

# Rappresentare i dati geografici

## Importare i dati delle mappe

Utilizzeremo i pacchetti del tidyverse.

```
library(tidyverse)
```

Come prima cosa, servono i dati dei poligoni: nell'esempio usiamo quelli scaricati dal sito Istat, decompressi in una cartella (qui, in una sotto-cartella *dati* della cartella di lavoro).

Installiamo il pacchetto per la gestione dei dati spaziali:

```
install.packages("sf")  
library(sf)
```

Gli shapefiles possono essere importati con la funzione `st_read()` e indicando il nome della cartella (e non dei files: `Reg01012019_g`):

```
#leggere shapefiles (DATI ISTAT)  
reg2019 <- st_read("dati/Reg01012019_g")
```

Il file viene importato in un formato (*simple feature*), che è anche di classe dataframe:

```
head(reg2019)
```

Simple feature collection with 6 features and 5 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: 313279.3 ymin: 4847853 xmax: 819674.3 ymax: 5220292

Projected CRS: WGS 84 / UTM zone 32N

|   | COD_RIP | COD_REG | DEN_REG             | Shape_Leng | Shape_Area  |
|---|---------|---------|---------------------|------------|-------------|
| 1 | 1       | 1       | Piemonte            | 1236787.2  | 25393891060 |
| 2 | 1       | 2       | Valle d'Aosta       | 310968.1   | 3258837561  |
| 3 | 1       | 3       | Lombardia           | 1410222.9  | 23862316681 |
| 4 | 2       | 4       | Trentino-Alto Adige | 800893.7   | 13607548167 |
| 5 | 2       | 5       | Veneto              | 1054697.6  | 18343532672 |
| 6 | 1       | 7       | Liguria             | 819640.4   | 5414544646  |

geometry

```
1 MULTIPOLYGON (((457749.5 51...  
2 MULTIPOLYGON (((390652.6 50...  
3 MULTIPOLYGON (((595652.4 51...
```

```
4 MULTIPOLYGON (((743267.7 52...
5 MULTIPOLYGON (((768124 5175...
6 MULTIPOLYGON (((459010.2 49...
```

## Le funzioni per i grafici

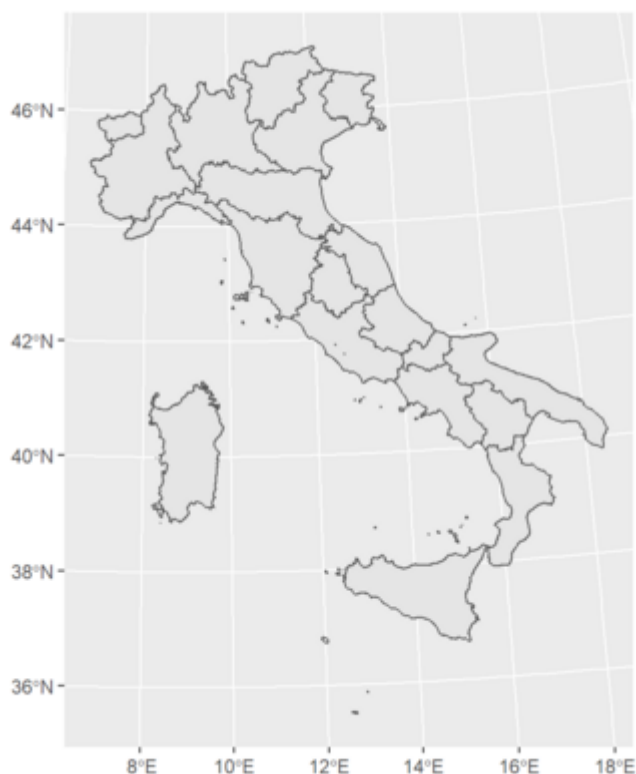
Il pacchetto comprende funzioni per rappresentare questi dati con ggplot:

### geom\_sf: le aree

```
ggplot(data = reg2019) +
  geom_sf()
```

o

```
reg2019 %>%
  ggplot() +
  geom_sf()
```



È dunque possibile usare tutte le funzioni di ggplot2. Qui, ad esempio, coloriamo le aree delle regioni in base alla variabile `DEN_REG` (denominazione della regione: `aes(fill = DEN_REG)`), eliminiamo la legenda, e usiamo il tema “vuoto” (`theme_void()`; vedi [i temi predefiniti](#)):

```
ggplot(reg2019) +  
  geom_sf(aes(fill = DEN_REG),  
          show.legend = FALSE) +  
  theme_void()
```



## geom\_sf\_text: le etichette

Per aggiungere etichette alle mappe, usiamo le funzioni `geom_sf_text()` e `geom_sf_label()`, analoghe a `geom_text()` e `geom_label()` (Vedi la voce: [Etichette dei valori](#)).

Usiamo ad esempio il nome della regione (`aes(label = DEN_REG)`), e riduciamo la dimensione delle etichette (`size = 3`):

```
ggplot(reg2019) +  
  geom_sf() +  
  geom_sf_text(aes(label = DEN_REG),  
              size = 3) +  
  theme_void()
```



## Mappe tematiche

Per rappresentare nelle mappe i valori di altre variabili, basterà aggiungere i dati al dataframe dei dati spaziali.

Ad esempio, avendo un dataframe di dati su base regionale, bisognerà prima di tutto accertarsi che esso preveda una variabile comune con il dataframe dei poligoni (COD\_REG = codice regione):

```
head(df)
```

```
# A tibble: 6 x 2
  COD_REG totale_casi
  <dbl>      <dbl>
1     13      12543
2     17       2788
3     18       6092
4     15      69613
5      8      62914
6      6      12264
```

Uniamo i due dataframe con `right_join` (vedi [dplyr\\_join](#)):

```
right_join(reg2019, df, by = "COD_REG") %>%
  head()
```

Simple feature collection with 6 features and 6 fields

geometry type: MULTIPOLYGON

dimension: XY

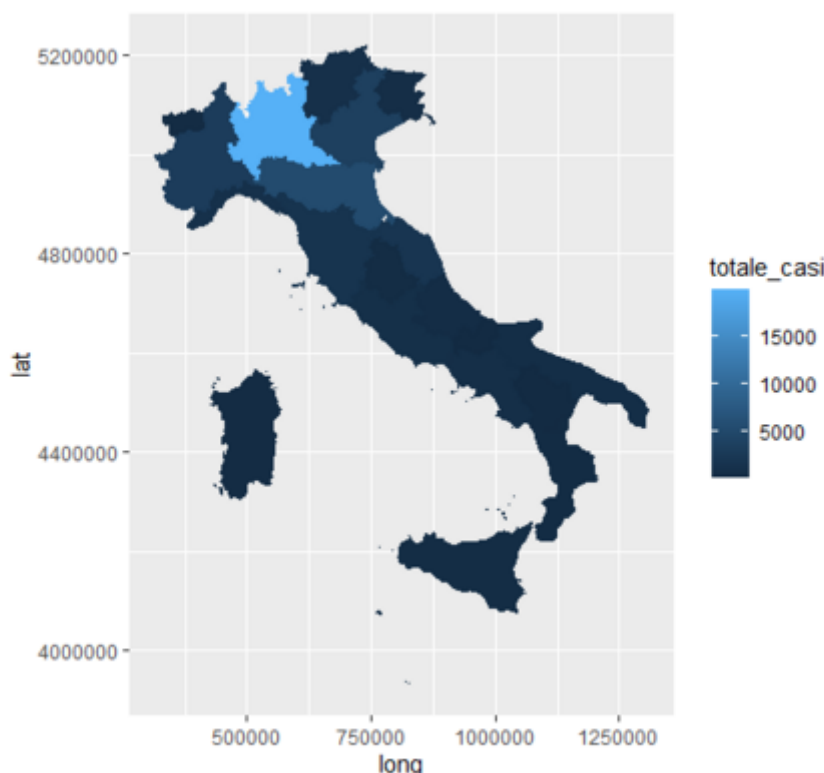
bbox: xmin: 313279.3 ymin: 4879338 xmax: 819674.3 ymax: 5220292

projected CRS: WGS 84 / UTM zone 32N

|             | COD_RIP      | COD_REG          | DEN_REG             | Shape_Leng | Shape_Area  |
|-------------|--------------|------------------|---------------------|------------|-------------|
| totale_casi |              |                  |                     |            |             |
| 1           | 1            | 1                | Piemonte            | 1236787.2  | 25393891060 |
| 81409       | MULTIPOLYGON | ((457749.5 51... |                     |            |             |
| 2           | 1            | 2                | Valle d'Aosta       | 310968.1   | 3258837561  |
| 3720        | MULTIPOLYGON | ((390652.6 50... |                     |            |             |
| 3           | 1            | 3                | Lombardia           | 1410222.9  | 23862316681 |
| 224191      | MULTIPOLYGON | ((595652.4 51... |                     |            |             |
| 4           | 2            | 4                | Trentino-Alto Adige | 800893.7   | 13607548167 |
| 10097       | MULTIPOLYGON | ((743267.7 52... |                     |            |             |
| 5           | 2            | 4                | Trentino-Alto Adige | 800893.7   | 13607548167 |
| 9872        | MULTIPOLYGON | ((743267.7 52... |                     |            |             |
| 6           | 2            | 5                | Veneto              | 1054697.6  | 18343532672 |
| 65531       | MULTIPOLYGON | ((768124 5175... |                     |            |             |

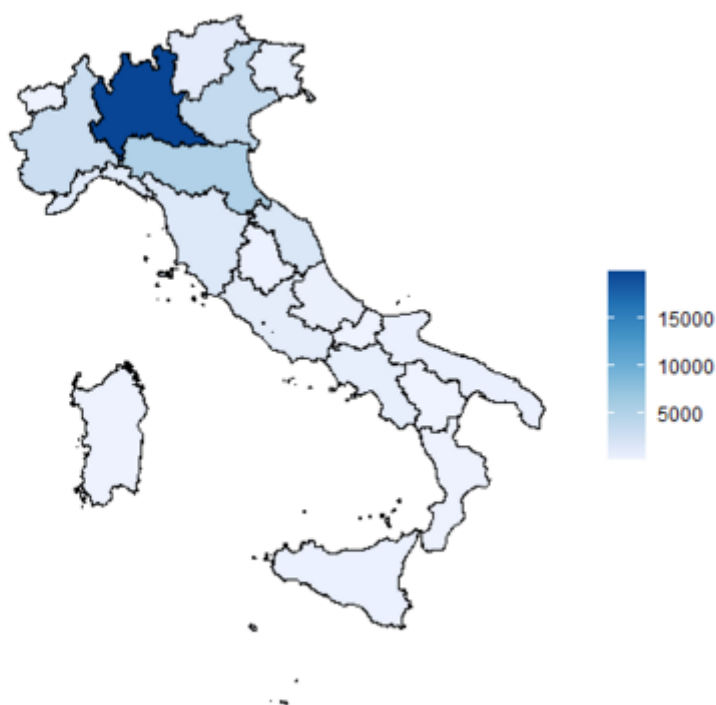
Ora procediamo alla costruzione della mappa:

```
right_join(reg2019, df, by = "COD_REG") %>%
  ggplot(aes(fill = totale_casi)) +
  geom_sf()
```



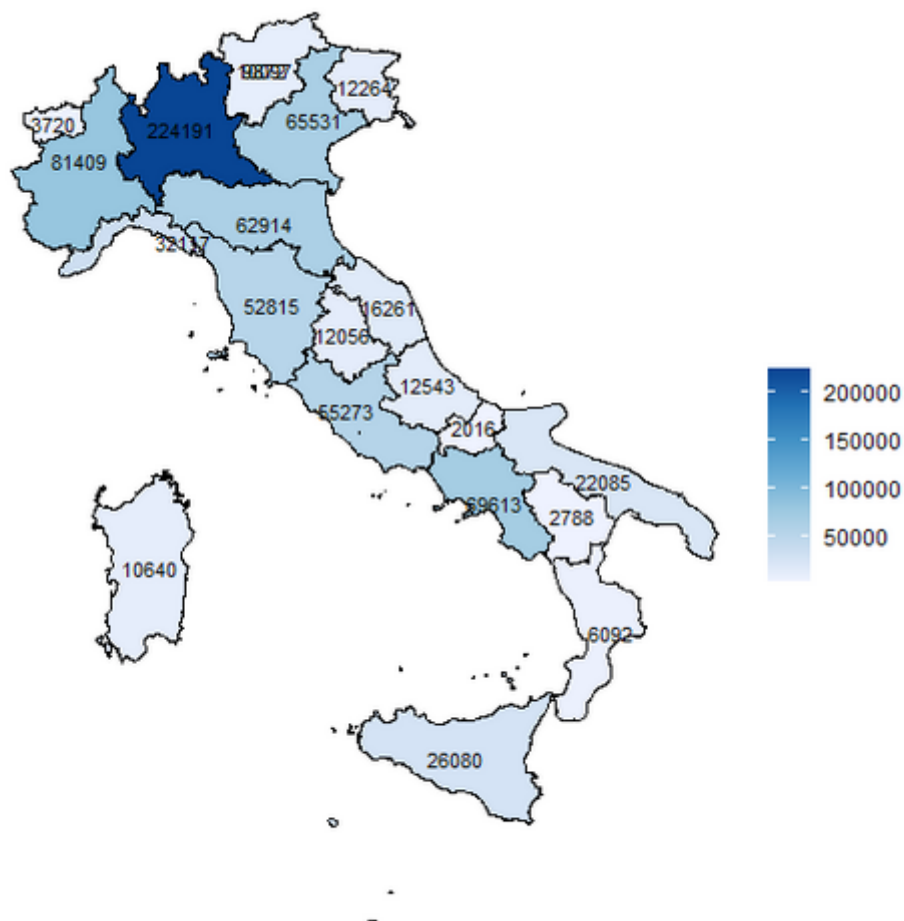
Miglioriamo la visualizzazione, ad esempio:

```
right_join(reg2019, df, by = "COD_REG") %>%
  ggplot(aes(fill = totale_casi)) +
  geom_sf(col = "black" # bordo dei poligoni
  ) +
  # tema vuoto
  theme_void() +
  # scala dei colori
  scale_fill_distiller(direction = 1) +
  # eliminare il titolo della legenda
  theme(legend.title=element_blank())
```



Infine, aggiungiamo le etichette:

```
right_join(reg2019, df, by = "COD_REG") %>%
  ggplot(aes(fill = totale_casi)) +
  geom_sf(col = "black") +
  theme_void() +
  scale_fill_distiller(direction = 1) +
  theme(legend.title=element_blank()) +
  # etichette
  geom_sf_text(aes(label = totale_casi),
               size = 3)
```



Vedi:

- <https://r-spatial.github.io/sf/index.html>
- <https://yutannihilation.github.io/ggspatial/index.html>

Grafici, ggplot2, Tidyverse, Mappe

From:

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

Permanent link:

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

Last update: **05/09/2025 16:48**

