

## Tema 3: Descripción numérica y gráfica de datos. Ejercicios resueltos.

- Supongamos que disponemos de un conjunto de estaturas de los individuos de tres ciudades. Los datos son:

**Elche** : 1.65, 1.55, 1.78

**Alicante** : 1.81, 1.90, 1.5, 1.52

**Albacete** : 1.95, 1.6, 1.72, 1.8

Definir y construir dos variables, estatura y ciudad, de forma que ciudad sea un factor de clasificación. Obtener una descriptiva de las estaturas, tanto globalmente como por ciudades. Investigar el comando `by` e utilizarlo para calcular la media de la altura en cada ciudad.

Solución:

```
estatura<- c(1.65,1.55,1.78,1.81,1.90,1.5,1.52,1.95,1.6,1.72,1.8)
ciudad<-as.factor( rep(c(1,2,3), c(3,4,4)) )
levels(ciudad)<-c("Elche","Alicante","Albacete")
mean(estatura); by(estatura, ciudad, mean)
mean(estatura[ciudad=="Elche"]), etc.
```

- En el banco de datos `orientaciones.dat` tenemos los datos codificados de la orientación de 1047 árboles en los que se analizó la presencia de procesionaria. Leer los datos en un vector y averiguar cuantos hay de cada tipo. Representar dichos datos.

Teniendo en cuenta que en realidad los valores del 1 al 8 representan los 8 puntos cardinales (E, SE, S, SW, W, NW, N, NE) tratar de buscar una mejor representación de ellos. Utilizar para ello la función `radial.plot()` de la librería `plotrix`. Guardar el gráfico resultante en un pdf.

```
Ayuda: orient<-scan("orientaciones.dat"); table(orient)
freqs<-c(75, 119, 151, 289, 152, 76, 89, 96)
nom<-c("E", "SE", "S", "SW", "W", "NW", "N", "NE")
pdf("desc-orientaciones.pdf", paper="special", width=13, height=7)
radial.plot(freqs,labels=nom,rp.type="p",radial.lim=c(0,300),
            poly.col="blue", show.grid=FALSE,
            clockwise=TRUE, main="Orientacion arbol")
dev.off()
```

- La librería `lattice` contiene una serie de funciones que nos permiten visualizar datos multivariados de gran utilidad para la exploración de relaciones e interacciones entre variables. En la línea de la función `coplot` pero con mucha más flexibilidad, la idea principal detrás de `lattice` es hacer gráficas condicionales múltiples: una gráfica bivariada se divide en varias gráficas con respecto a los valores de una tercera variable.

- a) Investigar en la ayuda la función `matplot()`, que produce varios dibujos  $x$ - $y$ , en un mismo gráfico, con las columnas de una matriz. Investigar también cómo con la función `matpoints()` podemos dibujar puntos y cómo con `matlines()` podemos dibujar líneas.

Utilizar estas funciones para dibujar una curva que muestre el comportamiento de `sin()` sobre el rango  $-\pi$  a  $\pi$ . Dibujar `sin()` y `cos()` juntos en el mismo rango. Distinguir las gráficas por tipo de línea y color mostrando una leyenda a tal efecto.

**Ayuda:**

```
x<- seq(-pi,pi,length=100)
aux <- matrix(c(sin(x),cos(x)),ncol=2)
matplot(x, aux , type="l", col=c(1,2), ylab = "seno, coseno",
        ylim=c(-2, 2))
legend(-pi,2, c("sin(x)", "cos(x)"), lty=c(1,1), col=c(1,2))
```

- b) Investigar las funciones `bwplot()`, `xyplot()` y `splom()` y utilizarlas para explicar el banco de datos `iris`.

**Ayuda:** ver apartado 4.6 del libro R para debutantes.

```
bwplot(Sepal.Length+Sepal.Width+Petal.Length+Petal.Width~Species,
       outer=TRUE)
```

4. Aproximación asintótica a la normalidad de la distribución binomial. Vamos a tratar de demostrar dicha aproximación considerando diversos casos de una distribución  $Bi(n, p)$ :

- a)  $p = 0.1$ ;  $n = 5, 10, 30, 50, 100$   
 b)  $p = 0.3$ ;  $n = 5, 10, 30, 50, 100$   
 c)  $p = 0.5$ ;  $n = 5, 10, 30, 50, 100$   
 d)  $p = 0.7$ ;  $n = 5, 10, 30, 50, 100$   
 e)  $p = 0.9$ ;  $n = 5, 10, 30, 50, 100$

Simular 100 datos en cada uno de los supuestos y representarlos con histogramas. Superponer, en cada uno de los gráficos, una curva que represente la función de densidad normal de la correspondiente aproximación asintótica. Comparar y comentar los resultados. ¿Cuándo se aprecia la convergencia asintótica?

**Ayuda:** `par(mfrow=c(5,5))`

```
for(p in c(0.1,0.3,0.5,0.7,0.9)) {
  for(n in c(5,10,30,50,100)){
    x<-rbinom(100,n,p)
    y<-seq(0,100,by=.2)
    hist(x,prob=T)
    lines(y, dnorm(y,mean=n*p,sd= sqrt(n*p*(1-p)))) } }
```