Writing to CSV using Python

Teun van Gils

17th January 2019

1 Writing to CSV

When moving from Python to R, it might be convenient to be able to export your data to CSV – a format that is very easily imported as a `DataFrame` in R. There are two ways to go about this:

1.1 The manual way

You can simply write text to a file in a way that matches the format of a CSV file. This means:

- Start the file with a comma-separated list of **column names** (which is called a header), and
- Write each row of data to a separate line as a comma-separated list of values, matching the order of the headers

An example of how you could do this:

```
1  teachers = [
2      {'fname': 'Teun', 'lname': 'van Gils', 'email': 'edu@teunvg.nl'},
3      {'fname': 'Joska', 'lname': 'de Langen', 'email': 'J.deLangen@uu.nl'},
4      {'fname': 'Lucie', 'lname': 'Kattenbroek', 'email': 'L.Kattenbroek@uu.nl'},
5      {'fname': 'Joris', 'lname': 'Vincent', 'email': 'J.Vincent@uu.nl'}
6  ]
7
8  with open('teachers.csv', 'w') as file:
9      # write the header
10     file.write('first_name, last_name, email\n') # the \n is for newline
11     for teacher in teachers:
12         file.write(f'{teacher.fname}, {teacher.lname}, {teacher.email}\n')
```

1.2 The way other people have fixed it for you

Of course, you are not the first one to run into this issue. Which is why other people have written code to automate this. The csv package also contains a DictWriter function. The could below would do the exact same as above. One adaptation is that automatic generation of all keys is also included.

```
1  import csv
2
3  ## Header
4  # If there's a lot of keys in our dictionary, we'll want to not have to type those
       out.
5  # This automatically gets all the keys that are in a list of dictionaries
6
7  header = []
8
9  for teacher in teachers:
10     for key in teacher.keys():
11         if key not in header:
12             header.append(key)
13
14 ## Writing
15 # We now write the file dictionary by dictionary
16 with open('teachers.csv', 'w') as file:
```

```
17      # Define a 'writer' object: this will handle the writing for us, with the
            specifications we want
18      writer = csv.DictWriter(file, fieldnames=header, lineterminator='\n',
            delimiter=',')
19
20      writer.writeheader()
21
22      for teacher in teachers:
23          writer.writerow(teacher)
```

In the end, the approach you take is up to you. Feel free to check what works best for you!

2 Countrynames in R

Everyone has different ways of writing country names. Some will call the UK Great Britain, for instance, of the Netherlands Holland. This is why it is often useful to be able to standardize country names when working with country data.

If you run into that scenario - and you don't have to for this assignment, but just in case you do - someone else ran into that same issue and wrote a package to help you out. The package is called `countrycode`, and can be installed and loaded like any package. The important function will be `countrycode()` - check out the help function to see how it works.