

Sessió 5

Anàlisi exploratòria d'una sola variable

Presentem la part més elemental amb gràfiques “austeres” que aprendrem a editar més endavant i amb les característiques bàsiques. En properes pràctiques analitzarem amb cura l'entorn gràfic i ampliarem les anàlisis.

- Creem un petit fitxer de dades que utilitzarem en els propers exemples. Són les notes i els grups de 9 estudiants:

```
x1<-c(9,4,8,2,1,7,8,10,1); x2<-c(5,8,8,0,4,5,5,8,4)
x3<-c("A","B","B","B","A","A","A","B","B")
df1<-data.frame(alg=x1,cal=x2, grup=x3)
```

La funció attach

Quan hem de treballar molt en un mateix data.frame, per exemple df1, a vegades es fa farragosa la notació que hem après, df1\$alg, per referir-nos a la variable alg dins del data.frame df1.

Només si estem treballant amb un sol data.frame no pot haver-hi confusió de noms. En aquest cas pot ser útil la funció attach. Ens permet accedir a les variables del fitxer de dades df1 sense avisar que estan dins de df1 amb \$.

```
> alg      # no ho entén
Error: object 'alg' not found
> attach(df1)  # fem accessibles les variables de df1 amb el seu nom només
> alg          # ara sí que entén que es tracta del vector numèric
[1]  9  4  8  2  1  7  8 10  1
```

- Si es pot evitar, millor no utilitzar attach, perquè sense adonar-nos podem tenir dos vectors que es diguin igual.

```
# si a més de df1 tenim un vector anomenat també grup
grup<-c('A','B', 'AB', '0','A','A','B','0',0)
attach(df1)
grup
[1] "A"  "B"  "AB" "0"  "A"  "A"  "B"  "0"  "0"
```

- Observem que en fer `attach(df1)` els noms de les variables no apareixen en el nostre entorn de treball (environment o `ls()`), encara que si escrivim només `alg`, **R** entén que ens referim al vector `df1$alg`.

```
attach(df1)
ls()
[1] "df1" "x1"  "x2"  "x3"
```

- Per desfer l'adjunció de les variables (l'`attach`) només cal utilitzar `detach`

```
detach(df1)
> alg
Error: object 'alg' not found
```

5.1 Caraterístiques numèriques

- Els descriptius o característiques numèriques més importants d'una variable quantitativa són

```
length(df1[[1]]); length(df1$alg) # nombre d'elements de la variable
mean(df1[[1]]); mean(df1$alg)    # la mitjana (aritm.) de "alg"
var(df1$cal); sd(df1$cal)        # variància i desv. típica de "cal"
median(df1$alg)                  # la mediana
range(df1$cal)                   # mínim i màxim, és un vector de long. 2
(R<-range(df1$cal)[2]-range(df1$cal)[1]) # Calculem R=max-min
```

Atenció: la variància és la corregida i la desviació típica la seva arrel quadrada.

$$\text{var} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} \quad \text{sd} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

- Fem `attach` de l'arxiu per escurçar les comandes i repetim l'anterior:

```
attach(df1)
length(alg); mean(alg); median(alg); var(cal); sd(cal)
range(cal); (R<-range(cal)[2]-range(cal)[1])
# d'una altra manera:
(R<-max(cal)-min(cal))
```

- Es poden imprimir resultats amb noms, amb la funció `print` i paste o bé amb `list`:

```
print(paste("mitjana de càlcul=", mean(cal)))
print(paste("rang=", R))
print(list("mitjana càlcul"=mean(cal), "rang càlcul"=R)) # ahora
```

- Fem algunes comprovacions:

```
# "var=sd^2"?
var<- var(cal); s<- sd(cal)
var==s^2
```

```
## per evitar errors d'arrodoniment seria millor fer:
isTRUE(all.equal(var,s^2))
### ATENCIÓ que són la variància i la desviació típiques mostrals
# comprovem la Md ordenant i agafant el valor que ocupa la pos.central:
median(alg)
sort(alg)      # mediana: valor que ocupa la posició 5 (de 9)
```

- La suma de tots els valors, relació amb la mitjana aritmètica

```
sum(alg)      # la suma
# la mitjana és la suma de tots els valors dividit per n?
sum(alg)/length(alg)==mean(alg)
## per evitar errors d'arrodoniment seria millor fer:
suma<-sum(alg); n<-length(alg)
isTRUE(all.equal(suma/n,mean(alg)))
```

- Quantiles: percentils, decils, quartils, ...

```
?quantile
quantile(alg,probs=.5)    # quantila0.5=percentil50=mediana
quantile(alg,probs=.2)    # quantila0.2=percentil20=decil2
quantile(alg,probs=.333)  # quantila0.333
quantile(alg,probs=seq(0,1,by=.1)) # mínim, decils i màxim
quantile(alg,probs=seq(0,1,by=.25)) # mínim, quartils i màxim
```

Atenció amb la mediana calculada amb R: tal com l'hem definit a teoria, si es tracta d'una variable discreta amb pocs valors, millor afegir l'opció `type=2`.

```
quantile(alg,probs=0.5,type=2)
```

- Resums ràpids de diverses variables alhora (funcions `ColMeans`, `rowMeans`, `summary`):

```
colMeans(df1)          # atenció, 3a variable és de caràcters!
## info. error: "Error en colMeans(df1): 'x' debe ser numérico"
colMeans(df1[, -3])     # ara sí
rowMeans(df1[, -3])     # mitjanes per files
colSums(df1[, -3])      # suma per columnes
rowSums(df1[, -3])      # suma per files

# Resum complet:
summary(df1)            # cada variable segons el tipus
      alg          cal          grup
Min.   : 1.000   Min.   :0.000   A:4
1st Qu.: 2.000   1st Qu.:4.000   B:5
Median : 7.000   Median :5.000
Mean    : 5.556   Mean    :5.222
3rd Qu.: 8.000   3rd Qu.:8.000
Max.    :10.000   Max.    :8.000
# de la variable grup dóna les freqs. dels valors
```

Pràctica:

• Recupera l'script utilitzat per obtenir el data.frame `PerTur17` de la pràctica 4 (Enquesta sobre la percepció del turisme pels ciutadans de Barcelona ciutat). Pots obtenir-lo directament del Moodle (tot l'script de la pràctica, `sess4-practs`, o bé només el data.frame `PerTur17.RData`). Genera dos sub-data.frames que continguin les variables `SEXE`, `EDAT`, `DISTRICTE`, `SECTOR_RIQUESA` i `ALLOTJA_TUR_NRE`: Un per als homes (`PerTur17H`) i un per a les dones (`PerTur17D`).

a) Troba la mitjana, desviació típica, mediana, el rang i els quartils de l'edat pels homes i per les dones.

b) Fes un resum totes les variables pels homes i per les dones.

c) Imprimeix una frase que digui: "En l'enquesta sobre la percepció del turisme al 2017, entre els veïns que van allotjar algun turista a casa les dones en van allotjar una mitjana de x i els homes de y " substituint x i y pel valor que surti (arrodonint a un decimal).

La variable del nombre de turistes que han allotjat a casa és `ALLOTJA_TUR_NRE` i recorda que has de treure la no resposta.

• Genera una matriu de números aleatoris amb dos decimals entre 6 i 200 (amb llavor 1) anomenada `DESPESA` que representarà la despesa en lleure de 15 estudiants durant els 12 mesos de l'any. Anomena les columnes amb els noms dels mesos en anglès i les fileres amb `Alumne.1` fins `Alumne.15`.

a) Troba la despesa del total de l'any i la despesa mitjana per mes de cada alumne (arrodoneix a dos decimals).

b) Troba la despesa del total mensual i la despesa mitjana per cada mes (arrodoneix a dos decimals).

c) Fes un resum dels principals descriptius de cada mes i de cada alumne.

5.2 Taules de freqüències

- Taules de freqs., percentatges, etc. de variables qualitatives (o numèriques amb pocs valors diferents)

```
table(grup)      # var. qualitativa
table(cal)       # numèrica amb "pocs" valors diferents
#
df.cal<-as.data.frame(table(cal)) # reconvertim taula a arxiu
# ara completarem la taula ...
(ni<-df.cal$Freq) # freq. absolutes
n<-sum(ni)        # total d'observacions
(fi<-ni/n)        # freq. relatives
(pci<-fi*100)     # percentatges
(Ni<-cumsum(ni))  # freq. absolutes acumulades
(Fi<-Ni/n)       # freq. relatives acumulades
(Pci<-Fi*100)    # percentatges acumulats
(df.cal<-data.frame(df.cal,fi,pci,Ni,Fi,Pci))
options(digits=2)
df.cal           # és la taula completa
```

Noteu que R distingeix minúscules de majúscules (`ni` de `Ni`, etc.).

Pràctica:

• Seguirem treballant amb el data.frame `PerTur17`. Construeix la taula de freqüències absolutes i percentatges de les variables categòriques: `SECTOR_RIQUESA` (Quin sector econòmic creu que aporta més riquesa a la ciutat?) i `COMP_TUR` (Segons el que vostè veu diria que el comportament dels turistes a la ciutat és...?).

Té sentit fer el càlcul de les freqüències acumulades en els dos casos? Si la resposta és negativa, en quina de les dues variables té sentit? Per què?

5.3 Gràfiques estadístiques bàsiques

- Gràfiques d'una variable qualitativa (observeu que primer sempre cal fer table):

```
pie(table(df1$grup))      # sectors d'una variable categòrica
barplot(table(df1$grup))  # barres d'una variable categòrica
```

- Gràfiques d'una variable numèrica:

```
hist(df1$alg)      # histograma d'una variable numèrica
boxplot(df1$alg)   # boxplot d'una variable numèrica (la veurem a teoria)
```

- Diagrama de freqüències acumulades relatives:

```
plot(ecdf(cal))      # "ecdf" dona les freqs. relatives acumulades
plot.stepfun(cal, col.vert = "gray80") # anàleg
```

- Dues o més gràfiques alhora (les podem exportar en un sol fitxer gràfic) i altres modificacions:

```
op<-par(mfrow=c(1,2),cex=0.7) # paràmetres gràfics nous
## mfrow=c(1,2) vol dir: 1 fila i 2 columnes
## cex=0.7 canvia la mida de les etiquetes (al 70% de la mida estàndard)
plot(sort(unique(cal)),df.cal$pci,type="h",ylim=c(0,40))
plot(ecdf(cal))
par(op) # tornem als paràmetres gràfics per defecte
```

- Histograma amb intervals de diferent amplitud.

Suposem que la variable edat agrupada en intervals té la taula següent

Interval	Freqüència	
[0,16)	10	Per fer un histograma ens hem d'inventar unes dades que tinguin la mateixa taula de freqüències. Ho fem repetint la marca de classe de cada interval tantes vegades com indica la freqüència de l'interval. Per exemple, repetim 10 cops el 8 (la marca de la primera classe és 8 i la classe té freqüència 10), després 15 cops el valor 18, etc.
[16,20)	15	
[20,24)	20	
[24,40)	50	
[40,65)	40	
[65,100]	20	

```
# considerem els intervals d'edat següents:
## [0,16), [16,20) [20,24) [24,40), [40,65) [65,100)
# introduïm (manualment) les marques de classe
xi<-c(8,18,22,32,52.5,82.5) # punts mitjos
ni<-c(10,15,20,50,40,20)
edat<-rep(xi,ni)
talls<-c(0,16,20,24,40,65,100)
hist(edat,breaks=talls,ylim=c(0,0.1))
```

Pràctica:

Treballarem amb el data.frame `PerTur17`

- Fes una gràfica adequada per a cada una de les variables `TUR_BENEF`, `SEXE` i `EDAT`
- Crea un nou data.frame amb els enquestats del `DISTRICTE de CIUTAT VELLA` i un data.frame amb els enquestats del `DISTRICTE de SANT ANDREU`. amb la funció `par(mfrow=c(1,2))` fes dos gràfics, un al costat de l'altre, de la variable `TUR_BENEF` en cadascun dels dos barris. S'hi observen diferències?
- Crea dos data.frames, un amb els enquestats joves (de menys de 30 anys) i un amb les persones grans (més de 65). Estudia gràficament la variable `TUR_BENEF` en cada grup d'edat. Observes diferències?

5.4 Recodificació de dades.

Dins de la gestió de dades, sovint cal recodificar els valors d'una o més variables. Entenem per “recodificació” qualsevol canvi en els valors d'una variable per definir una nova variable o la mateixa variable. Ja n'hem vist alguns exemples: agrupar valors en intervals, definir nous nivells d'un factor. Ara veurem com fer-ho utilitzant la combinació d'operadors lògics amb la indexació de vectors.

- Recodificar en la mateixa variable però de manera que es perdi la variable inicial. En algun cas pot ser interessant perdre els valors antics, si aquests són erronis o no ens interessa analitzar-los. Vegem dos exemples:

- Tenim un vector d'edats amb valors superiors a 150 (sabem segur que són erronis). Els volem declarar valors perduts. Recodificaríem així

```
> edat<-c(10,20,32,15,3,160,200,44)
> edat[edat>130]<- NA
> edat
[1] 10 20 32 15 3 NA NA 44
```

- En l'enquesta del baròmetre del CIS del maig del 2019 hi ha la pregunta següent:

P.19 Como Ud. sabe, en España hay distintos partidos o coaliciones políticas a las que puede votar en unas elecciones. Me gustaría que me dijera cuál es la probabilidad de que Ud. vote a cada uno de los que le voy a mencionar, utilizando para ello una escala de 0 a 10, sabiendo que el 0 significa que "con toda seguridad, no lo votaría nunca" y el 10 significa que "con toda seguridad, lo votaría siempre". **(MOSTRAR TARJETA G)**.

	Escala	NS	NC
	00 -10	98	99
- PSOE.....	_____	98	99 (80)(81)

La variable `P1901` del fitxer que us podeu descarregar conté les respostes a aquesta pregunta. Volem recodificar-la de manera que els NS i NC, que estan ara codificats, respectivament, com a 98 i 99, passin a ser NA. Si el data.frame l'hem importat a l'objecte `enquesta`, escrivim

```
> summary(enquesta$P1901) # veiem que el màxim és 99
> enquesta$P1901[enquesta$P1901==98 | enquesta$P1901==99]<- NA
> summary(enquesta$P1901) # veiem el resultat de recodificar
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
0.000  0.000   5.000   4.572  8.000  10.000    198
```

- Un altre exemple

```
x<-c(1:4,99,99,NA,-1,-2,-99); x # parteix d'un vector
x[is.na(x)]<-0; x # recodifica els NA per 0
x[x<0]<- -x[x<0]; x # canvia el signe de les components negatives
x[x==99|x== -99]<-NA; x # si x val 99 o bé -99, es canvia per NA
```

- Recodificar en una nova variable, sense perdre la variable inicial (molt similar). Fixeu-vos en que cal inicialitzar el vector nou.

```
x<-c(1:4,99,99,NA,-1,-2,-99); x # parteix d'un vector
y<-numeric(length(x)) # inicialment y és un vector de 0 de la long de x
y[is.na(x)]<- 0 # recodifica alguns casos: si x és NA --> y és 0
y[!is.na(x)]<- -x[!is.na(x)] # recodifica els altres casos:
# si x no és NA --> y és -x
y[x==99|x== -99]<-NA # si x val 99 o bé -99, y val NA
x; y #compareu l'inicial x i el final y
```

Nota: Mantenim l'antiga x i hem creat la nova y.

- Per recodificar valors alfanumèrics, podem fer:

```
x<-sample(letters[1:6],20,rep=T) # creem una mostra de lletres
length(x)
X<-numeric(length(x))
X[x=="a"|x=="b"]<-1
X[x=="c"|x=="d"]<-2
X[x=="e"|x=="f"]<-3
x;X # observem com hem codificat
```

- En cas que vulguem recodificar amb només dos valors de sortida podem utilitzar la funció `ifelse` de R.

```
x<-c("dona","home","home","dona","home","dona","dona")
(x.cod<-ifelse(x=="dona",1,2))
[1] 1 2 2 1 2 1 1
y<-sample(0:10,20,replace=TRUE)
(y.cod<-ifelse(y>=5,"Aprovat","Suspès"))
[1] "Aprovat" "Suspès" "Aprovat" "Suspès" "Suspès" "Aprovat" "Aprovat" "Aprovat"
```

- També podem utilitzar funcions específiques per recodificar d'alguns paquets, com ara la funció `recode` del paquet `car`. Però n'hi ha d'altres, com ara les funcions `revalue` i `mapvalues` del paquet `plyr`.

```
library(car)
respostes<-sample(1:10,100,replace = T)
respostes.factor<-recode(respostes,"1:4='desacord';5='indiferent';
6:10='acord'", as.factor.result=T)
# as.factor.result=T quan volem que el resultat sigui un factor.
```

Pràctica: Recupera l'script utilitzat per obtenir el data.frame `PerTur17` de la pràctica 4 (Enquesta sobre la percepció del turisme pels ciutadans de Barcelona ciutat), o bé carrega directament el fitxer `PerTur17.RData` que trobaràs a l'apartat de pràctiques a "Fitxers sessió 5 i lliurament 5".

(a) Crea una nova variable (`HUT_MOLESTIES_PUNTS`) on recodificar la variable `HUT_MOLESTIES` ("Creu que els habitatges d'ús turístic ocasionen moltes, bastantes, poques o cap molèstia als veïns?") assignant les següents puntuacions:

Moltes molèsties = 10; Bastantes molèsties = 7.5; Poques molèsties = 2.5; Cap molèstia = 0

(Els codis "No ho sap" i "No contesta" han de quedar com a NA).

La nova variable s'ha de construir com a part del data.frame `PerTur17`

(b) Genera els següents sub-data.frames que continguin les variables `SEXE`, `EDAT`, `DISTRICTE`, `HUT_MOLESTIES`, `COMP_TUR` i `HUT_MOLESTIES_PUNTS`:

- Un pels que creuen que al seu edifici hi ha algun habitatge d'ús turístic (`df.hiHaHUT`) i un pels que no (`df.NoHiHaHUT`) (necessitaràs la variable `HUT_HIHA`);

- Un pels que valoren positivament la gestió de l'Ajuntament (`df.BonaGestio`) i un pels que no (`df.MalaGestio`) (necessitaràs la variable `GEST_TUR`).

(c) Calcula la mitjana i la desviació típica de la variable recodificada `HUT_MOLESTIES_PUNTS`, per a cada un dels segments (recorda que cal treure els NA). Podem dir que hi ha diferències en l'opinió els segments analitzats?

5.5 Més sobre gràfiques (opcional).

S'anomenen d'*alt nivell* aquelles funcions que creen una gràfica nova, com ara `plot`, `hist`, `pie`, `barplot`, `boxplot`. Són de *baix nivell* aquelles funcions que afegeixen elements a una gràfic (creat abans amb una funció d'alt nivell). Exemples de funcions gràfiques de baix nivell són `lines`, `axis`, `text`, `title`, `abline`. En veurem unes quantes. Podeu consultar la pag. 62 del manual *An Introduction to R*.

5.5.1 Funcions gràfiques d'alt nivell

La funció `plot` fa diferents gràfiques, adaptant-se al tipus de l'objecte, diríem que és genèrica:

- Plot d'un vector

```
dades1 <- c(1,2.4,5,2,3); dades1 # definim un vector
# plot d'un vector numèric: sèrie de valors.
plot(dades1)
plot(dades1, type="b")
# tipus de grafica: "p" punts, "l" linies, # "b" punts i linies,
# "s" esglaonat, "h" barres verticals, etc.
# "n" no dibuixa els punts, prepara les dimensions dels eixos..
# un exemple d'ús de "n", sense punts només etiquetes de text
names(dades1) <- letters[1:5] # posem noms als elements del vector
plot(dades1,type="n") # buit!
text(dades1,labels=names(dades1)) # el text a la posició dels punts
plot(dades1,type="l"); text(dades1,labels=names(dades1)) # tot alhora!
```

- Plot d'una funció

```
plot(sin,-pi,pi) # plot d'una funció (valors entre dos límits)
```


- Plot d'un parell de vectors (diagrama de dispersió)

```
x <- 1:10; y <- x^2          # definim dos vectors d'igual longitud
plot(x,y)                   # fa un diagrama de dispersió
```

- Plot d'un factor (barres)

```
f<-c(rep(c("a","b"),rep(5,2)),rep("c",2)) # vector de caràcters
f<-factor(f); f                          # ara f és un factor
plot(f)                                  # plot d'un factor: diagrama de barres
```

- Plot d'un factor i un vector numèric (boxplot)

```
x <- 1:12
plot(f,x)          # plot d'un factor i d'un vector numèric: boxplot
plot(x~f)          # idem
```

Alguns paràmetres gràfics

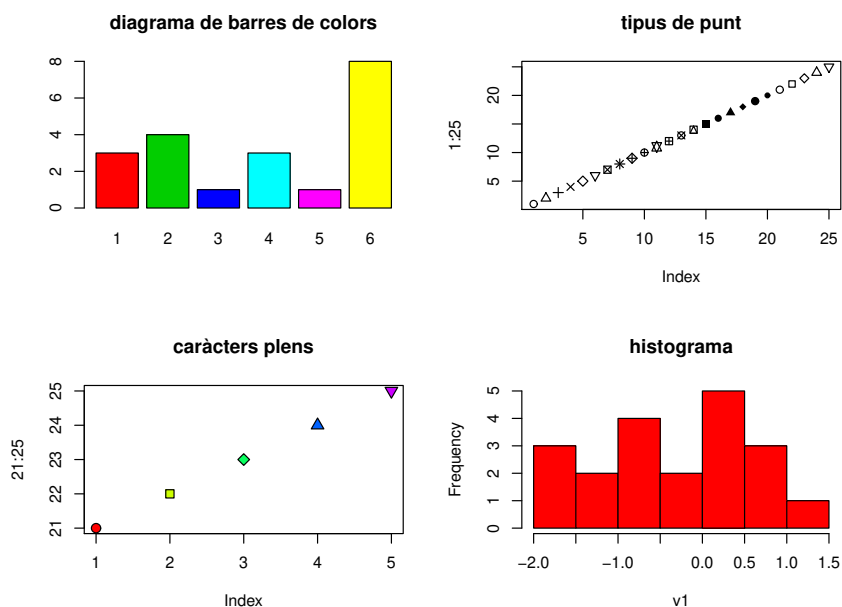
Mostrem alguns arguments de la funció plot (consulteu l'ajuda per a veure'n més)

- Paràmetres xlab, ylab, main, sub, xlim, ylim

```
plot(dades1,type="s",ylab="nom eix y",xlab="x",
      main="Nom de la figura", sub="Altres comentaris")
plot(dades1,xlim=c(0,20),ylim=c(0,10),pch=18)
```

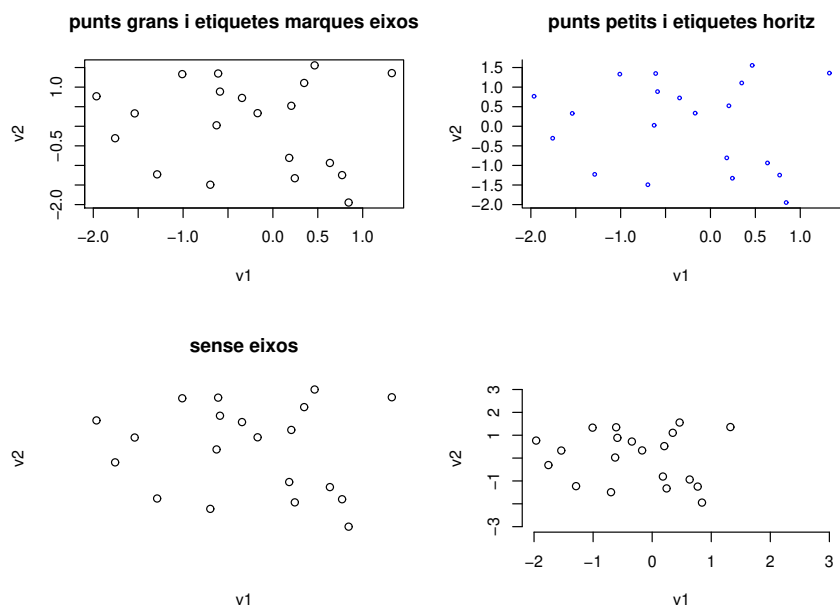
- Paràmetres pch, bg, col, cex

```
x<-sample(1:6,20,replace=T)
v1<-rnorm(20)
v2<-rnorm(20)
par(mfrow=c(2,2))
barplot(table(x),col=2:7, main="diagrama de barres de colors")
plot(1:25,pch=1:25,cex=1,main="tipus de punt")
plot(21:25,pch=21:25,bg=rainbow(5),cex=1.3,main="caràcters plens")
hist(v1,col="red",main="histograma")
par(mfrow=c(1,1))
```



- las, col, bty, axes=FALSE

```
par(mfrow=c(2,2))
plot(v1,v2,las=0,cex=1, main="punts grans i etiquetes marques eixos")
plot(v1,v2,las=1,bty="l", cex=0.5,col="blue",
      main="punts petits i etiquetes horitz")
plot(v1,v2,axes = FALSE,main="sense eixos")
plot(v1,v2,bty="n",xlim=c(-2,3),ylim=c(-3,3))
par(mfrow=c(1,1))
```



5.5.2 Afegir elements a una gràfica oberta

Després de crear una gràfica amb una funció d'alt nivell (plot, per exemple), la gràfica es manté oberta en el dispositiu gràfic de **RStudio** (la pestanya **plots**). Mentre no tornem a executar una altra funció gràfica d'alt nivell podem afegir nous elements a la gràfica oberta, com per exemple punts, línies, eixos, etiquetes, ombres, etc., i crear gràfiques pròpies, no estàndard, que es van afegint als bàsics. Ho veiem amb exemples

```
#primer el dibuix d'alt nivell
plot(0:10,0:10,axes=F,xlab="",ylab="") # punts sense eixos ni nom d'eix
# hi afegim elements
title(main="fals!!!", sub="Van Gogh") # títol i subtítol
points(runif(7,0,10),runif(7,0,10),cex=3,col="red")
#hi afegim punts aleatoris vermells, cex és la mida del punt
text(1,7,"que bonic!",col=rainbow(2)) # afegim text al punt (1,7)
abline(h=2,lty=1) # recta horitzontal a altura 2
# lty és tipus de línia, h vol dir horitzontal
abline(v=3,lty=3) # recta vertical a x=3, discontinua
abline(a=2,b=1,col="red") # recta amb intercept=2 i pendent 1
lines(c(1,3,5),c(10,2,0)) # segments que uneixen punts abscises, ordenades
polygon(c(0,3,4),c(0,1,7),col="yellow") # polígon ple fix
polygon(runif(4,1,7),runif(4,1,7), col="green") # polígon aleatori
legend(8,8,lty=c(1,2),legend=c("a","b")) # afegim llegenda
```

5.5.3 Els paràmetres gràfics globals

Hi ha paràmetres gràfics que es posen com a arguments dins d'una funció d'alt nivell que crea un gràfic (com per exemple `cex` dins de `plot(x,y,cex=0.5)`). En canvi d'altres paràmetres només es poden canviar globalment a través de la funció `par()` (ja hem vist l'exemple `par(mfrow=c(2,2))`).

La funció `par` també és fonamental per adaptar tot un conjunt de gràfiques al gust de l'usuari.

Si executem la funció `par()` ens surt la llista de tots els paràmetres gràfics globals i quin valor tenen en aquest moment. Per exemple

```
par()
...
$bg
[1] "white"
$btty
[1] "o"
...
$mar
[1] 5.1 4.1 4.1 2.1
```

Busqueu ajuda d'aquests paràmetres i d'altres.

Vegem un exemple. Abans de modificar els paràmetres, guardarem els paràmetres vells en una variable, i així podrem recuperar-los.

```
x<-rnorm(10)
```

```

y<-rnorm(10)
par()$mar      # quant val el paràmetre mar ara mateix?
[1] 5.1 4.1 4.1 2.1 # són els marges externs del gràfic
par()$bg       # quant val el paràmetre bf?
[1] "white"     # és el color de fons del gràfic

param.vells<-par() # guardem els paràmetres actuals abans de canviar-los
par(bg="lightgray",mar=c(2.5, 2.5, 2.5, 0.7)) # canviem bg i mar
# A partir d'ara tots els gràfics tindran fons de color gris
plot(x, y, type="n", xlab="", ylab="", xlim=c(-2, 2),
      ylim=c(-2, 2), xaxt="n", yaxt="n")
rect(-3,-3,3,3,col="cornsilk")
points(x,y,pch=10,col="red",cex=2)
axis(side=1,c(-2,0,2),tcl=-0.2,labels=FALSE)
axis(side=2, -1:1, tcl=-0.2, labels=FALSE)
title("personalitzar un gràfic",font.main=4,adj=1,cex.main=1)
mtext("Deu números a l'atzar", side=1, line=1, at=1, cex=0.9, font=3)
mtext("Uns altres deu números", side=2,line=1, at=0.5, cex=0.9, font=3)
mtext(c(-2, 0, 2), side=1, las=1, at=c(-2, 0, 2), line=0.3,
      col="blue", cex=0.9)
mtext(-1:1, side=2, las=1, at=-1:1, line=0.2, col="blue", cex=0.9)
## els costats del gràfic són 1 el de baix, 2 el de l'esquerra,
# 3 el de dalt i 4 el de la dreta.

par(param.vells) # tornem als valors vells dels paràmetres
par()$bg # veiem que torna a valdre "white"

```

Pràctica:

• Passem la matriu `DESPESA` de la pràctica 5.1. a format `data.frame` per treballar-la gràficament. Pots recuperar-la de l'arxiu `DESPESA.RData` que trobàs al Moodle.

Fes un gràfic amb punts i línies de la despesa mitjana mensual (arrodonint el resultat a dos decimals) que tingui les característiques següents:

Títol: Despesa mensual dels alumnes

L'etiqueta de l'eix x ha de ser "Mesos" i el de l'eix y "Despesa".

A l'eix x han d'aparèixer el nom dels 12 mesos de l'any.

L'eix de les y ha d'anar de 0 a 150 amb etiquetes cada 50.

Que cada punt estigui representat per una creu

Que cada punt tingui l'etiqueta del valor, amb mida de lletra al 70% de la que té per defecte, en vermell i negreta, i posicionada a sobre dels punts.

Afegir tres línies horitzontals: una línia contínua blava i amb gruix 2 que sigui la mitjana total i dues línies grises discontinües que representin la mitjana \pm desviació típica.

• Carrega a R l'arxiu de dades `mundo.RData` d'un estudi fet l'any 95 de diverses variables als països del món. Trobaràs el fitxer al Moodle (pràctiques → fitxers pràctica 5).

(a) Fes un boxplot de la variable `espvida` de l'arxiu de dades `mundo` (esperança de vida femenina).

(b) Afegeix al boxplot anterior línies horitzontals vermelles discontinües als quartils (Q_1 , Q_2 i Q_3).

(c) Calcula el rang interquartílic RI i fes dues línies horitzontals discontinües en verd als números $Q_1 - 1.5 * RI$ i $Q_3 + 1.5 * RI$.

5.5.4 Guardar gràfiques directament en un fitxer (sense utilitzar menús)

Cada gràfica es pot guardar en format pdf, entre altres (png, eps, etc.) `plot(c(0,1,2),c(1,1,1))`

```
pdf("nomquevull.pdf")      # obre el buidat en pdf
plot(runif(10,0,10),runif(10,0,10))
plot(c(0,1,2),c(1,1,1))
# altres gràfiques si es vol
dev.off()                  # acaba el buidat en pdf
                             # el fitxer té dos gràfics
```

En l'exemple anterior obtenim un fitxer pdf amb dues pàgines. No passa el mateix si el tipus de fitxer que obrim és PNG. Aleshores només es guarda el darrer gràfic.

```
png("nomquevull.png")      # obre el buidat en pdf
plot(runif(10,0,10),runif(10,0,10))
hist(runif(20,0,10),col="red")
dev.off()                  # el fitxer té un sol gràfic
```

Comproveu que s'han creat els fitxers pdf i png

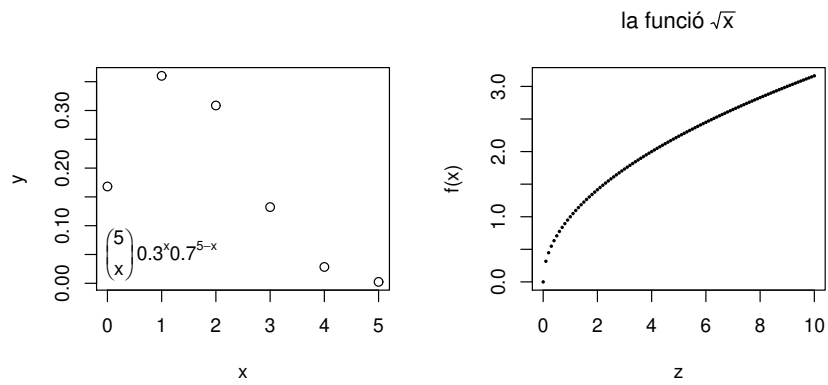
Pràctica: • Carrega el data frame `iris` del paquet base de R. Fes dues gràfiques apropiades per a les variables `Sepal.length` i dues gràfiques apropiades per a la variable `Species`, totes juntes en dues files per dues columnes. Guarde-les en un fitxer `iris.pdf`.

5.5.5 Afegir fórmules a una gràfica

```
x<- 0:5
y <- choose(5,x)*0.3^x*0.7^(5-x); y

#ara posarem l'expressió al títol
plot(x,y)
text(1,0.05,expression(paste(bgroup("(",atop(5,x),")"),0.3^x,0.7^{5-x})))
plot(x,y)
title(main=expression(paste(bgroup("(",atop(5,x),")"),
                                0.3^x,0.7^{5-x})))

x<-seq(0,10,by=0.1)
plot(x,sqrt(x),pch=16,ylab="f(x)")
title(main=expression(paste("la funció ",sqrt(x))))
```



Consulteu ajudes de `help(plotmath)` `example(plotmath)` `demo(plotmath)`