Data to answer a question: COVID-19 and...

"Data! Data! Data! [...] I can't make bricks without clay!", famously screams Sherlock Holmes.[1]

So, let's get some data! We are going to start with the collection of COVID-19 cases and deaths counts compiled by the New York Times. This is shared with the public in a GitHub repository[2]. To clone a GitHub repository like the one mantained by the NYT in Rstudio is very easy (click 'file' select 'new project', choose 'version control, then 'Git' and follow instructions...). You can also only download the one file you need for the analysis below. In order to re-run the commands below on your machine, you might need to modify them to make sure that Rstudio is looking for files in the right place (cloning the directory might take some time, so do not worry)

[1] Arthur Conan Doyle (1892) "The Adventure of the Copper Beeches" in *The Adventures of Sherlock Holmes*

[2] `https://github.com/nytimes/covid-19-data`

Intro to code chunks

R chunks are portions of R code that are included in the R markdown and executed every time the .Rmd is knitted. An R chunk starts on a new line with three backquotes followed by `{r}` and and ends with three backquotes on a new line. We can choose to make the commands visible, or to view only the results, including warnings and echos, depending on the options we specify in the chunk options after `r` and in between the curly brackets.

The following code chunk loads one of the datasets available in the directory.

```
us<-read_csv("us-states.csv")
```

We now take a look at the data, using a different option in the specifying the code-chunk.

```
names(us)
```

```
## [1] "date"   "state"  "fips"   "cases"  "deaths"
```

```
us[1:10,]
```

```
## # A tibble: 10 x 5
##    date       state      fips  cases deaths
##    <date>     <chr>      <chr> <dbl>  <dbl>
##  1 2020-01-21 Washington 53        1      0
##  2 2020-01-22 Washington 53        1      0
##  3 2020-01-23 Washington 53        1      0
##  4 2020-01-24 Illinois   17        1      0
##  5 2020-01-24 Washington 53        1      0
```

```
##  6 2020-01-25 California 06        1      0
##  7 2020-01-25 Illinois   17        1      0
##  8 2020-01-25 Washington 53        1      0
##  9 2020-01-26 Arizona    04        1      0
## 10 2020-01-26 California 06        2      0
```

Using Python in Rmd

You can also run Python code chunks in the same `.Rmd` document. To do this, you need to have installed the `reticulate` package and loaded it — which we did in the "invisible" set-up chunk. Note that the choice of using Python for this class is entirely up to you. If you do not want to install Python on your computer, please comment out (or erase) the following code chunks.

We start by loading `pandas` and reading the data.

```
import pandas as pd
us_py = pd.read_csv("us-states.csv")
```

Now we use Python to compute daily new cases and find, for each state, the date with the highest number of new cases.

```
##          state       date  new_cases
##     California 2022-01-10   227972.0
##        Florida 2022-01-04   193786.0
##          Texas 2022-01-03   164902.0
## North Carolina 2022-01-18   121315.0
##       Michigan 2022-01-19    98299.0
##       Illinois 2022-01-18    93423.0
##       New York 2022-01-08    90132.0
##      Wisconsin 2022-01-17    83187.0
## South Carolina 2022-01-18    72445.0
##  Massachusetts 2022-01-10    64715.0
```

Looking a graphs

Finally we graphically display part of the data. Note that the chunk options here include details on the size of the figure. While you can avoid specifying these, often, in order to obtain effective displays you will need to customize these values. We are using 'ggplot', which is a fairly sophisticated way of making graphics, and the commands might look a bit mysterious to you. It is always a good idea to look up the meaning of commands you might find using the Help option in Rstudio.

This is a large graph, for which we decided to use the entire width of the html file. Take a look at the code to see how we achieved that.

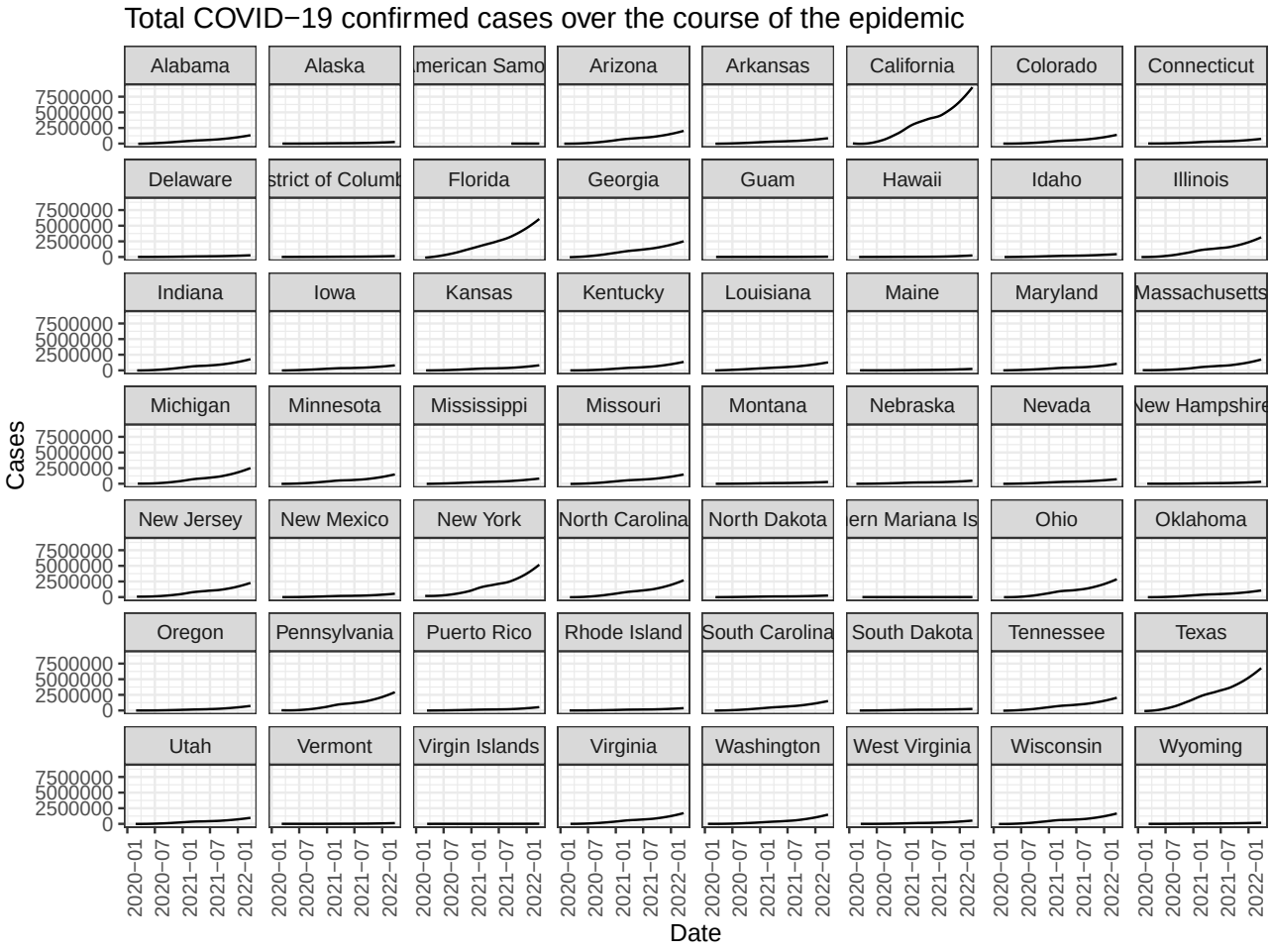

Figure 1: Cumulative confirmed COVID-19 cases over time by state (New York Times data).

The default place for a figure is in the main column. This is an OK placement for a figure that is really part of your discourse, but keep in mind that figures are not part of your text. Where ever they are placed, you need to bring them into the text by precisely referring to them. To do so you need be able to refer to their number.

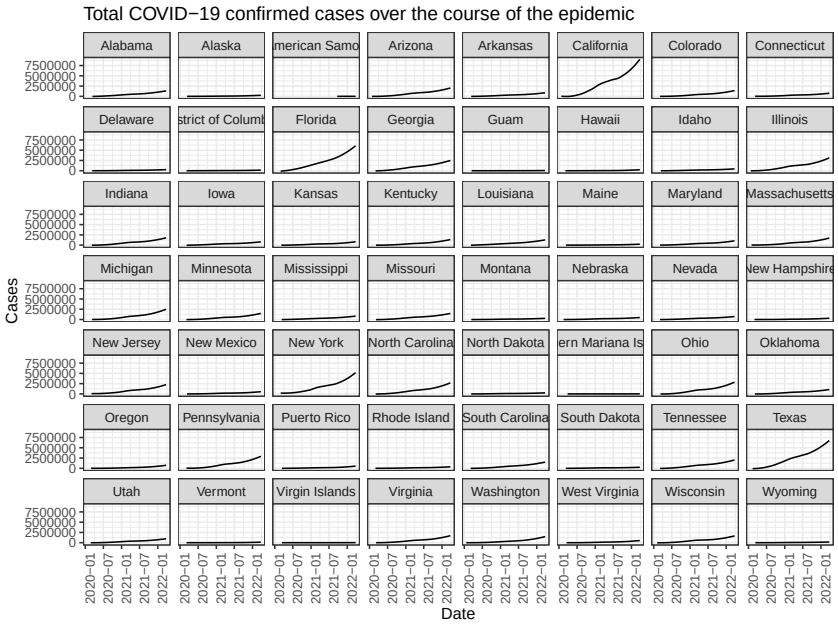

Figure 2: This is how I put a legend

This can be tedious and error prone, fortunately, by the time we will need to do this, you will move to a version of the software that relies on the `latex` engine and it will take care of numbering for you.

Often, we might be interested in smaller figures. With the `tufte` package we have the option to display them on the side of the text, by appropriately modifying the chunk options. The display on the side focuses on the state of New York, and looks at daily new cases, rather than cumulative counts. This requires us to work on the data a bit before producing a display. In the code for both graphics, there are a number of options, specifying labels, themes etc. They are an invitation for you to explore.

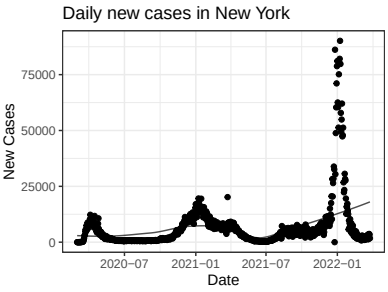

Figure 3: This side graph focuses on the state of New York

Finally, let's play around with sizes to figure out what makes for a best display of information. First of all, note that in order to have good control of your displays, you will want to have one r-chunk for each figure.

A final note on this document. It was created with a main goal to give you an overview of how to use R markdown, so I tried to pack it with a number of commands. In order for it to be useful, you should go back and forth between the .Rmd file and the .html one.

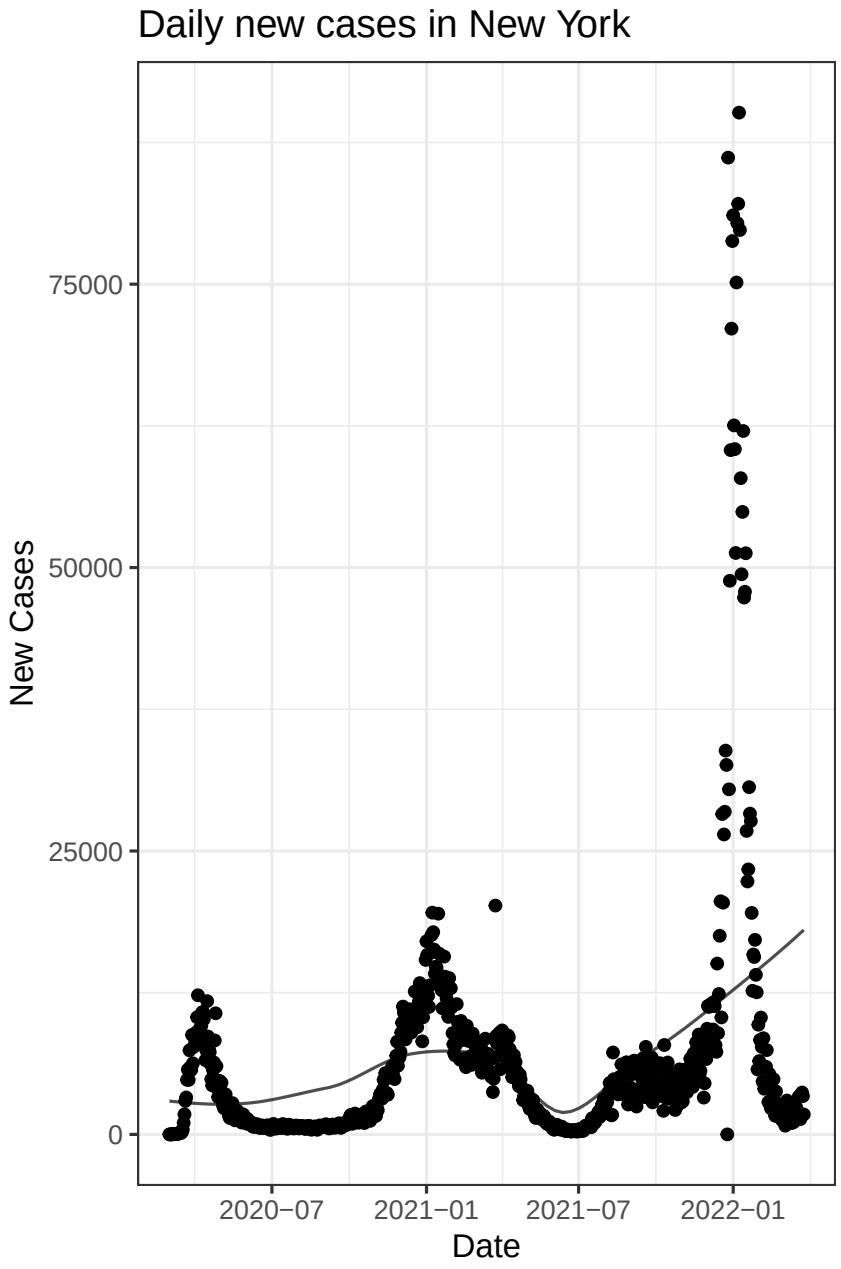

Figure 4: Note the effect of some size parameters

- Look at how to make titles of different sizes
- Make sure you note how to include links
- Find out how to write in *italic* and **bold**
- How to make a bullet points list

1. And a numbered one