



# Plotting Corona Virus Data

## Overview

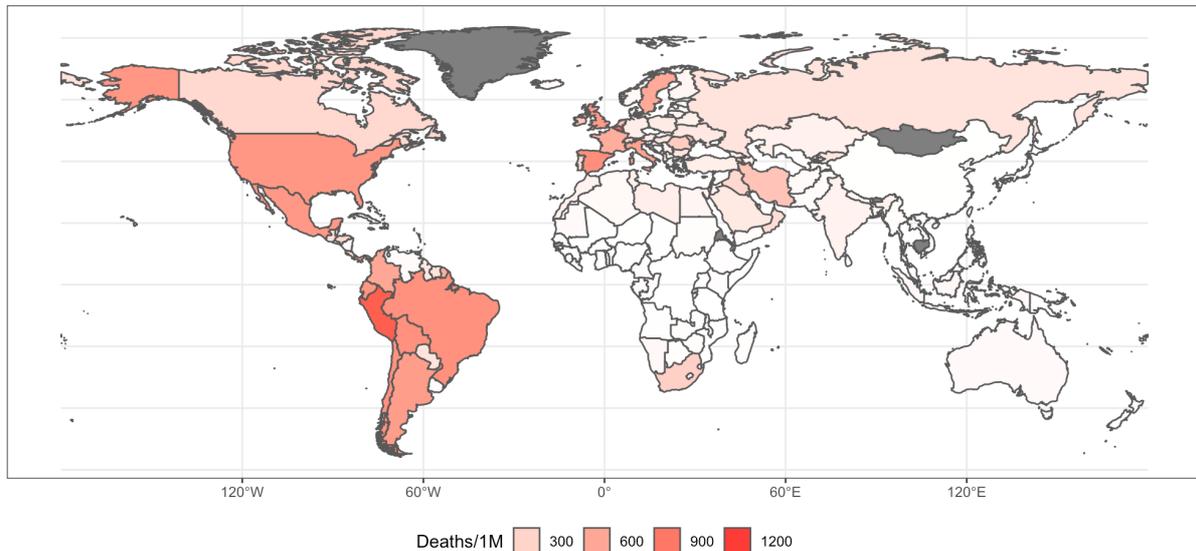
After following this guide you will be able to download the latest COVID 19 data from the web and plot it on both two-dimensional and three-dimensional world maps.

Our goal is to get something that looks like this. The video tutorial can be found here

[https://www.youtube.com/watch?v=ruYBUwP\\_RC4&feature=youtu.be](https://www.youtube.com/watch?v=ruYBUwP_RC4&feature=youtu.be)

Covid Deaths per 1 Million People

10/30/2020



<https://www.worldometers.info/coronavirus/>

## Download the Data

In order to get the data we will work with, you can either download the file I'll provide which is updated as of October 30th 2020, or you can get the most updated data by using the web scraping tool you prefer.

I suggest using *Data Miner*, an open source extension of Google Chrome. You can find the extension at this link

<https://data-miner.io/>.

Then, visit the following page <https://www.worldometers.info/coronavirus/>. This website contains a table full of data, select it and right click on it. Then select 'get data' and download it as a csv file.

I recommend doing some cleaning on the data from excel. Change the Country/Other column name to just **Country** and the cases and death per million column names to **Cases1M** and **Deaths1M**. It will make our life simpler.

Now import the data in R.

## Install required packages and get going

We will need the following R packages to proceed. Let's install them.

```
install.packages("ggplot2", "ggspatial", "sf", "rnaturalearth", "rnaturalearthdata")
```

Then, call the libraries.

```
library("ggplot2")  
library("sf")
```

```
library("rnaturalearth")
library("rnaturalearthdata")
```

Set the theme to black and white.

```
theme_set(theme_bw())
```

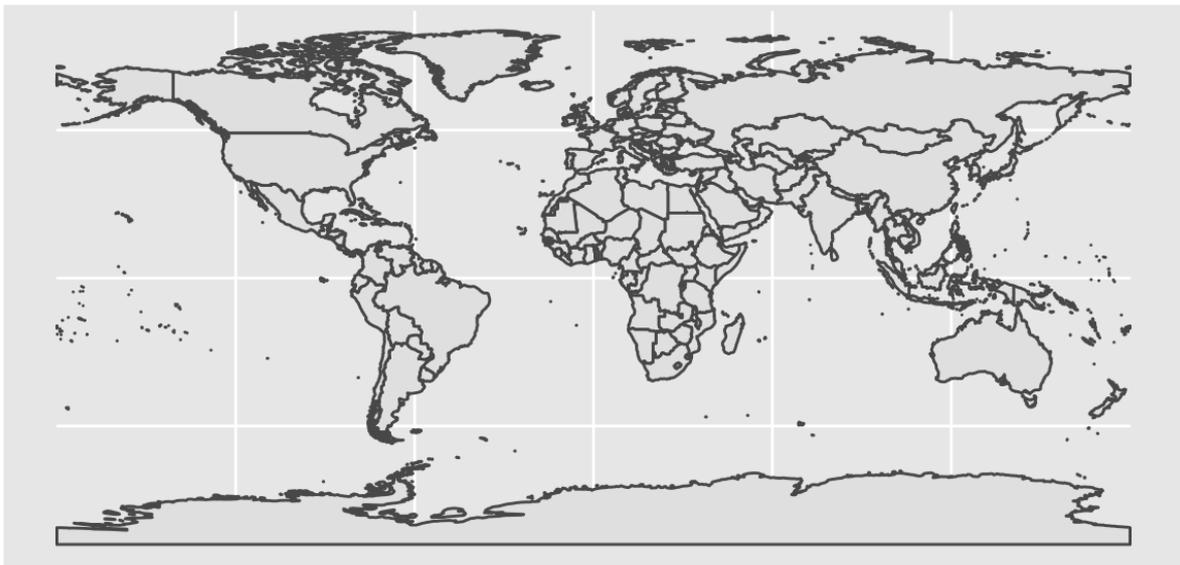
## Take a look at the map

Import the countries data necessary to plot the map.

```
world <- ne_countries(scale = "medium", returnclass = "sf")
```

Then plot the blank map.

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



## Merge Data Sets

We need to merge our corona virus data to the world data frame. Note that the countries name are called **name** in the world data frame and **Country** in the corona data set.

Also, we need to assess some names problems. In the corona data frame, USA needs to be **United States** and UK **United Kingdom**.

```

corona$Country = ifelse(corona$Country == 'USA', 'United States', corona$Country)
corona$Country = ifelse(corona$Country == 'UK', 'United Kingdom', corona$Country)

corona_map = merge(world, corona,
                    by.x = "name",
                    by.y = "Country")

```

Now we have everything we need in our *corona\_map* data frame.

## Let's Plot!

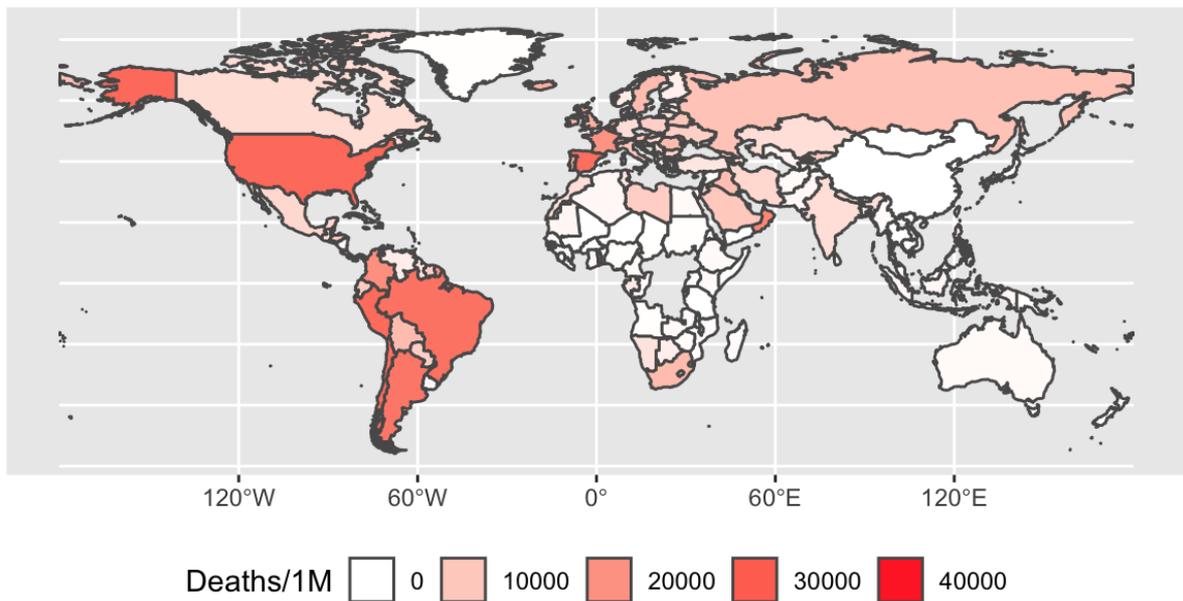
Using the *corona\_map* data frame, let's plot our first map.

```

ggplot(data = corona_map) +
  geom_sf(aes(fill = Cases1M)) +
  scale_fill_gradient(low = "white", high = "firebrick1", guide = "legend",
                     limits = c(0,40000),breaks=c(0,10000,20000,30000,40000))+
  theme(legend.position="bottom")+
  labs( title = 'Covid Cases per 1 Million People',
        subtitle = '10/30/2020',
        fill = 'Deaths/1M',
        caption = 'https://www.worldometers.info/coronavirus/')

```

## Covid Cases per 1 Million People 10/30/2020



<https://www.worldometers.info/coronavirus/>

Feel free to play with the code. Some things you can do:

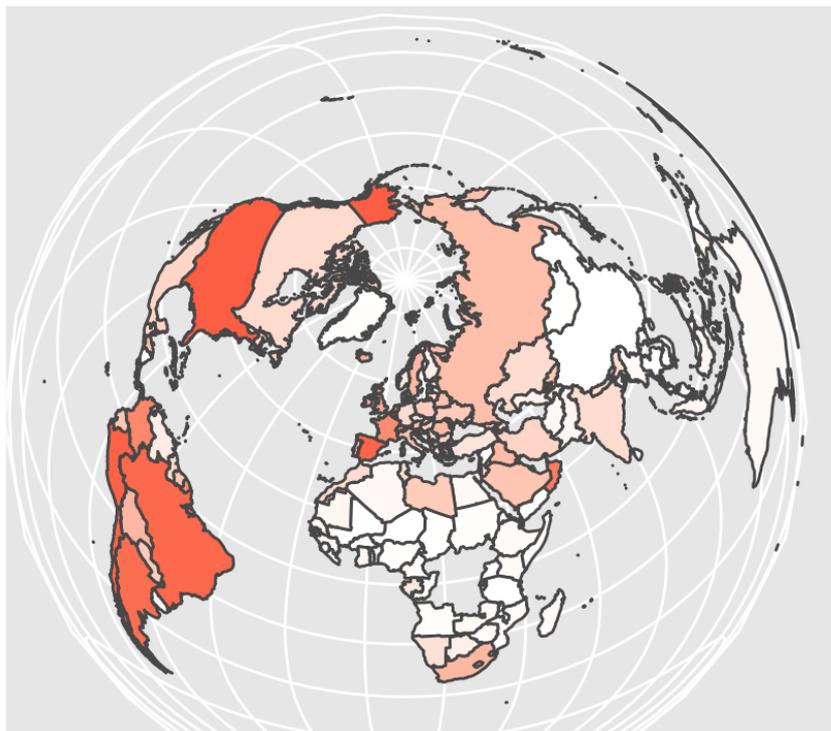
- change to `Death1M` if you want to plot deaths per 1 million instead of cases
- change the color schema. Low and high are the starting and ending point of the color gradient. Be fancy.
- change the bin size of the legend

## 3D Plot

Let's get fancy. Plot the 3D globe. We use the same code as before except one line.

```
ggplot(data = corona_map) +  
  geom_sf(aes(fill = Cases1M)) +  
  coord_sf(crs = "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs ") +  
  scale_fill_gradient(low = "white", high = "red", guide = "legend", limits = c(0,40000), breaks=c(0,10000,20000,30000,40000)) +  
  labs(title = 'Covid Cases per 1 Million',  
       caption = 'https://www.worldometers.info/coronavirus/')
```

### Covid Cases per 1 Million



Cases1M  0  10000  20000  30000  40000

<https://www.worldometers.info/coronavirus/>

Play with the latitude and longitude values. It will change the angle you look at the globe from!

## Zoom In

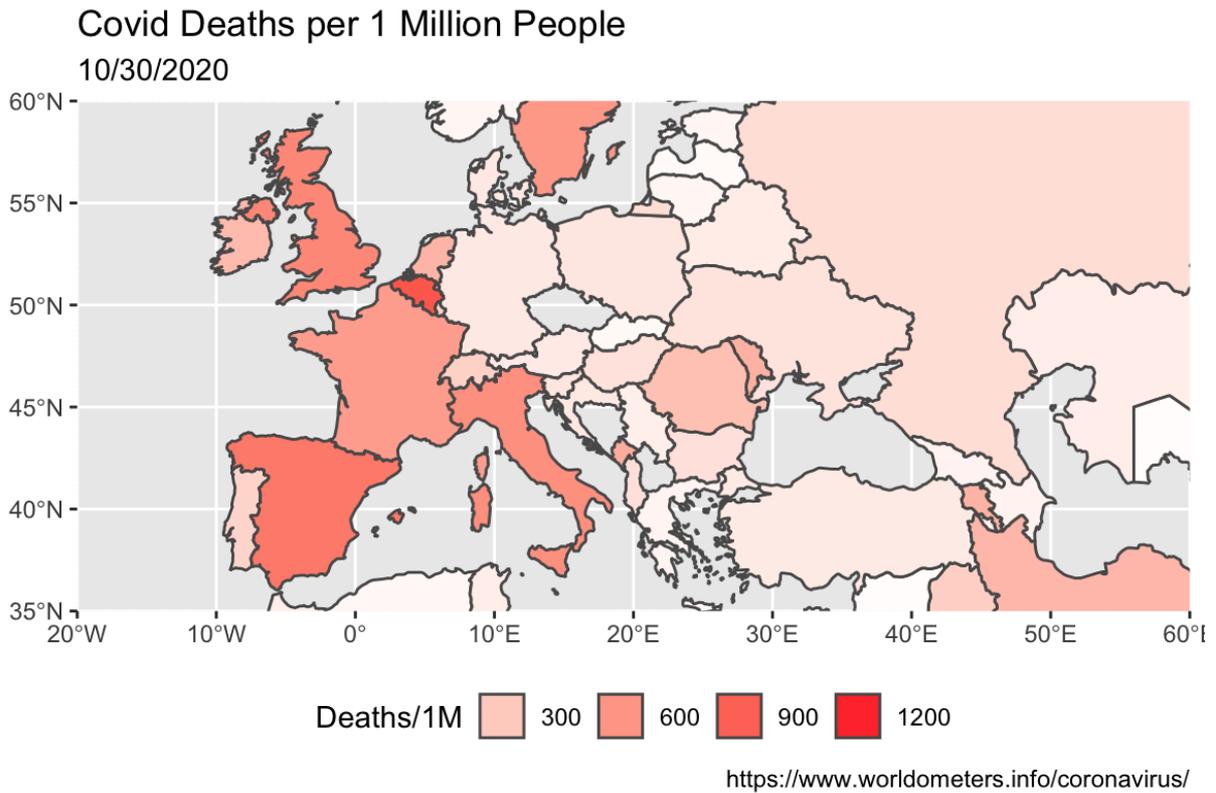
What if we are interested in a particular region and not the whole globe.

We can zoom in by setting latitude and longitude values.

Let's look at Europe.

```
ggplot(data = corona_map) +  
  geom_sf(aes(fill = Deaths1M)) +  
  coord_sf(xlim = c(-20, 60), ylim = c(35, 60), expand = F) +  
  scale_fill_gradient(low = "white", high = "firebrick1", guide = "legend") +  
  theme(legend.position="bottom") +  
  labs(title = 'Covid Deaths per 1 Million People',  
       subtitle = '10/30/2020',
```

```
fill = 'Deaths/1M',
caption = 'https://www.worldometers.info/coronavirus/')
```



## Conclusion

Visualization bridges the gap between the analyst and the viewer. There are many things we can do with maps and I thought this could be interesting, especially during these tough times.

Please don't hesitate to contact me if you have any questions or proposals. Thank you.

Mattia