

Lesson 1

Plotting coronavirus data of a country

Download R and Rstudio.

Start up Rstudio.

Import the data. First look at the table `time_series_covid19_deaths_global.csv`.

`csv` stands for comma separated values. To see where the name comes from we can look at the raw data, that is, at the data how it looks to the computer, without pretty formatting. To do this, either find the **Raw** button on this webpage or just click here for the raw data.

To import the raw data click the play button on the right below:

```
mydata = read.csv(url("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_"))
```

The console should be in the lower left on Rstudio. You can just copy and paste from above.

To verify that Rstudio imported the data click the play button on the right below:

```
View(mydata)
```

Extract the data for one country Search the table for “Italy”.

I found “Italy” in row 138.

Extract this row from the data and show the first two cells:

```
View(mydata[138,1:2])
```

Ok, now we can extract some data from this row. Let us extract the last 30 days and save the result in a variable ‘`deaths_italy`’:

```
deaths_italy <- mydata[138,(ncol(mydata)-30):ncol(mydata)]
```

Here, `ncol` is a function that computes the number of columns of the table. This is important, because the number of columns in the table changes every day when the data is updated. `ncol(mydata)` is the number of columns of `mydata`. We can verify this.

First, look again at `time_series_covid19_deaths_global`. Find the row for Italy (note that it may have a different number) and then the last cell in that row. Remember the number in this cell. Now run

```
View(mydata[138,ncol(mydata)])
```

You should see that same number. To display the last 30 days in that row enter

```
View(mydata[138,(ncol(mydata)-30):ncol(mydata)])
```

Of course, since we already saved this data in the variable `deaths_italy` we can also simply run

```
View(deaths_italy)
```

for the same result.

Plot the data Next, we want to plot the data.

First we need to remove the commas using the function `gsub`:

Convert the data to a list of numbers and plot it.

```
deaths_italy <- gsub(",", "", deaths_italy)
```

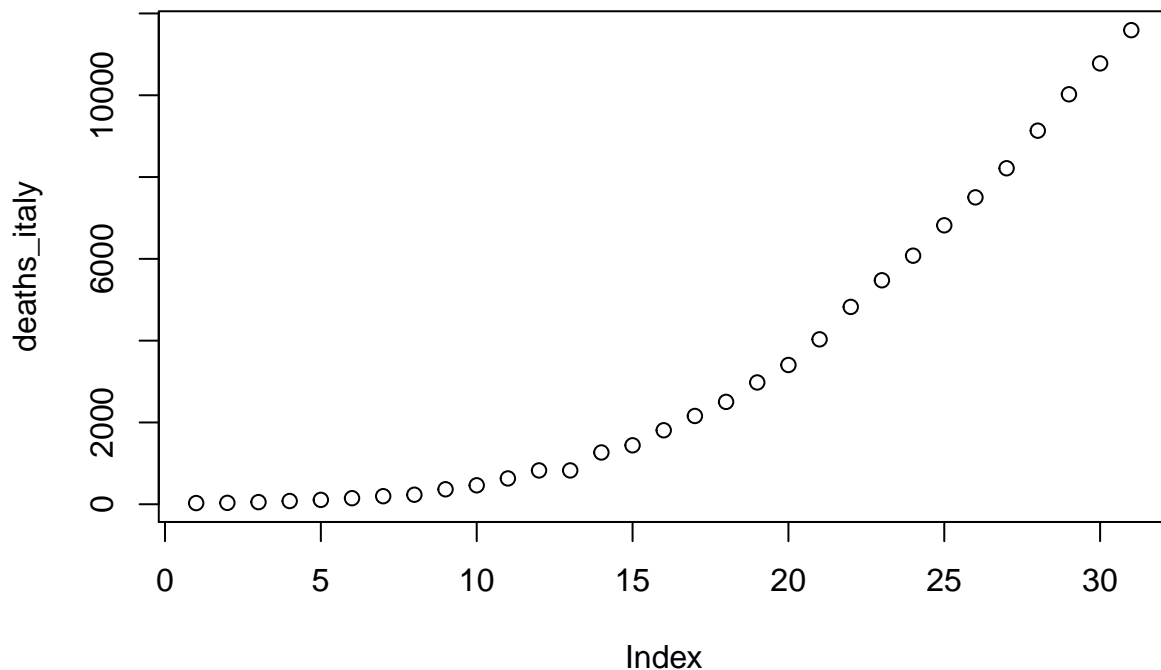
We read this line as: Subtract the “,” from the data stored in the variable `deaths_italy` and store the result again in `deaths_italy`.

Next, we come to subtle point. Whenever you see 0 you may just think of the number zero. But to a computer, the symbol 0 in a text is different from the number 0. In a text, 0 is just another letter, or *character* as one says. It doesn't make sense to arithmetic on characters. Computers only know how to do arithmetic on numbers. So they do need to convert the characters into numbers before they can computer with them.¹

```
deaths_italy <- as.numeric(deaths_italy)
```

Now, we can plot the data.

```
plot(deaths_italy)
```



That's

it for the first lesson. I felt good when I got this going.

Storing the program as a file Now it is play time. First, let us save everything in a file `lesson-01`.

Download this file, save it in a local folder and open it in Rstudio. It should appear in the upper left window. You can now

- go through the program line by line with control-enter (windows) or cmd-enter (macos);
- select all and click on run to execute all in one go;
- change the program and plot your own data ...

¹If you want to understand this in more detail, look at how characters are encoded in a machine. This code is called ASCII. For example, you see that `a` is encoded as `110 0001` (the space is just for readability) and `0` as a character is encoded as `011 0000`. But `0` as a number is encoded differently, namely as a sequence of 64 0s.

Suggested exercises Make your own exercises. Choose different countries, or different time ranges. Look at the data with your naked eyes. You will find that not for all countries there is reliable data. Etc, etc, ...

I wrote this listening to The Clash.