

Contents

O traballo con gráficos en R	1
Alguns parámetros gráficos	8

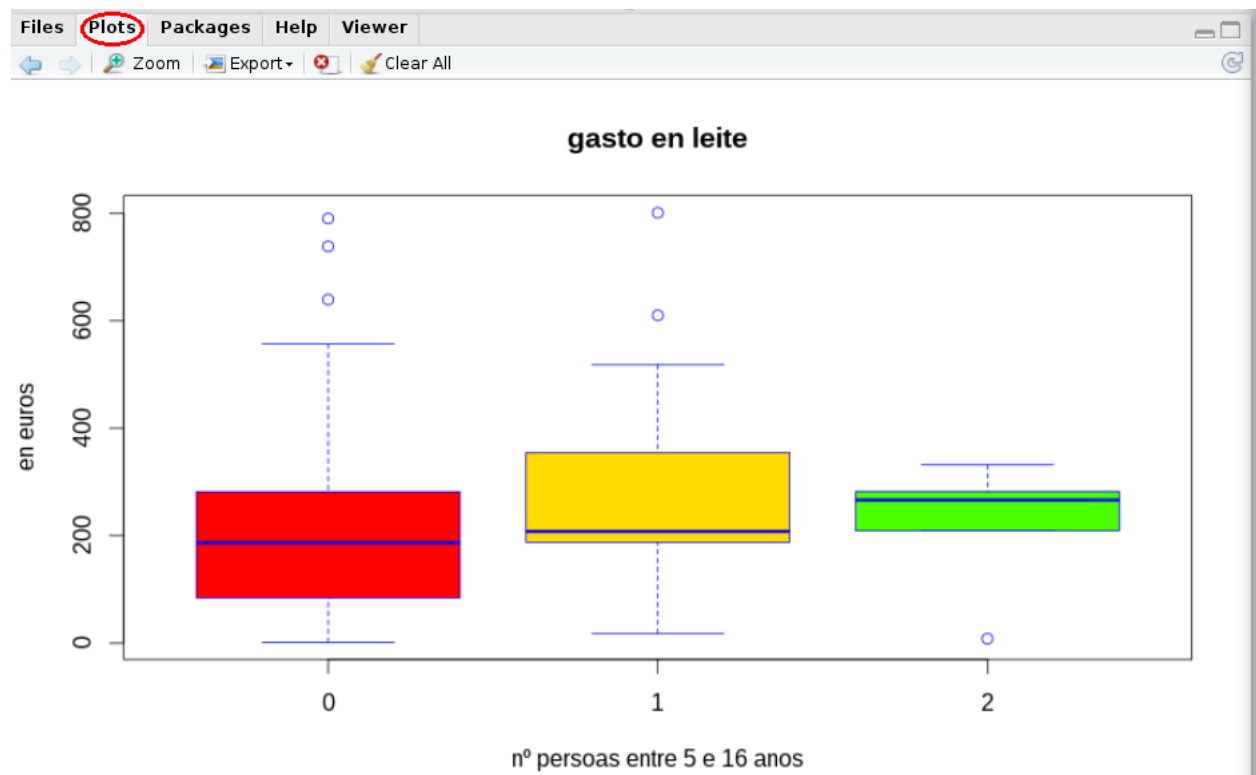
O traballo con gráficos en R

Esta parte esta pensada máis como anexo para consulta que como material de clase, xa que a elaboración básica de gráficos, que é o que correspondería a este cursoño xa está cuberto dentro da parte de Estatística básica.

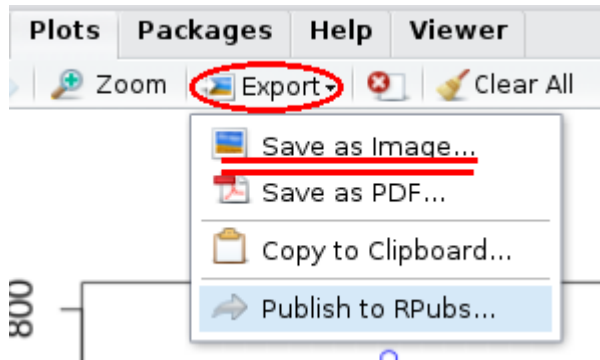
Así, esta parte inclúe referencias para coñecer aspectos técnicos do traballo con gráficos en R, pero carece de exemplos explicados da súa aplicación que se deben buscar noutras fontes.

Gardar gráficos con RStudio

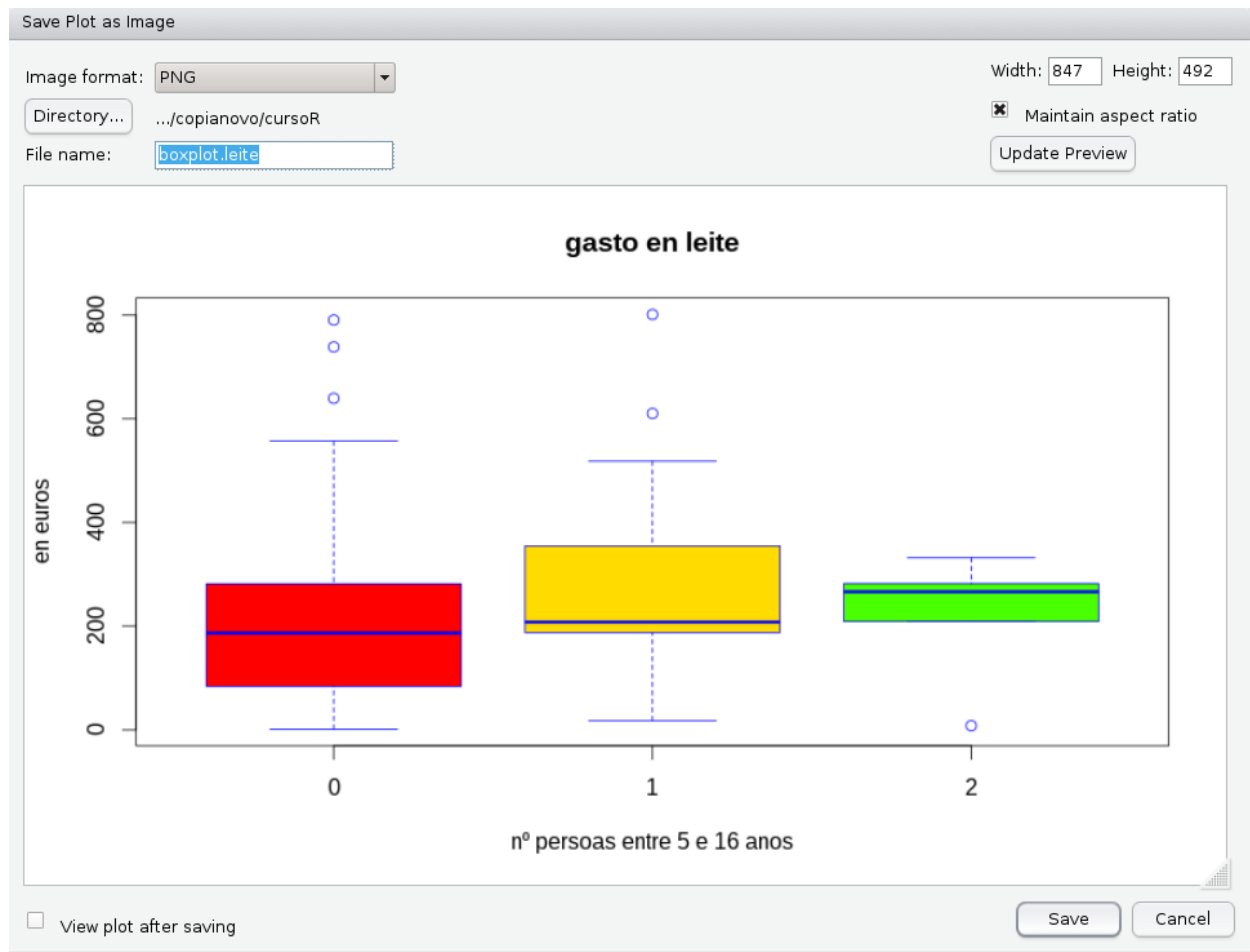
O espazo para os gráficos en RStudio está na pestana **Plots**.



Nesta pestana tamén se permite unha xestión básica dos gráficos, como por exemplo gardalos en diferentes formatos, no botón **Export**:



aí vaise poder escoller entre diferentes formatos de gráfico, cambiar o nome por defecto, modificar o tamaño e algunhas cousas máis:



Gardar gráficos con R

Ao usar R con RStudio como interface a maneira habitual de gardar gráficos será mediante esa interface, polo menos cando é un numero pequeno de gráficos..

As interfaces propias de R tamén adoitan vir con algún sistema de menu para gardar os gráficos (polo menos en Windows), pero non melloran o proporcionado por RStudio.

Ademais existe a posibilidade de gardar os gráficos mediante comandos, e isto si aporta unha vantaxe, posto

que permite automatizar o proceso, e quizais para un ou dous gráficos non mereza a pena, pero para 17 gráficos o aforro de traballo pode ser importante.

Dispositivos Gráficos (Devices)

R produce os gráficos nun *dispositivo gráfico (Device)*, que varía en función do sistema operativo ou de como se pretenda obter o gráfico.

En RStudio a xanela na que se produce o gráfico esta xestionado por un dispositivo do propio RStudio, pero cando non se usa esa interface os dispositivos son **X11** para Unix/Linux, **Windows** en Windows e **Quartz** en Mac. Estes dispositivos limítanse a amosar a saída gráfica, usando o menu da interface. Para gardar os gráficos R dispoñe doutros dispositivos, que varían en función do tipo de formato gráfico que se queira empregar: *pdf()*, *postscript()*, *pictex()*, *svg()*, *png()*, *jpeg()*, *bmp()*, *tiff()**

Un exemplo de uso é o seguinte, que garda o gráfico nun ficheiro con formato *png*. Con este código o gráfico débuxase directamente no ficheiro.

```
#Este primeiro comando abre un 'device' png, ou sexa, o dispositivo que manexa os gráficos
#vai escribilos directamente nun ficheiro dese tipo.
#ademais faino cun fondo transparente (bg="transparent")
#ancho de 1024 pixels e altura de 768
png(file = "myplot.png", bg = "transparent", width = 1024, height = 768)
barplot(c(3,1,1,4)) #realiza o gráfico, coma sempre
dev.off() #pecha o "device", agora os gráficos apareceran onda sempre
```

Usando ese sistema é necesario pechar o dispositivo para que se vexa o gráfico. Ademais, calquera outro gráfico que se realizase iría a parar ao mesmo sitio, co que eliminaría o xa feito.

Outra cuestión a ter en conta é o nome que se asigna: R crea un gráfico con ese nome, e non pregunta para borrar calquera outro que exista e que xa use ese mesmo nome.

Por esas cuestións habería que abrir e pechar o *device* para cada novo gráfico, cambiando o nome para cada ocasión.

Outra posibilidade é pedirlle que vaia numerando os ficheiros graficos a medida que os crea. Iso faise incluíndo no nome do ficheiro a expresión *%d*:

```
png(file = "myplot%d.png")
barplot(c(3,1,1,4)) #realiza o gráfico myplot1.png
hist(rnorm(100))#realiza o gráfico myplot2.png
plot(rnorm(20),runif(20))#realiza o gráfico myplot3.png
dev.off() #pecha o "device", agora os gráficos apareceran onda sempre
```

Sistemas gráficos no R

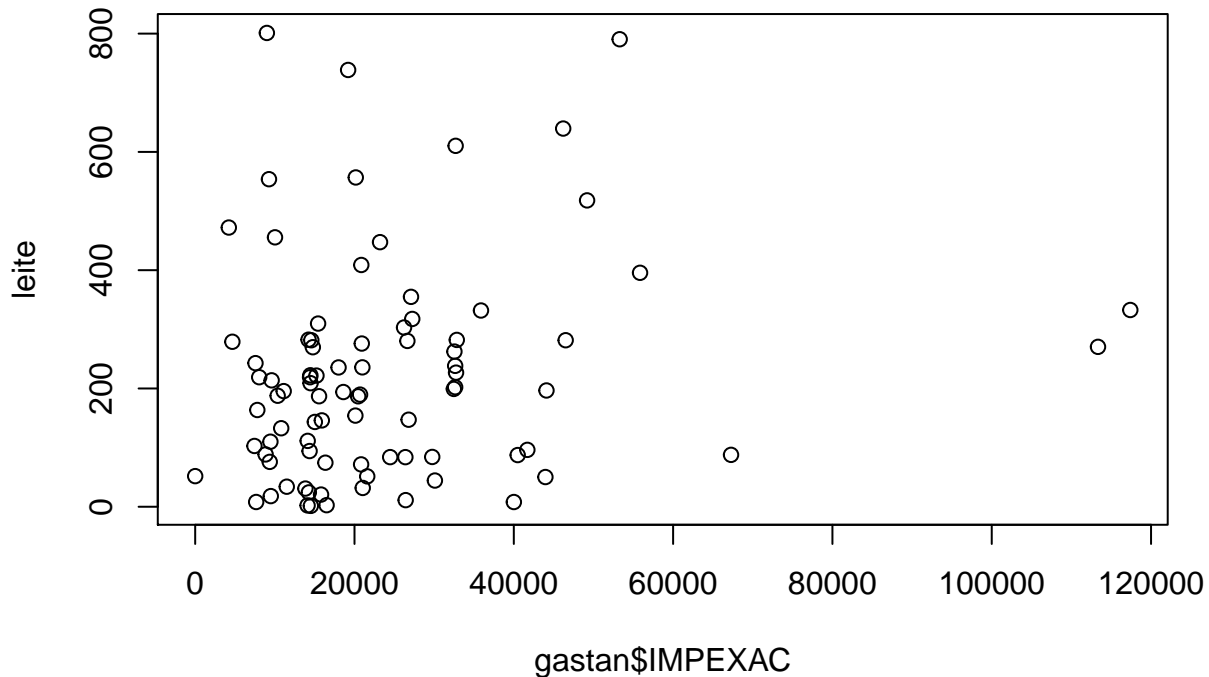
R manexa tres sistemas gráficos diferentes. Cada un deles permite elaborar gráficos, en ocasións incluso os mesmos tipos, pero con diferentes comandos e sobretodo con diferentes maneiras de traballar e de conseguir os obxectivos.

- Sistema básico

É o sistema orixinario do R. Aquí cada comando produce un resultado simple, e permite ir modificando o resultado sobre a marcha. Ademais ten diferentes comandos para engadir diferentes características dun mesmo gráfico.

```
plot(gastan$IMPEXAC,leite)
#engade o título
title(main = "Relación entre gasto en leite e ingresos")
```

Relación entre gasto en leite e ingresos



O primeiro comando produce o gráfico, non ten título, que se engade co segundo comando. Este sistema produce gráficos moi elementais pero que son altamente persoalizables modificando diferentes comandos e parámetros.

```
par(bg = "azure2")
plot(gastan$IMPEXAC,leite, type = "n",xlab="Ingresos",ylab="gasto en leite")
rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], col = "aliceblue")

points(gastan$IMPEXAC,leite,col=c("red","blue")[unclass(lugar)],
       pch=c(4,2)[unclass(gastan$SEXOSP)],cex=0.7)
auxi=lugar=="outro"
abline(lm(leite[auxi]~IMPEXAC[auxi],data=gastan),col="blue")
```

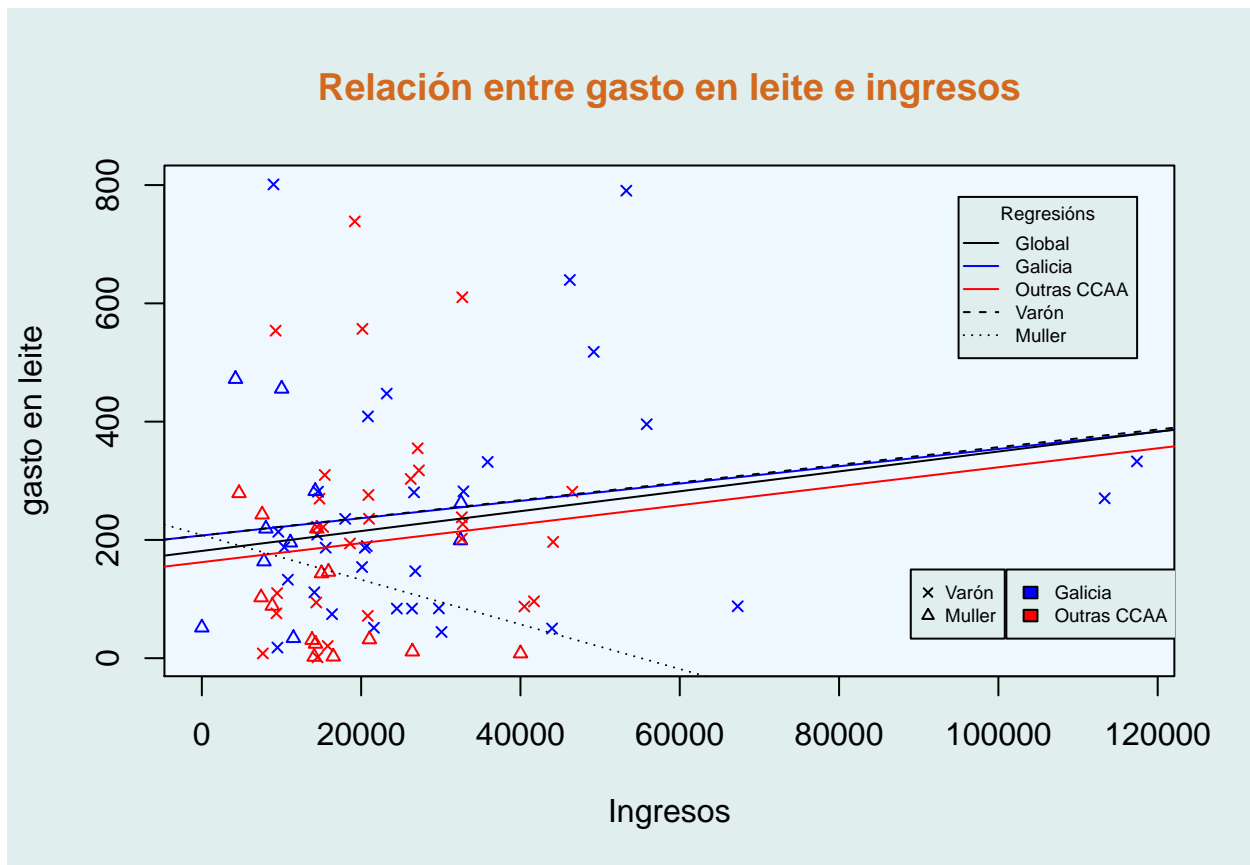
```

auxi=lugar=="gal"
abline(lm(leite[auxi]~IMPEXAC[auxi],data=gastan),col="red")
auxi=gastan$SEXOSP==1
abline(lm(leite[auxi]~IMPEXAC[auxi],data=gastan),lty="dashed")
auxi=gastan$SEXOSP==6
abline(lm(leite[auxi]~IMPEXAC[auxi],data=gastan),lty="dotted")

abline(lm(leite~IMPEXAC,data=gastan))
#engade o título
title(main = "Relación entre gasto en leite e ingresos",col.main="chocolate")

legend(101000,150,
      legend=c("Galicia", "Outras CCAA"),
      cex=.6,
      fill=c("blue","red"))
legend(89000,150,
      legend=c("Varón", "Muller"),
      cex=.6,
      pch=c(4,2))
legend(list(x = 95000,y = 780), title = "Regresións",
      legend = c("Global","Galicia","Outras CCAA","Varón","Muller"),
      col = c("black","blue","red","black","black"), lty = c(1,1,1,2,3),
      merge = TRUE,cex=0.6)

```



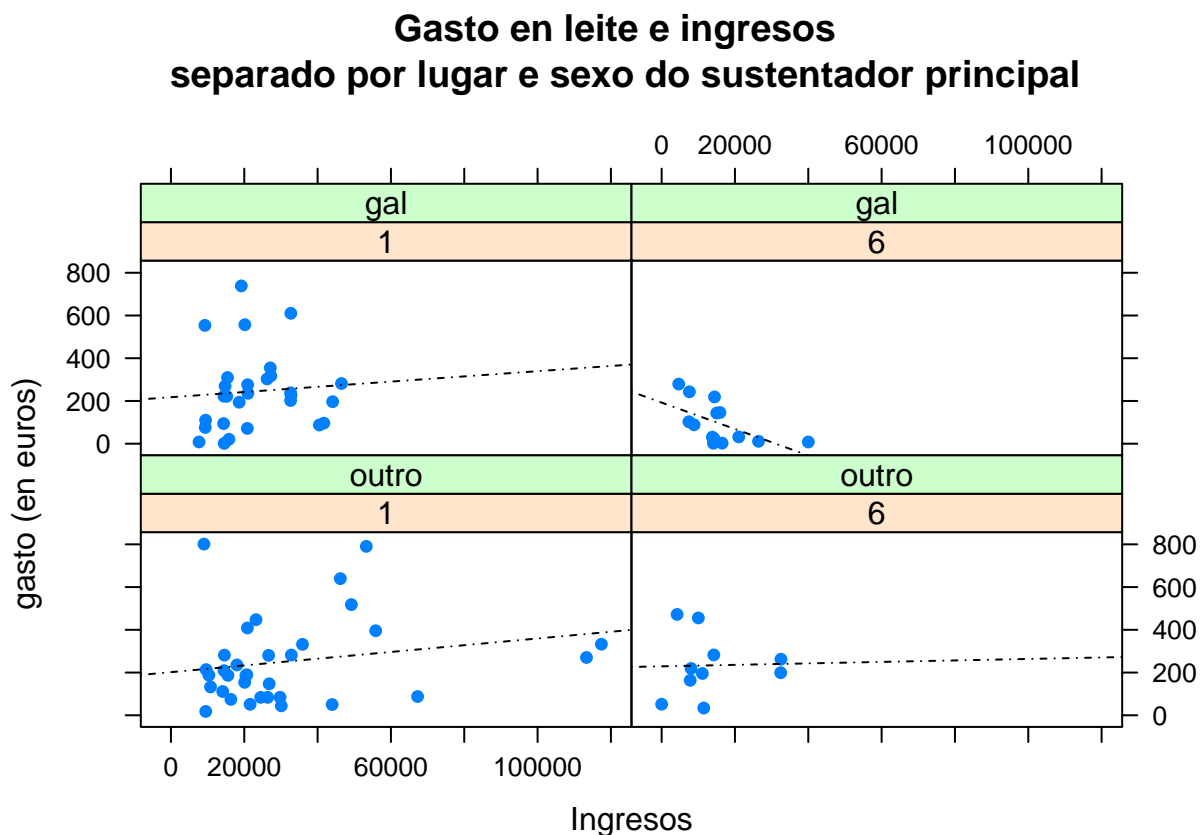
- Sistema Lattice

Está implementado no paquete *Lattice*. Esta incluído entre os paquetes contribuídos que se inclúen ao descargar a versión básica do R, pero non se carga en memoria ao iniciarse, se non que hai que cargalo a posteriori, cando se necesita.

Neste sistema a idea é que os comandos se ocupen de organizar o gráfico desde o principio, tamén se poden modificar os diferentes aspectos dunha gráfica de *lattice*, pero hai que incluír esas modificacións coa gráfica que o crea.

```
library(lattice)

xyplot(leite ~ IMPEXAC | SEXOSP+lugar, data = gastan, layout = c(2, 2),
       xlab = "Ingresos", ylab = "gasto (en euros)",
       main =
         "Gasto en leite e ingresos \n separado por lugar e sexo do sustentador principal",
       panel = function(x, y) {
         panel.xyplot(x, y, pch = 16, cex = 0.8) #pch:tipo de punto
         # cex:tamaño
         panel.lmline(x, y, lty = 4) #lty: tipo de liña
       }, as.table = T)
```



Unha vantaxe deste sistema é a súa facilidade para gráficos condicionados, ou sexa, gráficos nos que se representa un gráfico diferente para grupos de individuos clasificados por un ou varios factores.

- Sistema ggplot

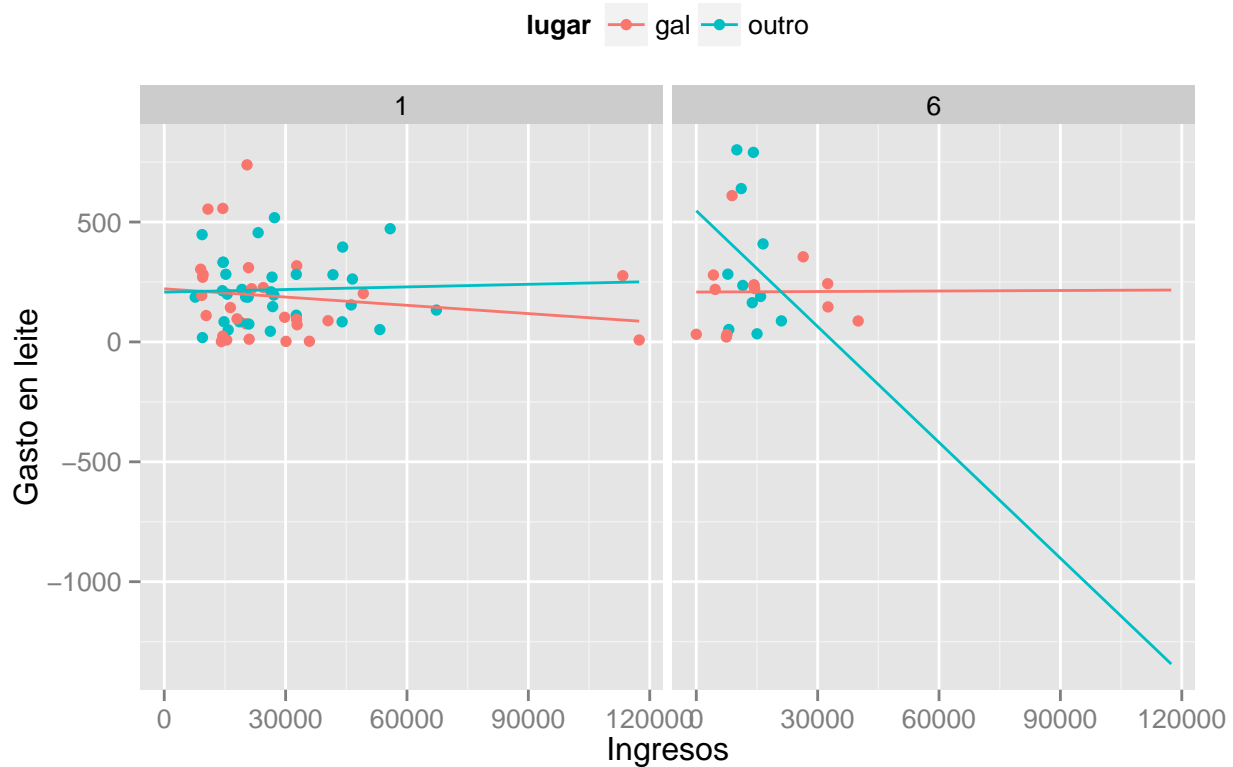
A idea detras deste sistema é a de manexar *unha gramática* que describa os diferentes aspectos dun gráfico.

Na súa implementación comparte aspectos do sistema base e de Lattice, xa que o sistema encárgase de organizar todos os aspectos do gráfico, como o lattice, pero permitindo ampliacións posteriores como o sistema base.

Usa o paquete *ggplot2*, que non está incluído na instalación básica do R polo que é necesario instalalo e cargalo despois en memoria, cando se vaia utilizar.

```
library(ggplot2)

ggplot(data=gastan, aes(IMPEXAC, leite,color=lugar))+
  geom_point( size=2)+ xlab("Ingresos")+ylab("Gasto en leite")+ facet_grid(.~SEXOSP)+
  geom_smooth(method=lm, se=FALSE, fullrange=TRUE)+
  theme(legend.position="top")
```



```
#Adaptado de: http://blog.espol.edu.ec/nemo/2011/01/16/creando-grafico-dispersion-ggplot2-part3/  
#sthash.GbXrBcxfb.dpuf
```

Algúns parámetros gráficos

Parámetros dos gráficos elementais

Os seguintes son algúns dos parámetros que se manejan cos diferentes gráficos elementais de R, cambian diferentes aspectos, como poden ser as cores ou os títulos dos eixes. Máis información sobre eles aparece buscando a *axuda* dos comandos, se ben en algun caso esta axuda pode redirixirnos á axuda do comando *par()*.

- pch: símbolo para crear puntos, por exemplo en diagramas de dispersión.

1: ○	2: △	3: +	4: ×	5: ◇	6: ▽	7: ☒	8: ✱
9: ⬠	10: ⊕	11: ⚡	12: ⊞	13: ☒	14: ☒	15: ■	16: ●
17: ▲	18: ◆	19: ●	20: ●	21: ○	22: □	23: ◇	24: △
25: ▽							
33: !	34: "	35: #	36: \$	37: %	38: &	39: '	40: (
41:)	42: *	43: +	44: ,	45: -	46: .	47: /	48: 0
49: 1	50: 2	51: 3	52: 4	53: 5	54: 6	55: 7	56: 8
57: 9	58: :	59: ;	60: <	61: =	62: >	63: ?	64: @

- lty: Tipo de liña, (sólida, de puntos, de raias, ...)

0.	blank	
1.	solid	——
2.	dashed	----
3.	dotted
4.	dotdash	-.-.-. .
5.	longdash	- - - -
6.	twodash	- . - . - .

- lwd: O ancho de liña
- col: cores aplicadas no gráfico. Sobre a utilización de cores en R hai exemplos e información na web **R Graphics Gallery**:<http://research.stowers-institute.org/efg/R/index.htm>, na sección *Color*
- main: Título para o gráfico
- xlab: Título para o eixe das X
- ylab: Título para o eixe das Y

Parámetros do sistema base

Estes parámetros configuran o espazo no que se vai debuxar o gráfico. Determinarían características de aspecto previas á realización do gráfico, se ben algunhas poden ser modificadas a posteriori, cando se completa o debuxo,

Estes parámetros modifícanse empregando o comando *par()*, sendo na axuda deste comando onde se pode atopar máis información sobre cada un deles.

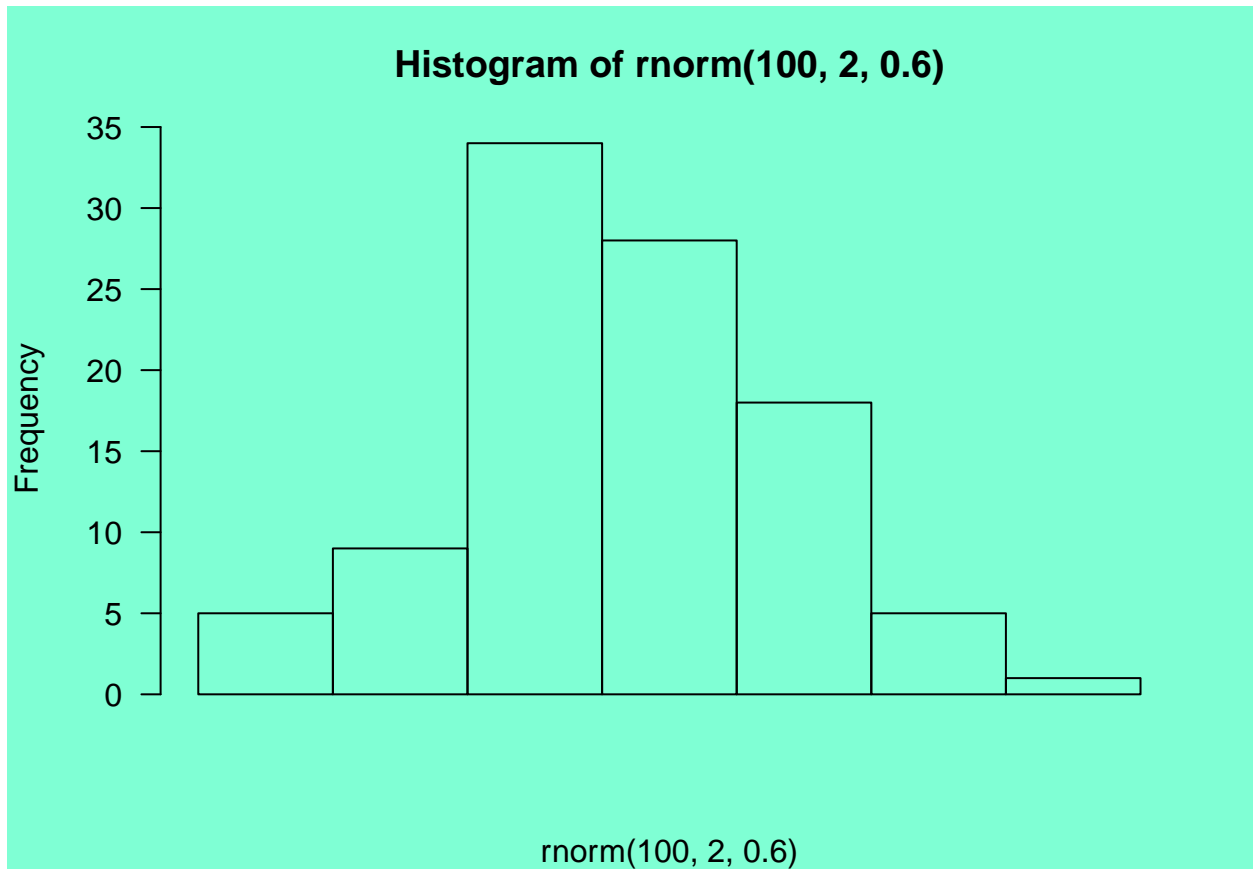
```
par(nome.de.parámetro)=algo #cambia o parámetro
par(nome.de.parámetro) #da o valor por defecto do parámetro
```

- las: A orientación das etiquetas no eixe
- bg: A cor de fondo
- mar: O tamaño da marxe
- oma: O tamaño da marxe externa, que é 0 por defecto
- xaxt: Especifica se debuxar (=“s”) ou non (=“n”) o eixe X
- yaxt: Especifica se debuxar (=“s”) ou non (=“n”) o eixe Y

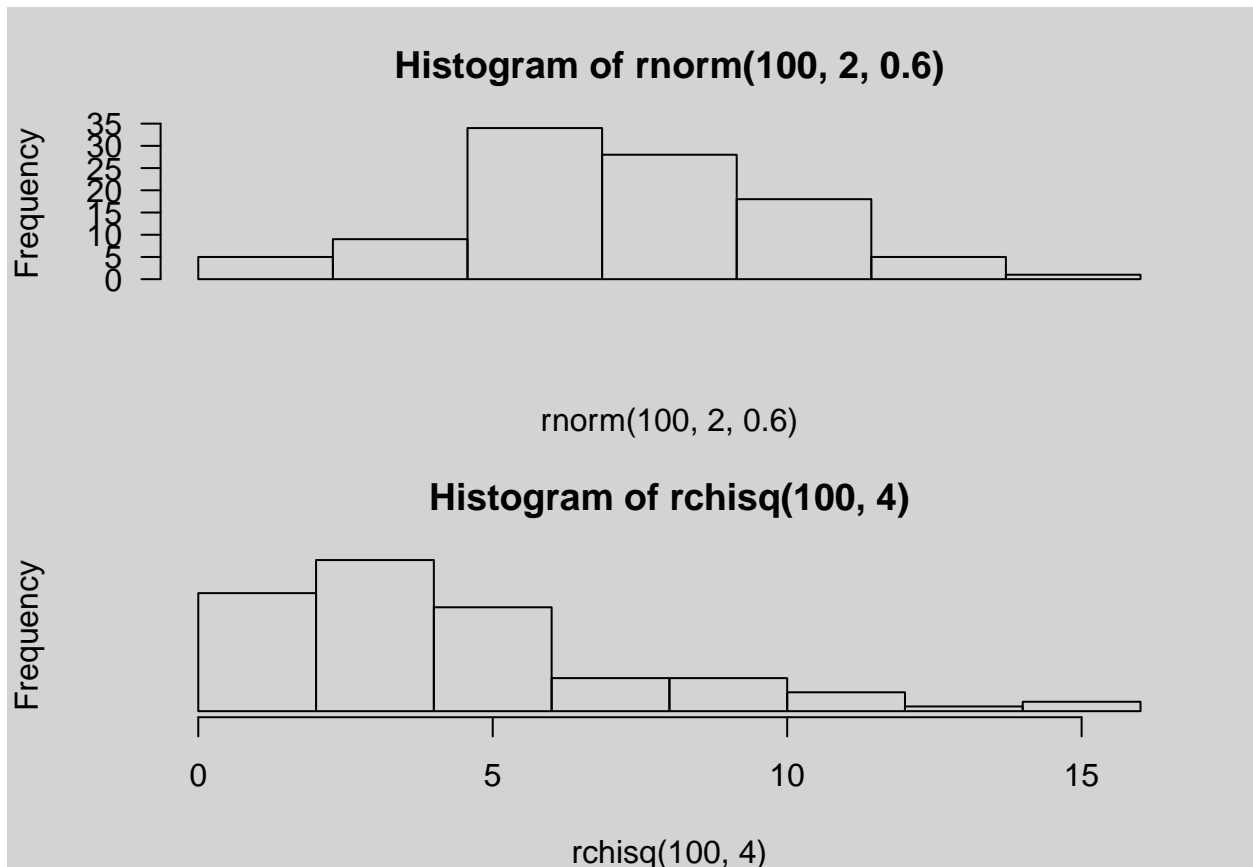
Os dous seguintes parámetros permiten combinar varios gráficos nunha única vista:

- mfrow: n^o de gráficos por liña e columna (1^o completa unha liña, despois as seguintes)
- mfcoll: n^o de gráficos por columna e liña (1^o completa unha columna, despois as seguintes)

```
set.seed(872345)#parámetro para fixar os valores aleatorios producidos por rbinom
par(mar=c(4, 4, 3, 1), oma=c(0,0,0,1),bg="aquamarine", xaxt="n", yaxt="s", las=2)
hist(rnorm(100,2,0.6))
```



```
set.seed(872345)#parámetro para fixar os valores aleatorios producidos por rbinom
par(mfrow=c(2,1))
par(mar=c(4, 4, 3, 1), oma=c(0,0,0,1),bg="light gray", xaxt="n", yaxt="s", las=2)
hist(rnorm(100,2,0.6))
par(xaxt="s", yaxt="n", las=1)
hist(rchisq(100,4))
```



```
par(mfrow=c(1,1)) #Para retornar a 1 gráfico por vista
```

Elementos para engadir aos gráficos básicos

- lines: debuxa liñas no gráfico, a partir dun vector de valores X e outro de valores Y.
- abline: representa unha liña de regresión
- points: engade puntos no gráfico
- text: engade etiquetas de texto nun punto determinado
- title: engade etiquetas de texto para o título principal, eixes X e Y, subtítulo.
- mtext: engade texto nas marxes da gráfica
- axis: engade un eixe, podendo especificar as súas características, como por exemplo se é o eixe X ou Y, as etiquetas que apareceran, etc.
- polygon: debuxa unha liña poligonal, determinada por dous vectores de puntos (X e Y)
- rect: debuxa un rectángulo nas coordenadas dadas

A información sobre estes comandos pode conseguirse na axuda de cada un deles.

```
x = seq(-5,5, length=250)
y = dnorm(x) #dnorm produce os valores da densidade Normal, nos puntos indicados
plot(x,y, las=1, ylab='dnorm', type='n', yaxs='i'
     , ylim=c(-0.2, 0.5), axes = F,xlab=NA) #Plot é un comando xenérico para gráficos
x2 = seq(qnorm(0.95), 5, length=50)
y2 = dnorm(x2)
polygon(c(x2[1], x2, x2[length(x2)]), c(0, y2, 0), border=NA, col='grey')

lines(x, y)
cex=0.5
lines(c(2.5, 2.1), c(0.1, 0.02))
#'expression' permite incluír símbolos no texto dun gráfico
#máis información na axuda de 'plotmath'
text(2.8, 0.175, expression(alpha=="nivel de"))
text(2.8, 0.125, "significación")

text(1.645,-0.025,expression("z"[alpha]))
text(0.6,0,"x")
lines(c(-5, 5), c(0, 0))
lines(c(1, 0.6), c(-0.05, 0))
text(1.05,-0.05,"D")
lines(c(1.645, 5), c(0, 0),col="red",lwd=4)

text(-0.1,0.47,expression(paste(H[0]," correcta")),cex=1.5)

text(3.6,-0.05,"Rexión crítica", col="red")

lines(c(-5,1.645), c(0, 0),col="blue",lwd=2)
text(-2,-0.05,"Rexión de aceptación", col="blue")
text(1.05,-0.1,"Guai!!")

text(2.1,0,"x",col="red")
lines(c(2.5, 2.1), c(-0.1, 0))
text(2.52,-0.125,"D",col="red")
text(2.52,-0.17,"Erro tipo I",col="red")

x2 = seq(0.6, 5, length=100)
y2 = dnorm(x2)
```

```

polygon(c(x2[1], x2, x2[length(x2)]), c(0, y2, 0), border=NA
, density=10,col='green')
text(2,0.3,"p-valor",col="green")
lines(c(2,0.9), c(0.28, 0.18))

```

H_0 correcta

