Curso de R y estadística básica

[Felipe de J. Muñoz González]

fmunoz@lcg.unam.mx

Introducción
Descargar Presentación

Programación

Operadores lógicos y comparativos

Relaciónes

- **>** Mayor que
- **<** Menor que
- **<=** Menor o igual que
- **>=** Mayor o igual que
- **==** Igual
- **!=** Diferente

Lógicos

- **!x** Negación (no x)
- **x & z** Conjución (x y z)
- **x && z** Conjunción(*)
- **x|y** Disyuncion
- **==** Disyuncion(*)
- **xor(x, y)** O exclusivo (∗∗)
- **identical()** Comparar dos objetos

(∗) Si se escriben dos símbolos repetidos, estos tienen el mismo significado que si apareciese uno, la diferencia consiste en que se evalúa primero la parte de la izquierda y, si ya se sabe el resultado no se sigue evaluando, por lo que pueden ser mas rapidos y eliminar errores

(∗∗) Da como valor verdadero si uno y sólo un argumento es válido.

Programación

Operadores lógicos y comparativos

```
> x<-10; x # Asignamos a x el valor 10
> x<5 # Le preguntamos si x es menor que 5
> x>=5 # Le preguntamos si x es mayor o igual que 5
> x==5 # Le preguntamos si x vale 5
> x!=5 # Le preguntamos si x es distinto de 5
```

```
> y<-1:3; z<-3:1 # Creamos dos vectores
> identical(y,z) # Le preguntamos si son iguales
> y==z # Vemos los elementos que coinciden
> x<-1:5 # Renombramos x e y
> y<-c(2,4,3,6,5)
```

```
> x==y
> x!=y
> x[x==y]
> x[x!=y]
```

Probabilidad

Probabilidad

Espacio muestral

Espacio muestral (denotado S) consiste en el conjunto de todos los posibles resultados de un experimento aleatorio.

Moneda

```
> S<-data.frame(pos=c("H","T"))
```

Espacio Muestral de una moneda que se lanza 3 veces

```
> expand.grid(t(S),t(S),t(S))
```

Dado

```
> S<-data.frame(pos=c(1:6))
```

```
>  sample(x=c("H","T"), size=5, replace=T)
```

Espacio Muestral de un dado

```
>  expand.grid(t(S),t(S),t(S))
```

Probabilidad

Espacio muestral

Espacio Muestral de Cartas Inglesas

```
> palos<-c("D","P","T","C")

> numeros<-c(1:10,"J","Q","R")

> cartas<-as.vector(outer(numeros, palos, paste, sep=""))

> cartas<-sample(cartas)
```

Espacio Muestral de Muestreo de urnas

```
> urna=c("roja","azul","amarilla","violeta","negra","blanca")
```

```
> urnsample<-sample(urna,size=20, replace=T)

> table(urnsample)
```

Probabilidad

Subsets de datos

%in% #busqueda por elementos

```
> x <- 1:10
> y <- 8:12
> y %in% x
> y[y %in% x]
```

isin

```
> isin(x,y) #todo el vector
```

all

```
> x <- 1:10
> y <- c(3, 3, 7)
> unique(c(y %in% x))
```

¿Por que isin y all tienen esos resultados?

Probabilidad

Union, Interseccion y diferencia

Elementos que existen en el Evento A, en el Evento B o en ambos union(A,B)

```
> S <- expand.grid(numeros, palos)
> colnames(S)<-c("numero", "palo")
> A <- subset(S, palo == "C")
> B <- subset(S, numero %in% as.character(7:9))

> union(apply(A,1,paste, collapse=""), apply(B,1,paste, colla
```

Elementos que existen en el Evento A y en el Evento B intersect(apply(A,1,paste, collapse=""), apply(B,1,paste, collapse=""))

```
>  intersect(apply(A,1,paste, collapse=""), apply(B,1,paste, c
```

Elementos que existen en el Evento A pero no en el Evento B

```
> setdiff(apply(A,1,paste, collapse=""), apply(B,1,paste, col
```

Nota setdiff no es simetrico y podemos calcular el complemento de todos los eventos Ej. setdiff(S,A)

Probabilidad

Probabilidades de frecuencias relativas

P(A) ≈ observados / posibles ≈ S_n/n

```
> S<-data.frame(pos=c(1:6))
> posibles<-expand.grid(t(S),t(S),t(S))
> posibles[which(posibles[,1] == posibles[,2] & posibles[,3] =
> obsv<-length(which(posibles[,1] == posibles[,2] & posibles[
> prob= obsv/length(posibles)
```

Ej. Moneda no balanceada

```
> S<-c("H","T")
> p<-c(1/3,2/3)
> sample(S, prob=p, size=1, replace=T)
> sample(S, prob=p, size=200, replace=T)
```

WARNING: RAM memory y probabilidades infinitecimales

Probabilidad

Conteo con urnas

	ordered = TRUE	ordered = FALSE
replace = TRUE	n^k	$\frac{(n-1+k)!}{(n-1)!k!}$
replace = FALSE	$\frac{n!}{(n-k)!}$	$\binom{n}{k}$

Numeros Factoriales

```
> factorial(n)
```

Coeficiente binomial (Combinaciones)

```
> choose(n,k)
```

Probabilidad

Probabilidad Condicional

```
> S<-1:6
> space <- sample(S, size=100, replace= TRUE)
> head(S) # first few rows
```

```
> E <- expand.grid(t(S),t(S),t(S))
> A <- subset(E, Var1 == Var2)
> B <- subset(E, Var1 + Var2 >= 8)
```

```
> prob(A, given = B) #no Code
> prob(B, given = A) #no Code
```

Probabilidad

Variables Aleatorias

Definición: Una variable aleatoria X es una función X:S -> R que asocia para cada w ∈ S exactamente X(ω) = x.

Se define como S todos los posibles resultados de el evento E

Ejemplo:

Definimos la variable aleatoria X como "numero de aguilas cuando se tira una moneda".

Por lo tanto si **S** es nuestro espacio muestral y **w** los sucesos posibles

w∈ S	AA	AS	SA	SS
X(w) = x	2	1	1	0

Probabilidad

Variables Aleatorias

Escribir una formula que define una variable aleatoria dentro de una función, agregando una columna a un data.frame.

Tiramos un dado de 4 lados 3 veces y definimos nuestra variable $U = X1 - X2 + X3$

Ahora podemos preguntar, ¿Cual es la probabilidad de que $U > 6$?

Distribuciones de datos

Distribuciones de datos

Centroide: Conjunto de datos está asociado con un número que representa una tendencia media o general de los datos.

La **Dispersión** de un conjunto de datos está asociada con su variabilidad; Los conjuntos de datos con una dispersión grande tienden a cubrir un gran intervalo de valores, mientras que los conjuntos de datos con dispersión pequeña tienden a agruparse fuertemente alrededor de un valor central.

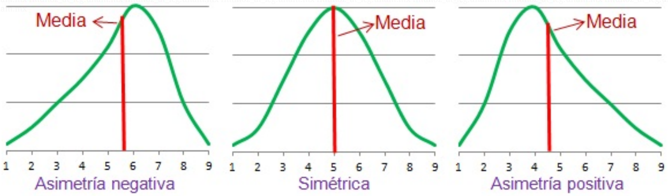
Forma: Forma exhibida por una pantalla gráfica asociada. La forma puede decirnos mucho sobre cualquier estructura subyacente a los datos, y puede ayudarnos a decidir qué procedimiento estadístico debemos usar para analizar los.

Distribuciones de datos

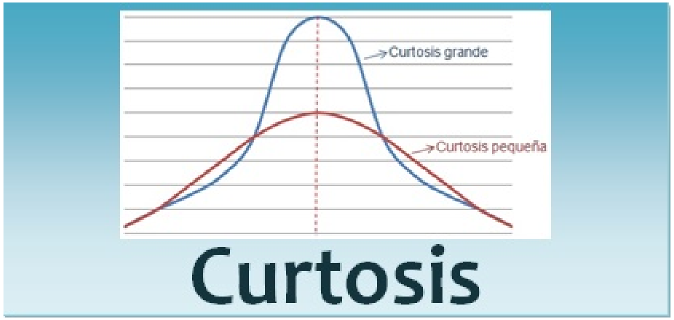
Forma

Simetría y **asimetría**

- **positivamente sesgada**
- **negativamente sesgada**



La **curtosis** (o apuntamiento) es una medida de forma que mide cuán escarpada o achatada está una curva o distribución.



Estadistica descriptiva

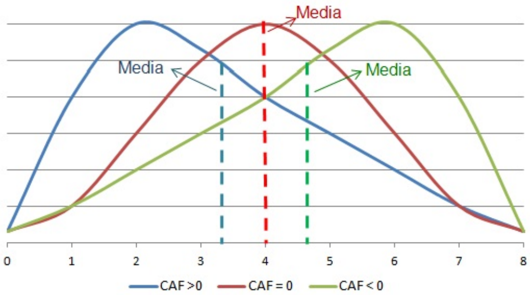
Medidas de Forma

La **asimetría** (Fisher) de la muestra, se define por la fórmula

$$CA_F = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^3}{Ns^3}$$

siendo $\overline{x}$ la media y s la desviación típica

donde S es la desviación estandar (o tipica)



Estadistica descriptiva

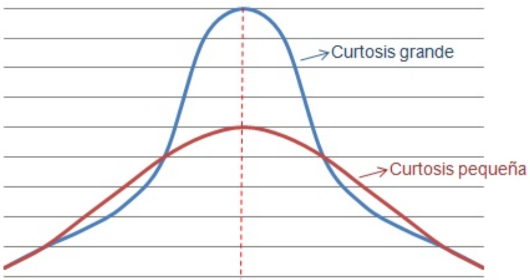
Medidas de Forma

La **curtosis** de la muestra, se define por la fórmula

$$Curtosis = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^4}{Ns^4} - 3$$

siendo $\overline{x}$ la media y s la desviación típica

donde S es la desviación estandar (o tipica)



Estadistica descriptiva

Medidas de Forma

Asimetria

```
> library(e1071)
> skewness(discoveries)
```

```
> 2 * sqrt(6/length(discoveries))
```

Nota si 2 * sqrt(6/n) < skewness(x) => existe un sesgo dado el signo del calculo.

Curtosis

```
> kurtosis(UKDriverDeaths)
```

```
> 4 * sqrt(6/length(UKDriverDeaths))
```

Nota abs(4 * sqrt(6/n)) < kurtosis(x) => presenta curtosis

Estadistica descriptiva

Utilizando R. Calcula las siguientes cosas del vector

```
x<-round(runif(20, min=1, max=100))
```

- rango
- media
- mediana/media recortada
- quantiles/quintiles/septiles
- varianza
- desviación estandar

Nota Rcmdr

Statistics > Summaries > Numerical Summaries

calculamos los cuantiles automaticamente

Estadistica descriptiva

Rangos intercuantiles y MAD

```
> tr=c(3,4,5,3,4,5,4,3,2,3,12,11,3,4,89)
> quantile(tr)
> quantile(tr,.25)
> quantile(tr,.10)
```

suceptibilidad de la media, mediana a valores extremos.

Rango intercuartil (**IQR**) definido por IQR = q*{0.75} - q*{0.25}

Otro método más robusto que el IQR es la Media de la desviación absoluta (**MAD**).

1. Calculamos la media (prom)
2. mediana(|x{i} - prom(X)|) , para toda i

Observaciones Extremas

Problemas que pueden implicar estimaciones exageradas e "inestabilidad" estadística. Podemos considerar que estos datos pueden ser:

- Error tipográfico (typoo)
- Observaciónes que no eran para el estudio. (Ej. Complicaciones medicas)
- Indican un fenomeno o una tendencia más profunda

Estadistica descriptiva

Grafica de caja

Estas gráficas son buenas para visualizar mucha información descriptiva de nuestros datos al mismo tiempo:

Centroide (estimada por la mediana)

Dispersión

Forma

Observaciones extremas

Outliers Observaciones que pasan 1.5 veces el tamaño de la caja para cualquier extremo.

Para observar los valores outliers

```
> boxplot.stats(rivers)$out #1.5 default
```

```
> boxplot.stats(rivers, coef = 3)$out #coef=3
```

```
> boxplot(rivers, horizontal=T)
```

Estadistica descriptiva

Z-value

Valor estandarizado, cuando queremos comparar datos en escala que es independiente a la medida.

Dado X=x[1], x[2], x[3], … ,x[n] los z-scores son z[1], z[2],..n se ven definidos como

z[i]=(x[i]-median(x))/s

donde s es la sd()

```
> ?scale
```

Lectura y escritura de datos.

Read table, View, fix

```
> # Leemos el archivo tabla.csv y lo nombramos misdatos
> misdatos <- read.table("Pathway", header=FALSE, sep="", na.s
```

Con la función "View" visualizamos los datos que hemos cargado en memoria anteriormente.

```
> View(Datos) #ver los datos

> fix(Datos) # editarlos datos
```

Lectura y escritura de datos.

write

```
> Datos1 <- edit(as.data.frame(NULL)) # Creamos una tabla en b
```

```
> Datos1 # Vemos si realmente tenemos lo datos
> Datos1$var1->A # Vemos las columnas y las renombramos
> Datos1$var2->B
```

```
> A
> B
> A+B
> write(A*B,"sumaAyB.dat") # Lo guardamos en un fichero .dat c
```

Ejemplos de funciones

Funciones elementales

Calcular la media

```
> media<-function(x=NA)
+ {
+ x<-x[!is.na(x)]
+ sum(x)/length(x)
+ }
> media(c(2,4,1,3,6,7))
> media(c(2,4,1,3,6,NA))
```

Calcular la varianza

```
> Varianza<-function(x=NA)
+ {
+ n<-length(x)
+ v<-sum((x-(sum(x)/n))^2)/n
+ return(v)
+ }
```

Ejemplos de funciones

```
### Funciones el
```

Calcular la desviación estandar

```
> DT<-function(x=NA)
+ {
+ n<-length(x)
+ v<-sqrt(sum((x-(sum(x)/n))^2)/n)
+ return(v)
+ }
> DT(1:3)
> DT(c(1,3,4,2,6,4))
```

Calcular la covarianza

```
> Varianza<-function(x=NA)
+ {
+ n<-length(x)
+ v<-sum((x-(sum(x)/n))^2)/n
+ return(v)
+ }
```

Ejercicio. Crear una funcion llamada fact2 que genere el factorial de cualquier numero.

That's all folks (for now)!

Slideshow created using remark.