

## Sección 2

# Vectores: Ordenar vectores

- La función `sort` ordena los elementos de un vector en orden creciente.

```
> x<- c(31,4,15,92,65)
> sort(x)
[1]  4 15 31 65 92
```

- En el ejemplo del dataset de asesinatos

```
> library(dslabs)
> data(murders)
> sort(murders$total)
 [1]  2  4  5  5  7  8 11
 [8] 12 12 16 19 21 22 27
[15] 32 36 38 53 63 65 67
[22] 84 93 93 97 97 99 111
[29] 116 118 120 135 142 207 219
[36] 232 246 250 286 293 310 321
```

# Ordenar vectores: Función Order

- La función `order` toma un vector y nos retorna los índices que ordenan a ese vector de manera creciente.

```
> x<- c(31,4,15,92,65)
> index<-order(x)
> index
[1] 2 3 1 5 4
> x[index]
[1] 4 15 31 65 92
```

- El orden en un `dataSet` se mantiene en todos los vectores incluidos en el

```
> murders$state[1:5]
[1] "Alabama" "Alaska" "Arizona" "Arkansas" "California"
> murders$abb[1:5]
[1] "AL" "AK" "AZ" "AR" "CA"
```

# Ordenar un dataSet

- Para obtener los estados ordenados por la cantidad de asesinatos

```
> index<-order(murders$total)
> murders$state[index]
[1] "Vermont"           "North Dakota"
[3] "New Hampshire"    "Wyoming"
[5] "Hawaii"           "South Dakota"
[7] "Maine"            "Idaho"
[9] "Montana"          "Rhode Island"
```

# Máximos y mínimos

- Para obtener valores mínimos y máximos de un array sin tener que ordenarlo

```
> max(murders$total)
[1] 1257
> i_max<-which.max(murders$total)
> i_max
[1] 5
> murders$state[i_max]
[1] "California"
> min(murders$total)
[1] 2
> murders$state[which.min(murders$total)]
[1] "Vermont"
```

# Ranking

- La función `rank()` nos retorna un ranking del vector que le pasemos como parámetro

```
> x <- c(31,4,15,92,65)
> x
[1] 31  4 15 92 65
> rank(x)
[1] 3 1 2 5 4
```

## Resumen- funciones para ordenar en R

original	sort	order	rank
31	4	2	3
4	15	3	1
15	31	1	2
92	65	5	5
65	92	4	4