Resumiendo datos con dplyr

dplyr

- Entender la importancia de resumir los datos en el análisis exploratorio de los mismos
- Poder utilizar la funciones summarize() y group_by() para reducir los datos
- Poder acceder a valores utilizando el dot
- Poder examinar los datos luego de ordenarlos utilizando la función arrange()

Función Resumen - Summarize

- Crea un nuevo dataset con los elementos importantes de la entrada.
- Resumen estadístico de los datos referenciales.
- Funciona de forma análoga a la función mutate, excepto que en lugar de añadir nuevas columnas crea un nuevo data frame.

Ejemplo función summarize

- Calculamos el promedio y la desviación estándar para los hombres

```
library(tidyverse)
library(dslabs)
data(heights)
s <- heights %>% filter(sex == "Male") %>%
summarize(average = mean(height), standard_deviation =
sd(height))
```

```
> s$average
[1] 69.31475
> s$standard_deviation
[1] 3.611024
> s
    average standard_deviation
1 69.31475           3.611024
```

Operador punto - dot

- Acceder a valores de datos pipeados

```
> us_murder_rate <- murders %>% summarize(rate =
sum(total) / sum(population) * 100000) %>% .$rate
> us_murder_rate
[1] 3.034555
```

```
> us_murder_rate <- murders %>% summarize(rate =
sum(total) / sum(population) * 100000)
```

```
> us_murder_rate
      rate
1 3.034555
```

```
> us_murder_rate %>% .$rate
[1] 3.034555
```

```
> us_murder_rate$rate
[1] 3.034555
```

```
> us_murder_rate %>% pull(rate)
[1] 3.034555
```

Función group_by

- Agrupa un conjunto de filas de acuerdo con los valores de una o más columnas o expresiones

```
group_by(x, cond): dplyr
```

  - x = data frame
  - cond = Condición para agrupar datos

- Escritura utilizando el pipe

```
x %>% group_by(cond)
```

Ejemplo group_by : dplyr

- Agrupar de acuerdo a una expresión lógica:

```
alturas<-group_by(heights,height>70)
```

	sex	height	height > 70
1	Male	75.00000	TRUE
2	Male	70.00000	FALSE
3	Male	68.00000	FALSE
4	Male	74.00000	TRUE
5	Male	61.00000	FALSE
6	Female	65.00000	FALSE
7	Female	66.00000	FALSE

```
> alturas %>% group_by(sex)
# A tibble: 1,050 x 3
# Groups:   sex [2]
   sex     height `height > 70`
   <fct>    <dbl> <lgl>
 1 Male        75 TRUE
 2 Male        70 FALSE
 3 Male        68 FALSE
 4 Male        74 TRUE
 5 Male        61 FALSE
 6 Female      65 FALSE
 7 Female      66 FALSE
 8 Female      62 FALSE
 9 Female      66 FALSE
10 Male        67 FALSE
# ... with 1,040 more rows
```

group_by y summarize

- Creamos un nuevo data frame con el promedio de alturas de acuerdo al género

```
heights %>%
    group_by(sex) %>%
    summarise(alturas = mean(height))
```

```
# A tibble: 2 x 2
  sex    alturas
  <fct>    <dbl>
1 Female    64.9
2 Male      69.3
```

Summarize y pull

- La mayoría de funciones de la librería dplyr retornan data frames como resultado
- Para acceder al valor de un objeto data frame utilizamos la función pull

```
> indice_us <- murders %>%
+     summarize(rate = sum(total) / sum(population) * 100000)
> indice_us
      rate
1 3.034555
> class(indice_us)
[1] "data.frame"
> indice_us<-pull(indice_us)
> indice_us
[1] 3.034555
> class(indice_us)
[1] "numeric"
```