Sessió 7

Dades multivariants 1: relació entre una variable numèrica i un factor

Veurem com s'explora numèricament i gràficament la relació d'una variable numèrica i un factor o, el que és el mateix, com es veu l'"efecte" del factor sobre la variable numèrica. Fem una anàlisi exploratòria, per tenir els resultats de la variable numèrica "estratificats" pels nivells del factor.

7.1 Informació i descripció ràpida del fitxer

- Comencem per carregar i demanar informació general del fitxer.

```
data(mtcars)    # carreguem el data frame al nostre espai de treball
                                                #(environment)
?mtcars  # demanem informació sobre l'arxiu, les variables que té, etc
dim(mtcars)    #nombre de casos i de variables
head(mtcars)   #capçalera i les primeres files
sapply(mtcars,class)  # una manera ràpida de saber les classes de les
           #  variables introduïda a la pràctica anterior

summary(mtcars)
```

- Les variables `cyl, vs, am, gear, carb`, convé definir-les com a factors. Podeu convertir-les amb `as.factor`. Per canviar a factor la variable `cyl` de l'arxiu de dades `mtcars` fem el següent:

```
class(mtcars$cyl)
[1] numeric      # comprovem que és un vector numèric
mtcars$cyl<-as.factor(mtcars$cyl)    # el convertim a factor
class(mtcars$cyl)
[1] factor      # ara és un factor.
# Feu el mateix amb les variables vs, am, gear, carb.
# o bé (optatiu)  de manera més ràpida podem convertir les cinc variables
  #  de cop. Volem que la sortida sigui una llista, per això cal lapply
mtcars[,c("cyl", "vs", "am", "gear", "carb")] <-
     lapply(mtcars[,c("cyl", "vs", "am", "gear", "carb")],as.factor)

# comprovem com ha canviat la sortida de summary:
summary(mtcars)
```

- Adjuntem les variables de l'arxiu de dades per treballar-hi més còmodament

```
attach(mtcars)    # a partir d'ara ja no cal escriure, per exemple,
                  # mtcars$cyl sinó només cyl
```

- Fem un resum ràpid de les variables numèriques del fitxer

```
summary(data.frame(mpg,disp,hp,drat,wt,qsec))
library(psych)    # amb aquesta llibreria tenim un resum més complet
describe(data.frame(mpg,disp,hp,drat,wt,qsec))
```

- Però també podem fer el nostre resum personalitzat definint abans una funció adequada

```
resum<-function(x){
  tipus<-class(x)
  valors.perduts<-sum(is.na(x))
  quartils<-quantile(x)
  minim<-quartils[1]
  maxim<-quartils[5]
  q1<-quartils[2]
  Md<-quartils[3]
  q3<-quartils[4]
  rang<-max(x)-min(x)
  R.I<- q3-q1
  mitjana<-mean(x)
  desv<-sqrt(mean((x-mean(x))^2))
  desv.med<-mean(abs(x-median(x)))
  CV<-desv/mitjana
  alpha.sim<-((q3-Md)-(Md-q1))/R.I
  mu4<-mean((x-mean(x))^4)
  Kurtosis<-mu4/desv^4
  llista<-list(min=minim,Q1=q1,mediana=Md,mitjana=mitjana,
       Q3=q3,max=maxim,rang=rang,R.I=R.I,desv=desv,desv.med=desv.med,
       CV=CV,alpha=alpha.sim,curtosi=Kurtosis,NAs=valors.perduts)
  lapply(llista,round,2)
}
# apliquem la funció resum a les variables numèriques de mtcars
mtcars.num<-data.frame(mpg,disp,hp,drat,wt,qsec)
t(sapply(mtcars.num,resum))
```

	min	Q1	Md	mitj	Q3	max	rang	R.I	desv	CV	alpha	curtosi	NAs
mpg	10.4	15.4	19.2	20.1	22.8	33.9	23.5	7.4	5.9	0.3	0	2.8	0
disp	71.1	120.8	196.3	230.7	326	472	400.9	205.2	122	0.5	0.3	1.9	0
hp	52	96.5	123	146.7	180	335	283	83.5	67.5	0.5	0.4	3.1	0
drat	2.8	3.1	3.7	3.6	3.9	4.9	2.2	0.8	0.5	0.1	-0.5	2.4	0
wt	1.5	2.6	3.3	3.2	3.6	5.4	3.9	1	1	0.3	-0.4	3.2	0
qsec	14.5	16.9	17.7	17.8	18.9	22.9	8.4	2	1.8	0.1	0.2	3.6	0

Pràctica: Treballarem amb el fitxer `aux_Cardio_Train.csv` que trobaràs al Moodle. Aquest fitxer conté informació objectiva del pacient, els resultats de proves mèdiques i informació subjectiva donada pel pacient.
A l'adreça `https://www.kaggle.com/sulianova/cardiovascular-disease-dataset` trobaràs la informació de les variables que necessites per treballar.
a) Importa el fitxer i anomena'l `Cardio.Train`.
b) Quants casos i quantes variables conté el fitxer?
c) De quin tipus són les variables? Utilitza la funció `sapply` per poder executar la instrucció per totes les variables alhora.
d) Hi ha alguna variable que caldria convertir a factor? Fes-ho utilitzant la funció `lapply`.
e) Crea una nova variable al data.frame que sigui l'edat en anys (la variable `age` està en dies) i anomena-la `age.years`. Arrodoneix al valor enter de l'edat complerta (utilitza la funció `trunc`).
f) Fes un resum ràpid de les variables del data.frame amb les funcions `summary` i `describe`. Quina creus que t'aporta millor informació?

7.2 Descriptius de la variable numèrica, estratificant pel factor

En aquesta secció i en la propera estudiarem la variable numèrica d'interès és `mpg` (milles per galó), és a dir consum. En aquest cas, més milles per galó significa menys consum. Considerarem el factor `cyl`. Afecta el consum? Com?

Podem calcular la mitjana, la mediana o els quartils de la variable numèrica `mpg`, estratificant per `cyl`, amb la funció **by**

```
by(mpg,cyl,mean)
by(mpg,cyl,mean,na.rm=T) # si hi haguessin valors perduts
by(mpg,cyl,mean,na.rm=T,trim=0.05) # per treure NA i el 10% extrem
by(mpg,cyl,median)
by(mpg,cyl,quantile)
by(mpg,cyl,summary)
# podem treure més informació de cop amb una funció que torni una llista:
by(mpg,cyl,function(x) list(mean=mean(x),sd=sd(x),median=median(x)))
```

Podem fer-ho amb el paquet `psych`

```
library(psych)
describeBy(mpg,cyl)
```

També podíem haver seleccionat els casos amb `cyl` igual a 4 i calcular els descriptius, després seleccionar els casos amb `cyl` igual a 6, ...

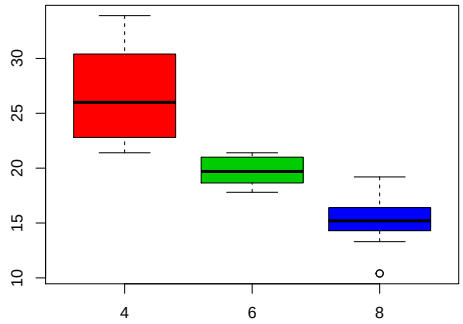
```
summary(subset(mtcars,cyl==4,select=c("mpg","disp","hp","drat","wt","qsec")))
summary(subset(mtcars,cyl==6,select=c(1,3,4,5,6,7)))
summary(subset(mtcars,cyl==8,select=c(1,3,4,5,6,7)))
```

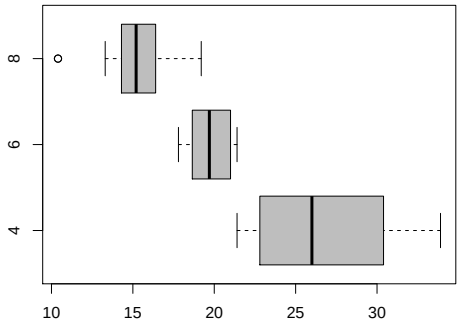
Pràctica: Seguim treballant amb el data.frame `Cardio.Train`
a) Crea una nova variable al data.frame a partir de `age.years` que sigui un factor amb 4 categories que recullin una freqüència semblant. Anomena-la `age.cat`.
b) Descriu numèricament les variables alçada i pes en funció dels factors `sexe` i la nova variable `age.cat`. Quin dels factors influeix més en l'alçada i el pes d'un individu, el sexe o l'edat?

7.3 Representació gràfica: boxplot i plotmeans

- El gràfic que millor ens ajuda a entendre com el factor `cyl` afecta a la variable `consum` és el **boxplot**:

```
boxplot(mpg~cyl)
boxplot(mpg~cyl,col=2:4) # amb diferents colors
boxplot(mpg~cyl, col="gray",horizontal = TRUE) # grisos i horitzontals
```



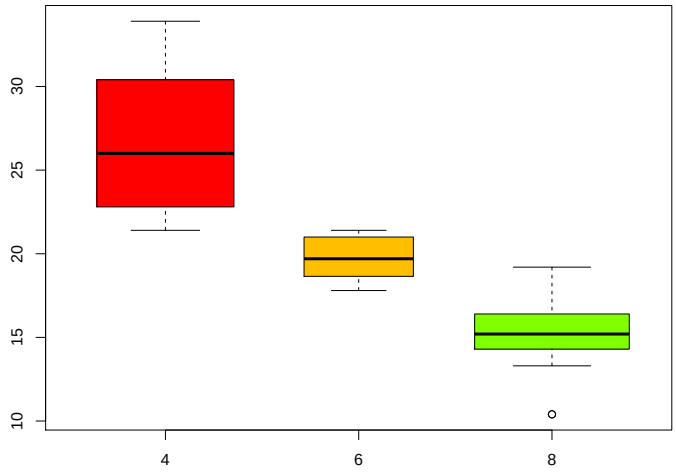


Fixem-nos que ara podem interpretar els boxplots que hem obtingut, i veure que són coherents amb els valors dels descriptius per grups. Per exemple veiem que la variable `mpg` és més baixa en els cotxes de 8 cilindres que en els de 6, i que és més baixa en els de 6 que en els de 4.

- En cas que el nombre de casos dels grups siguin molt diferents entre sí, podem fer boxplots d'amplades diferents. De manera automàtica, amb l'opció `varwidth=TRUE`

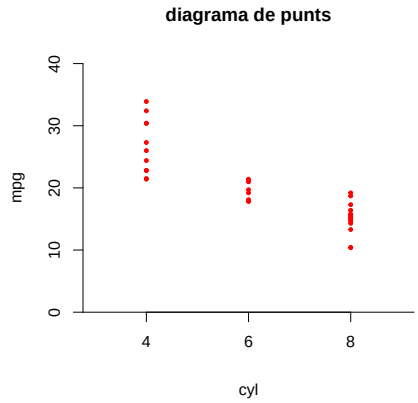
```
c(sum(cyl==4),sum(cyl==6),sum(cyl==8))
[1] 11  7 14     # veiem que els grups són desiguals
boxplot(mpg~cyl,col=rainbow(8),varwidth=TRUE)
```

El gràfic que obtenim té els diagrames de caixa de diferents amplades.



- Quan el nombre de casos és baix, podem fer un gràfic de punts, passant la variable categòrica a numèrica:

```
plot(as.numeric(cyl),mpg,cex=0.8,pch=20,axes=F,xlim=c(0.5,3.5),
     xlab="cyl",ylim=c(0,40),col=2,main="diagrama de punts")
axis(side=2)
axis(side=1,at=c(1,2,3),pos=0,labels=c(4,6,8))
abline(h=0)
```



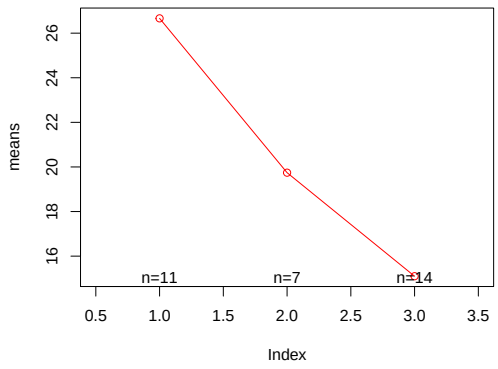
D'aquesta manera representem també si hi ha diferències de mida dels grups definits pel factor.

- Un altre gràfic que ens ajuda a comparar la variable numèrica en els diferents grups definits pel factor és **plotmeans**

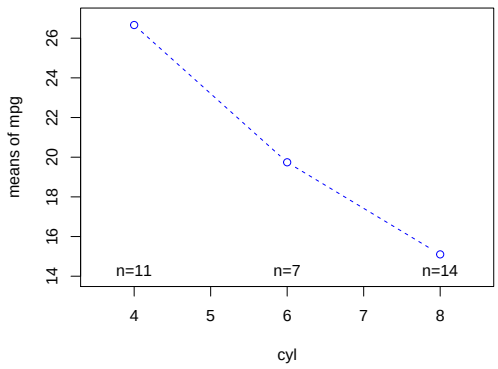
```
library(gplots)
plotmeans(mpg~cyl,mean.labels=T,digits=2,pch=1,col="red",bars=F,connect=T)
```

El mateix gràfic el podíem haver fet sense cap paquet amb el codi següent que construeix un gràfic a la carta

```
mitj1<-by(mpg,cyl,mean)[[1]] # mitjana de mpg a cada grup
mitj2<-by(mpg,cyl,mean)[[2]]
mitj3<-by(mpg,cyl,mean)[[3]]
mitjanes<-c(mitj1,mitj2,mitj3)
nivells<-as.numeric(levels(cyl))
plot(nivells,mitjanes,type="b", xlim=c(3.5,8.5),ylim=c(14,27),
           xlab="cyl",ylab="means of mpg",col="blue",lty=2)
# ara afegirem les etiquetes n=11, n=7,...
n1<-by(mpg,cyl,length)[[1]]  # calculem el nombre de casos a cada grup
n2<-by(mpg,cyl,length)[[2]]
n3<-by(mpg,cyl,length)[[3]]
etiquetes<-paste("n=",c(n1,n2,n3),sep="")
text(nivells,rep(14.3,3),etiquetes)
```



Gràfic amb plotmeans



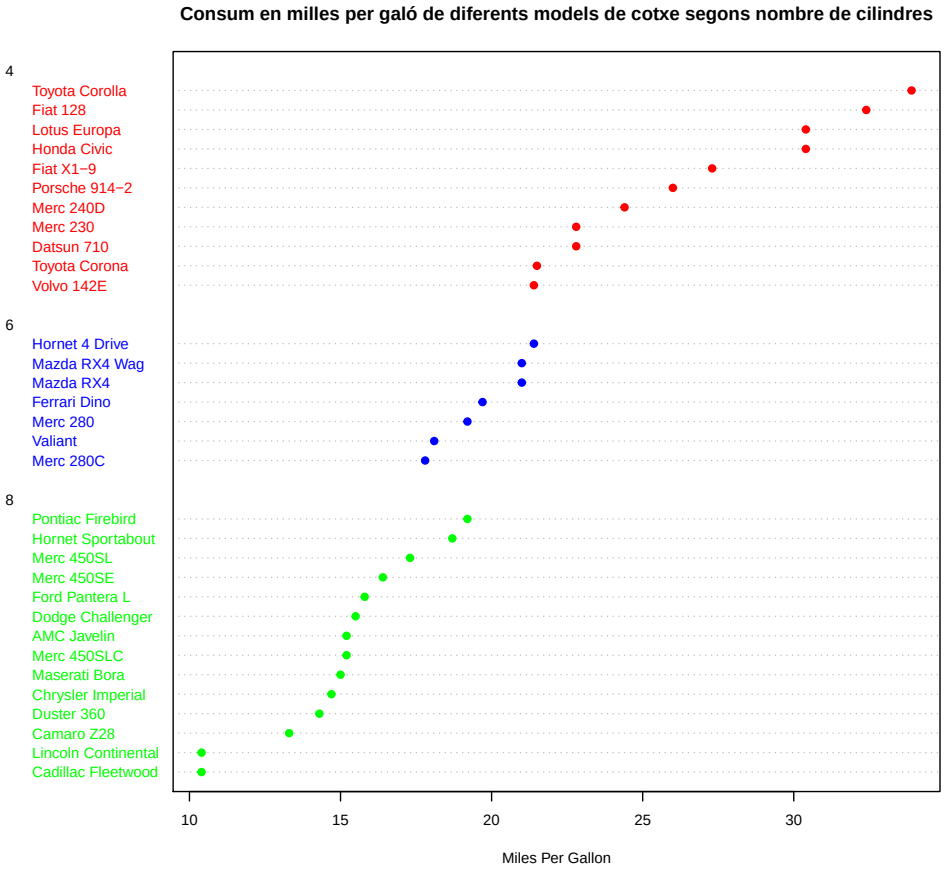
gràfic "a la carta"

Pràctica: Fes un boxplot d'amplades diferents i un `plotmeans` per a la variable pes en funció dels hàbits de la persona en fumar, beure alcohol i fer activitat física. Quin factor influeix més en el pes d'un individu?

7.4 Altres gràfics

Diagrama de punts amb colors per categories

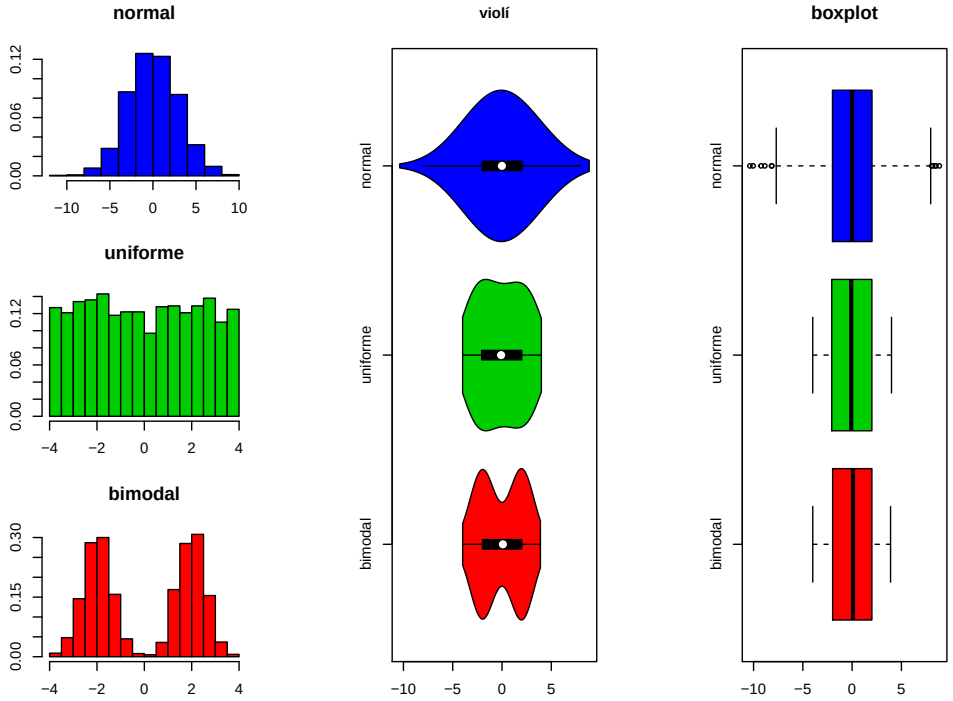
```
x<-mtcars[order(mtcars$mpg),]
x$color[x$cyl==4]<-"red"
x$color[x$cyl==6]<-"blue"
x$color[x$cyl==8]<-"green"
head(x)

dotchart(x$mpg,labels=row.names(x),cex=.7,groups=x$cyl,
     gcolor="black",color=x$color,pch=19,
     main = "Consum en milles per galó de diferents models de cotxe segons
     nombre de cilindres", xlab = "Miles Per Gallon")
```



Gràfics de violí

Els gràfics de violí són boxplots modificats segons la forma de la distribució. També permeten representar diferents variables.

Per entendre els gràfics de violí hem de comparar-los amb els histogrames. Veurem un exemple de comparació de tres variables generades aleatòriament que tenen histogrames de formes molt diferents.



Fixem-nos que a dins de cada violí hi ha el boxplot. En aquest cas, però, els dos primers boxplots són molt semblants i donen poca informació.

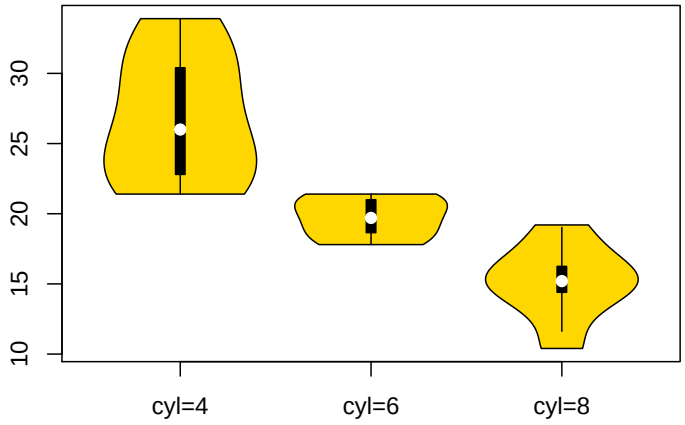
El codi que genera els gràfics és el següent:

```
# generem les variables
mu<-2
si<-0.6
set.seed(100)
bimodal<-c(rnorm(1000,-mu,si),rnorm(1000,mu,si))
uniforme<-runif(2000,-4,4)
normal<-rnorm(2000,0,3)
# el layout és una manera més sofisticada que par(mfrow=...) per
# organitzar els gràfics:
# per posar-los en una fila de tres columnes i una fila de 1 columna:
layout(matrix(c(1,2,3,4,4,4,5,5,5),nrow=3))
#  1    4    5   el primer gràfic es posarà al rectangle 1, el segon al 2, ...
#  2    4    5   el quart ocupant l'espai dels 4
#  3    4    5   el cinquè l'espai dels 5
par(mar=c(2,4.1,4.1,4.1))
hist(normal,col=4,main="normal",xlab="",probability=T,ylab="")
hist(uniforme,col=3,main="uniforme",xlab="",probability=T,ylab="")
hist(bimodal,col=2,main="bimodal",xlab="",probability = T,ylab="")
vioplot(bimodal,uniforme,normal,col=2:4,horizontal=T,main="violí",
   names=c("bimodal","uniforme","normal"))
boxplot(bimodal,uniforme,normal,horizontal=T,col=2:4,main="boxplot",
```

```
       names=c("bimodal","uniforme","normal") )
```

Les variables que comparem amb el gràfic de violí també poden ser la mateixa variable en els diferents grups determinats per un factor. Per exemple, la variable `mpg` en els grups de 4, de 6 i de 8 cilindres.

```
library(vioplot)
x1 <- mtcars$mpg[mtcars$cyl==4]
x2 <- mtcars$mpg[mtcars$cyl==6]
x3 <- mtcars$mpg[mtcars$cyl==8]
vioplot(x1, x2, x3, names=c("cyl=4", "cyl=6", "cyl=8"), col="gold")
```



Boxplots amb punts

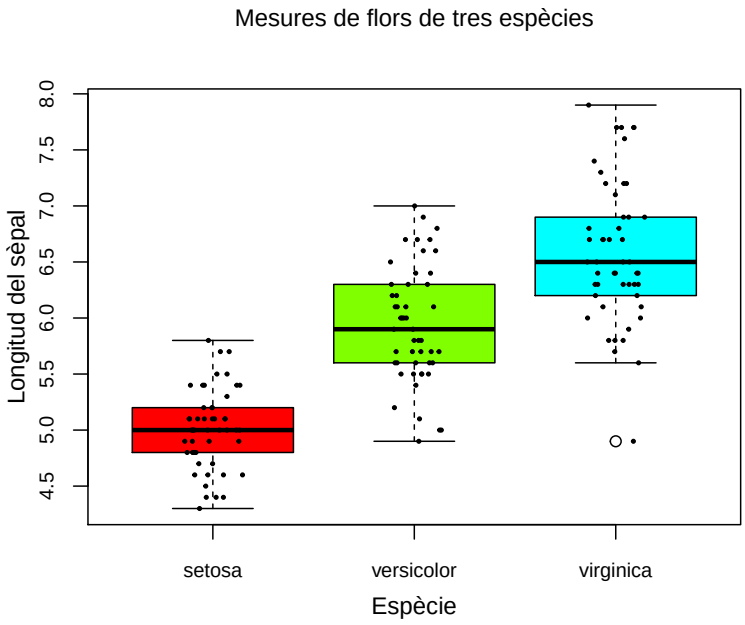
En mostres no molt grans podem afegir als boxplots els punts individualitzats. Perquè no quedin tots en la mateixa línia, s'utilitza la funció `jitter` de R, que modifica a l'atzar un número amb una amplitud màxima. Amb aquest gràfic es visualitza la distribució que queda amagada al boxplot.

```
jitter(rep(1,5),amount=0.2)
[1] 0.9791345 0.9863539 0.8126787 0.8977229 0.8664743
```

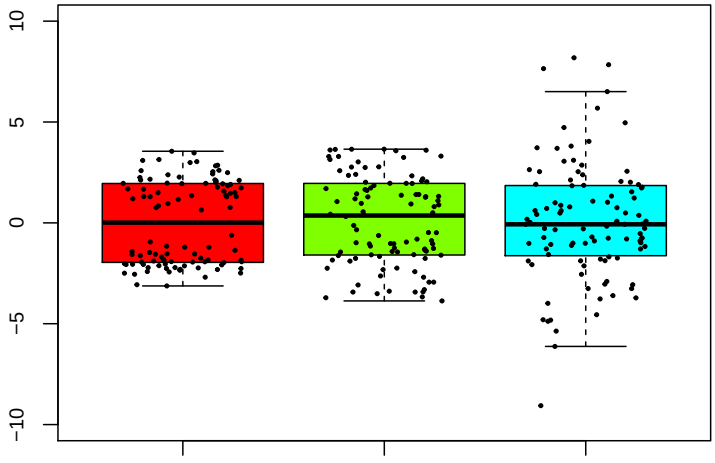
En l'exemple següent s'ha representat la variable `Sepal.Length` en les tres `Species` del data frame `iris`

```
boxplot(iris$Sepal.Length~iris$Species,col=rainbow(4),xlab="",ylab="",cex.axis=0.8)
n<-nrow(iris)
frequencies<-as.vector(table(iris$Species))
desvia1<-jitter(rep(1,frequencies[1]),amount=0.15)
points(desvia1,iris$Sepal.Length[iris$Species=="setosa"],cex=0.5,pch=20)
desvia2<-jitter(rep(2,frequencies[2]),amount=0.15)
points(desvia2,iris$Sepal.Length[iris$Species=="versicolor"],cex=0.5,pch=20)
desvia3<-jitter(rep(3,frequencies[3]),amount=0.15)
points(desvia3,iris$Sepal.Length[iris$Species=="virginica"],cex=0.5,pch=20)
mtext(1,line=2.3,text="Espècie")
mtext(2,line=2,text="Longitud del sèpal")
```

```
mtext(3,line=2,text="Mesures de flors de tres espècies")
```



Com en els tres gràfics de violí corresponents a una distribució bimodal, una uniforme i una normal, podem fer un boxplot amb punts. En aquest gràfic s'han utitlizat mostres de 100 valors només i dins del boxplot les opcions `outline=FALSE` (per no dibuixar els otuliers en un primer moment) i `ylim=c(-10,10)` (per tal de fer lloc per a l'outlier de la normal que surt quan dibuixem els punts).



Pràctica: Crea un gràfic de punts dels 50 primers casos de les variables de pressió arterial (`ap_hi` i `ap_lo`), separant en dos grups segons si han patit una malaltia cardiovascular o no.
Fes el mateix amb un gràfic de violí agafant tots els casos. Interpreta els resultats.

7.5 Annex: Generar informes amb Rmarkdown.

Els informes de R o documents dinàmics de R són fitxers word, pdf o html que inclouen els resultats d'anàlisi de dades (taules, gràfics, etc) produïts amb R i alhora comentaris i interpretacions d'aquests resultats. Tot això de manera dinàmica, així que si canviem les dades només canviaran els resultats, però les instruccions de R i els comentaris poden ser els mateixos. Per generar aquest tipus d'informes necessitem el paquet **rmarkdown** i el paquet `knitr`.

Hi ha tres tipus de fixers de R que permeten generar informes:

- **Script normal de R** (`nom.R`): *Quan al script no hi ha cap error*, es genera l'informe com es mostra a la Figura 1 o bé amb la icona següent:

  A l'informe generat apareixen en requadres grisos els codis de R i les sortides, així com els gràfics. Els comentaris que posem radera d'un coxinet surten en un altre color.

  Si no posem res a la capçalera, quan generem l'informe ens ofereix si volem que sigui tipus Word, PDF o HTML. Perquè funcioni el PDF cal tenir instal·lat el LaTeX. A vegades només surt en HTML, i perquè la sortida sigui en Word, que és el més recomanable *si després volem acabar de pulir el fitxer editant-lo*, convé posar al començament del script les línies següents:

```
#' ---
#' title: "Informe de la pràctica 6"
#' output: word_document
#' author: "Rosa Camps"
#' ---
```

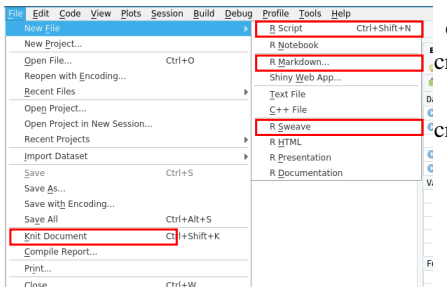
- **Script tipus Rmarkdown** (`nom.Rmd`): Conté més format. Per crear-lo, vegeu Figura 1). La diferència és que els codis de R els escrivim dins dels que s'anomenen **chunks**. Per instertar un nou chunk des de Rstudio tenim una icona que apareix quan estem treballant amb un fitxer tipus Rmarkdown.

  Fora dels chunks hi posem text explicatiu. Podem posar negreta (entre $**$), títols (#), llistes (amb $*$), etc.

- **Fitxer tipus Sweave** (`nom.Rnw`): És un fitxer de LaTeX que inclou, com els de rmarkdown, chunks amb codi R. Per la resta, són fitxers de LaTeX. Per utilitzar aquest tipus de fitxer cal tenir instal·lat el LaTeX a l'ordinador i tenir un mínim de coneixements d'aquest llenguatge de composició de textos matemàtics.

Per crear un fitxer d'un dels tres tipus i generar l'informe corresponent podem utilitzar menús de RStudio:

generar un informe bàsic del script (en word, htm o pdf)
Si l'arxiu és .Rmd funciona igual.
Es pot generar també amb "CONTROL+Shift+K" o amb "Compile report"



crear un script (.R)
crear un document Rmarkdown(.Rmd)
crear un document LaTeX (.Rnw)

Figura 1: Creació des de RStudio de diferents tipus d'arxius R i com generar informes