a dar un breve resumen estadistico para las variables cuantitativas, en donde encontrará el valor mínimo, máximo, la media y resumen de los 5 Números para cada variable del DataFrame. Si usted tiene variables categoricas marcadas como factor le hará un conteo de los datos de esa variable.

```
summary(Wage2) #Resumen DF
```

```
##       wage            hours             IQ             KWW
##  Min.   : 115.0   Min.   :20.00   Min.   : 50.0   Min.   :12.00
##  1st Qu.: 669.0   1st Qu.:40.00   1st Qu.: 92.0   1st Qu.:31.00
##  Median : 905.0   Median :40.00   Median :102.0   Median :37.00
##  Mean   : 957.9   Mean   :43.93   Mean   :101.3   Mean   :35.74
##  3rd Qu.:1160.0   3rd Qu.:48.00   3rd Qu.:112.0   3rd Qu.:41.00
##  Max.   :3078.0   Max.   :80.00   Max.   :145.0   Max.   :56.00
##       educ           exper           tenure            age        married
##  Min.   : 9.00   Min.   : 1.00   Min.   : 0.000   Min.   :28.00   0:100
##  1st Qu.:12.00   1st Qu.: 8.00   1st Qu.: 3.000   1st Qu.:30.00   1:835
##  Median :12.00   Median :11.00   Median : 7.000   Median :33.00
##  Mean   :13.47   Mean   :11.56   Mean   : 7.234   Mean   :33.08
##  3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:11.000   3rd Qu.:36.00
##  Max.   :18.00   Max.   :23.00   Max.   :22.000   Max.   :38.00
##  black   south   urban
##  0:815   0:616   0:264
##  1:120   1:319   1:671
##
##
##
##
```

Por último se va a fijar la base de datos con el comando **attach**, con este comando se indica que todas las variables que se van a trabajar se encuentran en esta base de datos.

```
attach(Wage2)
```

Argumentos generales.

Por lo general los graficos son muy sencillos de hacer, con el comando normal se obtiene un gráfico muy básico, sin embargo, estos comandos tienen argumentos con los cuales se pueden modificar algunas caracteristicas del gráfico. Algunos de esos argumentos son los siguientes:

- **xlab:** Con este argumento se le puede asignar una titulo o nombre al eje X.
- **ylab:** Con este argumento se le puede asignar una titulo o nombre al eje Y.
- **main:** Permite cambiar el titulo del gráfico.
- **col:** Este argumento permite poner colores en el gráfico.
- **ylim:** Se usa para modiicar los limites o rangos del eje Y.
- **ylim:** Se usa para modiicar los limites o rangos del eje X.

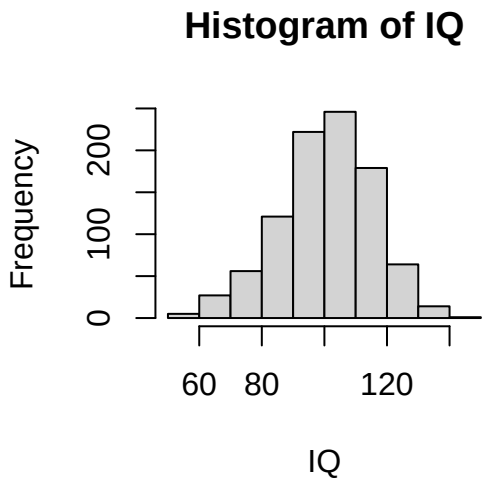
Mas adelante los pondremos en practica.

Histogramas.

Los histogramas son gráficos que indican la frecuencia de los datos mediante una distribución de rangos. Este gráfico solo se puede realizar para variables cuantitativas, es decir, variables medibles, tales como el peso, la estatura, la edad, salario, etc.

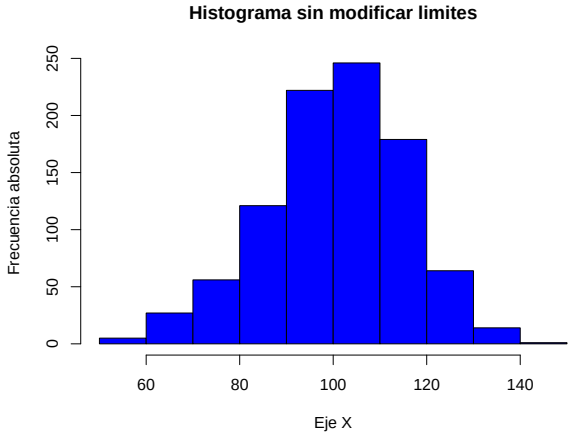
En R, estos histogramas son muy sencillos de hacer, un histograma básico se puede obtener empleando el comando **hist** de la siguiente manera.

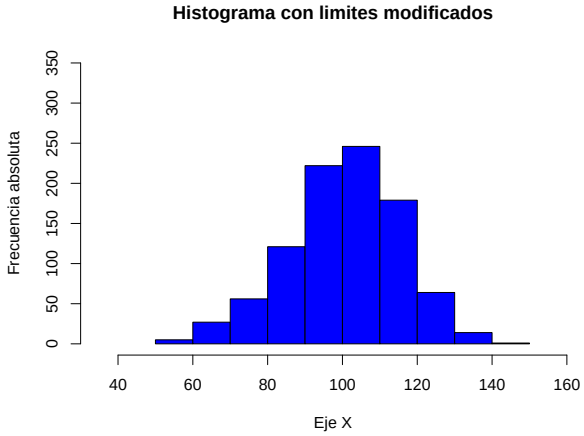
```
hist(IQ) #Histograma para la variable IQ
```



Para modificarlo se pueden usar los argumentos que se mencionaron anteriormente.

```
hist(IQ, xlab = "Eje X", ylab = "Frecuencia absoluta", col = "Blue",
     main = "Histograma sin modificar limites")

hist(IQ, xlab = "Eje X", ylab = "Frecuencia absoluta", col = "Blue",
     main = "Histograma con limites modificados", ylim = c(0,350), xlim = c(35,160))
```





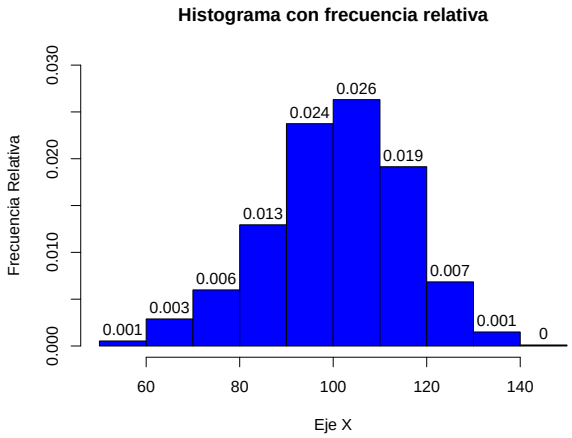
Los Histogramas también tienen otros argumentos adicionales:

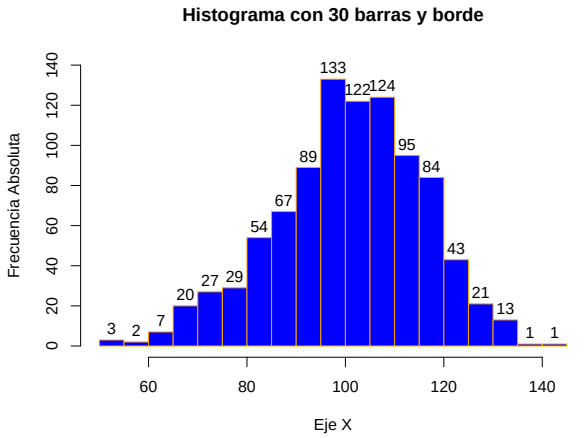
Si usted quiere mostrar los datos en frecuencia relativa o en términos porcentuales puede usar el argumento *freq* = *F*. Este comando también afecta al argumento *labels* = *T*, con el cual se agregan etiquetas sobre cada barra del histograma.

Por defecto, el numero de barras del histograma se determinan con el metodo "Sturges", pero si usted desea modificar esta caracteristica y poner un número específico de barras puede hacerlo con el comando *breaks*.

Si quiere poner el borde de las barras en otro color use *border*.

```
hist(IQ, xlab = "Eje X", ylab = "Frecuencia Relativa", col = "Blue",
     main = "Histograma con frecuencia relativa", freq = F, labels = T, ylim = c(0,0.03))

hist(IQ, xlab = "Eje X", ylab = "Frecuencia Absoluta", col = "Blue",
     main = "Histograma con 30 barras y borde", freq = T, breaks = 30, border = "orange",
     labels = T, ylim = c(0,140))
```

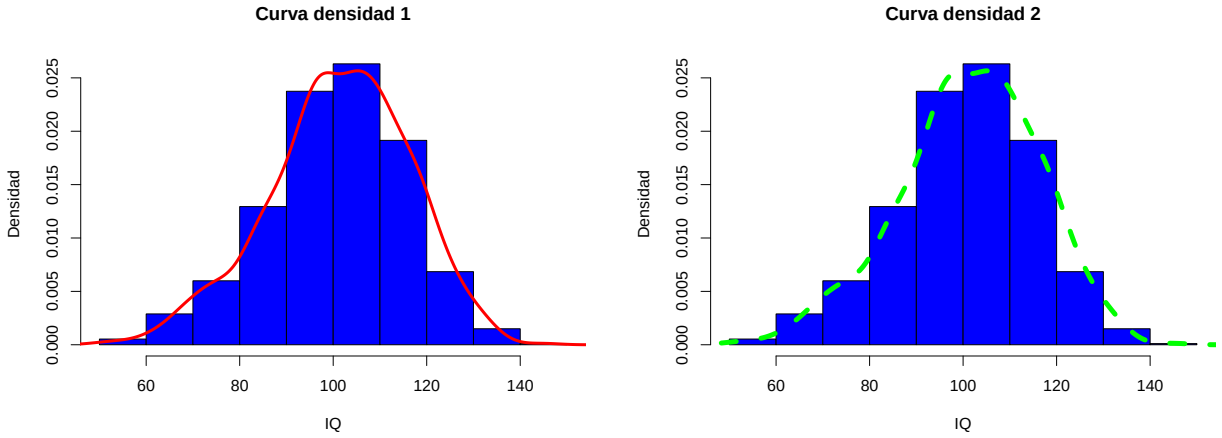




Algunas veces los histogramas vienen con lineas de densidad, esto con el fin de saber la probabilidad de que la variable tome un determinado valor. Para hacer esto en R se agrega el código **lines** después del codigo **hist**.

El primer argumento de este código es definir sobre cual variable se va a calcular la densidad, para ello se pone el argumento *density()* y entre parentesis se va a colocar la variable. Con el argumento *lwd* usted puede definir el tamaño o ancho de la curva. También está el argumento *lty* para establecer el estilo de la linea de densidad.

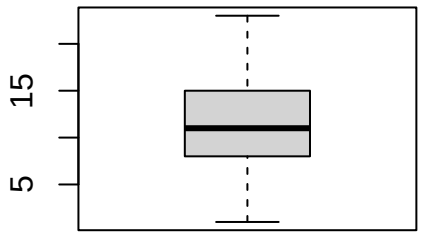
```
hist(IQ, freq = F, main = "Curva densidad 1", ylab = "Densidad", col = "blue")
lines(density(IQ), lwd = 3, col = 'red')


hist(IQ, freq = F, main = "Curva densidad 2", ylab = "Densidad", col = "blue")
lines(density(IQ), lwd = 5, col = 'green', lty = 8)
```



Boxplot

El diagrama de caja o Boxplot también puede ayudar a dar una imagen sobre la distribución de los datos en los quartiles, los datos atípicos y la ubicación de la mediana. Estos graficos se hacen con la función **boxplot**.

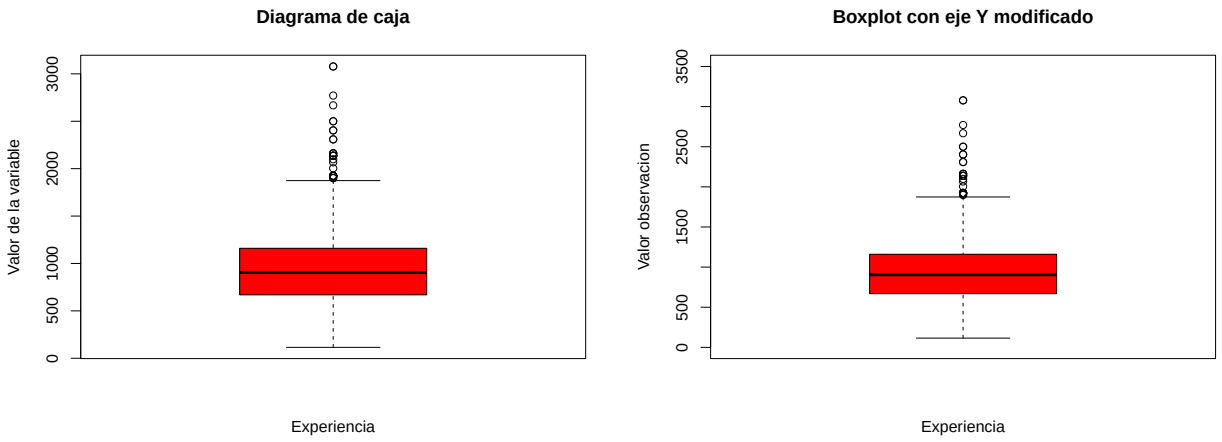
```
boxplot(exper)
```



A este diagrama también se le pueden aplicar los argumentos generales que se describieron al inicio de la guía.

```
boxplot(wage, xlab = "Experiencia", ylab = "Valor de la variable", col = "Red",
     main = "Diagrama de caja")

boxplot(wage, xlab = "Experiencia", ylab = "Valor observacion", col = "Red",
```

```
     main = "Boxplot con eje Y modificado", ylim = c(0,3500))
```



La función **boxplot** también permite modificar algunos atributos propios:

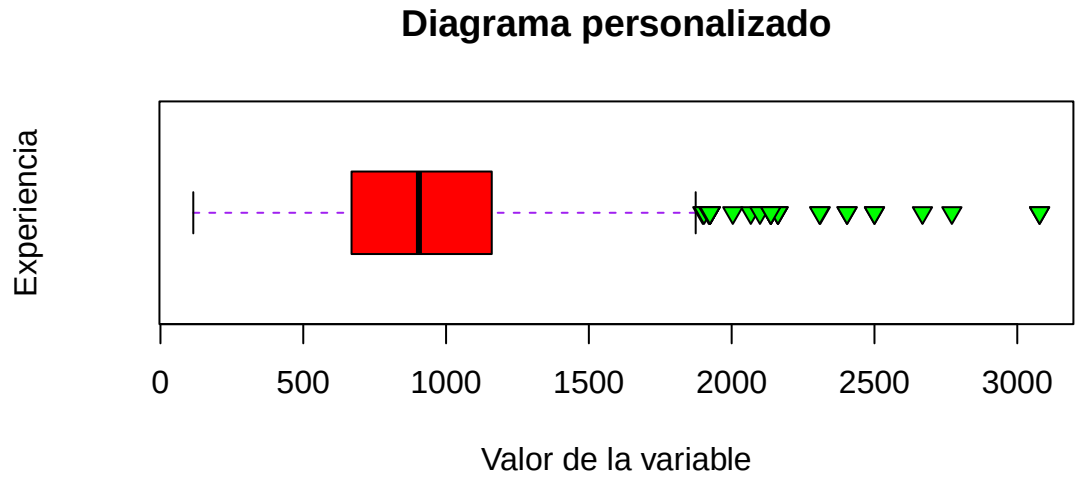
El argumento *horizontal* = *T* permite mostrar las cajas de modo horizontal.

Si quiere agrandar un poco los bordes de la caja y de los datos atípicos lo puede hacer con el argumento *lwd*.

Con el argumento *outpch* se pueden cambiar los símbolos que representan los valores atípicos. Para cambiar el color de dicho símbolo use al argumento *outbg*.

Para personalizar los bigotes de la caja use los argumentos *whiskcol* y *whisklty*, con los cuales usted podrá definir el color y tipo de linea, respectivamente.

```
boxplot(wage, xlab = "Valor de la variable", ylab = "Experiencia", col = "Red",
     main = "Diagrama personalizado", horizontal = T, outpch = 25, outbg = "green",
     whisklty = 2, whiskcol = "purple")
```



También sirve para comparar la distribución de los datos de una variable cuantitativa y una variable cualitativa. En el siguiente ejemplo se va a comparar si el color piel (black) afecta al salario (wage). Primero se coloca la variable cuantitativa y luego la variable categórica.

Si desea poner colores diferentes a cada caja puede hacerlo con la función col, especificando los colores que desea poner.

```
boxplot(wage ~ black, col = "bisque", main = "Un color y lineas grandes", lwd = 2)

boxplot(wage ~ black, col = c("brown", "yellow"), main = "Varios colores", outpch = 7)
```

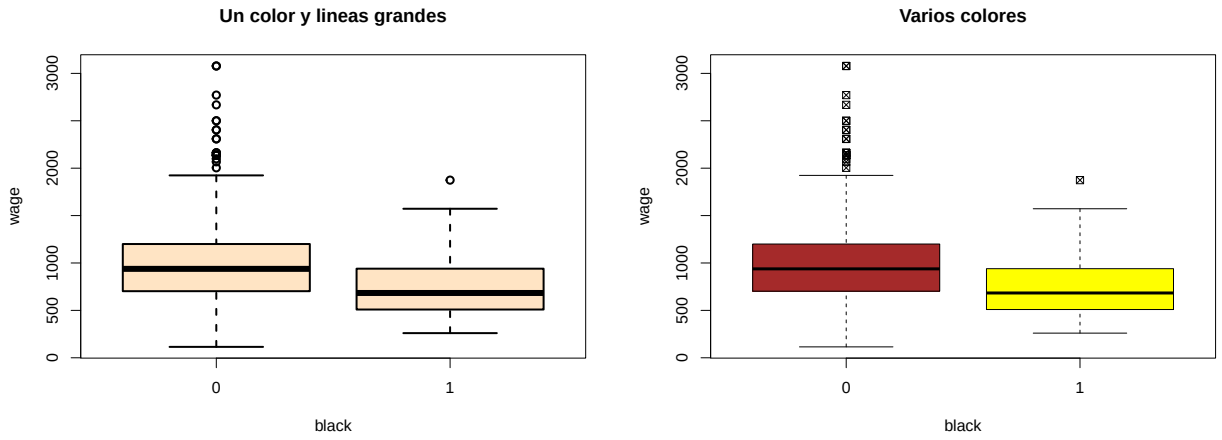


Diagrama de barras

Los diagramas de barras permiten representar gráficamente un conjunto de datos por medio de barras, el tamaño de estas barras serán proporcional a los valores representados.

Para realizar este gráfico en R es necesario resumir o agrupar los datos, para ello se utiliza el comando **table**, el cual resume en una tabla los datos de la variable.

```
tablacasados <- table(married) #Se guarda la informacion en la variable tablacasados.
tablacasados # Se imprime el objeto "tablacasados".
```

```
## married
##   0   1
## 100 835
```

Al aplicar esta función sobre la variable "married", se observa que hay 100 personas que están solteras y 835 personas que están casadas, ya se ha agrupado la información dependiendo de la categoría de la variable.

Si usted desea mostrar los datos en términos porcentuales puede usar el comando prop.table.

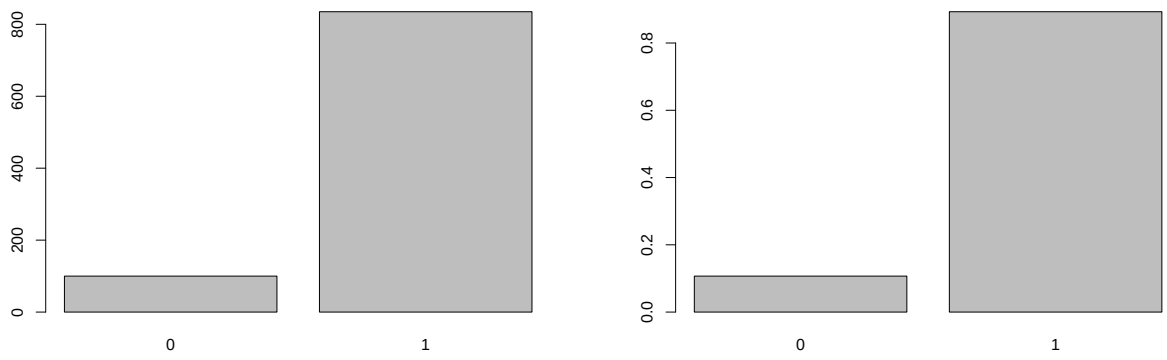
```
Porcentaje_casados <- prop.table(tablacasados) #Se guarda la informacion en porcentajes
Porcentaje_casados
```

```
## married
##         0         1
## 0.1069519 0.8930481
```

Ahora tenemos la información expresada de una manera diferente, se puede afirmar que el 11 % de la población total está soltera, y el 89 % restante está casada.

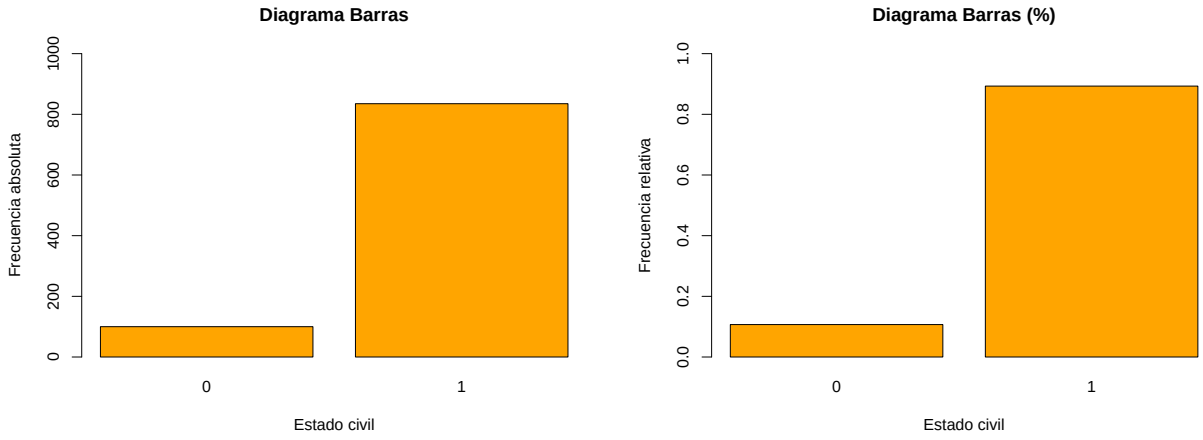
Para realizar el gráfico de barras en R se puede usar el comando **barplot**.

```
barplot(tablacasados) # Frecuencia absoluta
barplot(Porcentaje_casados) # Frecuencia relativa
```



Ahora se van a modificar los atributos de estos gráficos con los comandos iniciales.

```
barplot(tablacasados, xlab = "Estado civil",
        ylab = "Frecuencia absoluta",
        col = "orange ", main = "Diagrama Barras", ylim = c(0,1000))

barplot(Porcentaje_casados, xlab = "Estado civil",
        ylab = "Frecuencia relativa",
        col = "orange ", main = "Diagrama Barras (%)", ylim = c(0,1))
```



Al igual que los diagramas anteriores, este también puede ser personalizado.

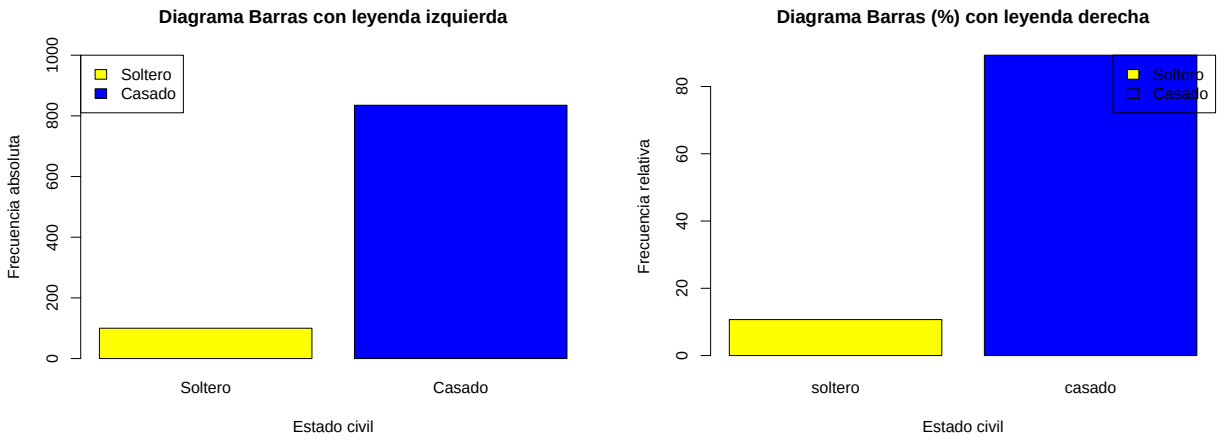
Se puede asignar un color diferente para cada una de las barras, para ello se tiene que asignar manualmente los colores. Como es más de un color se va a usar c(), recuerde que de esta manera se pueden agrupar datos. El primer valor que aparece tanto en la gráfica como en la tabla es el 0, entonces este sería el primer color que se asignará, luego está el 1, será el segundo color. Supongamos que vamos a poner amarillo a 0 y azul a 1, se haría de la siguiente manera: *col = c("yellow", "blue")*.

Si usted quiere cambiar las etiquetas de las barras lo puede hacer con el argumento *names.arg*, el cual funciona igual que la asignación de los colores, para este caso, el 0 significa soltero, el 1 significa cadasos, entonces se haría de la siguiente manera: *names.arg = c("Soltero", "Casado")*.

Al igual que el boxplot, este grafico también permite representarlo horizontalmente, para ello use el código *horiz = T*.

Usando el argumento *legend.text* puede colocar una leyenda o cuadro donde especifique el color que representa cada dato de la variable. Funciona exactamente igual que el argumento *names.arg*. Para modificar la posicion de este cuadro use el argumento *args.legend = list(x = " ")*, dentro de las comillas deberá establecer la posición, las opciones son: top, bottom, topleft, topright, bottomleft y bottomright.

```
barplot(tablacasados, xlab = "Estado civil", ylab = "Frecuencia absoluta",
        col = c("yellow", "blue"), main = "Diagrama Barras con leyenda izquierda",
        ylim = c(0,1000),
        names.arg = c("Soltero", "Casado"),
        legend.text = c("Soltero", "Casado"),
        args.legend = list(x = "topleft"))

barplot(Porcentaje_casados*100, xlab = "Estado civil",
        ylab = "Frecuencia relativa",
        main = "Diagrama Barras (%) con leyenda derecha",
        col = c("yellow", "blue"),
        names.arg = c("soltero", "casado"),
        legend.text = c("Soltero", "Casado"),
        args.legend = list(x = "topright"))
```



Para hacer gráficos de barras agrupadas se utiliza el argumento *beside*, si las quiere por separado ponga T, si la quiere unido en una barra ponga F.

A continuación haremos dos diagramas de barras agrupadas, en donde tendremos información de los años de educación de las personas que viven en el sur y las personas que viven en el norte.

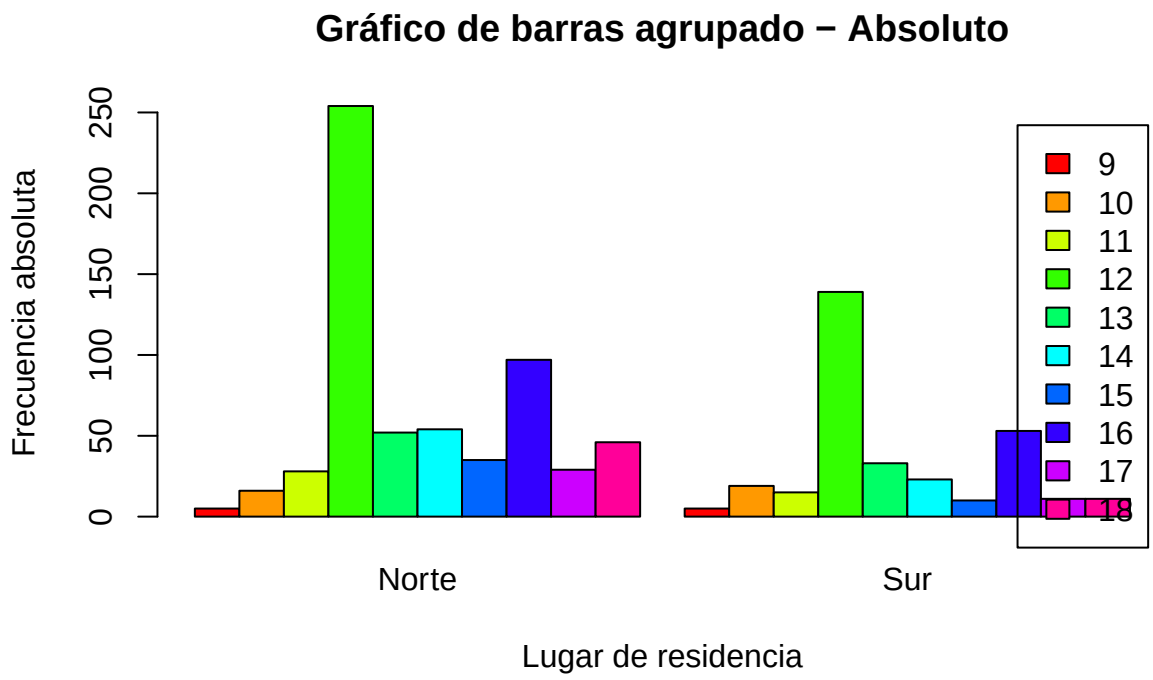
```
tabla_variables = table(educ,south)
tabla_variables ## Tabla con la información a graficar
```

```
##     south
## educ   0   1
##   9    5   5
##   10  16  19
##   11  28  15
##   12 254 139
##   13  52  33
##   14  54  23
##   15  35  10
##   16  97  53
##   17  29  11
```

```
##   18  46  11
```

R también trae integradas algunas paletas de colores, para no escribir un color para cada barra se pueden usar estas herramientas, en este caso se usará la paleta *rainbow*, y entre paréntesis se colocará el número de colores a utilizar, como son 10 barras para cada grupo entonces se pondrá 10 entre paréntesis.

```
barplot(tabla_variables,
        main = "Gráfico de barras agrupado - Absoluto",
        xlab = "Lugar de residencia", ylab = "Frecuencia absoluta",
        col = rainbow(10),
        names.arg = c("Norte", "Sur"),
        legend.text = rownames(tabla_variables),
        beside = TRUE)
```




```
barplot(prop.table(tabla_variables)*100,
        main = "Gráfico de barras agrupado - Relativo",
        xlab = "Lugar de residencia", ylab = "Frecuencia relativa",
        ylim = c(0,70),
        col = rainbow(10),
        names.arg = c("Norte", "Sur"),
        legend.text = rownames(tabla_variables),
        beside = F)
```

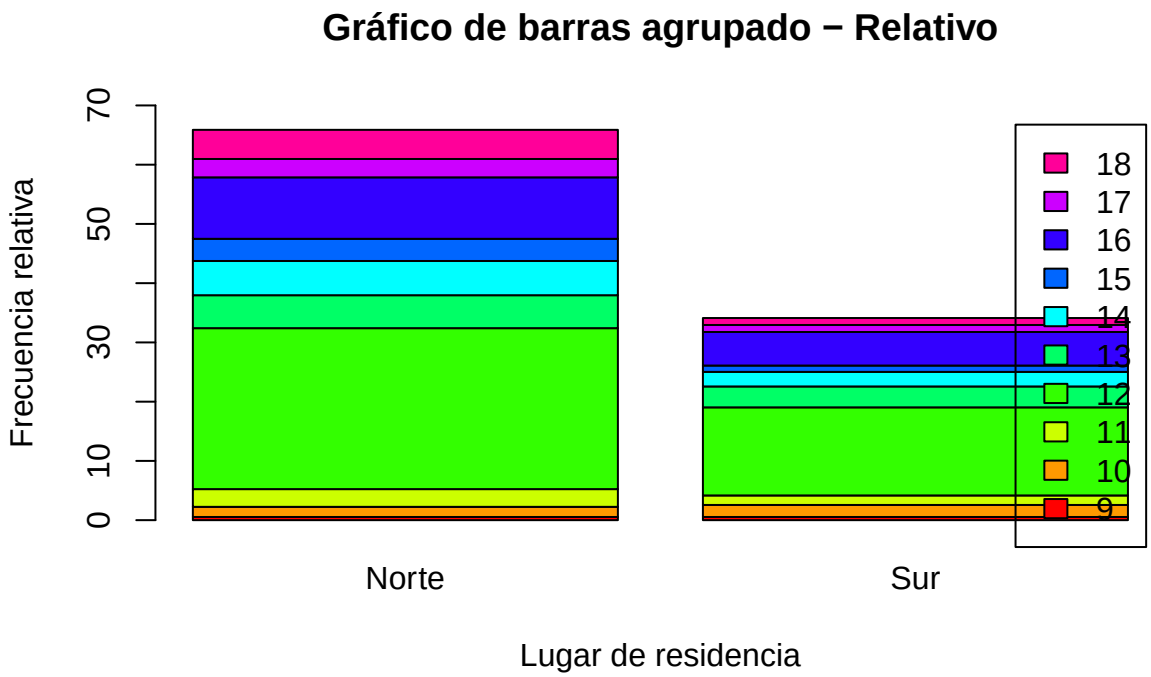
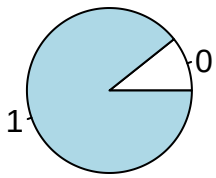


Diagrama de torta

El diagrama de torta es un gráfico circular que representa porcentajes sobre la composición de la variable, donde el área y la longitud del arco de cada sector es proporcional a la cantidad representada.

Para hacer estos gráficos se debe hacer el mismo proceso que con los diagramas de barras, se debe tener una tabla que resuma y contabilice los datos, sobre esta tabla se emplea la función **pie** . (Se van a usar las mismas tablas del ejemplo anterior).

```
pie(tablacasados)
```



Ahora vamos a personalizarlo un poco:

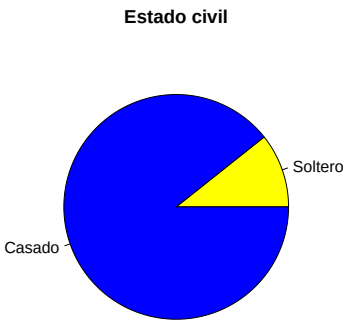
La asignación de colores y etiquetas de los sectores se realiza practicamente igual al diagrama de barras, con la única diferencia de que para las etiquetas acá se utiliza el argumento *labels*. Con la etiqueta labels podrá poner ya sea el grupo al que pertenece el sector de gráfico o el valor numérico que desea representar.

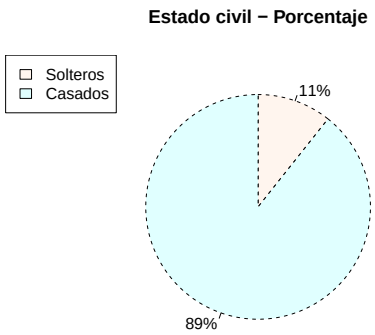
También se puede modificar la dirección del gráfico con el argumento *clockwise* = *T*, los datos de la tabla también se irán organizando en orden como si fuera un relog.

El argumento *lyt* le permite modificar el tipo de línea de los bordes del gráfico.

Para agregar una leyenda a este gráfico debemos poner el comando **legend** después de terminado el código **pie**, en el argumento *legend* coloque el nombre de los grupos, y con el argumento *fill* indique los colores de esos grupos.

```
pie(tablacasados, col = c("yellow","blue"), labels = c("Soltero", "Casado"),
    main = "Estado civil")

pie(Porcentaje_casados, col = c("seashell","lightcyan"), labels = c("11%", "89%"),
    main = "Estado civil - Porcentaje",clockwise = T, lty = 2)
legend("topleft", legend = c("Solteros", "Casados"), fill =  c("seashell","lightcyan"))
```





Diagramas de dispersión.

Los diagramas de dispersión se utilizan para mirar el comportamiento de dos variables **cuantitativas**, ayudan a determinar si una variable tiene influencia sobre la otra. Acá se manejan variables depedientes e independientes. La variable dependiente es aquella que se ve afectada por las otras variables, y las independientes son aquellas que pueden influenciar o afectar el comportamiento de la variable dependiente.

Un ejemplo de esto podría ser el peso y la estatura, se dice que la estatura afecta el peso, entre más alta sea la persona más pesada debería ser. Para este caso, la variable dependiente sería el peso, mientras que la variable independiente sería la estatura, ya que el peso de la persona no afecta la estatura.

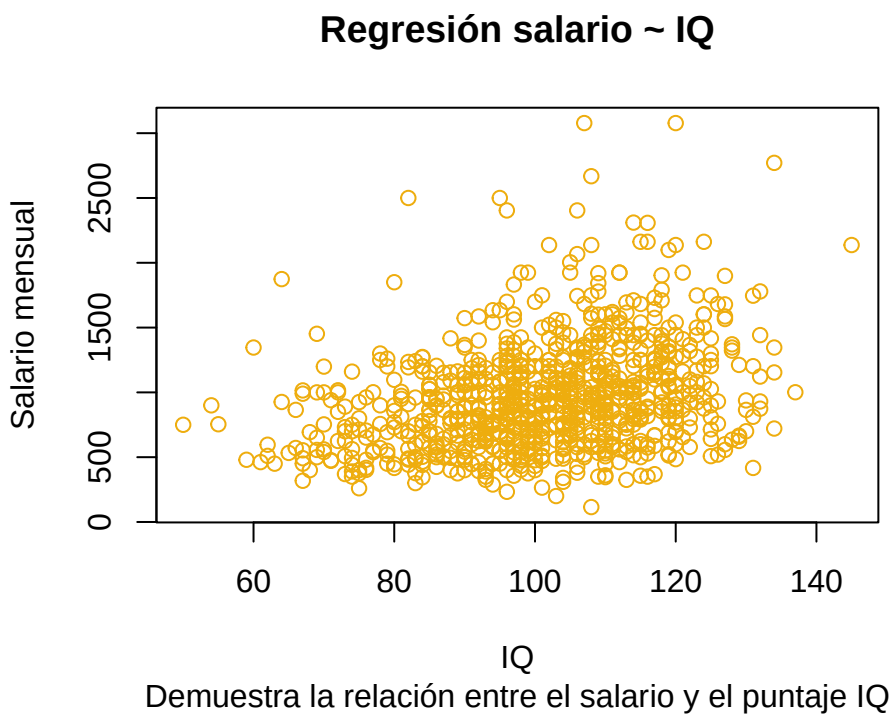
- Variable dependiente (Y) = Peso.
- Variable independiente (X) = Estatura.

Suponga que con los datos que se tienen en la base de datos ***Wage2*** se quiere determinar si el puntaje IQ de una persona afecta al salario.

- Variable depediente (Y): wage (Salario)
- Variable independiente (X): IQ

Para hacer el gráfico de dispersión se utiliza la función **plot**

```
plot(wage~IQ, xlab = "IQ", ylab = "Salario mensual",
     main= "Regresión salario ~ IQ", col= "darkgoldenrod2",
     sub= "Demuestra la relación entre el salario y el puntaje IQ")
```



##Para la funcion *type=*

#type="p": Dibuja puntos individuales (opcion por defecto) #type="l": Dibuja lineas #type="b": Dibuja puntos y lineas #type="o": Dibuja puntos atravesados por lineas #type="h": Dibuja con lineas verticales #type="s": Dibuja a base de funciones escalera