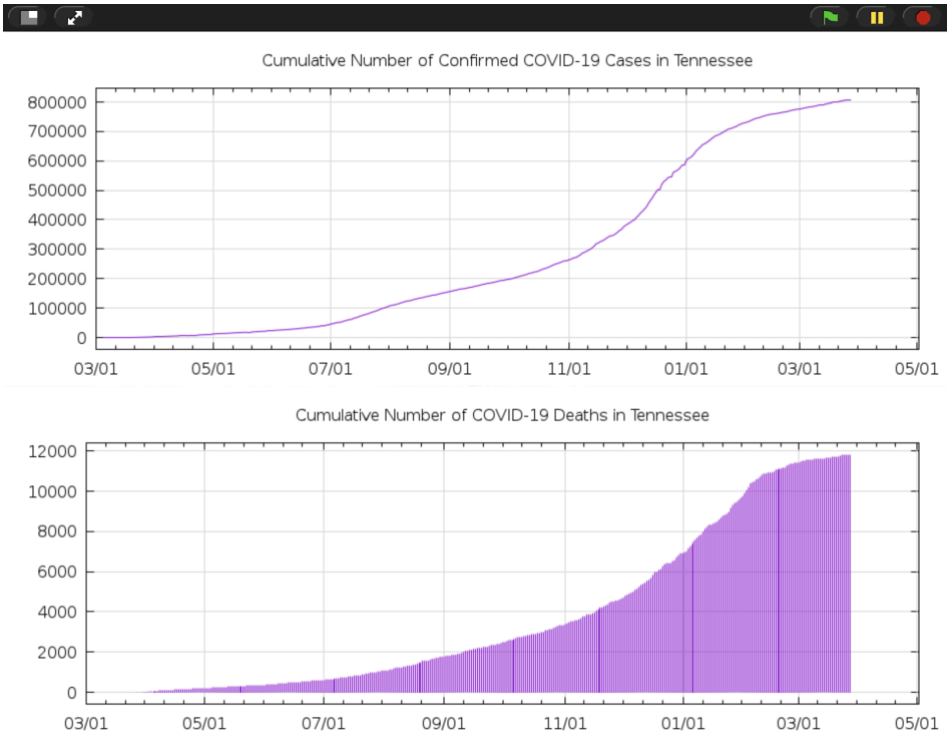
# NetsBlox Lesson: Plotting COVID-19 data

This lesson builds on the Weather project and the introductory lesson on plotting. We will display the same map as the weather app. But when we click on it, it will show the plot of COVID-19 pandemic data. This is how it will look like:

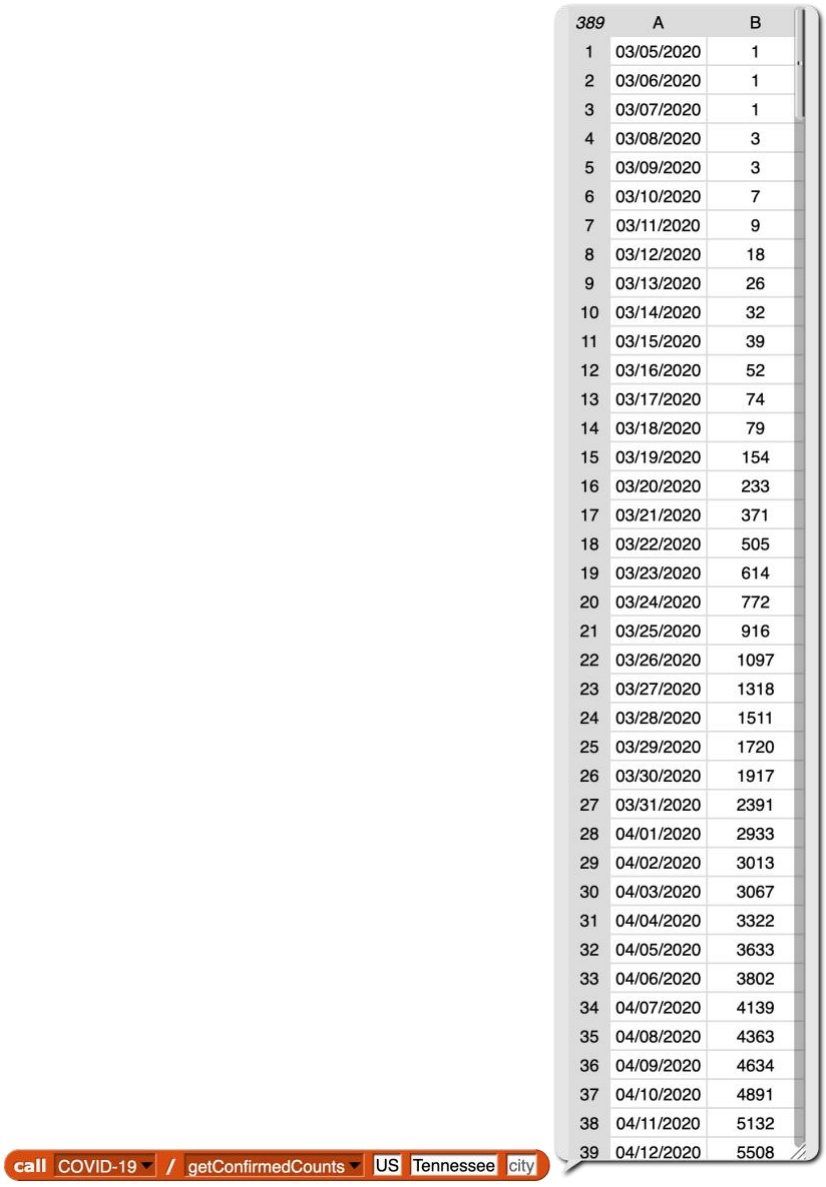



Start from the weather app. The stage code for displaying an interactive Google Map background is exactly what we need, so we do not need to modify it. However, you can change the default zoom level to something smaller to start with the map of the entire US.

We'll have two sprites with very similar code: one for the cases and one for the death counts. Start with the cases, which is the top plot in the figure above. The first part of sprite code is almost identical to the weather app:

We get the location of the mouse click and figure out the country and state where the click occurred. Note we created a number of variable including **`country`** and **`state`**.

Next, we need to consider the data the COVID-19 service returns. As you can see below, the first column does not contain numbers, but actual dates. So, plotting it is a bit more complicated.

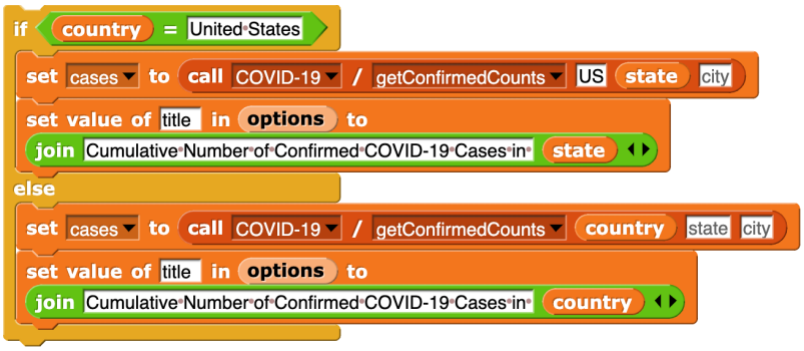



| 389 | A | B |
|---|---|---|
| 1 | 03/05/2020 | 1 |
| 2 | 03/06/2020 | 1 |
| 3 | 03/07/2020 | 1 |
| 4 | 03/08/2020 | 3 |
| 5 | 03/09/2020 | 3 |
| 6 | 03/10/2020 | 7 |
| 7 | 03/11/2020 | 9 |
| 8 | 03/12/2020 | 18 |
| 9 | 03/13/2020 | 26 |
| 10 | 03/14/2020 | 32 |
| 11 | 03/15/2020 | 39 |
| 12 | 03/16/2020 | 52 |
| 13 | 03/17/2020 | 74 |
| 14 | 03/18/2020 | 79 |
| 15 | 03/19/2020 | 154 |
| 16 | 03/20/2020 | 233 |
| 17 | 03/21/2020 | 371 |
| 18 | 03/22/2020 | 505 |
| 19 | 03/23/2020 | 614 |
| 20 | 03/24/2020 | 772 |
| 21 | 03/25/2020 | 916 |
| 22 | 03/26/2020 | 1097 |
| 23 | 03/27/2020 | 1318 |
| 24 | 03/28/2020 | 1511 |
| 25 | 03/29/2020 | 1720 |
| 26 | 03/30/2020 | 1917 |
| 27 | 03/31/2020 | 2391 |
| 28 | 04/01/2020 | 2933 |
| 29 | 04/02/2020 | 3013 |
| 30 | 04/03/2020 | 3067 |
| 31 | 04/04/2020 | 3322 |
| 32 | 04/05/2020 | 3633 |
| 33 | 04/06/2020 | 3802 |
| 34 | 04/07/2020 | 4139 |
| 35 | 04/08/2020 | 4363 |
| 36 | 04/09/2020 | 4634 |
| 37 | 04/10/2020 | 4891 |
| 38 | 04/11/2020 | 5132 |
| 39 | 04/12/2020 | 5508 |

Fortunately, the chart service accepts options that specify that we want to plot time series data and how to interpret the dates. Here are the options we need to set for the COVID-19 data:

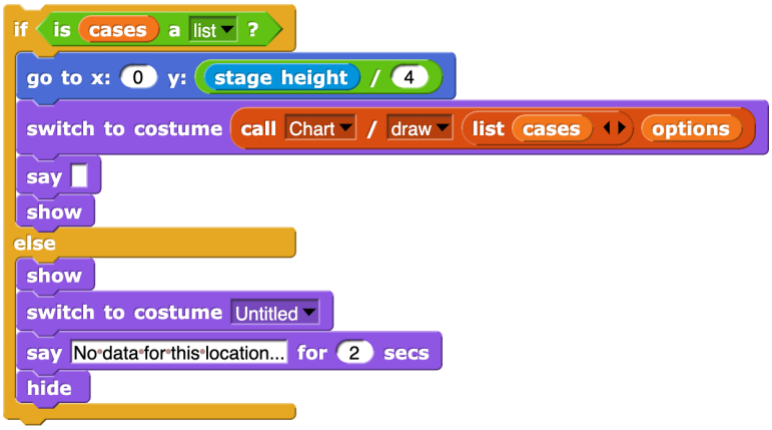

Specifically, we need to set the `isTimeSeries` option to `true` and the `timeInputFormat` to `%m/%d/%Y`. This rather cryptic text means that the date is to be interpreted as a month (`%m`) first, followed by a slash, then a day (`%d`), followed by a slash, and finally a year value (`%Y`). The `timeDisplayFormat` specifies how the labels should look like in a similar fashion. The rest of the options specify that we want a line plot (not a bar chart for example) and the height of the image should be half of the stage. Remember, we have two plots to display.

Next, we have to treat US states and countries differently since within the US we want to see data for individual states, while for the rest of the world, we stick to country level data.




We call the `getConfirmedCounts` RPC of the COVID-19 service accordingly. We use the `state` variable in the first call (US), but not the second (rest of the world). We also set the title of the plot with the name of the area we clicked at.

What if we clicked on a place that has no data, for example, an ocean? Instead of the data list, we would get an error message from the RPC. So, we need to handle that case:



If the returned result is a list, we are in business. We go to the center point of where we want the plot: centered on x, but up a quarter height in y. Then we simply call the chart service **`draw`** RPC and change the costume and show ourselves.

But if there was an error, we just display an error message for 2 seconds in the else branch and hide ourselves to show the map background again. Here is the entre script in one place.


We are not quite done. When we click the map, we show the plots, but when we click the plot or press any key, we should hide the plots and show the map again. It takes a few simple scripts:


That's it for the first sprite! The second sprite displays a plot with the death counts. It is almost identical to the first sprite. It calls a different COVID-19 RPC of course. Also, it makes a bar chart as opposed to a line plot:


And finally, it does not need to display an error message in case there is no data since that was taken care of by the first sprite already.

The final solution is below. It also includes a little helper sprite that displays some helpful directions when the project starts:

https://editor.netsblox.org/?action=present&Username=ledeczi&ProjectName=COVID-19&