

## Sección 3.2

## dplyr

- `library(dplyr)`: Este paquete contiene funciones con las operaciones más comunes para la manipulación de datos.

## mutate

- Para cambiar o agregar una nueva columna a un data table utilizamos `mutate`.

# Ejemplo - mutate

- Agregamos al data frame murders una nueva columna con la frecuencia de asesinatos 'rate'
  - Primer argumento data frame
  - Segundo argumento la nueva columna con formato: nombre = valor

```
library(dplyr)
library(dslabs)
data(murders)
murders<-mutate(murders,rate=total / population * 100000)
```

```
> murders
```

	state	abb	region	population	total	rate
1	Alabama	AL	South	4779736	135	2.8244238
2	Alaska	AK	West	710231	19	2.6751860
3	Arizona	AZ	West	6392017	232	3.6295273
4	Arkansas	AR	South	2915918	93	3.1893901
5	California	CA	West	37253956	1257	3.3741383
6	Colorado	CO	West	5029196	65	1.2924531
7	Connecticut	CT	Northeast	3574097	97	2.7139722
8	Delaware	DE	South	897934	38	4.2319369
9	District of Columbia	DC	South	601723	99	16.4527532

# Función Filter()

- Filtra datos creando subconjuntos de filas.
- Ejemplo: Obtenemos una tabla con los estados que tienen una frecuencia de asesinatos  $\leq 0.71$ 
  - Primer argumento el data table original
  - Segundo argumento la condición a filtrar

```
> filter(murders, rate <= 0.71)
```

	state	abb	region	population	total	rate
1	Hawaii	HI	West	1360301	7	0.5145920
2	Iowa	IA	North Central	3046355	21	0.6893484
3	New Hampshire	NH	Northeast	1316470	5	0.3798036
4	North Dakota	ND	North Central	672591	4	0.5947151
5	Vermont	VT	Northeast	625741	2	0.3196211

# Función Select

- Selecciona columnas específicas para realizar subconjuntos de datos.
- Ejemplo: Creamos una tabla que contenga solo 3 columnas de murders
  - Primer parámetro es el data frame original
  - Luego se ingresan cuáles serán las columnas de la tabla a crear

```
> new_table<- select(murders,state, region, rate)
> new_table
```

	state	region	rate
1	Alabama	South	2.8244238
2	Alaska	West	2.6751860
3	Arizona	West	3.6295273
4	Arkansas	South	3.1893901
5	California	West	3.3741383
6	Colorado	West	1.2924531

## Todo junto

```
> new_table<- select(murders,state, region, rate)
> filter(new_table, rate <= 0.71)
```

	state	region	rate
1	Hawaii	West	0.5145920
2	Iowa	North Central	0.6893484
3	New Hampshire	Northeast	0.3798036
4	North Dakota	North Central	0.5947151
5	Vermont	Northeast	0.3196211

- La primer función se aplica al data frame 'murders'
- luego trabajamos con la recién creada 'new\_table'

# Pipe

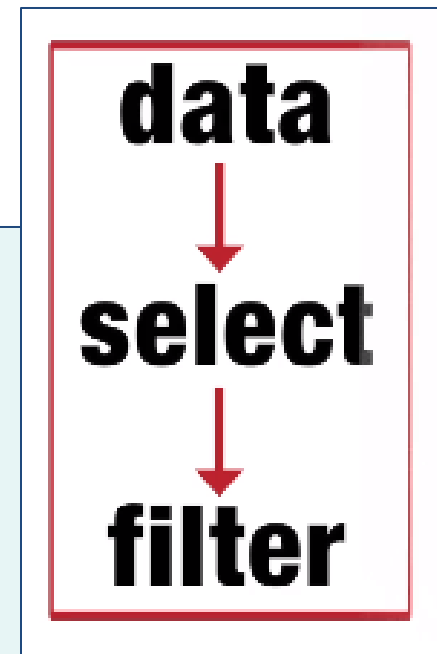
- Operador Pipe `%>%`: Es el nexo entre funciones, sirve para enviar los resultados de una función como entrada de otra.

# Pipe - %>%

- Podemos trabajar directamente con los datos originales y obtener el resultado que queremos

```
> murders %>% select(state, region, rate) %>% filter(rate <= 0.71)
```



	state	region	rate
1	Hawaii	West	0.5145920
2	Iowa	North Central	0.6893484
3	New Hampshire	Northeast	0.3798036
4	North Dakota	North Central	0.5947151
5	Vermont	Northeast	0.3196211





# Crear data frames

```
> notas <- data.frame(nombres = c("John", "Juan", "Jean", "Yao"),
+                     exam_1 = c(95, 80, 90, 85),
+                     exam_2 = c(90, 85, 85, 90))
> notas
  nombres exam_1 exam_2
1   John     95     90
2   Juan     80     85
3   Jean     90     85
4    Yao     85     90
> class(notas$nombres)
[1] "factor"
> notas <- data.frame(nombres = c("John", "Juan", "Jean", "Yao"),
+                     exam_1 = c(95, 80, 90, 85),
+                     exam_2 = c(90, 85, 85, 90),
+                     stringsAsFactors = FALSE)
> class(notas$nombres)
[1] "character"
```



A partir de la versión 4.0 de R no es necesario