



udp UNIVERSIDAD
DIEGO PORTALES

Curso Programación y Datos

ICO 09112

- Universidad Diego Portales
- Coordinación Programación

Temario

I. Indexar

II. Indexar condicional



udp

R, RStudio y RStudio Cloud

Indexar

Cargue en un nuevo data frame la base de datos HairEyeColor, de la siguiente forma.

```
tipo_ojos <- data.frame(HairEyeColor)
```

Para obtener el valor de una fila y columna en específico, utilice:

```
tipo_ojos[ fila_nro , columna_nro ]
```

Por ejemplo `tipo_ojos[3,4]` nos muestra lo que se encuentre en la fila 3 y columna 4.

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Si queremos obtener parte de una columna utilizamos

```
tipo_ojos[ a:b , columna_nro ]
```

Donde **a** y **b** indican la posición de inicio y final de las filas a seleccionar.

Así por ejemplo, si queremos seleccionar desde el quinto al décimo elemento de la tercera columna, hacemos

```
tipo_ojos[ 5:10 , 3 ]
```

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Si queremos obtener parte de una fila utilizamos

`tipo_ojos[fila_nro , a:b]`

Donde a y b indican la posición de inicio y final de las columnas a seleccionar

Así por ejemplo, si queremos seleccionar desde el primer al tercer elemento de la tercera fila, hacemos

`tipo_ojos[3 , 1:3]`

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Si queremos obtener parte de una fila y columna

`tipo_ojos[a:b , c:d]`

Donde a y b indican la posición de inicio y final de las filas y c y d indican la posición inicial y final de las columnas a seleccionar

Así por ejemplo, si queremos seleccionar desde la cuarta a la octava fila y desde la tercera a cuarta columna

`tipo_ojos[4:8 , 3:4]`

	▲	Hair ▲	Eye ▲	Sex ▲	Freq ▲
1		Black	Brown	Male	32
2		Brown	Brown	Male	53
3		Red	Brown	Male	10
4		Blond	Brown	Male	3
5		Black	Blue	Male	11
6		Brown	Blue	Male	50
7		Red	Blue	Male	10
8		Blond	Blue	Male	30
9		Black	Hazel	Male	10
10		Brown	Hazel	Male	25
11		Red	Hazel	Male	7
12		Blond	Hazel	Male	5
13		Black	Green	Male	3
14		Brown	Green	Male	15
15		Red	Green	Male	7
16		Blond	Green	Male	8

Si queremos obtener una fila entera

`tipo_ojos[fila_nro ,]`

Así por ejemplo, si queremos seleccionar la fila 10

`tipo_ojos[10 ,]`

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Si queremos obtener una columna entera

`tipo_ojos[, columna_nro]`

Así por ejemplo, si queremos seleccionar la columna 4

`tipo_ojos[, 4]`

Otra forma, dado que es un dataframe, para conseguir trabajar con una columna podemos llamarla así

`tipo_ojos$nombre_columna`

En el ejemplo anterior, el equivalente es

`tipo_ojos$Freq`

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Filtrar con sentencias lógicas

Usaremos las sentencias lógicas para obtener los datos que queramos de nuestro dataframe.

Veamos el siguiente ejemplo: Queremos obtener el color de ojos de únicamente los hombres de nuestro dataset

```
tipo_ojos[ tipo_ojos[, 3]=="Male", "Eye"]
```

Estamos usando la sentencia lógica `==` para obtener las filas de la columna 3 que digan "Male" y luego la indexamos con la columna "Eye"

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Filtrar con sentencias lógicas

Suponga ahora que quiere obtener de las dos primeras columnas, aquellos datos donde la frecuencia es mayor a 50.

Se tiene:

```
tipo_ojos[ tipo_ojos[,4]>50, 1:2]
```

Estamos usando la sentencia lógica “>” para seleccionar aquellas filas donde la frecuencia sea mayor a 50 y en el campo de columna indicamos que son de la primera a la segunda

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Indexar o filtrar por más de una condición

Se tienen los operadores lógicos **AND & OR** |
o el **NOT** !

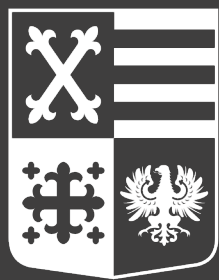
Suponga que quiere el color de cabello de aquellas personas que son hombres y además su color de ojos es azul, se tiene entonces

```
tipo_ojos[(tipo_ojos[,2]=="Blue") &  
(tipo_ojos[,3]=="Male"),"Hair"]
```

	Hair	Eye	Sex	Freq
1	Black	Brown	Male	32
2	Brown	Brown	Male	53
3	Red	Brown	Male	10
4	Blond	Brown	Male	3
5	Black	Blue	Male	11
6	Brown	Blue	Male	50
7	Red	Blue	Male	10
8	Blond	Blue	Male	30
9	Black	Hazel	Male	10
10	Brown	Hazel	Male	25
11	Red	Hazel	Male	7
12	Blond	Hazel	Male	5
13	Black	Green	Male	3
14	Brown	Green	Male	15
15	Red	Green	Male	7
16	Blond	Green	Male	8

Trabajo en clases

- Abra el dataset “trees” en un dataframe llamado “arbol”
- Obtenga las alturas de aquellos árboles que tienen volumen mayor a 25.
- Obtengamos el promedio de las alturas de los árboles que tienen volumen mayor a 25, usando la función `mean()` donde entre paréntesis coloque el código anterior
- La función `length()` nos indica cuantos elementos tiene el argumento. ¿Cuántos árboles tienen diámetro menor a 13 y volumen mayor a 20?



udp UNIVERSIDAD
DIEGO PORTALES