Datascience

Rodda John

06/23/2020

1 Datascience

- What is datascience?

1.1 Definition

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, deep learning and big data.

1.2 So what?

- Why do we care about datascience?
- Where might we see datascience in practice?

1.3 What we are going to do

(these are all quite real)

- A new type of a loop: `for`
- File management in Python
  - And some associated functions
- Library use in Python (specifically numpy)
- How to combine file management and numpy to do some simple data analysis
- Graphing in matplotlib

2 `For` Loops in Python

- A `for` loop is a special structure for iterating through an iterable

2.1 Examples

Assuming:

```
l = [1, 2, 3, 4, 5, 6]
```

Check out:

```
for e in l:
    print(e)
```

Or using `range`:

```
for i in range(10):
    print(i)
```

2.2 So if we rewrite `sum_of_list`:

```
def sum_of_list(l):
    to_return = 0

    for e in l:
to_return += e

    return to_return
```

2.3 Conclusion

Computer Scientists are lazy.

3 File Management in Python

- What is a file?
- Are there different types of files?
- Anyone know what a `.csv` is?

3.1 `.csv` - Comma Separated Value

- An excel spreadsheet can be exported as a `.csv`
- A `.csv` is in plaintext
- Sample:

```
fname,lname,age,gpa
Bart,Simpson,30,3.6
Homer,Simpson,32,3.7
Spongebob,Squarepants,20,4.0
```

This is really:

`fname,lname,age,gpa\nBart,Simpson,30,3.6\nHomer,Simpson,32,3.7...`

3.2 For Our Purposes

fname	lname	age	gpa
Bart	Simpson	30	3.6
Homer	Simpson	32	3.7
Spongebob	Squarepants	20	4.0

3.3 `open(file, mode)`

- Accepts two arguments, `file` (name of the file), and `mode` (see table)
- A function that returns a `file pointer` to a file

char	meaning
`r`	open for reading (default)
`w`	open for writing (truncation)
`a`	open for writing (appendation)

3.4 `read()`

- A function that operates on a file pointer (`fp.read()`)
- Returns the contents of the file as a string

3.5 `write(contents)`

- `fp.write(content)` Writes `contents` to `fp`.

3.6 Examples

(assuming `csv` is `data.csv`)

```
print(open('data.csv'))

contents = 'stuff,to,write\nsecond,line'

open('data.csv').write(contents)
```

3.7 `split(char)`

- `str.split(char)`
  - Runs on a string and splits the string into a list of various elements delimined by `char`

3.8 Another example

```
data = open('data.csv').read()

sum_of_elements = 0

for line in data.split('\n'):
    parsed_line = line.split(',')

    sum_of_elements += int(parsed_line[2])

print(sum_of_elements / len(data.split('\n')))
```

4 Library Use in Python

- A library is a collection of functions written by others that supplement what is natively provided by Python.
- We will be using `numpy`, a scientific processing library

4.1 Loading files in `numpy`

```
import numpy

np_array = numpy.genfromtxt('gpas.csv', delimiter=',', skip_header=1)
```

```
print(np_array)
```

This creates a two dimensional array.

4.2 Advanced Array Splicing

```
# This returns a list of all values in the column with index 3

gpas = np_array[:,3]
```

4.3 Statistics Functions in `numpy`

function	description	sample use
`median(a)`	Finds the median of an array	`median(gpas)`
`average(a)`	Finds the average of an array	`average(gpas)`
`mean(a)`	Finds the mean of an array	`mean(gpas)`
`std(a)`	Finds the std of an array	`std(gpas)`
`var(a)`	Finds the var of an array	`var(gpas)`

4.4 Thus

```
import numpy

all_data = numpy.genfromtxt('gpas.csv', delimiter=',', skip_header=1)

gpas = all_data[:,3]

print (gpas)

print (numpy.median(gpas))
print (numpy.average(gpas))
print (numpy.mean(gpas))
print (numpy.std(gpas))
print (numpy.var(gpas))
```

5 A Small Assignment

- Given a file, `gpa.csv`, find all the statistical values for each column:
  - Median

- Mean
- Stddev
- Variance

6 Graphing through Matplotlib

To REPL we go!