

mutate, transmute (dplyr)

`mutate()` aggiunge nuove variabili ad un dataset, indicandone il nome e il valore, in questo modo.

```
mutate(dataset, nuova_variabile = ...)
```

oppure

```
dataset %>%  
  mutate(nuova_variabile = ...)
```

Ricodifica con mutate()

La funzione può essere usata per aggiungere un vettore di valori, o un vettore (nuova variabile) risultato della trasformazione di un'altra variabile. È dunque una funzione fondamentale per le operazioni di ricodifica. Ad esempio:

```
library(tidyverse)
```

```
starwars %>%  
  # selezioniamo due sole variabili per visualizzare meglio i  
  # risultati  
  select(name, mass) %>%  
  # creiamo la variabile "mass_norm"  
  mutate(mass_norm = mass / mean(mass, na.rm = TRUE))
```

```
# A tibble: 87 x 3  
  name          mass mass_norm  
  <chr>         <dbl>   <dbl>  
1 Luke Skywalker    77    0.791  
2 C-3P0             75    0.771  
3 R2-D2             32    0.329  
4 Darth Vader     136    1.40  
5 Leia Organa      49    0.504  
6 Owen Lars       120    1.23  
7 Beru Whitesun lars  75    0.771  
8 R5-D4             32    0.329  
9 Biggs Darklighter  84    0.863  
10 Obi-Wan Kenobi    77    0.791  
# ... with 77 more rows
```

Oppure, per riprendere l'esempio in [Calcolare una nuova variabile](#), creiamo la variabile Totale (della popolazione) sommando Totale.Maschi e Totale.Femmine, nel dataset Istat16¹⁾:

```
library(tidyverse)
mutate(Istat16, Totale = Totale.Maschi + Totale.Femmine)
```

Applicando altre altre funzioni

Generalmente, la ricodifica richiede l'uso di altre funzioni, all'interno di `mutate()`. In questo caso, si scriverà:

```
dataset %>%
  mutate(nuova_variabile = fct_recode(vecchia_variabile, ...))
```

e non

```
dataset %>%
  fct_recode(vecchia_variabile, ...)
```

Quindi, ad esempio:

```
starwars <- starwars %>%
  mutate(sex = as.factor(sex))
```

equivale a:

```
starwars$sex <- as.factor(starwars$sex)
```

Mentre è scorretto:

```
starwars %>%
  as.factor(sex)
```

Con variabile di raggruppamento

I valori della nuova variabile possono essere specificati in base ad una variabile di raggruppamento: ad esempio il valore normalizzato della massa può essere calcolato nei gruppi definiti dalla variabile "homeworld", usando `group_by()`

```
starwars %>%
  select(name, mass, homeworld) %>%
  group_by(homeworld) %>%
```

```
mutate(mass_norm = mass / mean(mass, na.rm = TRUE))
```

```
# A tibble: 87 x 4
# Groups:   homeworld [49]
  name          mass homeworld mass_norm
  <chr>         <dbl> <chr>         <dbl>
1 Luke Skywalker    77 Tatooine      0.902
2 C-3P0             75 Tatooine      0.878
3 R2-D2             32 Naboo        0.499
4 Darth Vader      136 Tatooine      1.59
5 Leia Organa       49 Alderaan     0.766
6 Owen Lars        120 Tatooine      1.41
7 Beru Whitesun lars 75 Tatooine      0.878
8 R5-D4             32 Tatooine      0.375
9 Biggs Darklighter 84 Tatooine      0.984
10 Obi-Wan Kenobi    77 Stewjon       1
# ... with 77 more rows
```

transmute()

Con `mutate()`, la variabile viene direttamente aggiunta al dataset.

Con **`transmute()`**, la nuova variabile sostituisce invece la o le variabili precedenti. Si consideri l'esempio:

[es_mutate.R](#)

```
# costruiamo le variabili standardizzate
mutate(cars, zspeed = as.numeric(scale(speed)),
       zdists = as.numeric(scale(dist))) %>%
  as_tibble() # per
visualizzare
```

```
# A tibble: 50 x 4
  speed dist zspeed zdists
  <dbl> <dbl> <dbl> <dbl>
1     4     2 -2.16 -2.16
2     4    10 -2.16 -2.16
3     7     4 -1.59 -1.59
4     7    22 -1.59 -1.59
5     8    16 -1.40 -1.40
6     9    10 -1.21 -1.21
7    10    18 -1.02 -1.02
```

```
8      10      26 -1.02  -1.02
9      10      34 -1.02  -1.02
10     11      17 -0.832 -0.832
# ... with 40 more rows
```

es_transmute.R

```
library(tidyverse)

# nuovo file
zcars <- transmute(cars, zspeed = as.numeric(scale(speed)),
                  zdist = as.numeric(scale(speed)))
as_tibble(zcars)
```

```
# A tibble: 50 x 2
  zspeed  zdist
  <dbl>  <dbl>
1 -2.16  -2.16
2 -2.16  -2.16
3 -1.59  -1.59
4 -1.59  -1.59
5 -1.40  -1.40
6 -1.21  -1.21
7 -1.02  -1.02
8 -1.02  -1.02
9 -1.02  -1.02
10 -0.832 -0.832
# ... with 40 more rows
```

Vedi [RDocumentation: mutate](#)

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

¹⁾

Scaricabile da Bitbucket: <https://bitbucket.org/agnesevardanega/rwiki/downloads/>

From:

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

Permanent link:

https://www.agnesevardanega.eu/wiki/r/tidyverse/dplyr_mutate?rev=1670000943

Last update: **02/12/2022 17:09**

