**Sesión 10 functions, Data Frame *and dplyr Package***

Contenido: select, filter, arrange, rename, mutate, group\_by, %>%

R. **Actividad 10**

Aplicaciones en R utilizando los datos de esta sesión.

**Introducción a dplyr**

**Para qué sirve dplyr**

El paquete dplyr proporciona una forma bastante ágil de manejar los ficheros de datos de R. Además, el código escrito usando este paquete (especialmente usando la sintaxis en cadena que veremos más adelante) es mucho más legible y fácil de entender que el habitual.

El paquete incluye un conjunto de comandos que coinciden con las acciones más comunes que se realizan sobre un conjunto de datos (seleccionar filas filter, seleccionar columnas select, ordenar arrange, añadir nuevas variables mutate, resumir mediante alguna medida numérica summarise). Lo que hace que la sintaxis sea especialmente clara es la correspondencia tan nítida entre el comando y la acción. Para llevar a cabo estas acciones debemos tener en cuenta algunas características comunes:

* El primer argumento siempre es un *data.frame*
* El resto de argumentos indican lo que queremos hacer con el *data.frame*.
* El resultado siempre tiene también la estructura de *data.frame*

**Un fichero para los ejemplos**

Usaremos los famosos datos de medidas del pétalo y sépalo de tres especies de lirios, que se encuentran en el *data.frame* iris, ya incluido en R. Como este fichero tiene 150 observaciones, creamos otro muy sencillo (con solo 15 filas) para que los ejemplos se entiendan mejor. Seleccionamos los 5 primeros lirios de cada una de las tres especies.

lirios <- iris[c(1:5,51:55,101:105),]

lirios

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 5.1 3.5 1.4 0.2 setosa

## 2 4.9 3.0 1.4 0.2 setosa

## 3 4.7 3.2 1.3 0.2 setosa

## 4 4.6 3.1 1.5 0.2 setosa

## 5 5.0 3.6 1.4 0.2 setosa

## 51 7.0 3.2 4.7 1.4 versicolor

## 52 6.4 3.2 4.5 1.5 versicolor

## 53 6.9 3.1 4.9 1.5 versicolor

## 54 5.5 2.3 4.0 1.3 versicolor

## 55 6.5 2.8 4.6 1.5 versicolor

## 101 6.3 3.3 6.0 2.5 virginica

## 102 5.8 2.7 5.1 1.9 virginica

## 103 7.1 3.0 5.9 2.1 virginica

## 104 6.3 2.9 5.6 1.8 virginica

## 105 6.5 3.0 5.8 2.2 virginica

**Seleccionar filas: filter()**

Esta acción consiste en seleccionar las observaciones (filas) que cumplen las condiciones que nos interesan.

Tres ejemplos: el primer comando selecciona todos los lirios de la especie *setosa*, el segundo selecciona los lirios de la especie *setosa* o *virginica*, el tercero selecciona los lirios de la especie *setosa* cuya longitud de sépalo es inferior a 5 mm.

filter(lirios, Species=='setosa')

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 5.1 3.5 1.4 0.2 setosa

## 2 4.9 3.0 1.4 0.2 setosa

## 3 4.7 3.2 1.3 0.2 setosa

## 4 4.6 3.1 1.5 0.2 setosa

## 5 5.0 3.6 1.4 0.2 setosa

filter(lirios, Species=='setosa' | Species=='virginica')

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 5.1 3.5 1.4 0.2 setosa

## 2 4.9 3.0 1.4 0.2 setosa

## 3 4.7 3.2 1.3 0.2 setosa

## 4 4.6 3.1 1.5 0.2 setosa

## 5 5.0 3.6 1.4 0.2 setosa

## 6 6.3 3.3 6.0 2.5 virginica

## 7 5.8 2.7 5.1 1.9 virginica

## 8 7.1 3.0 5.9 2.1 virginica

## 9 6.3 2.9 5.6 1.8 virginica

## 10 6.5 3.0 5.8 2.2 virginica

filter(lirios, Species=='setosa', Sepal.Length < 5)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 4.9 3.0 1.4 0.2 setosa

## 2 4.7 3.2 1.3 0.2 setosa

## 3 4.6 3.1 1.5 0.2 setosa

**Seleccionar columnas: select()**

Esta acción consiste en elegir un subconjunto de las variables (columnas) del fichero. Por ejemplo,

select(lirios, Sepal.Length, Sepal.Width)

## Sepal.Length Sepal.Width

## 1 5.1 3.5

## 2 4.9 3.0

## 3 4.7 3.2

## 4 4.6 3.1

## 5 5.0 3.6

## 51 7.0 3.2

## 52 6.4 3.2

## 53 6.9 3.1

## 54 5.5 2.3

## 55 6.5 2.8

## 101 6.3 3.3

## 102 5.8 2.7

## 103 7.1 3.0

## 104 6.3 2.9

## 105 6.5 3.0

Es posible seleccionar un rango de variables utilizando : o elegir todas las variables menos algunas. Por ejemplo, el primero de los siguientes comandos selecciona todas las variables entre la longitud del pétalo y la del sépalo, mientras que el segundo selecciona todas las variables menos la especie:

select(lirios, Petal.Length:Sepal.Length)

## Petal.Length Sepal.Width Sepal.Length

## 1 1.4 3.5 5.1

## 2 1.4 3.0 4.9

## 3 1.3 3.2 4.7

## 4 1.5 3.1 4.6

## 5 1.4 3.6 5.0

## 51 4.7 3.2 7.0

## 52 4.5 3.2 6.4

## 53 4.9 3.1 6.9

## 54 4.0 2.3 5.5

## 55 4.6 2.8 6.5

## 101 6.0 3.3 6.3

## 102 5.1 2.7 5.8

## 103 5.9 3.0 7.1

## 104 5.6 2.9 6.3

## 105 5.8 3.0 6.5

select(lirios, -Species)

## Sepal.Length Sepal.Width Petal.Length Petal.Width

## 1 5.1 3.5 1.4 0.2

## 2 4.9 3.0 1.4 0.2

## 3 4.7 3.2 1.3 0.2

## 4 4.6 3.1 1.5 0.2

## 5 5.0 3.6 1.4 0.2

## 51 7.0 3.2 4.7 1.4

## 52 6.4 3.2 4.5 1.5

## 53 6.9 3.1 4.9 1.5

## 54 5.5 2.3 4.0 1.3

## 55 6.5 2.8 4.6 1.5

## 101 6.3 3.3 6.0 2.5

## 102 5.8 2.7 5.1 1.9

## 103 7.1 3.0 5.9 2.1

## 104 6.3 2.9 5.6 1.8

## 105 6.5 3.0 5.8 2.2

Otra posibilidad es seleccionar las variables cuyo nombre contenga ciertos términos:

select(lirios, contains('Petal'))

## Petal.Length Petal.Width

## 1 1.4 0.2

## 2 1.4 0.2

## 3 1.3 0.2

## 4 1.5 0.2

## 5 1.4 0.2

## 51 4.7 1.4

## 52 4.5 1.5

## 53 4.9 1.5

## 54 4.0 1.3

## 55 4.6 1.5

## 101 6.0 2.5

## 102 5.1 1.9

## 103 5.9 2.1

## 104 5.6 1.8

## 105 5.8 2.2

En lugar de *contains*, se puede hacer un uso similar con las siguientes expresiones: *starts\_with*, *ends\_with* o *matches*.

**Ordenar: arrange()**

Ordena las filas de menor a mayor valor de la variable elegida. Si escribimos un signo menos, ordena de mayor a menor. Por ejemplo para ordenar de acuerdo con la longitud del sépalo:

arrange(lirios, Sepal.Length)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 4.6 3.1 1.5 0.2 setosa

## 2 4.7 3.2 1.3 0.2 setosa

## 3 4.9 3.0 1.4 0.2 setosa

## 4 5.0 3.6 1.4 0.2 setosa

## 5 5.1 3.5 1.4 0.2 setosa

## 6 5.5 2.3 4.0 1.3 versicolor

## 7 5.8 2.7 5.1 1.9 virginica

## 8 6.3 3.3 6.0 2.5 virginica

## 9 6.3 2.9 5.6 1.8 virginica

## 10 6.4 3.2 4.5 1.5 versicolor

## 11 6.5 2.8 4.6 1.5 versicolor

## 12 6.5 3.0 5.8 2.2 virginica

## 13 6.9 3.1 4.9 1.5 versicolor

## 14 7.0 3.2 4.7 1.4 versicolor

## 15 7.1 3.0 5.9 2.1 virginica

arrange(lirios, -Sepal.Length)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 7.1 3.0 5.9 2.1 virginica

## 2 7.0 3.2 4.7 1.4 versicolor

## 3 6.9 3.1 4.9 1.5 versicolor

## 4 6.5 2.8 4.6 1.5 versicolor

## 5 6.5 3.0 5.8 2.2 virginica

## 6 6.4 3.2 4.5 1.5 versicolor

## 7 6.3 3.3 6.0 2.5 virginica

## 8 6.3 2.9 5.6 1.8 virginica

## 9 5.8 2.7 5.1 1.9 virginica

## 10 5.5 2.3 4.0 1.3 versicolor

## 11 5.1 3.5 1.4 0.2 setosa

## 12 5.0 3.6 1.4 0.2 setosa

## 13 4.9 3.0 1.4 0.2 setosa

## 14 4.7 3.2 1.3 0.2 setosa

## 15 4.6 3.1 1.5 0.2 setosa

Es posible ordenar respecto de una variable y resolver los empates de acuerdo con una segunda variable. El siguiente comando ordena los lirios según la especie (por orden alfabético) y dentro de cada especie ordena de menor a mayor longitud del sépalo:

arrange(lirios, Species, Sepal.Length)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 1 4.6 3.1 1.5 0.2 setosa

## 2 4.7 3.2 1.3 0.2 setosa

## 3 4.9 3.0 1.4 0.2 setosa

## 4 5.0 3.6 1.4 0.2 setosa

## 5 5.1 3.5 1.4 0.2 setosa

## 6 5.5 2.3 4.0 1.3 versicolor

## 7 6.4 3.2 4.5 1.5 versicolor

## 8 6.5 2.8 4.6 1.5 versicolor

## 9 6.9 3.1 4.9 1.5 versicolor

## 10 7.0 3.2 4.7 1.4 versicolor

## 11 5.8 2.7 5.1 1.9 virginica

## 12 6.3 3.3 6.0 2.5 virginica

## 13 6.3 2.9 5.6 1.8 virginica

## 14 6.5 3.0 5.8 2.2 virginica

## 15 7.1 3.0 5.9 2.1 virginica

**Sintaxis en cadena**

El paquete incorpora una sintaxis encadenada que permite escribir las acciones en un orden natural, en oposición a la forma anidada en la que lo haríamos normalmente. Primero se escribe el nombre del fichero y luego las acciones en el orden en que se realizan separadas por el operador %>%(que podríamos leer como *entonces*). Por ejemplo, si queremos seleccionar las variables que contienen las medidas del pétalo, seleccionar los lirios para los que la longitud del pétalo es mayor que 4 mm y ordenarlos de menor a mayor longitud del pétalo, podemos escribir:

*# Sintaxis en cadena*

lirios %>%

select(contains('Petal')) %>%

filter(Petal.Length > 4) %>%

arrange(Petal.Length)

## Petal.Length Petal.Width

## 1 4.5 1.5

## 2 4.6 1.5

## 3 4.7 1.4

## 4 4.9 1.5

## 5 5.1 1.9

## 6 5.6 1.8

## 7 5.8 2.2

## 8 5.9 2.1

## 9 6.0 2.5

El término a la izquierda de cada operador %>% es el primer argumento del término de la derecha (y por lo tanto el primer argumento se puede omitir).

Podemos comparar el código anterior con el código anidado habitual:

arrange(filter(select(lirios, contains('Petal')), Petal.Length > 4), Petal.Length)

## Petal.Length Petal.Width

## 1 4.5 1.5

## 2 4.6 1.5

## 3 4.7 1.4

## 4 4.9 1.5

## 5 5.1 1.9

## 6 5.6 1.8

## 7 5.8 2.2

## 8 5.9 2.1

## 9 6.0 2.5

La sintaxis encadenada permite un código significativamente más fácil de leer y entender. Una observación importante es que una vez cargado *dplyr* **podemos usar la sintaxis en cadena con otros comandos de R**. Esta característica hace que *dplyr* sea un paquete que puede influir bastante en la manera de escribir código en R. Veamos un ejemplo en el que se calcula la distancia euclídea entre dos vectores:

x1 <- 1:6; x2 <- 7:12

*# Sintaxis anidada*

sqrt(sum((x1-x2)^2))

## [1] 14.7

*# Sintaxis en cadena*

(x1-x2)^2 %>% sum() %>% sqrt()

## [1] 14.7

**Añadir nuevas variables: mutate()**

Seguimos con las acciones básicas implementadas en el paquete. Veamos como crear nuevas variables que son función de las ya existentes. En este ejemplo creamos una variable que corresponde al cociente entre la anchura y la longitud del pétalo (que podría corresponder a algún aspecto de la forma del pétalo):

lirios %>%

mutate(forma = Petal.Width/Petal.Length)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species forma

## 1 5.1 3.5 1.4 0.2 setosa 0.1429

## 2 4.9 3.0 1.4 0.2 setosa 0.1429

## 3 4.7 3.2 1.3 0.2 setosa 0.1538

## 4 4.6 3.1 1.5 0.2 setosa 0.1333

## 5 5.0 3.6 1.4 0.2 setosa 0.1429

## 6 7.0 3.2 4.7 1.4 versicolor 0.2979

## 7 6.4 3.2 4.5 1.5 versicolor 0.3333

## 8 6.9 3.1 4.9 1.5 versicolor 0.3061

## 9 5.5 2.3 4.0 1.3 versicolor 0.3250

## 10 6.5 2.8 4.6 1.5 versicolor 0.3261

## 11 6.3 3.3 6.0 2.5 virginica 0.4167

## 12 5.8 2.7 5.1 1.9 virginica 0.3725

## 13 7.1 3.0 5.9 2.1 virginica 0.3559

## 14 6.3 2.9 5.6 1.8 virginica 0.3214

## 15 6.5 3.0 5.8 2.2 virginica 0.3793

**Resumir (subconjuntos de) variables: group\_by() + summarise()**

Usamos summarise() para aplicar comandos a variables. Normalmente se usa en combinación con group\_by() de manera que se calculen estadísticos para subgrupos de observaciones. En el siguiente ejemplo se calcula la media de la longitud del pétalo para los lirios de cada una de las especies:

lirios %>%

group\_by(Species) %>%

summarise(mean(Petal.Length))

## Source: local data frame [3 x 2]

##

## Species mean(Petal.Length)

## 1 setosa 1.40

## 2 versicolor 4.54

## 3 virginica 5.68

Una variación viene dada por la acción summarise\_each() en la que se consideran varias variables a la vez. Un ejemplo es el siguiente, en el que se calculan las medias de cada una de las medidas del pétalo y para cada una de las tres especies:

lirios %>%

group\_by(Species) %>%

summarise\_each(funs(mean), contains('Petal'))

## Source: local data frame [3 x 3]

##

## Species Petal.Length Petal.Width

## 1 setosa 1.40 0.20

## 2 versicolor 4.54 1.44

## 3 virginica 5.68 2.10

**Otras funciones**

Para **extraer aleatoriamente** algunas observaciones

*# Extrae 4 obs. sin reemplazamiento*

lirios %>% sample\_n(4)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 5 5.0 3.6 1.4 0.2 setosa

## 2 4.9 3.0 1.4 0.2 setosa

## 51 7.0 3.2 4.7 1.4 versicolor

## 4 4.6 3.1 1.5 0.2 setosa

*# Extrae un 25% de obs con reemplazamiento*

lirios %>% sample\_frac(0.25, rep=TRUE)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species

## 51 7.0 3.2 4.7 1.4 versicolor

## 2 4.9 3.0 1.4 0.2 setosa

## 54 5.5 2.3 4.0 1.3 versicolor

## 54.1 5.5 2.3 4.0 1.3 versicolor

Para ver la **estructura** de un fichero:

glimpse(lirios)

## Variables:

## $ Sepal.Length (dbl) 5.1, 4.9, 4.7, 4.6, 5.0, 7.0, 6.4, 6.9, 5.5, 6.5,...

## $ Sepal.Width (dbl) 3.5, 3.0, 3.2, 3.1, 3.6, 3.2, 3.2, 3.1, 2.3, 2.8,...

## $ Petal.Length (dbl) 1.4, 1.4, 1.3, 1.5, 1.4, 4.7, 4.5, 4.9, 4.0, 4.6,...

## $ Petal.Width (dbl) 0.2, 0.2, 0.2, 0.2, 0.2, 1.4, 1.5, 1.5, 1.3, 1.5,...

## $ Species (fctr) setosa, setosa, setosa, setosa, setosa, versicol...