



# Prácticas Propuestas Tema 1 R

Universidad de Granada





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 Selección
- 3 Recodificación de variables
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 Operaciones con data.frames
- 5 Archivos
  - Función `read.table()`
- 6 Gráficos





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 **Selección**
- 3 **Recodificación de variables**
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 **Operaciones con data.frames**
- 5 **Archivos**
  - Función `read.table()`
- 6 **Gráficos**





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 **Selección**
- 3 **Recodificación de variables**
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 **Operaciones con data.frames**
- 5 **Archivos**
  - Función `read.table()`
- 6 **Gráficos**





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 **Selección**
- 3 **Recodificación de variables**
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 **Operaciones con data.frames**
- 5 **Archivos**
  - Función `read.table()`
- 6 **Gráficos**





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 **Selección**
- 3 **Recodificación de variables**
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 **Operaciones con data.frames**
- 5 **Archivos**
  - Función `read.table()`
- 6 **Gráficos**





# Prácticas iniciales

- 1 **Uso de operadores**
  - Operadores de generación
  - Operadores lógicos
- 2 **Selección**
- 3 **Recodificación de variables**
  - Operaciones usuales de recodificación de datos
  - Agrupación en intervalos
- 4 **Operaciones con data.frames**
- 5 **Archivos**
  - Función `read.table()`
- 6 **Gráficos**





# Ejemplos de uso de operadores

Uso de los operadores **:**, **rep()**, **seq()**

## Operadores

```
> seq(1,5,0.5)
 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> rep(8,10)
 8 8 8 8 8 8 8 8 8 8
> 20:40
 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
 35 36 37 38 39 40
> seq(length=10,1,30)
1.000000  4.222222  7.444444 10.666667 13.888889
17.111111 20.333333 23.555556 26.777778 30.000000
```



## Operadores de generación

## Use los comandos siguientes

## :=, rep(), seq()

- 1 n1:n2
- 2 rep(n1,n)
- 3 seq(n1,n2,incremento)
- 4 seq(length=n,n1,n2)

## Ejemplo

- Genere números consecutivos de 6 a 20
- Repita 5 veces el valor 10
- Genere números de 2 a 20 con incrementos de 4
- genere 10 valores reales entre 1 y 5

?



# Operadores lógicos

Con el vector `x` de valores

## Ejemplo de uso de operadores lógicos

Operadores: `==`, `!=`, `<`, `>`, `>=`, `<=`

```
> x
```

```
[1] 4 4 2 2 1 5 5 4 3 5
```

```
> x==4
```

```
[1] TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE
```

```
> x!=4
```

```
[1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
```

```
> x>3
```

```
[1] TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE FALSE
```

```
> x<=5
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```



# Selección de elementos de variables y data.frames

## Selección

- Selección de elementos de un vector
- Selección de elementos de un data.frame
- Selección de filas de un data.frame
- Selección de columnas de un data.frame



# Selección de elementos de variables y data.frames

Con los objetos **dtf** y **vect**

## Selección

- Selecciona todas las columnas del dataframe **dtf** menos las 3 a 7

```
dtf[, -(3:7)]
```

- Selecciona las columnas 2 a 5 del dataframe **dtf**

```
dtf[, (2:5)]
```

- Selecciona los elementos del vector **vect** menos los 3 a 7

```
vect[-(3:7)]
```

- Selecciona los elementos en posiciones pares de **vect**

```
vect[2*(1:5)]
```



# Ejemplo selección de elementos

## Selección

Con el data.frame **USArrests** disponible en package **data()** de R, formado por 50 observaciones sobre 4 variables

- Selección de primeras 10 filas de la variable **UrbanPop**
- Selección de filas 4 y 7 de la variable **UrbanPop**
- Selección del elemento de fila 3 columna 4 del data.frame
- Selección de las últimas 10 filas del data.frame
- Selección de columnas 1 y 4 del data.frame

Realice ejercicios similares con el data.frame **swiss** disponible también en package **data** de R



# Ejemplo selección de casos que cumplen condiciones

## Selección

Con el data.frame **USArrests** disponible en package `data()` de R, formado por 50 observaciones sobre 4 variables

- Selección de filas cuyo valor de la variable **UrbanPop** supera 80
- Selección de casos con  $100 < Assault < 170$  y  $UrbanPop > 70$
- Selección de filas que cumple  $100 < Assault < 170$  o  $UrbanPop > 70$
- Selección de los casos con valores en todas las variables por encima de los correspondientes cuartiles terceros. Use la función **summary(USArrests)** para determinarlos previamente.



# Ejemplo selección de casos que cumplen condiciones

## Selección de casos

Se hace más manejable con un nombre más corto.

`b= USArrests` (Otra opción es usar la función `attach` de R que permite acceder a las variables sin especificar el nombre del `data.frame`)

- Selección de filas cuyo valor de la variable **UrbanPop** supera 80.

`b[b$UrbanPop > 80, ]` (el operador lógico selecciona filas)

- Selección de casos con  $100 < Assault < 170$  y  $UrbanPop > 70$

El operador lógico `oplog` es

`oplog=(100 < b$Assault & b$Assault < 170 ) & (b$UrbanPop > 70)`

`b[oplog, ]`





# Ejemplo continuación

## Selección de casos

- Selección de filas que cumple  $100 < Assault < 170$  o  $UrbanPop > 70$
- Selección de los casos con valores en todas las variables por encima de los correspondientes cuartiles terceros. Use la función **summary(USArrests)** para determinarlos previamente.

Realice ejercicios similares con el data.frame **swiss** disponible también en package **data** de R





# Recodificar variables

Operaciones usuales de recodificación de datos

Sustitución de categorías de un factor por otros niveles.

Sustituye los niveles h y m de sexo por hombre mujer

```

> sexo
[1] m h m h m m m h
Levels: h m
> sexo2=factor(sexo,levels=c("h","m"),
               labels=c("Hombre","Mujer"))
> sexo2
[1] Mujer  Hombre Mujer  Hombre Mujer  Mujer  Mujer  Hombre
Levels: Hombre Mujer

```





# Variables expresadas en intervalos

## La función `cut()`

Permite agrupar en intervalos con extremos especificados

### Ejemplo

```
> x
[1] 1 4 5 5 2 4 4 1 2 3
> cut(x,br=c(0,2,5))
[1] (0,2] (2,5] (2,5] (2,5] (0,2] (2,5] (2,5]
(0,2] (0,2] (2,5]
Levels: (0,2] (2,5]
> cut(x,br=c(1,2,5))
[1] <NA> (2,5] (2,5] (2,5] (1,2] (2,5] (2,5]
<NA> (1,2] (2,5]
Levels: (1,2] (2,5]
>
```



# Operaciones corrientes con data.frames

## Las más usadas

- Combinar data.frames por filas y columnas. **rbind()** y **cbind()**
- Uso de **subset()** para extraer parte de información del data.frame que cumple una determinada condición.  
subset(data.frame, operador lógico, select=var)
- División del data.frame con la función **split** según los niveles de un factor. split(data.frame, factor)
- División del data.frame con la función **by()**.  
by(data.frame, factor o lista de factores, función)
- División de data.frames con **aggregate()**.  
aggregate(data.frame, factor o lista de factores, función)
- Mezcla de data.frames con **merge()**. Una variable como clave permite relacionar los casos de ambos.  
merge(dt1, dt2, by.x="var", by.y="var")



# Ejemplos

## Data.frames con el mismo número de filas

```
> a1
```

	Provincia	TasaParo
--	-----------	----------

1	A	10
2	B	12
3	C	13
4	D	16

```
> a2
```

	Poblacion
--	-----------

1	500
2	340
3	20
4	300



# Ejemplos con `cbind()`

## Resultado de unión de Data.frames por columnas

```
> cbind(a1,a2)
  Provincia TasaParo Poblacion
1         A        10       500
2         B        12       340
3         C        13        20
4         D        16       300
>
```



# Ejemplos: Uso de `rbind()`

## Data.frames con el mismo número de columnas

```
> a1
  Provincia TasaParo
1         A        10
2         B        12
```

```
> a2
  Provincia TasaParo
3         C        13
4         D        16
```

```
> rbind(a1,a2)
  Provincia TasaParo
1         A        10
2         B        12
3         C        13
4         D        16
```





# Ejemplos: Uso de merge()

## Data.frames con variable Provincia identificando filas

```
> b
```

```
Provincia partido
```

1	A	x
2	B	x
3	C	y
4	D	y
5	E	y

```
> a
```

```
Provincia TasaParo Poblacion
```

1	A	10	500
2	B	12	340
3	C	13	20
4	D	16	300



# Ejemplos: Resultado de mezcla `merge()`

## Data.frames con variable Provincia identificando filas

```
> c=merge(a,b,by.x="Provincia",by.y="Provincia",all=T)
> c
```

	Provincia	TasaParo	Poblacion	partido
1	A	10	500	x
2	B	12	340	x
3	C	13	20	y
4	D	16	300	y
5	E	NA	NA	y

```
>
```



# Ejemplos: Resultado de división `split()`

## división de data.frame c por partido

```
split(c,c$partido)
```

\$x

	Provincia	TasaParo	Poblacion	partido
1	A	10	500	x
2	B	12	340	x

\$y

	Provincia	TasaParo	Poblacion	partido
3	C	13	20	y
4	D	16	300	y
5	E	NA	NA	y

>



# Ejemplos: Resultado de división **by()**

## división de data.frame **c** por partido y resumen columnas 2 y 3

```
> by(c[,2:3],c$partido,summary)
```

```
c$partido: x
```

TasaParo	Poblacion
Min. :10.0	Min. :340
1st Qu.:10.5	1st Qu.:380
Median :11.0	Median :420
Mean :11.0	Mean :420
3rd Qu.:11.5	3rd Qu.:460
Max. :12.0	Max. :500

```
c$partido: y
```

TasaParo	Poblacion
Min. :13.00	Min. : 20
1st Qu.:13.75	1st Qu.: 90



# Ejemplos: Resultado de división `aggregate()`

división de data.frame **c3** por partido y resumen columnas 1 y 2

```
> c3
  TasaParo Poblacion partido
1         10         500      x
2         12         340      x
3         13          20      y
4         16         300      y

> aggregate(c3[,1:2],by=list(c3$partido),FUN="mean")
  Group.1 TasaParo Poblacion
1      x         11.0       420
2      y         14.5       160

>
$
```



# Lectura con read.table()

```
mes ventas
enero 1000
febrero 2000
marzo 1500
```

Lectura de archivo guardado con EXCEL como texto con tabulaciones

```
> a=read.table("Libro1.txt",head=T)
> dim(a)
[1] 3 2
> a
```

	mes	ventas
1	enero	1000
2	febrero	2000
3	marzo	1500



# Lectura con read.table()

El archivo Libro3 se guarda como texto con tabulaciones y contiene datos missing que se sustituyen por NA en variables numéricas

Lectura de archivo guardado con EXCEL como texto con tabulaciones

```
> a=read.table("Libro3.txt",head=T,sep="\t",
na.strings="NA")
```

```
> a
```

	edad	salario	formacion
1	23	1500	a
2	45	3000	b
3	NA	NA	
4	45	1500	
5	NA	3000	b



# Otras variantes de read.table llevan opciones por defecto

Guarde el file Libro4 como separado por comas desde Excel  
Luego lea con **read.csv2()** o con **read.delim2()** (aunque diga texto delimitado por comas lo guarda con punto y coma)

Lectura de archivo guardado con EXCEL como delimitado por comas

```
a=read.csv2("Libro4.csv")
> a=read.csv2("Libro4.csv")
> a
```

	edad	salario	formacion
1	23	1000.0	a
2	45	3000.0	b
3	NA	NA	
4	45	1500.0	
5	NA	3000.0	b





# Representación gráfica de datos

## Dispositivos gráficos

Un dispositivo gráfico es una ventana gráfica o un archivo. Los parámetros gráficos vienen dados por defecto, pero se pueden cambiar.

## Ejemplo histograma

```
> x=rnorm(100,30,5)
> hist(x)
> hist(x,freq=F)
> hist(x,freq=F,col="red")
> hist(x,freq=F,col="red",main="Edad de Grupo")
>
```



$\alpha$