

ggplot

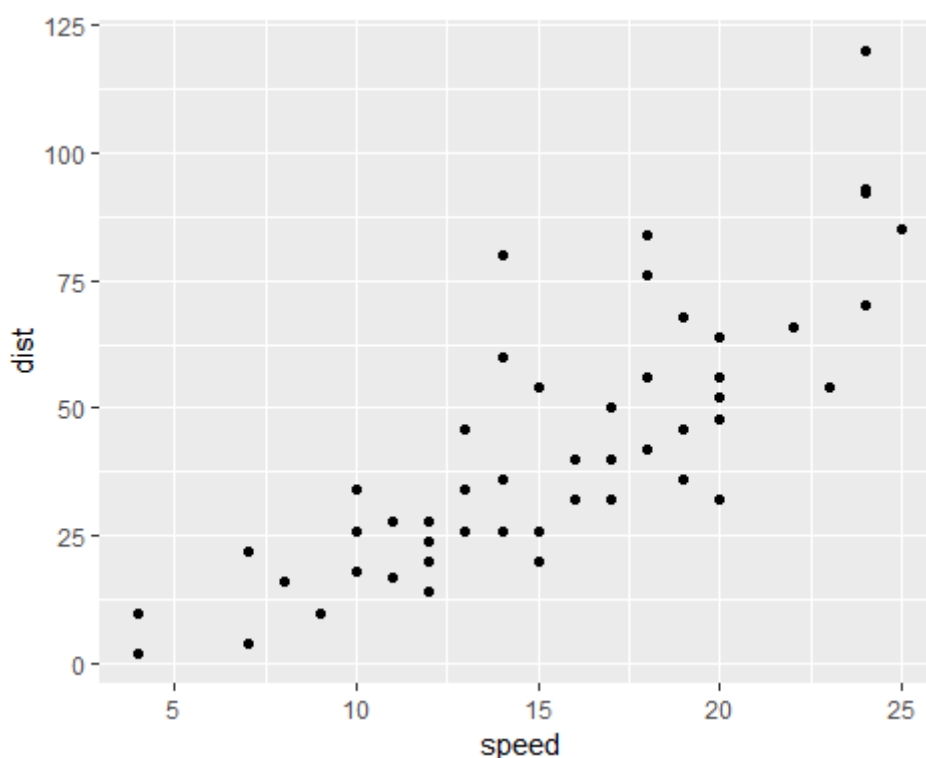
Fondamentalmente, per creare un grafico con ggplot2, dobbiamo indicare:

- un dataset;
- una mappatura delle variabili da rappresentare (``mapping = aes()``);
- una geometria (geom), ovvero il tipo di rappresentazione desiderata.

Mappatura (aes) e rappresentazioni geometriche (geom)

```
library(tidyverse)
```

```
cars %>%  
  ggplot(aes(x = speed, y = dist)) +      # mappatura delle  
  geom_point()                            # geometria "punti" per  
  # rappresentarle
```



`mapping` = può essere omesso.

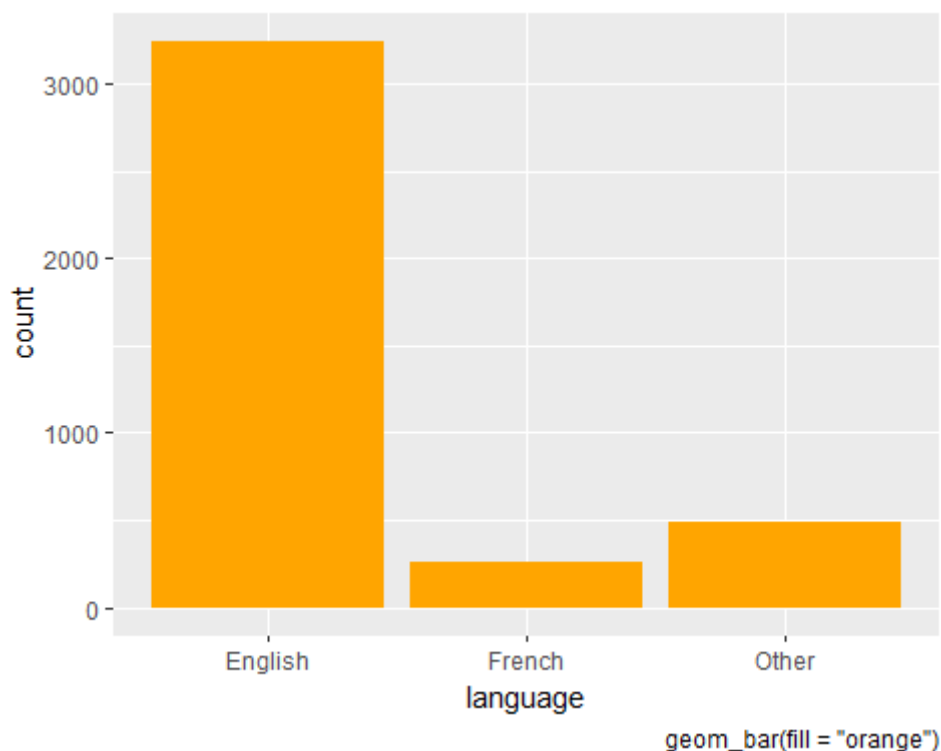
Possiamo ottenere lo stesso grafico scrivendo:

```
cars %>%
```

```
ggplot() +
  aes(x = speed, y = dist) +           # aes come
  comando a parte
  geom_point()

cars %>%
  ggplot() +
  geom_point(aes(x = speed, y = dist) ) # aes in geom
```

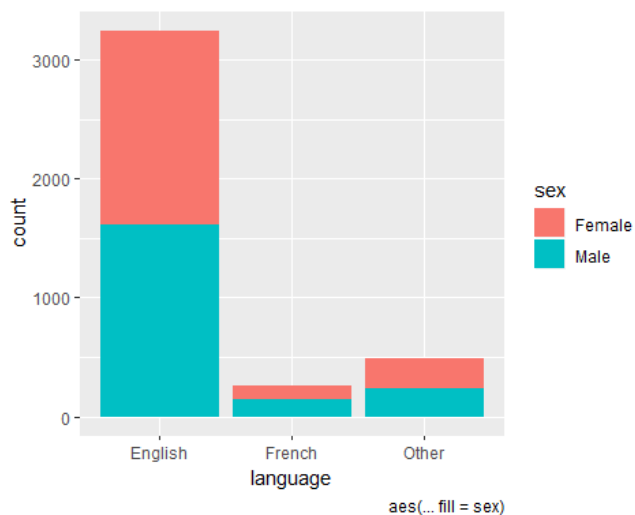
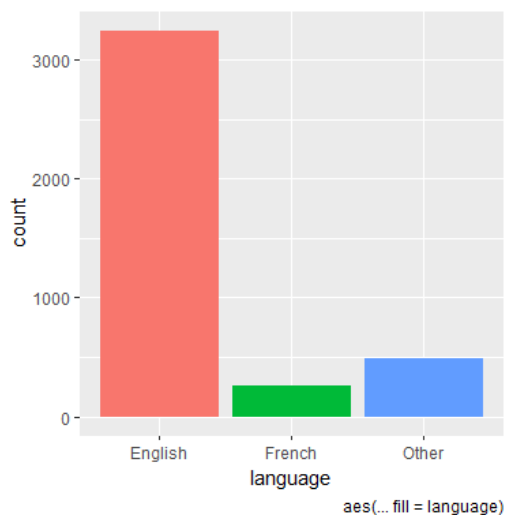
Vediamo la differenza fra mapping = aes e geom:



```
library(carData)

SLID %>%
  na.omit() %>%                         # ometto i casi
  mancanti
  ggplot(aes(x = language)) +
  geom_bar(fill = "orange")             # indico un
  colore
```

In geom() è possibile indicare un colore o un vettore di colori per le forme rappresentate, come in [graphics^{1\)}](#).



```
SLID %>%
  na.omit() %>%
  ggplot(aes(x = language, fill = language)) +           # mappa una
  geom_bar()                                             variabile
```

```
SLID %>%
  na.omit() %>%
  ggplot() +
  geom_bar(aes(x = language, fill = sex))                # mappa
  un'altra variabile
```

Con aes, invece, indico una variabile da “mappare”, per rappresentarla attraverso il colore: quindi anche una variabile diversa, come nel grafico a destra.

Per la variabile o le variabili mappate in questo modo viene prodotta automaticamente una legenda.

Script di esempio

E' possibile scaricare ed eseguire lo script dell'esempio:

[ggplot_base.R](#)

```
library(tidyverse)

# grafico base
cars %>%
  ggplot(aes(x = speed, y = dist)) +
  geom_point()

library(carData)
```

```

# differenza fra geom e aes
SLID %>%
  na.omit() %>%                                # ometto i
  casi mancanti
  ggplot(aes(x = language)) +
  geom_bar(fill = "orange")                     # indico un
  colore

SLID %>%
  na.omit() %>%
  ggplot(aes(x = language, fill = language)) +   # mappo una
  variabile
  geom_bar()

SLID %>%
  na.omit() %>%
  ggplot() +
  geom_bar(aes(x = language, fill = sex))         # mappo
  un'altra variabile

```

Grafici, ggplot2, Tidyverse

1)

In questo contesto, il **piping** (`%>%` - ctrl + alt + M in RStudio) è particolarmente comodo e semplice da usare. Non solo non è più necessario indicare il set di dati nei comandi successivi, ma le trasformazioni effettuate sul set di dati - come ad esempio eliminare i casi con valori mancanti - non sono permanenti. Il vantaggio sta nel fatto che non dobbiamo creare un subset per rappresentare i dati che ci interessano.

From:

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

Permanent link:

<https://www.agnesevardanega.eu/wiki/r/ggplot2/ggplot>

Last update: **15/10/2025 08:02**

