Programación científica en R

Manejo de datos con dyplr()

Marcos Ehekatzin García Guzmán

Agosto de 2024

Introducción

- Hasta ahora hemos hecho todas las modificaciones a nuestros datos sin utilizar ninguna paquetería extra.
- Esta forma de hacerlo no es eficiente, aunque es útil durante el aprendizaje.
- A partir de ahora utilizaremos la paquetería `tidyverse` para acceder a funciones que nos harán la vida más fácil.

- `tidyverse` es una paquetería diseñada para la ciencia de datos, entre ellas:
  - `dplyr`: Consiste en una *gramática* para la manipulación de datos. Proporciona *verbos* que ayudan a que la manipulación de los datos sea consistente.
  - `tidyr`: Provee de un conjunto de funciones diseñadas para solucionar los problemas más comunes durante la limpieza de datos.
  - `ggplot2`: Provee de una gramática para la creación de gráficas, además de que permite manipular fácilmente los aspectos visuales.
  - `purrr`: Provee de herramientas que mejoran las capacidades de programación de `R` permitiendo un uso fácil y entendible de funciones y vectores.

dplyr

`dplyr` contiene una gramática para la manipulación de datos basada en el uso de los siguientes verbos/funciones:

1. `mutate()`: Permite agregar nuevas variables (que son función de variables ya existentes).
2. `select()`: Hace eso, selecciona variables según sus nombres.
3. `filter()`: Filtra los datos según las condiciones que proveamos.
4. `summarise()`: Produce un resumen de las variables de un data frame.
5. `arrange()`: Cambia el orden de las filas de un df.

- Como siempre, debemos comenzar por la instalación de la paquetería y cargar la librería:

```
> install.packages('tidyverse')
> library('tidyverse')
```

- Las funciónes de `dyplr` siempre utilizan la misma sintaxis:
  1. El primer argumento es el df sobre el que trabajaremos,
  2. Los siguientes argumentos serán las instrucciones para manipular el df.
  3. El resultado será un nuevo data frame.

- Para ver cómo se utiliza cada función utilicemos un df precargado en la librería.

```
head(starwars)
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_colo
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>
## 1 Luke Sky~    172    77 blond      fair       blue
## 2 C-3PO        167    75 <NA>       gold       yellow
## 3 R2-D2         96    32 <NA>       white, bl~ red
## 4 Darth Va~    202   136 none       white      yellow
## 5 Leia Org~    150    49 brown      light      brown
## 6 Owen Lars    178   120 brown, gr~ light      blue
## # ... with 5 more variables: homeworld <chr>, species <c
## #   vehicles <list>, starships <list>
```

- Hagamos un `data frame` con todos los personajes humanos de starwars, conservando todas las demás variables.
- ¿Sugerencias?

```r
humans <- filter(starwars, species == "Human")
head(humans)
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_colo
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>
## 1 Luke Sky~    172    77 blond      fair       blue
## 2 Darth Va~    202   136 none       white      yellow
## 3 Leia Org~    150    49 brown      light      brown
## 4 Owen Lars    178   120 brown, gr~ light      blue
## 5 Beru Whi~    165    75 brown      light      blue
## 6 Biggs Da~    183    84 black      light      brown
## # ... with 5 more variables: homeworld <chr>, species <c
## #   vehicles <list>, starships <list>
```

- Ahora utilizaremos la función mutate calcularemos el índice de masa corporal para los humanos de sw.

```
humans <- mutate(humans, bmi=mass/((height/100)^2))
humans$bmi
```

```
##   [1] 26.02758 33.33007 21.77778 37.87401 27.54821 25.082
##   [9]       NA 24.69136 26.64360 33.95062 25.95156 23.350
##  [17]       NA       NA 23.89326       NA       NA 23.766
##  [25]       NA       NA 21.47709       NA 23.58984
##  [33]       NA       NA 16.52893
```

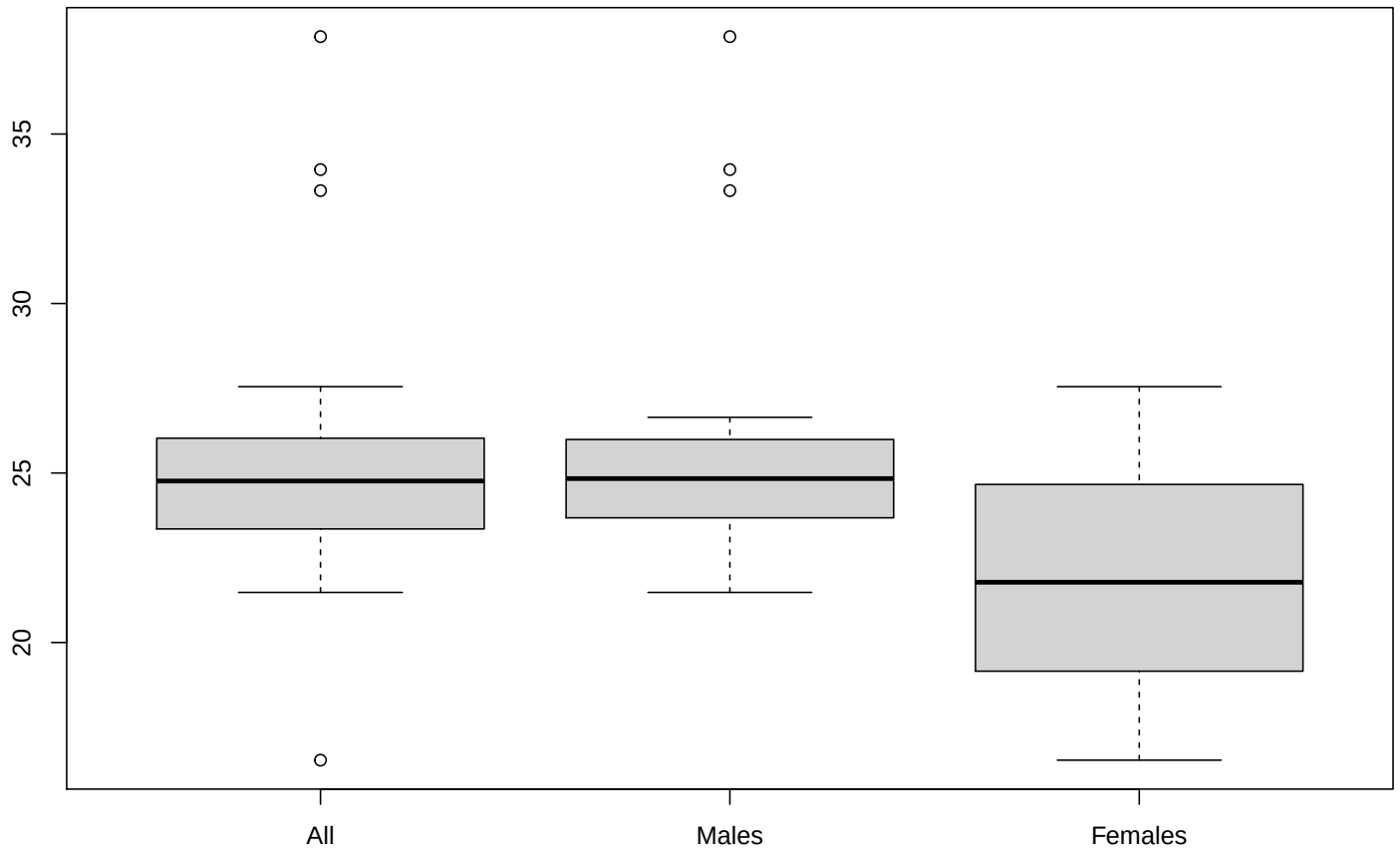
- Si no nos interesan todas las variables, podemos gardar solo las que queremos:

```
humans <- select(humans, name, mass, height, bmi, sex, home
head(humans)
```

```
## # A tibble: 6 x 6
##   name                mass height   bmi sex    homeworld
##   <chr>              <dbl>  <int> <dbl> <chr>  <chr>
## 1 Luke Skywalker        77    172  26.0 male   Tatooine
## 2 Darth Vader          136    202  33.3 male   Tatooine
## 3 Leia Organa           49    150  21.8 female Alderaan
## 4 Owen Lars            120    178  37.9 male   Tatooine
## 5 Beru Whitesun lars    75    165  27.5 female Tatooine
## 6 Biggs Darklighter     84    183  25.1 male   Tatooine
```

Ejercicio

- Hacer una gráfica de caja con el índice de masa corporal para todos, para hombres y para mujeres.



- Summarise nos ayuda a colapsar las variables-
- Ejemplo:

```
summarise(humans, bmi = mean(bmi, na.rm = T))
```

```
## # A tibble: 1 x 1
##     bmi
##   <dbl>
## 1  25.5
```

- Noten que esto no parece muy útil. Podríamos hacer lo mismo así:

```
mean(humans$bmi[!is.na(humans$bmi)])
```

```
## [1] 25.48618
```

- `summarise()` es más útil si lo combinamos con la función group_by(), que sigue la misma gramática.
- Ejemplo, guardemos la media, la mediana y la varianza del índice de masa corporal por planeta de origen.

```
by_world<- group_by(humans,homeworld)
humans_sum <- summarise(by_world, mean = mean(bmi, na.rm =
head(humans_sum)
```

```
## # A tibble: 6 x 4
##   homeworld      mean median    var
##   <chr>         <dbl>  <dbl>  <dbl>
## 1 Alderaan       22.1   22.1  0.165
## 2 Bespin         25.8   25.8 NA
## 3 Bestine IV     34.0   34.0 NA
## 4 Chandrila     NaN     NA   NA
## 5 Concord Dawn   23.6   23.6 NA
## 6 Corellia       25.7   25.7  1.91
```

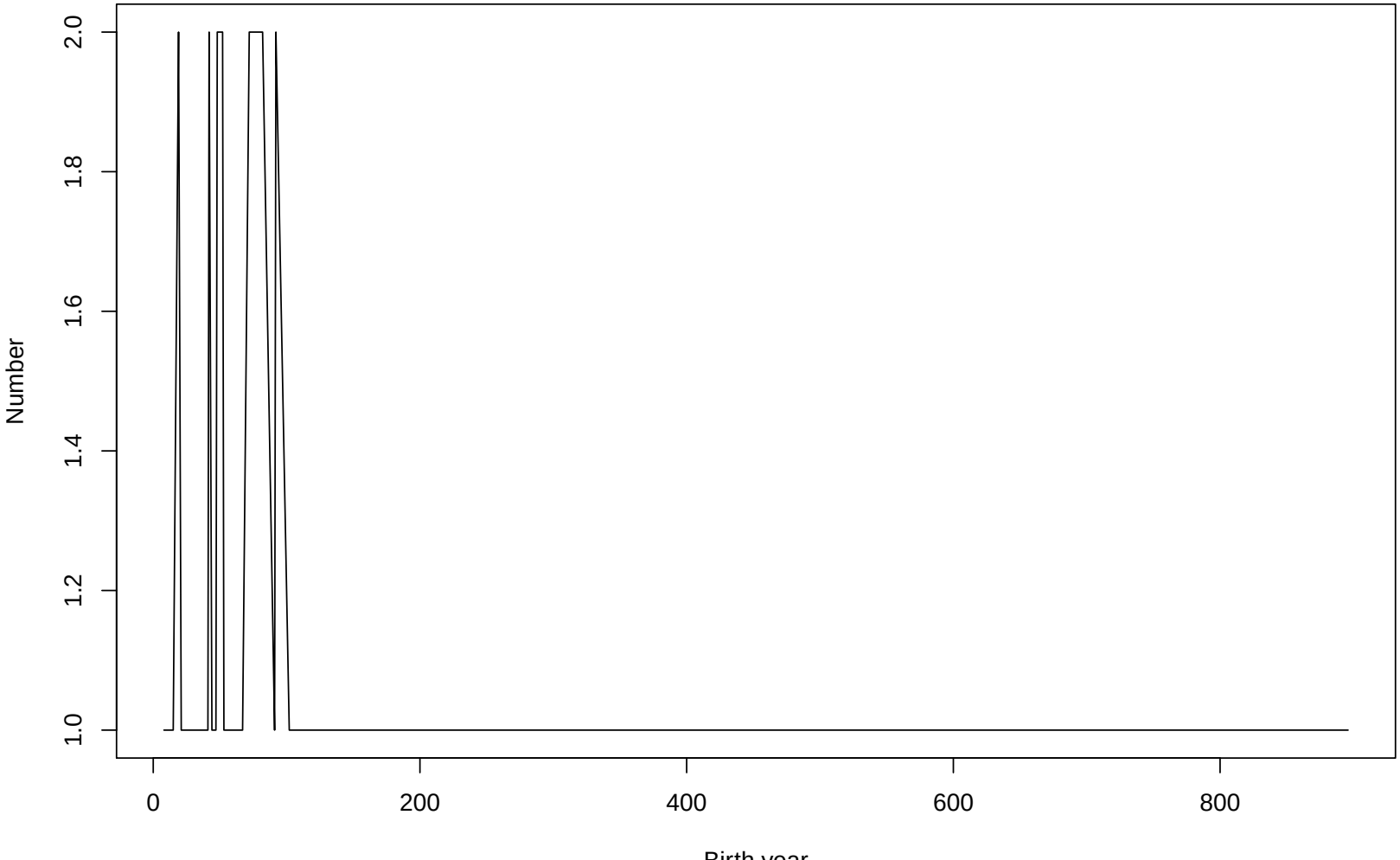
El operador %>%

- Una facilidad que nos da `dyplr`es que podemos utilizar varios verbos al mismo tiempo para modificar una base de datos.
- Esto lo podemos hacer con ayuda del operador %>%
- Ejemplo: Repliquemos el df `humans_sum`

```
humans_sum <- starwars %>%
  group_by(homeworld) %>%
  filter(species == "Human") %>%
  mutate(bmi  = mass / ((height / 100)  ^ 2)) %>%
  summarise(mean = mean(bmi, na.rm = T),
            median = median(bmi, na.rm = T),
            var = var(bmi, na.rm = T))
```

Ejercicio

1. Utilizando el df de starwars, graficar el número de personajes por año de nacimiento.



Ejercicios

Ejercicios

- Ejercicio 1: Utilizando la base de homicidios

1. Genere un data.frame con el total de homicidios en el país por año.
2. Haga una gráfica que muestre la evolución de los homicidios en México.

Ejercicios

- Ejercicio 2: Utilizando la base de la ENOE

1. Obtenga las estadísticas descriptivas del ingreso (ingocup) para:

a. El total de la muestra
b. Mujeres y hombres por separado (Hombres: sex == 1, Mujeres: sex == 2)

2. Haga una gráfica de caja sobre el ingreso (log) separando a hombres y mujeres.
3. Haga un vector con el promedio de el ingreso (log) y la edad por estado (ent) y otro con el promedio de la edad.
4. Haga una gráfica que muestre la relación entre el ingreso (log) y la edad (eda)