Guía de Trabajo 2

1- Plot

El comando *plot* para realizar gráficos es el más básico en R y genera gráficos cuya interpretación varía según el objeto al que se aplique.

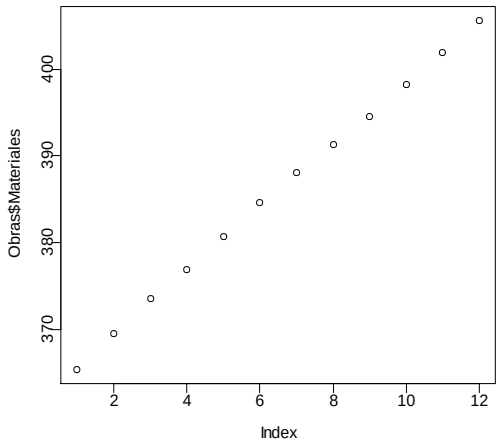
plot(x)

Si x es un vector grafica los valores de x en el eje y en función del identificador de cada caso

Ejemplo: para la variable Materiales del archivo Obras.csv

Primero leemos el archivo:

```
Obras<-read.csv2(file.choose())
plot(Obras$Materiales)
```



Podemos elegir el formato del punto con el argumento pch que toma valores de 1 a 25 que representan distintos tipos, algunos ejemplos son los siguientes:

pch=19 círculo sólido pch=10: círculo relleno

pch=20 círculo sólido más pequeño

pch=21 puntos sin rellenar

pch=22 cuadrados sin rellenar

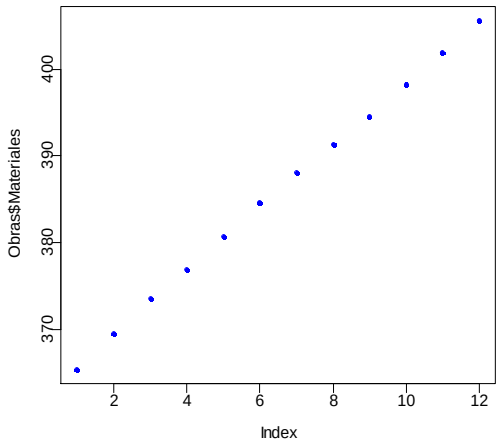
pch=23 rombos sin rellenar

pch=23 triángulos con punta hacia arriba

pch=23 triángulos con punta hacia abajo

Los pch de 21 a 25 pueden cambiarse de color, utilizando el comando col

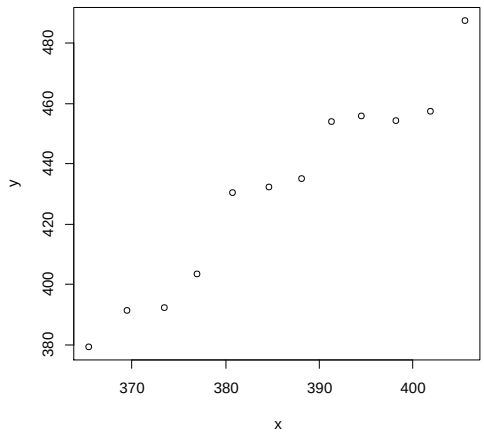
```
>plot(Obras$Materiales,col="blue",pch=20)
```



Si x e y son vectores genera el diagrama de dispersión entre x e y

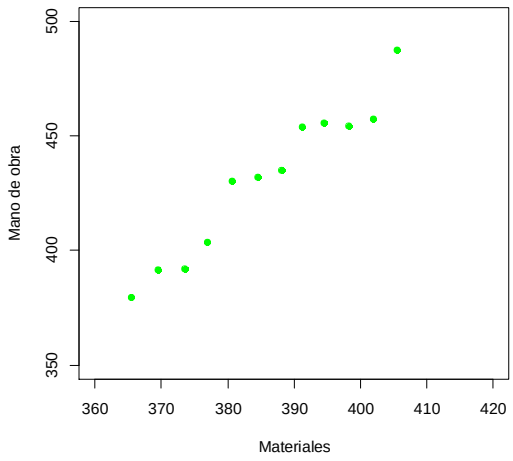
Definamos para el mismo ejemplo

```
x<-Obras$Materiales
y<-Obras$Mano.de.Obra
> plot(x,y)
```



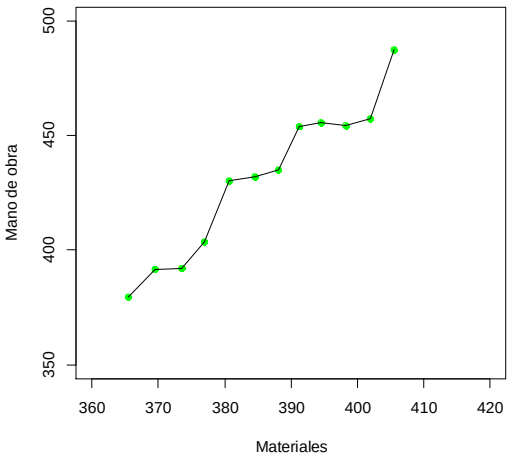
Con *xlab, ylab* ponemos nombre a los ejes, si además queremos fijar un rango para los ejes usamos *xlim* e *ylim*:

```
>plot(x,y,col="green",pch=19,xlab="Materiales",ylab="Mano de
obra",xlim=c(360,420),ylim=c(350,500))
```



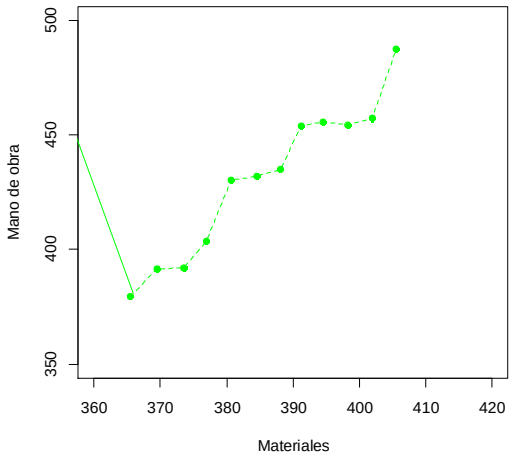
Si queremos unir con lineas, sin cerrar el gráfico usamos *lines*:

```
>lines(x,y)
```



Con *lty* podemos elegir el estilo de la línea, y con *col* el color de la misma

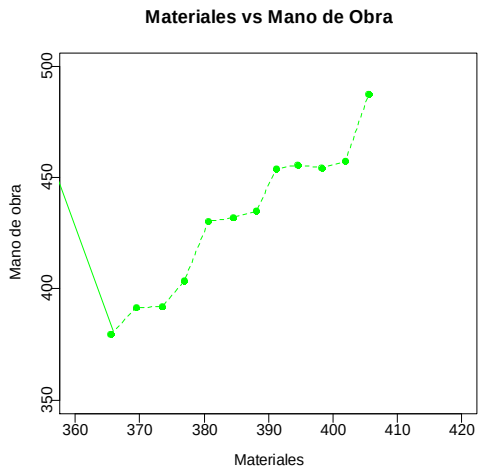
```
>plot(x,y,col="green",pch=19,xlab="Materiales",ylab="Mano de
obra",xlim=c(360,420),ylim=c(350,500))
> lines(x,y,lty=2,col="green")
```



Observación: Solo *lines* sin *plot* previamente no grafica nada.

Ponemos un título al gráfico

```
> title("Materiales vs Mano de Obra")
```



Para ver otras opciones gráficas *help(par)*

2-Distribuciones de Probabilidad

Distribución Binomial

Ejemplo: Sea X~Bi(5,0.1)

- P(X=3)

```
> dbinom(3,5,0.1)#probabilidad puntual
 [1] 0.0081
```

- P(x<=3)

```
>pbinom(3,5,0.1) #probabilidad acumulada a izquierda
[1] 0.99954
```

- P(x>3)

```
>pbinom(3,5,0.1,lower.tail=F)#probabilidad acumulada a derecha
[1] 0.00046
> pbinom(3,5,0.1,F)
[1] 0.00046
```

- k/P(x<=k)=0,9999

```
>qbinom(0.9999,5,0.1)
[1] 4
```

Comprobamos P(x<=4)

```
>pbinom(4,5,0.1)
[1] 0.99999
```

Si queremos generar "números aleatorios" con distribución binomial

```
>rbinom(10,5,0.1)#genera valores "al azar", pseudoaleatorios
 [1] 1 0 0 0 1 0 0 0 0 0
> rbinom(10,5,0.2)
 [1] 2 0 1 1 2 2 1 2 1 1

> rbinom(10,5,0.5)
 [1] 3 1 2 2 5 3 2 3 5 3
> rbinom(10,5,0.9)
 [1] 5 4 3 4 5 5 5 5 5 5
```

En general si X~Bi(n,p)

Para calcular:

- P(X=k)

```
dbinom(k, size=n, prob=p)
```

- P(X<=k)

```
pbinom(k, n, p)
```

- P(X>k)

```
pbinom(k, n, p,F)
```

- k/P(x<=k)=p2

```
qbinom(p2,n,p)
```

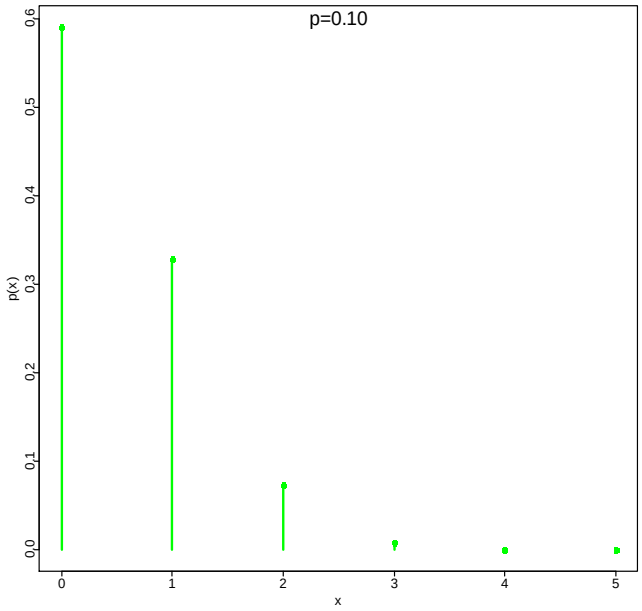
- m Valores al azar

```
rbinom(m,n,p)#genera m valores "al azar", pseudoaleatorios
```

Analogamente para las distribuciones discretas *help(distributions)*

Grafiquemos la función de probabilidad puntual

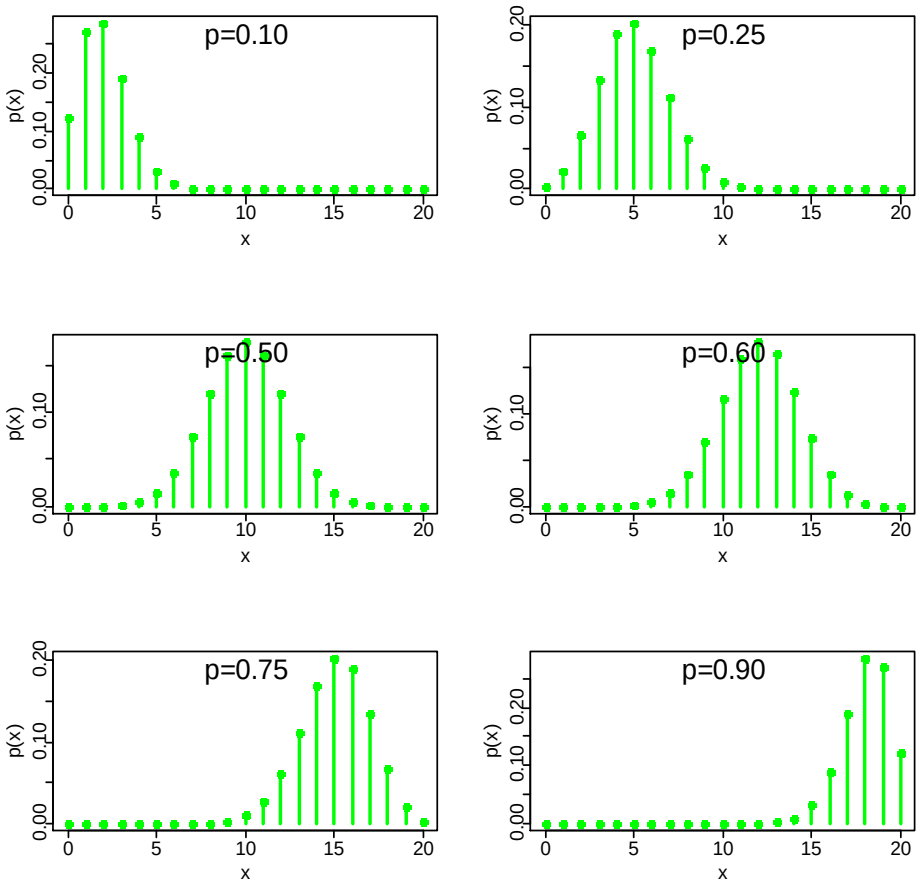
```
n<-5
x<-seq(0,n)
par(mex=0.7)
par(mgp=c(2,0.3,0))
par(cex=0.65)
p<-0.1
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.10",3,-2)
```



Sea X~Bi(20,p) , grafiquemos la función de probabilidad puntual para distintos valores de p

```
n<-20
x<-seq(0,n)
par(mfrow=c(3,2))
par(mex=0.7)
par(mgp=c(2,0.3,0))
par(cex=0.65)
p<-0.1
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.10",3,-2)
p<-0.25
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.25",3,-2)
p<-0.50
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.50",3,-2)
```

```
p<-0.60
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.60",3,-2)
p<-0.75
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.75",3,-2)
p<-0.90
y<-dbinom(x,n,p)
plot(x,y,ylab="p(x)",type="h",lwd = 2,col="green")
points(x,y,pch = 16,col="green")
mtext("p=0.90",3,-2)
```



Distribución Normal

Ejemplo: Sea X~N(mu, sigma)

```
dnorm(x, mean, sd ) evalua la función de densidad en x
pnorm(q, mean ,sd ) P(x<=q)
qnorm(p, mean ,sd ) q/P(x<=q)=p
rnorm(n, mean sd) genera n valores pseudoaleatorios normales con la
media y el desvío dado.
```

X~N(0,1)

- f(0)

```
> dnorm(0, 0, 1 )
[1] 0.3989423
> 1/sqrt(2*pi)
[1] 0.3989423
```

- P(X<=0)

```
> pnorm(0, 0, 1 )
[1] 0.5
```

- P(X<=2)

```
> pnorm(2, 0, 1 )
[1] 0.9772499
```

- P(X>2)

```
> pnorm(2, 0, 1 ,F)
[1] 0.02275013
```

```
> qnorm(0.99, 0, 1 )
[1] 2.326348
> qnorm(0.99, 0, 1 ,F)
[1] -2.326348
```

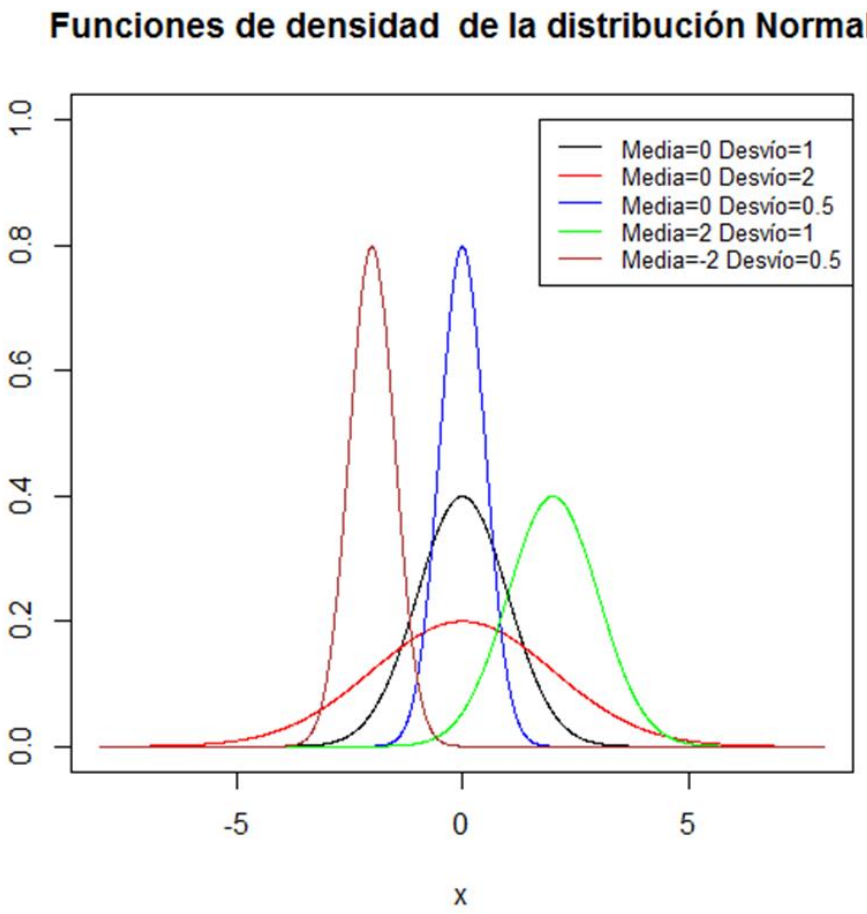
```
> rnorm(10, 0, 1)
 [1]  0.8864109  0.8167739  0.5039489 -0.1350530  1.0118410  0.5627892
 [7]  0.3238939 -1.2846347  1.5104558  1.2478405
```

Podemos fijar la semilla para obtener siempre los mismos valores, por ejemplo:

```
>  set.seed(34)
>   rnorm(10, 0, 1)
 [1] -0.138889971  1.199812897 -0.747722402 -0.575248177 -0.263581513
 [6] -0.455492149  0.670620044 -0.849014621  1.066804504 -0.007460534
```

Gráfico de densidades

```
x<-seq(-8,8,0.01)
plot(x,dnorm(x),type="l",ylim=c(0,1),ylab="")#normal estandar
lines(x,dnorm(x,0,2),type="l",ylim=c(0,1),col="red")#media 0 y desvío
2
lines(x,dnorm(x,0,0.5),type="l",ylim=c(0,1),col="blue") #media 0 y
desvío 0.5
lines(x,dnorm(x,2,1),type="l",ylim=c(0,1),col="green") #media 2 y
desvío 1
lines(x,dnorm(x,-2,0.5),type="l",ylim=c(0,1),col="brown") #media -2 y
desvío 0.5
legend(1.7,1, legend = c("Media=0 Desvío=1","Media=0
Desvío=2","Media=0 Desvío=0.5","Media=2 Desvío=1","Media=-2
Desvío=0.5"),
col = c(1,"red","blue","green","brown"),lty = 1, cex = .8, y.intersp =
1)
title("Funciones de densidad  de la distribución Normal")
```



3- Estadística descriptiva

Objetivo: Describir los datos buscando características y patrones en los mismos.

Las técnicas varían según el tipo de variable que se trate: cualitativa o cuantitativa.

Ejemplo 1: El siguiente ejemplo corresponde a datos filtrados de la EPH del 4to trimestre de 2013 importados en dbf4.

Se filtraron los datos correspondientes a la ciudad de Buenos Aires y a GBA (24 partidos del conurbano bonaerense) y otras variables que hacen a las características de los hogares como vivir en casa o departamento no usurpado, un solo hogar por vivienda, con cañerías y baño individual con inodoro dentro de la vivienda, en zonas no inundables, alejadas de basurales y no situadas en villas de emergencia, donde el ingreso familiar de los últimos 12 meses proviene del trabajo de personas mayores de edad.

Con estos filtros la base original de 789 hogares para CABA y 2076 hogares para GBA se redujo a 88 y 174 hogares respectivamente para CABA y GBA.

De todas las variables reportadas se conservaron las variables CODUSU, que identifica el hogar, ANO4, corresponde al año, TRIMESTRE, corresponde al trimestre, AGLOMERADO, identifica el aglomerado, ITF (Ingreso total familiar) y IPCF (Ingreso Familiar Per Cápita). El archivo se guardó en formato csv como Datos EPH.

Vamos a trabajar con las variables AGLOMERADO e ITF, observemos que el tratamiento de los datos correspondientes a cada una de las variables no puede ser el mismo dada la naturaleza diferente de las mismas.

Leemos los datos y le asignamos el nombre Datos EPH

```
> DatosEPH<-read.csv2(file.choose(), header=T)
```

¿Cómo podemos describir el comportamiento de una variable cualitativa? ¿Qué nos puede interesar?

Básicamente la frecuencia con que se presenta cada clase:

En R:

```
> table(DatosEPH$AGLOMERADO)
CABA  GBA
  88  174
```

tabulate realiza lo mismo pero no nos muestra las categorías, sino que es un vector que contiene las frecuencias absolutas.

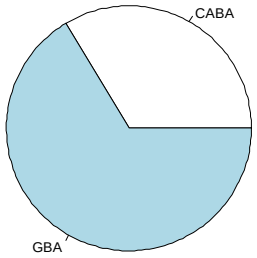
```
> tabulate(DatosEPH$AGLOMERADO)
[1]  88 174
```

Para una descripción gráfica podemos realizar un *diagrama circular* o un *diagrama de barras*:

Diagrama circular: el comando es *pie*

Podemos utilizar tanto el resultado de *table* como de *tabulate* para confeccionarlo.

```
>Aglomerado.frec<-table(DatosEPH$AGLOMERADO)
>pie(Aglomerado.frec) # genera un gráfico muy sencillo.
```



Si queremos un gráfico más elaborado, usamos las opciones *labels, col, main*:

```
>pie(Aglomerado.frec,labels=c("CABA","GBA"),col=c("red","blue"),main="
Gráfico circular para la variable Aglomerado")
```

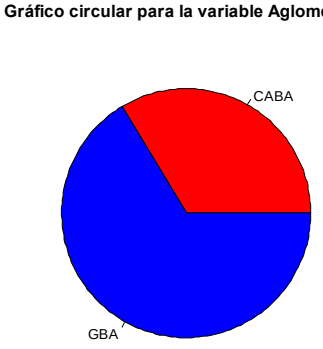
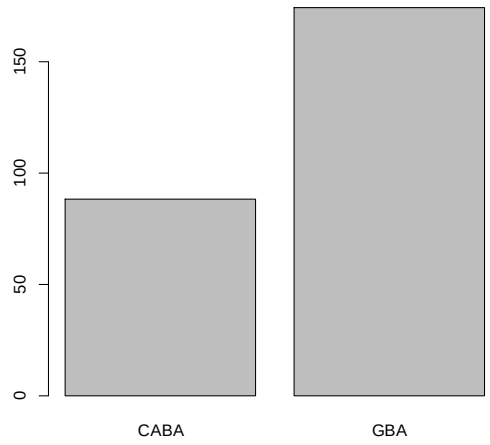
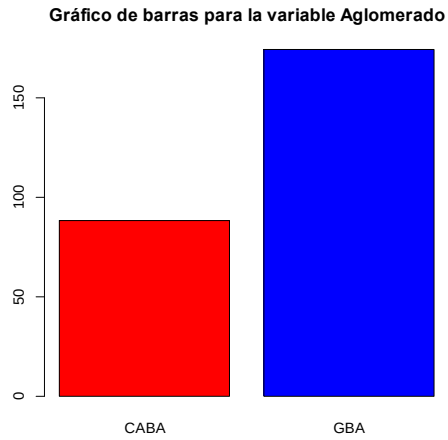


Diagrama de Barras: el comando es *barplot*

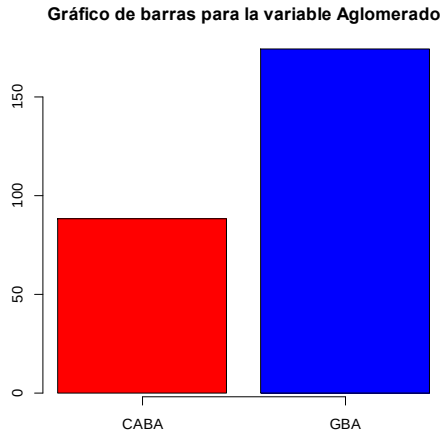
```
>barplot(Aglomerado.frec)
```




```
>barplot(Aglomerado.frec,col=c("red","blue"),main="Gráfico de barras
para la variable Aglomerado")
```




```
>barplot(Aglomerado.frec,col=c("red","blue"),main="Gráfico de barras
para la variable Aglomerado",axis.lty = 1)
```



Ya vimos que el comando *plot* es el más básico en R

Si x es el factor Aglomerado en el data frame DatosEPH

```
> plot(DatosEPH$AGLOMERADO)
```

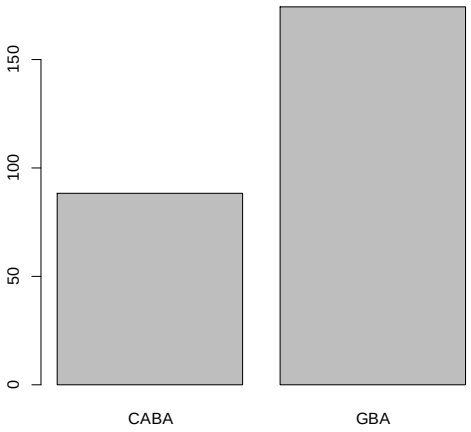
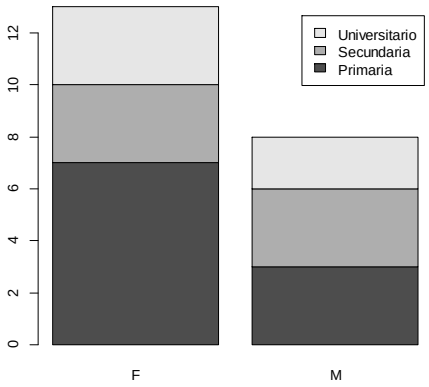


Gráfico de barras agrupadas

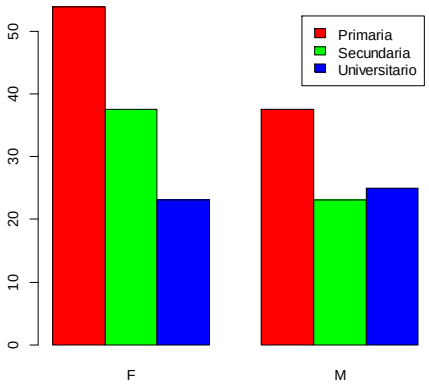
```
>Ejemplo<-read.csv2(file.choose(), header=T)
>Sex_Nivel<-table(Ejemplo$Nivel.Educativo,Ejemplo$Sexo)
> Sex_Nivel
                F M
  Primaria      7 3
  Secundaria    3 3
  Universitario 3 2
```

```
>barplot(Sex_Nivel,legend=rownames(Sex_Nivel))
```



Si queremos las barras al costado y cambiar el color

```
>barplot(SexNivel.porcentual,legend=rownames(SexNivel.porcentual),besi
de=T,col=c("red","green","blue"))
```



¿Cómo podemos describir el comportamiento de una variable cuantitativa? ¿Qué nos puede interesar?

En el caso de variables cuyos valores representan cantidades nos interesa graficar su distribución en el rango que se presentan los valores, y calcular medidas de resumen.

Medidas de resumen:

Medidas de posición

Media: promedio de los datos.(En R: *mean*)

Mediana: valor central de los datos.(En R: *median)*

Percentiles: percentil p es el valor que supera al p% de los valores de la muestra .(En R: *quantile*)

Medidas de dispersión

Rango: intervalo en el que están contenidos los datos

Varianza: varianza de los datos (variabilidad respecto a la media) .(En R: *var*)

Desvío Stándard: raíz cuadrada de la varianza .(En R: *ds*)

Rango Intercuartilico: distancia del primer al tercer cuartil (percentiles 25 y 75 respectivamente) .(En R: *IQR)*

MAD o desviación absoluta mediana .(En R: *MAD)*

Otras medidas:

Coeficiente de variación: Desvío Stándard/media

Coeficiente de asimetría (es necesario instalar un paquete)

Coeficiente de curtosis (es necesario instalar un paquete)

En nuestro ejemplo, para facilitar la sintaxis previamente le asignamos el nombre Ingresos a la variable ITF

```
> Ingresos<-DatosEPH$ITF
> mean(Ingresos)
[1] 9582.412
# Si hubiera valores faltantes daría NA, una forma de calcular la media descartando los valores faltantes es con na.rm=T
> mean(Ingresos,na.rm=T)
[1] 9582.412
> median(Ingresos,na.rm=T)
[1] 8000
> min(Ingresos,na.rm=T)
[1] 500
> max(Ingresos,na.rm=T)
[1] 47000
```

```
> quantile(Ingresos,na.rm=T)
 0% 25% 50% 75%  100%
  500  5000  8000 12000 47000
```

fivenum realiza lo mismo pero no nos muestra las %, sino que es un vector que contiene los valores de los percentiles.

```
> fivenum(Ingresos,na.rm=T)
[1] 500  5000  8000 12000 47000
```

Si queremos calcular otros percentiles distintos de los cuartiles debemos especificarlo

```
> quantile(Ingresos,na.rm=T,probs=c(0.10,0.30,0.90,0.95))

  10% 30% 90% 95%
 3200  6000 18450 21190
```

Con *summary* podemos calcular el mínimo, percentil 25, mediana, media, percentil 75 y el máximo.

```
> summary(Ingresos)
 Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
  500  5000  8000  9582 12000 47000
```

El comando *range* devuelve un vector de dos coordenadas que representan el máximo y el minimo de los datos

Luego, si queremos calcular el rango

```
> range(Ingresos)
[1] 500 47000
> diff(range(Ingresos))
[1] 46500
#Otra forma
> max(Ingresos)-min(Ingresos)
[1] 46500
```

Las restantes medidas podemos calcularlas directamente

```
> var(Ingresos)
[1] 40586133
```

```
> sd(Ingresos)
[1] 6370.725
> IQR(Ingresos)
[1] 7000
> mad(Ingresos)
[1] 4744.32
```

Para calcular el coeficiente de variación porcentual hacemos la cuenta

```
> 100*sd(Ingresos)/mean(Ingresos)
[1] 66.48352
```

Si queremos solo 2 cifras decimales

```
> round(100*sd(Ingresos)/mean(Ingresos),2)
[1] 66.48
```

Para calcular las medidas por grupos definidos por un factor (en este caso para cada Aglomerado) contamos con la función *tapply*

Definimos previamente para facilitar la notación la variable Región correspondiente a los 2 aglomerados

```
> Región<-DatosEPH$AGLOMERADO
> tapply(Ingresos, Región, summary)
$CABA
 Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
 1600  6000  8400 10080 11650 47000

$GBA
 Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
  500  5000  7675  9332 12580 31800

> tapply(Ingresos, Región, sd)
  CABA  GBA
6886.920 6098.472
> tapply(Ingresos, Región, mad)
 CABA   GBA
4225.41 5337.36
```

```
> 100*tapply(Ingresos, Región, sd)/tapply(Ingresos, Región, mean)
  CABA   GBA
68.33880 65.35035
```

¿Qué tipo de gráfico podemos utilizar para describir datos cuantitativos?

El comando plot visto anteriormente para una variable cuantitativa no suele ser un buen gráfico para captar la distribución de la variable. Para ello tenemos tres opciones mejores: histograma, diagrama tallo-hoja y boxplot

- **Histograma**

Es un gráfico de barras que representan las frecuencias con que aparecen las mediciones agrupadas en intervalos. Para construirlo se debe dividir el rango en que se van a graficar los datos en intervalos o clases que pueden o no ser de igual longitud (en general se recomienda que lo sean). La cantidad de clases no debe ser excesiva ya que de ese modo tanto detalle no permite visualizar un patrón, ni pocas de manera que la distribución resulte demasiado suavizada.. Se proponen distintas reglas:

1. Dixon y Kronmal: $10 \log_{10} n$
2. Velleman: $2\ n^{1/2}$
3. Sturgess: $1+\log_2 n = 1+3.3 \log_{10} n$

En todos los casos n representa la cantidad de datos y se elige como cantidad de clases la parte entera del cálculo realizado.

En R :*hist*

La función *hist* acepta varios argumentos:

x: el vector de datos a graficar

breaks: para especificar dónde queremos los límites de los intervalos o bien cuántos intervalos queremos, o sea que puede ser un vector o un número. Si no lo indicamos el default es la regla de Sturgess.

freq: especifica si el alto de las barras representa la frecuencia absoluta de los datos en el intervalo o la densidad de los mismos. El último criterio es más recomendable: como el histograma se utiliza para obtener una impresión visual de la distribución de una variable continua, dado que el área por debajo de un función de densidad es 1, es común y adecuado

graficar los histogramas de manera que el área total debajo del mismo sea también 1. Para ello el alto de las barras debe ser la frecuencia relativa porcentual (porcentaje de datos) dividido el ancho de la clase. Para que utilice este criterio debe especificarse freq=FALSE.

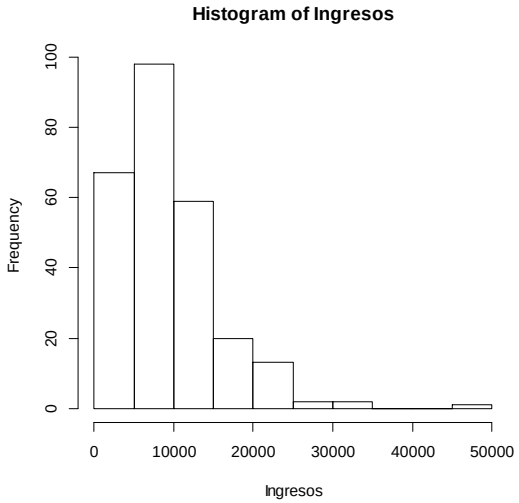
main; xlab, ylab,xlim, ylim como en los gráficos anteriores.

labels: añade etiquetas a cada barra indicanco la frecuencia.

plot: por default utiliza TRUE lo cual significa que aparece el gráfico. Si indicamos FALSE solo genera la tabla de frecuencias

Lo hacemos para la variable Ingresos del archivo Basa EPH, sin discriminar por aglomerado.

```
>hist(Ingresos)
```

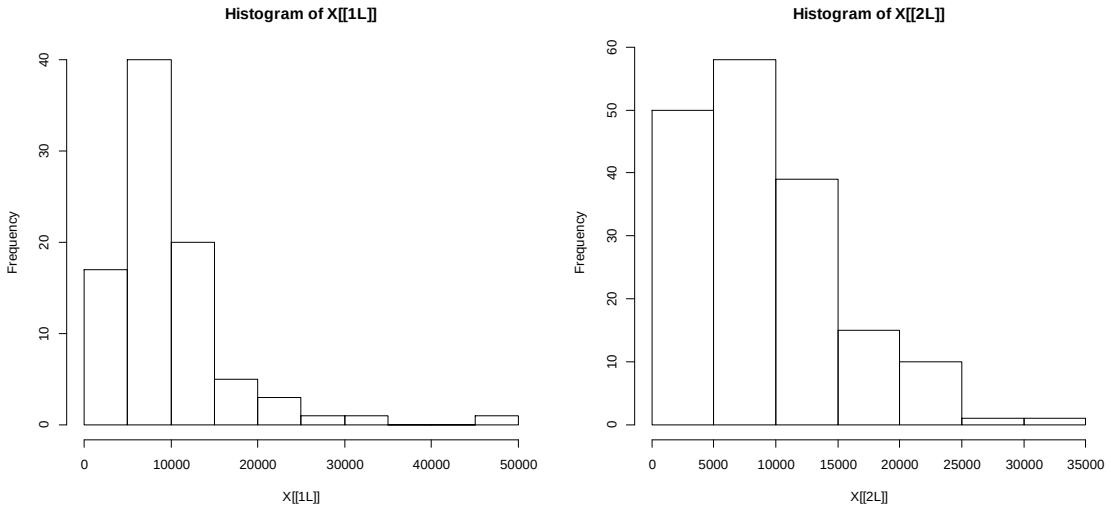


Si asignamos el histograma a un objeto:

```
> histo.ingresos<-hist(Ingresos)
> names(histo.ingresos)
[1] "breaks" "counts" "density"  "mids"   "xname"  "equidist"
> histo.ingresos$breaks #nos da los límites de los intervalos de clase
 [1]    0  5000 10000 15000 20000 25000 30000 35000 40000 45000 50000
> histo.ingresos$mids #nos da los puntos medios de cada intervalo de clase
 [1]  2500  7500 12500 17500 22500 27500 32500 37500 42500 47500
> histo.ingresos$counts #nos da la frecuencia absoluta de cada intervalo de
#clase
 [1] 67 98 59 20 13  2  2  0  0  1
```

Si discriminamos por aglomerado

```
>par(mfrow=c(1,2))# es para dividir la pantalla gráfica en 1 fila y
dos columnas para poder visualizar mejor.
> tapply(Ingresos,Región,hist)
$CABA
$breaks
 [1]    0  5000 10000 15000 20000 25000 30000 35000 40000 45000 50000

$counts
 [1] 17 40 20  5  3  1  1  0  0  1

$density
 [1] 3.863636e-05 9.090909e-05 4.545455e-05 1.136364e-05 6.818182e-06
 [6] 2.272727e-06 2.272727e-06 0.000000e+00 0.000000e+00 2.272727e-06


$mids
 [1]  2500  7500 12500 17500 22500 27500 32500 37500 42500 47500

$xname
[1] "X[[1L]]"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"

$GBA
$breaks
[1]    0  5000 10000 15000 20000 25000 30000 35000

$counts
[1] 50 58 39 15 10  1  1

$density
[1] 5.747126e-05 6.666667e-05 4.482759e-05 1.724138e-05 1.149425e-05
[6] 1.149425e-06 1.149425e-06

$mids
[1]  2500  7500 12500 17500 22500 27500 32500

$xname
[1] "X[[2L]]"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```



Para editar los gráficos resulta más sencillo si definimos las variables para cada aglomerado

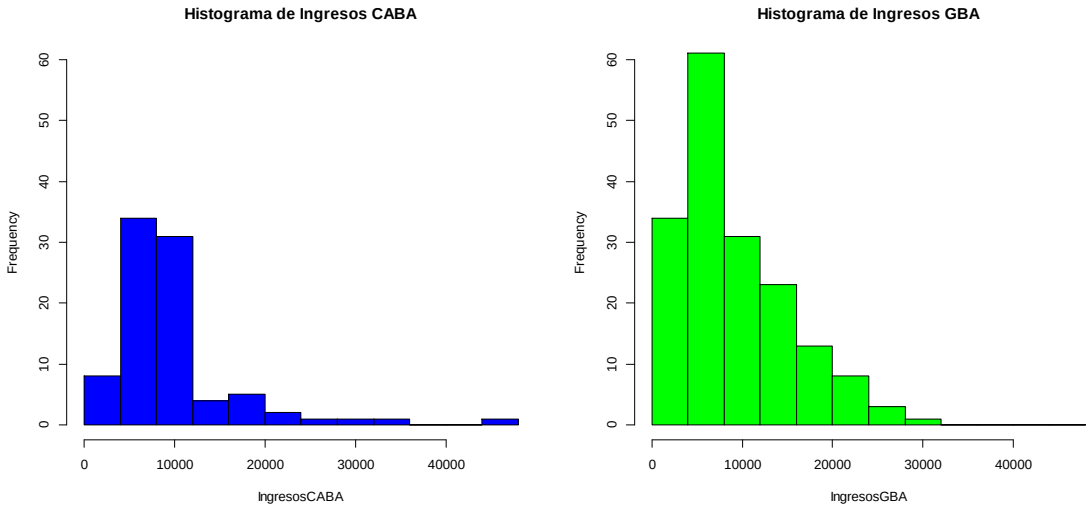
```
> IngresosCABA<-DatosEPH$ITF[DatosEPH$AGLOMERADO=="CABA"]
> IngresosGBA<-DatosEPH$ITF[DatosEPH$AGLOMERADO=="GBA"]
```

Histograma editado (fijamos la longitud de cada intervalo, los límites, elegimos color y ponemos títulos)

Para poder comparar ambos histogramas trabajamos para ambas variables con los mismos intervalos de clase y la misma escala para las frecuencias.

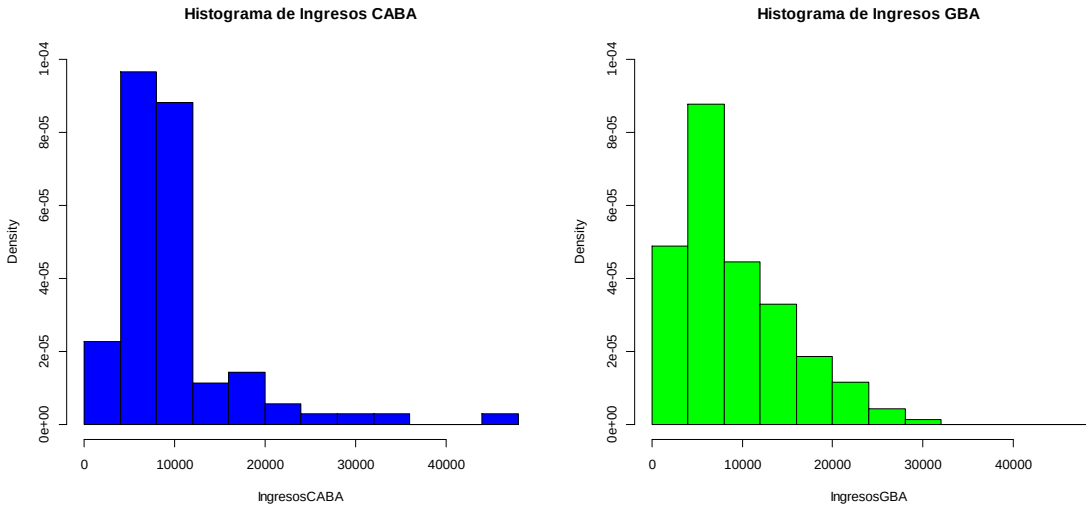
Vamos a fijar la longitud de cada intervalo 4000 (elegimos esta longitud de acuerdo a los datos observados).

```
>par(mfrow=c(1,2))# es para dividir la pantalla gráfica en 1 fila y
dos columnas para poder visualizar mejor.
>hist(IngresosCABA,breaks=seq(0,48000,4000),col="blue",ylim=c(0,60),
main="Histograma de Ingresos CABA")
>hist(IngresosGBA,breaks=seq(0,48000,4000),col="green",ylim=c(0,60),
main="Histograma de Ingresos GBA")
```



El default para graficar es una escala de frecuencias, si queremos trabajar con las probabilidades en el eje y, lo debemos especificar

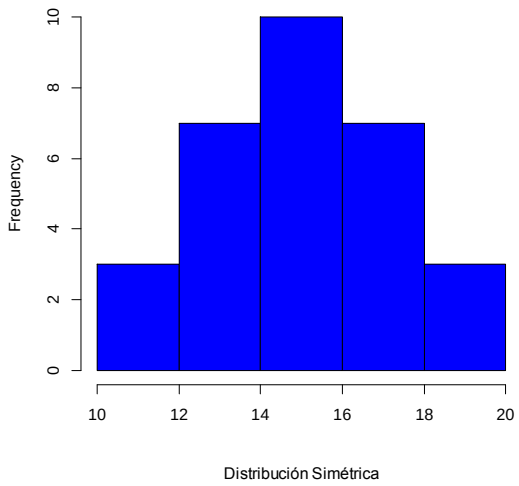
```
>hist(IngresosCABA,breaks=seq(0,48000,4000),freq=F,col="blue",ylim=c(0
,0.0001), main="Histograma de Ingresos CABA")
>hist(IngresosGBA,breaks=seq(0,48000,4000),freq=F,col="green",ylim=c(0
,0.0001), main="Histograma de Ingresos GBA")
```



¿Qué características nos interesan visualizar de la distribución?

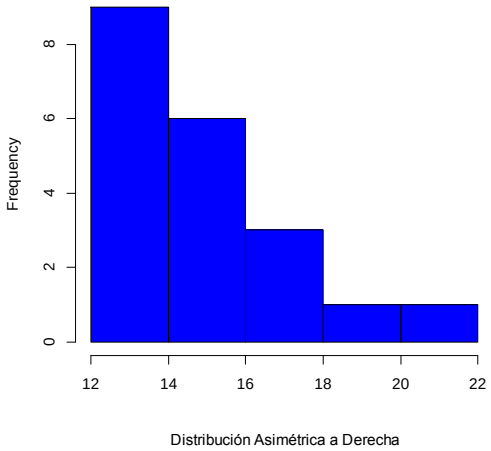
Principalmente analizar si la distribución es simétrica o asimétrica y detectar valores atípicos(outliers) .

Distribución simétrica:



La media, la varianza y el desvío stándard son buenas medidas de resumen.

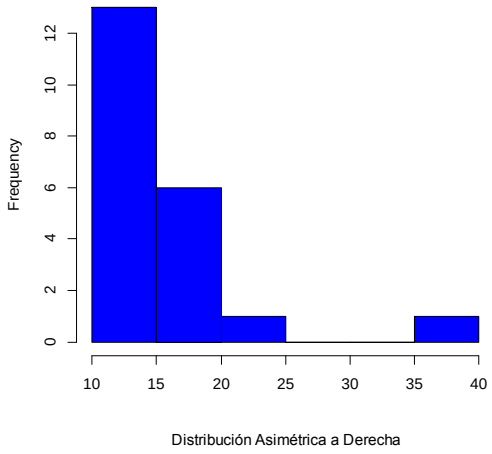
Distribución asimétrica:



La media y la varianza no son buenas medidas, se debe utilizar la mediana, y el rango intercuartil.

Datos atípicos o outliers

Supongamos que a los datos del gráfico anterior le agregamos un dato atípico, obtenemos el siguiente histograma:



Vemos claramente que hay un valor atípico, o sea un valor muy alejado del resto.

El histograma es muy influenciado por valores atípicos y puede dificultar su interpretación.

Los valores atípicos o outliers son valores muy alejados del resto, o lejano al patrón de los mismos. Estos valores no son muy visibles en un histograma y por el contrario pueden generar un rango muy grande y modificar engañosamente el aspecto de la distribución. Por ello hay dos tipos de gráficos que permiten una mejor visualización de la distribución: Gráfico Tallo-hoja y Boxplots.

- **Gráfico Tallo-Hoja**

Este gráfico es muy similar al histograma pero representa directamente los dígitos de los valores observados en vez de barras, por lo cual conservan mayor información.

Para el conjunto de datos simétrico:

```
The decimal point is at the |
  10 | 6
  11 | 16
  12 | 223
  13 | 1356
  14 | 13367
  15 | 33568
  16 | 2356
  17 | 256
  18 | 25
  19 | 7
```

Para el ejemplo del último histograma con el outlier:

```
The decimal point is 1 digit(s) to the right of the |
  1 | 222233444444
  1 | 5557779
  2 | 2
  2 |
  3 |
  3 | 6
```

Para los Ingresos de BaseEPH

```
> stem(Ingresos)

  The decimal point is 3 digit(s) to the right of the |
   0 | 58880000568
   2 | 00344555900000022289
   4 | 000000000002335555677880000000000000112455668
   6 | 00000000000000015678890000000001222555556677899
   8 | 000000000000011445555669000000 5578889
  10 | 000035556780000000000002566688
  12 | 000000000355567001556
  14 | 0000244555900
  16 | 0025556706
  18 | 056000145
  20 | 0700025
  22 | 006
  24 | 0000
  26 | 0
  28 | 5
  30 | 8
  32 |
  34 | 0
  36 |
  38 |
  40 |
  42 |
  44 |
  46 | 0
```

Por aglomerado

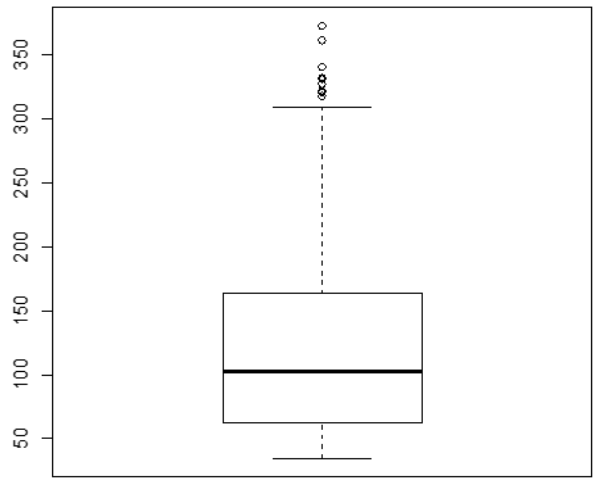
```
> tapply(Ingresos,Región,stem)
  The decimal point is 4 digit(s) to the right of the |
  0 | 233333444
  0 | 555555555666667777778888888888888888999999
  1 | 000000011111222222222344
  1 | 577899
  2 | 11
  2 | 5
  3 | 0
  3 | 5
  4 |
  4 | 7


  The decimal point is 3 digit(s) to the right of the |

 0 | 5888000058
 2 | 003445900022289
 4 | 00000000023555677800000000124568
 6 | 000000000000156789000000012255577
 8 | 0000000155690005788
  10 | 0558000000000268
  12 | 00035556001556
  14 | 0002455900
  16 | 00255706
  18 | 600145
  20 | 00025
  22 | 006
  24 | 000
  26 | 0
  28 |
  30 | 8
$CABA
NULL
$GBA
NULL
```

El diagrama tallo-hoja permite evidenciar de manera más clara:

- la simetría de los datos
- el centro de la distribución
- la dispersión de los datos
- la existencia de gaps o huecos en la distribución
- la existencia de valores muy frecuentes
- la existencia de valores atípicos
- visualizar concentración de datos

Una desventaja es que no es muy útil cuando el tamaño de los datos es grande ni muy práctico para comparar varias poblaciones.

- **Boxplots o diagrama de caja y bigotes (Box and Whiskers)**



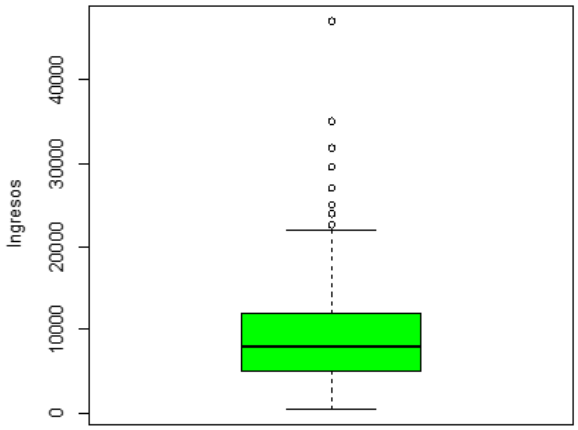
Un boxplot se construye dibujando :

- una caja cuyos extremos son los cuarteles inferior y superior y con una barra horizontal en la mediana.

- una línea desde el extremo superior de la caja hasta el máximo valor de la muestra que se encuentra de dicho extremo superior a una distancia menor o igual a 1.5 la distancia intercuartil
- una línea desde el extremo inferior de la caja hasta el mínimo valor de la muestra que se encuentra de dicho extremo inferior a una distancia a 1.5 la distancia intercuartil
- los valores que caen más allá de las líneas (bigotes) se los considera valores atípicos; moderados si la distancia es inferior a 3 veces la distancia intercuartil y severos si es mayor.

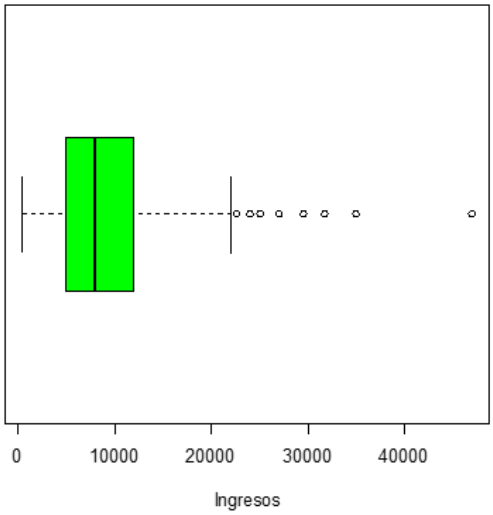
Podemos utilizar las opciones para modificar el aspecto:

```
>boxplot(Ingresos,col="green",ylab="Ingresos")
```



Si preferimos podemos construirlo en forma horizontal:

```
>boxplot(Ingresos,col="green",xlab="Ingresos",horizontal=T)
```

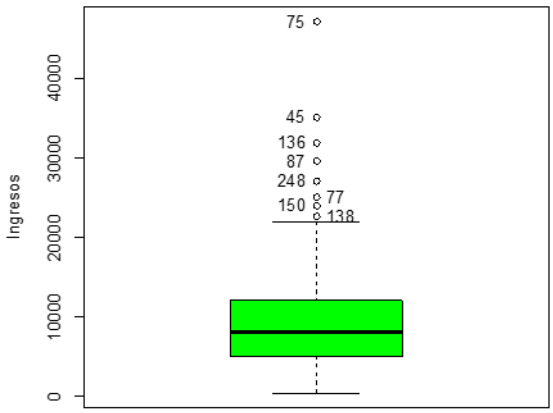


Nos gustaría identificar a los puntos que se muestran como atípicos; para ello hacemos uso de una de las *funciones interactivas*:

identify(x,y,labels): a partir de una ventana gráfica activa, identifica las coordenadas x,y de los puntos que el usuario marque sobre el gráfico con el botón izquierdo del mouse y le asigna la etiqueta especificada en labels.

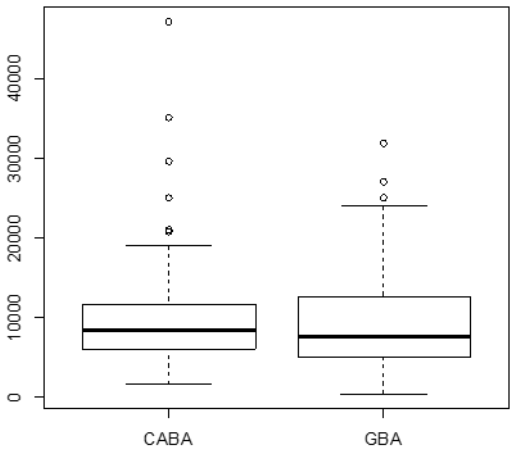
Para la coordenada x generamos un vector de 1's con longitud igual a la cantidad de filas del data.frame porque el boxplot toma como 1 la coordenada x, como coordenada y utilizamos lo que graficamos, que es la variable Ingresos, y como etiqueta, el nombre de la fila.

```
> boxplot(Ingresos,col="green",ylab="Ingresos")
> identify(rep(1,length(Ingresos)),Ingresos,rownames(DatosEPH))
 [1]  45  75  77  87 136 138 150 248
```

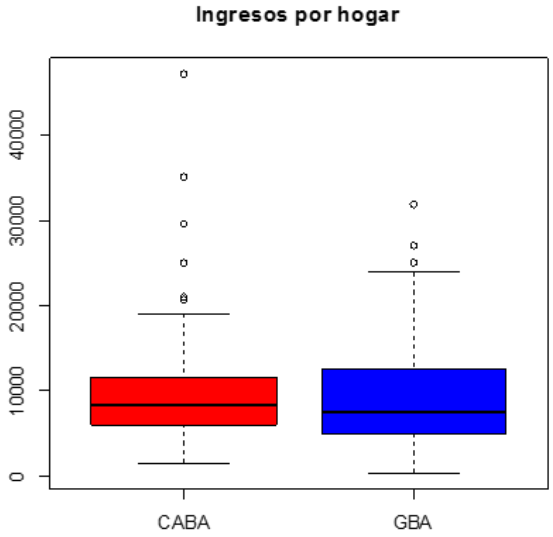


Por aglomerado podemos graficar los Boxplots en paralelo

```
>boxplot(Ingresos~Región)
```




```
>boxplot(Ingresos~Región,col=c("red","blue"), main="Ingresos por
hogar")
```



#Vamos a generar otra variable, Cant=número de integrantes del hogar

```
cant<-DatosEPH$ITF/DatosEPH$IPCF
cant
DatosEPH2<-cbind(DatosEPH,cant)
edit(DatosEPH2) #cerramos para continuar

#Guardamos la nueva base
write.csv2(DatosEPH2,"C:\\Users\\Usuario\\Desktop\\Curso de R\\Clase 2\\DatosEPH2.csv")
```