Web scraping and Covid-19: Covid 2

Welcome back to part two of lesson four of our web scraping to build social research data training series. In the first part of lesson four, we looked at scraping some very simple limited statistics about the Coronavirus situation globally. And this part, we're going to look a bit more in depth, we're going to try and scrape what we consider a data set.

So a fairly detailed table of statistics about Coronavirus. Globally, we implement pretty much all of the same techniques, there's a little bit more sophistication to how we scrape the data from the table. But it's still a very simple process. And at the end, we get quite a good result. As I said, we're working on with our Jupyter Notebook, which has lots of documentation or lots of examples for you to follow along with. So in this part, we keep it nice, nice and brief. I'll run through what we're trying to do, and I'll implement it. And then I'll give you some time to go away and review what we've done also and to practice and to change things around.

Also let's just quickly familiarise ourselves with the website. And once more. So here's the table of Coronavirus statistics, we're interested in scraping. So we've got 13 or 14 different variables or columns in the table, we've got a role for not just every country, but we've got an overall aggregate global set of statistics. And there's some content statistics mixed in as well. Just to remind you, this is the same website we used in the first part of lesson four. And here are the three pieces of information we scraped during that part of the lesson. But now we're interested down here, we're interested in this table. Also, you'll notice that the table is interactive. So you know, we can change some of the tabs, we can just specify we want countries and entities that relate to North America, for example, we can look at statistics yesterday, compare them to two days ago, etc. So there's different types of data that we could scrape from this table, we might call this table dynamics, or it changes based on our interaction with it, we're going to keep it simple. In this example, we're gonna just scrape information for all countries, and the most up to date information that we have. So again, let's run through the exact same process.

So in Python, we need to set it up so we can do a web scraping. Let's do that. So I've loaded in the modules I need for web scraping. And I've grabbed today's date and saved it in a variable called funnily enough date. So again, first step on the web scraping process, request the webpage and parse it so that Python knows it's dealing with a webpage. So excellent. That's all worked. And I've just condensed it down really quickly, because we want to see the end results of this of the scrape.

So previously, we tried to scrape information that was contained in div tags, or divider tags, which are basically sections of webpages. Now we're interested in is table. helpfully enough, a table in HTML is identified by the table tag. So we're asking to find the table tag in the web page. And again, handling the table has a unique ID, and it's called main table countries today, which is quite good. And then within that table, because we don't want the metadata associated with the table, we want to find the t body tag. And table body tag identifies the actual content of the table that we can see right here. And how I find how I found those tags is the same process before I went to the web page, you can maybe highlight a bit of the table just to make it a bit easier. And if I right click, and I go to inspect, it should bring me at least very close to the table tag that I need. Yep, so here's the the t bodyy tag containing all of the actual content of the table. And here's the table tag with the ID that I need. So so far, so good.

I want to extract the information contained in each row in the table. Here's where it gets slightly more sophisticated what we've seen before, essentially, we need to loop over every row of the table and then in every rule grabbed the information that's in every column. So we need a loop. So that's what we've got going on here. So for every row in the table, and find the column and extract the text that's contained within the column. And that's all really that's happening here. But there's a little bit more going on. I've provided some written explanations of what it is a bit more specifically. I've essentially run the scraper over the table extract all of the In the columns, and then stores and a different variable called global info, global info is basically a list of all of the rows in that table. And that makes it easier then to write it to a file. And shortly. So again, I just asked to print, you know, the first 10 rows of the global info variable. Or you can, it looks a bit messy, but you can kind of discern, you know that there's a list for every row. So the first row in this list refers to the world role in the table on the webpage. Exactly here.

So the first row in the new variable I've created contains all of the information contained in this row here. The second row has yet information for the US. And the third row has information relating to India. Exactly. So just with a quick visual inspection, I can see that my scrape has worked as intended. Because I don't want the world as statistics roll. And this little bit of code here says, delete the first row in this list, delete the first element of that list. So very simple. I think we've just done. And I've just printed, you know, I've asked to say, right, so how many roles we're in, we're in the table 211. I mean, that should line up. Yeah, it's not quite, I've clearly not scraped some of the information. And I think that will relate to the fact that, as I said, this is a dynamic table. So it'll produce some, you know, some records for non countries. So if you may remember, the diamond princess was was one of the cruise ships at the beginning, but couldn't dock anywhere. that's included in the official statistics, the MS Zaandam and other cruise ship, I believe. But I think I've grabbed pretty much all of the countries that I need. And I've deleted some records, you know, relating to the role of world statistics, I don't, I don't want those. But it's easier to export this to, you know, like a spreadsheet format, that we're familiar with a social scientist, and to work through that and to then see if there's any errors with the web scrape. So again, what I'm doing is creating a Downloads folder, if it doesn't exist, if it already exists, we just move on without execute that piece of code. I'm creating the first row of my spreadsheet, which is the variable names. I could have scraped the variable names, you know, here. But just to avoid, you know, spaces and gaps and commas, and maybe quotation marks, it wasn't that difficult. So I just thought I'd write the variable names, myself. So you can see there are some, you know, gaps gaps are fine. But you know, there's no forward slashes anymore. There's no columns within the variable names. It didn't take me that long, I thought it'd be easier, if I had control over what the variables were called in the spreadsheet, I'm creating the file for I'm going to actually save the results of the scrape, it goes in the Downloads folder. This is the name of the file. I'm appending, today's date and the name of the file. This is good practice. So I know when I when I scrub, scraped the data. And I'm going to save it as a CSV file, which has a comma separated values. And we'll see what that looks like very shortly.

Transcribed by https://otter.ai

So then I open the file that I want to save the information in. I basically just set up the the writing process. So I say, right, I'm writing to a CSV file, I feed it the variables first. So right start to the first row of the spreadsheet. And then in my global info variable, which contains all the rows of the table for every row in that list and write that row to the spreadsheet. So let's see if that worked. So that work worked really quickly, which is good. No, no really waiting around. I'll do the two means of demonstration as usual, which is we'll do the actual manual inspection. So everything I do is in this ncrm course folder in the code downloads, yep, so you can see. Yep, that's today's time. So you can see I didn't create this previously, and I'm trying to trick you. This is actually real time. web scraping, which is good.

So let's have a quick check if I actually wrote the information to the file correctly. So I was a, Yeah, so there's been it has worked quite successfully. But as you can see, there's some extra columns here. And that's all have a variable name, and I'll explain what's happening with that shortly. But just to kind of cursory glance. Yeah, it seems to work and as you can, as you can see, there's 211 rows. There are some rows of the table we didn't scrape so those rows again, as I said, referred to some of the cruise ships or some of the some of the entities or regions that are not countries, but scrolling device statistics are reported. And for. So what's going on with these extra variables is because it's a dynamic table that we're working with. This is what we can actually, you know, see. So this is what's actually sent back to us to our browser to view.

But if we looked at the underlying HTML, there was actually extra variables referring to each of the continents. So they're hidden from our view. But if I was to pick Europe, North America, Asia, essentially, those are the extra columns here. So there's information on North America, because the USA is in North America, India is in Asia. So there's an extra couple of statistics to do with Asia. And that we don't see in the table. But that exists in the underlying HTML says two ways of addressing this problem, we can do it in the scrape. So when we extract the tags, we can do a bit more data cleaning there to ignore those, those extra columns. Or we can scrape the information, just like we've done here, get to the stage of writing to the file, I think this is a bit easier if I personally I use data, I would just import this data set into Stata, and just drop those variables. So there's lots of ways of doing it, you don't have to be totally computational, or programmatic and how you deal with some of these data quality issues.

So let's very quickly get that data set back into Python. So we can have a quick look again. So we don't have to do the manual demonstration, we can we can pull it back into Python. So here we go. Here's our data set that we've scraped. Happily, you can see that when we return back into Python, it ignores those extra columns in the CSV file. So that's quite handy as well. But the pandas, which is identified by the PD acronym, so the pandas data module for handling datasets has taken care of that kind of surplus variable problem we were having. And this all looks pretty good. So this is the table, it's a bit more readable, I would say then the table as well, just here. And of course, more importantly, we have it now for our own statistical use outside of Python. And with us, again, we won't go too much into it. But the pandas module is excellent for for data management, and maybe not as intuitive as stated for me or confuse our SPSS. But it can, it can handle some unfamiliar data structures, some more complex data structures a bit better than those statistical programmes. And also, we can use Python to pick up particular roles in the data. So give us the role where the country variable takes this value here. So

Spain, for example, you may have noticed by now that I'm Irish, so I could look at the Irish record, etc, etc.

So that's lesson four. We've looked at two examples. One reasonably simple just getting you know three bits of information from a web page and writing to a file. And the second part, we've done a bit more of a realistic example, something that you're probably going to do yourself when you're scraping data, which is a table exists. So you know, the data has already kind of neatly arranged and you want to get that table and download it and put it into a file for your own future use. So that's the end of the practical examples we will do in this training series. The next lesson is going to focus on some of the ethical and legal considerations of engaging in web scraping. So looking forward to having you join us.