Fetching data in Python: os and Requests

Paul Bradshaw

This week:

- Python libraries for working with data ‘locally’: os
- Fetching documents from the web: Requests

The **os** library

- Navigates around the folders and files
- Useful for creating a list of files to then repeat actions on them all
- In Colab you are on a remote computer

os functions

- **os.getcwd( )** - where am I?
- **os.listdir( )** - what’s here?
- **os.chdir( )** - move (use “ . . ” to go up)
- **os.system( )** - run command line

Combining files with **cat**

```
os.system("cat *.csv > alldata.csv")
```

The **Requests** library

- **requests.get()** fetches document from URL
- Document can be webpage, or other files
- Requests ‘object’ has various properties

‘Objects’ and ‘classes’

- Objects created by certain functions often have certain properties
- E.g. A ‘makeacar’ object might have wheels, gears, current speed, top speed, etc.
- *(You might see this called a ‘class’ of object)*
- These can be accessed with built-in code

Requests ‘objects’

- The result of requests.get() has attributes:
- rqobject.status_code()
- .content()
- .text()

HTTP status codes

- 404 error - page not found
- 403: Forbidden
- 200: Success
- 400: Bad request, etcetera

Creating files: **open** and **.write()**

```
#create a new file - it's empty for now

fd = open("thisisanewpage.html","a")

#write the .text property of the
variable 'ojb' into that object

fd.write(ojb.text)
```

For PDFs: use .content

```
#create a new file - it's empty for now

fd = open("thisisanewpdf.pdf","a")

#write the .content property of the
variable 'ojb' into that object

fd.write(ojb.content)
```

Key points

- Use **requests** to fetch files from a URL
- Use **.text** to see the text of that file (e.g. HTML) or **.content** if it's encoded as a PDF
- Use **open** and **.write()** to create local copies of that text