GGPlot Advanced

Marco Torchiano

Version 1.0.0 - May 2021

License 

Joining data frames

Joining data frames

When related data are stored in distinct data frames, it is possible to merge them into a single data frame.

The `..._join()` methods take two data frames and produces a new one

- including all columns from the two data frames
- common (merged by) columns appear only once
- rows from the data frames are matched by
    - common columns
    - columns specified with parameter `by`

Joining data frames

- `inner_join()` : includes only matching rows
- `left_join()` : includes all rows from left df + matching rows
- `right_join()` : includes all rows from right df + matching rows
- `full_join()` : includes all rows from both dfs

Example data frames

Two data frames:

- `df1`: `id` and `name`
- `df2`: `id` and `day`

id	name	<- df1 - - - - - - - - - - - - - - - - - - df2 ->	id	day
100	Donald		101	Mon
101	Huey		102	Tue
102	Dewey		103	Wed
103	Louie		104	Thu

Inner join

```
df1 %>% inner_join(df2, by="id") %>% knitr::kable()
```

id	name	day
101	Huey	Mon
102	Dewey	Tue
103	Louie	Wed

- Rows from dataframes are matched by `id`
- Rows in either data frame with no corresponding `id` in the other are discarded.

Left join

```
df1 %>% left_join(df2, by="id") %>% knitr::kable()
```

id	name	day
100	Donald	NA
101	Huey	Mon
102	Dewey	Tue
103	Louie	Wed

- Rows from dataframes are matched by `id`
- All rows in *left* data frame are included
- Rows in *right* data frame with no corresponding `id` in the *left* are discarded.

Right join

```
df1 %>% right_join(df2, by="id") %>% knitr::kable()
```

id	name	day
101	Huey	Mon
102	Dewey	Tue
103	Louie	Wed
104	NA	Thu

- Rows from dataframes are matched by `id`
- All rows in *right* data frame are included
- Rows in *left* data frame with no corresponding `id` in the *right* are discarded.

Full join

```
df1 %>% full_join(df2, by="id") %>% knitr::kable()
```

id	name	day
100	Donald	NA
101	Huey	Mon
102	Dewey	Tue
103	Louie	Wed
104	NA	Thu

- Rows from dataframes are matched by `id`
- All rows from both data frames are included

Maps

Maps

There are several packages in R that allow drawing maps:

- ggplot2 using `geom_sf()`
- mapview interactive web-oriented maps
- leaflet based on the leaflet Javascript libray

Shape profiles

Often when showing maps we deal with shapes representing geografic regions.

Geografic shapes are often shared using the *Shapefile* file format.

- Usually consist of a main `.shp` file plus `.dbf`, `.prj`, `.shx` files

Shapefile sources

Often available online for official boundaries

- Italian administrative boundaries published by ISTAT
  - https://www.istat.it/it/archivio/222527
- Worldwide free spatial data at DIVA-GIS
  - https://diva-gis.org/Data

Simple format

Vector data is often encoded (internally) using the “simple features” standard

In R it is a dataframe containing a column named `geometry`

- Library `sf` can be used to read and manipulate
- `geom_sf()` in ggplot2 draws the layer
- `coord_sf()` predefined geographical coordinate system

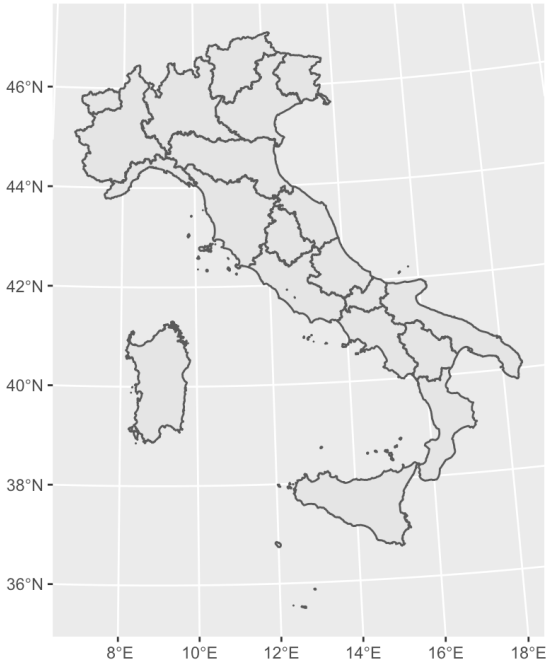
Load simple features from shapefile

```
it <- read_sf("Reg01012021_g/Reg01012021_g_WGS84.shp")
knitr::kable(head(it[,3:6],4))
```

DEN_REG	Shape_Leng	Shape_Area	geometry
Piemonte	1235512.1	25393901117	MULTIPOLYGON (((457749.5 51…
Valle d’Aosta	310968.1	3258837561	MULTIPOLYGON (((390652.6 50…
Lombardia	1410223.0	23862315006	MULTIPOLYGON (((485536.4 49…
Trentino-Alto Adige	800893.7	13607548167	MULTIPOLYGON (((743267.7 52…

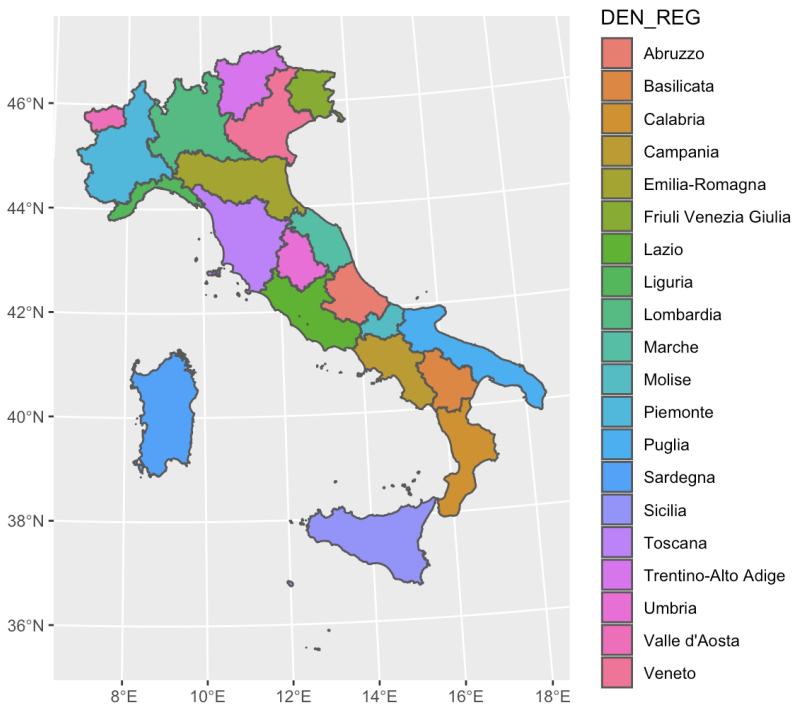
Plot simple features

```
ggplot(it,aes(geometry=geometry))+geom_sf()
```



Aesthetics of sf

```
ggplot(it,aes(fill=DEN_REG))+geom_sf()
```



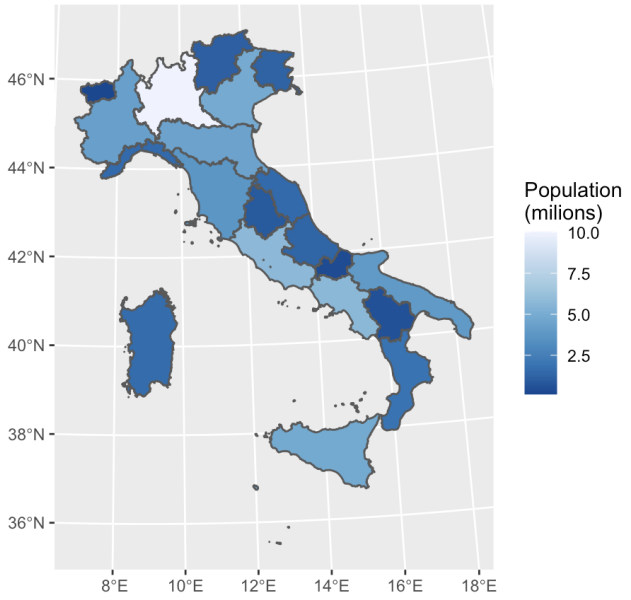
Italian Population per Region

Source: http://dati.istat.it/Index.aspx?QueryId=18460#

Territorio	maschi	femmine
Piemonte	2095058	2216159
Valle d'Aosta	61121	63913
Liguria	730371	794455
Lombardia	4912375	5115227
Trentino-Alto Adige	531506	546563
Veneto	2389717	2489416
Friuli Venezia Giulia	586719	619497
Emilia-Romagna	2173781	2290338

Merging sf with df

```
it %>% inner_join(pop_it,by=c("DEN_REG"="Territorio")) %>%
ggplot(aes(fill=(maschi+femmine)/1000000))+geom_sf()+
  scale_fill_distiller(name="Population\n(milions)")
```



Combining plots

Combining plots

Plots can be combined using the library `patchwork`

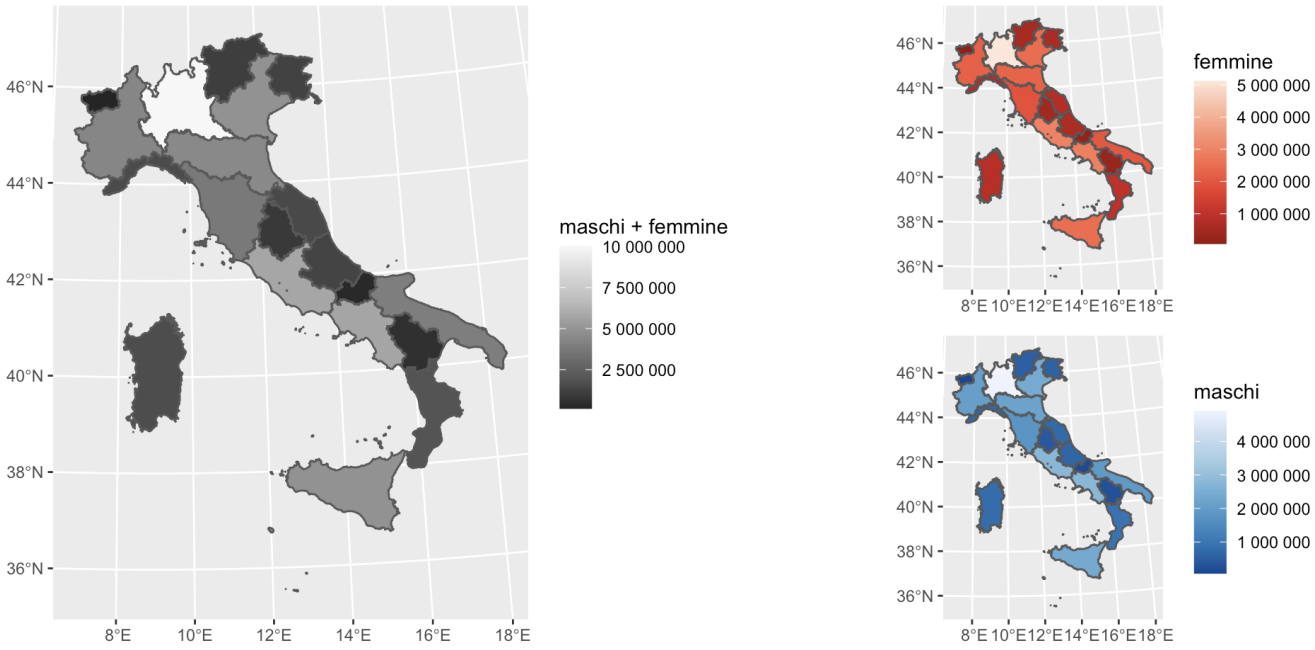
It works by combining *ggplot2* objects with operators:

- `g1 + g2` : places the plots side by side
- `g1 / g2` : places the plots one over the other
- `(` and `)`: groups plots

Merging sf with df

```
md <- it %>% inner_join(pop_it,by=c("DEN_REG"="Territorio"
pfm <- ggplot(md, aes(fill=maschi+femmine))+geom_sf()+
  scale_fill_distiller(palette="Greys",
                       labels=scales::label_number())
pf <- ggplot(md, aes(fill=femmine))+geom_sf()+
  scale_fill_distiller(palette="Reds",
                       labels=scales::label_number())
pm <- ggplot(md, aes(fill=maschi))+geom_sf()+
  scale_fill_distiller(palette="Blues",
                       labels=scales::label_number())
pfm + ( pf / pm )
```


Compising plots



References

- Nico Hahn, Making Maps with R
  - https://bookdown.org/nicohahn/making_maps_with_r5/docs/introduction.html
- Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen. “ggplot2: Elegant Graphics for Data Analysis”, in-progress
  - https://ggplot2-book.org/