

select, rename (dplyr)

Select() permette di selezionare, eventualmente rinominandole, le colonne di un dataframe; rename() serve solo a rinominarle.

Vedi anche [Indicare e scegliere le variabili](#); [dplyr_across](#)

select

Con select() si possono scegliere le variabili di un dataset, operando in pratica un [subset](#).

```
library(tidyverse)          # carica tutti i pacchetti
data("ChickWeight")
```

La funzione richiede come primo argomento il nome del dataset, e, a seguire, il nome delle variabili da selezionare (di cui si può anche cambiare l'ordine):

```
names(ChickWeight)
```

```
[1] "weight" "Time"   "Chick"  "Diet"
```

```
select(ChickWeight, Chick, weight)
```

oppure

```
ChickWeight %>% select(Chick, weight)
```

oppure

```
ChickWeight %>% select(c(3,1))
```

o anche

```
ChickWeight %>% select(3,1)
```

	Chick	weight
1	1	42
2	1	51
3	1	59
4	1	64
...		

select_if

Una variazione molto comoda di questa funzione è `select_if()`, che permette di selezionare le variabili in base a condizioni. Ad esempio, per selezionare solo le variabili numeriche di un dataset:

```
starwars %>%  
  select_if(is.numeric)
```

```
# A tibble: 87 x 3  
  height mass birth_year  
  <int> <dbl>      <dbl>  
1    172    77         19  
2    167    75        112  
3     96    32         33  
4    202   136        41.9  
5    150    49         19  
6    178   120         52  
7    165    75         47  
8     97    32         NA  
9    183    84         24  
10   182    77         57  
# ... with 77 more rows
```

Equivale a `select(where())`

Per eliminare le colonne di un dataset che contengono solo valori mancanti, possiamo usare questa funzione, in questo modo:

```
df %>%  
  select_if(~sum(!is.na(.x)) > 0)
```

oppure

```
df %>%  
  select(where(~sum(!is.na(.x)) > 0))
```

rename

Con la funzione `select()` possiamo anche rinominare le variabili:

```
ChickWeight %>%  
  select(Peso = weight,
```

```
Dieta = Diet)
```

	Peso	Dieta
1	42	1
2	51	1
3	59	1
4	64	1
...		

Per rinominare le variabili, c'è anche la funzione `rename()`.

```
mydata <- ChickWeight %>%
  rename(Peso = weight,
         Tempo = Time,
         Pollo = Chick,
         Dieta = Diet)
```

Le righe precedenti equivalgono a:

```
mydata <- rename(ChickWeight,
                 Peso = weight,
                 Tempo = Time,
                 Pollo = Chick,
                 Dieta = Diet)
```

Controlliamo i risultati usando `as.tibble()` (vedi [Tibble](#)), che mostra le prime righe, le intestazioni di colonna e alcune informazioni sulle variabili:

```
mydata %>%
  as_tibble()
```

```
# A tibble: 578 x 4
   Peso Tempo Pollo Dieta
* <dbl> <dbl> <ord> <fct>
1    42     0 1     1
2    51     2 1     1
3    59     4 1     1
4    64     6 1     1
5    76     8 1     1
6    93    10 1     1
7   106    12 1     1
8   125    14 1     1
9   149    16 1     1
10  171    18 1     1
# ... with 568 more rows
```

Quando le variabili da rinominare sono numerose, può forse essere più veloce usare `names()` (si veda [Rinominare le variabili](#)):

```
names(ChickWeight) <- c("Peso", "Tempo", "Pollo", "Dieta")
```

Vedi: [RDocumentation](#)

[Gestione dei dati](#), [Tidyverse](#), [funzioni](#)

From:

<https://www.agnesevardanega.eu/wiki/> - **Ricerca Sociale con R**

Permanent link:

https://www.agnesevardanega.eu/wiki/r/tidyverse/dplyr_select?rev=1658326711

Last update: **20/07/2022 14:18**

