6.S083 / 18.S190 Problem set 1: Data analysis

Submission deadline: 11:59pm on Tuesday, April 7

(Submission instructions to follow later.)

In this problem set we will explore the COVID-19 data set from

https://github.com/CSSEGISandData/COVID-19

in more detail.

Please use the notebook from lecture 1 as the basis for loading the data.

You should make a Jupyter notebook with your solutions. We will combine the best ones to make the model solutions.

Exercise 1: Visualizing the data over time

In this exercise we will practice data cleaning and plotting. Make a new plot for each question.

1. Extract the data and country names from the CSV file as we did in class. Call the country names `all_countries`.
2. Make a `Vector` called `countries` with a subset of countries that we wish to plot, say China, Japan, South Korea, US, United Kingdom, France, Germany. Be careful to check how they are written in the data set.
3. Define a variable `num_days` by extracting the number of days of data from the dataframe.
4. We need to accumulate the data for those places that are split up into territories. Make a zero vector of the correct length for each country, e.g. using the function `zeros`.

   Loop through all the countries and add the corresponding data to that country's data.

   You may use a dictionary (`Dict`), or a matrix, or a `Vector` containing `Vector`s, or a new `DataFrame` to store the data.
5. Plot all countries' data on the same graph by using a `for` loop. To do so, first make an empty plot:

   ```
   p = plot()
   ```

   Then run the loop, adding in the data using `plot!`

   Finally, display the plot by evaluating the plot object `p`. (Just type its name and evaluate.)

6. Now use a `log` scale on the y axis. In order to do so you will need to convert the vectors to contain `Float64` and replace any 0 values by `NaN` ("not a number") so that `Plots.jl` ignores those values.

   Is there exponential growth?

7. Turn this into an interactive visualization by adding a slider corresponding to the current day, varying between 1 and the total number of days for which you have data. You should draw only the data up to that particular day. As you move the slider the plot should update. Fix the horizontal axis using the keyword argument `xlim=(0, num_days)` inside the first `plot` command.

Exercise 2: Visualizing changes

Now let's try to reproduce the essence of the nice visualization from https://aatishb.com/covidtrends, which is a less usual point of view. Again the slider will represent a day during the epidemic.

The horizontal axis will show the *total* confirmed cases until the given day, while the vertical axis will show the *change* in confirmed cases during the past week (7 days).

1. Make a data set `total_cases_to_date`, representing for each day the *total* number of confirmed cases during the whole epidemic up until that point.
2. Make a set of data `new_cases` which is the total number of cases only during the past 7 days.

   You may use the `sum` function.

3. Make the visualization using a slider representing days as in the previous exercise, plotting the total number of cases on the x axis and new cases on the y axis.

   You need to take care that each vector being plotted has the same length.

4. Add a dot for each countries current position using the `scatter` function (which otherwise works like `plot`) and annotate the countries' dots using something like

   ```
   annotate!(x, y, text(country_name, 10, :black))
   ```

   where `x` and `y` are the positions at which to annotate and `10` is the font size.

Exercise 3: Helping with transcripts

The goal of this exercise is to improve the quality of the auto-generated subtitles / captions of the lecture videos via crowdsourcing. Although the natural language processing is very impressive, it's far from perfect!

1. Choose one of the lecture transcripts from lectures 1–4 in the Google Docs transcript folder.
2. Choose at least 10 lines, preferably contiguous, and correct the captions by listening to the relevant portion of the video, as given in the time-stamp. Modify the text according to what you hear, paying particular attention to the computational and mathematical words that the automatic transcription software tends to get wrong.

   Please do not modify the time-stamps or line numbers.
3. Change the colour of the text that you transcribed to a colour that is not black nor the same as the previous and next block.
4. Also check (proofread) the 10 previous lines and 10 following lines and correct things if you think they're not correct.
5. Write down in your pset submission which piece of which transcript you did. If there are none left, make sure you are first in the line for the following lectures!