CS1028: Programming for Sciences and Engineering

Lecture 11

Daniel Vogel

24/10/2019

University of Aberdeen

Outline

1) Recap: input from keyboard
2) “Make up” some data
3) Reading and writing files
4) Output to graphics

1) Recap: Entering data manually

At the OS command prompt:

The file Lec09_helloJimmy.py contains

```
import sys
print('Hi, ',end='')
print(sys.argv[1],end='')
print('. How are you?’)
```

At the command prompt:

```
> python Lec09_helloJimmy.py Daniel
```

This prints to the screen:

```
Hi, Daniel. How are you?
```

Recap: Entering data manually

Within Python (e.g. an IDE such as Spyder):

```
nstr = input("Give me a number: ")
```

This prints `Give me a number:` to the screen, and assigns whatever the user enters (finished by Enter) to the variable `nstr` (of type string).

```
n = int(nstr)
x = float(nstr)
```

These commands turn the input into a number (either integer or floating point number), they will produce errors if `nstr` does not "look like" a number.

Recap: Entering data manually

Entering arbitrary many numbers:

(e.g. for the min exercise, Q1 from Section 1.5 in the Princeton book)

```
numbers = []
a = int(input("Give me a number: "))
numbers = numbers + [a]

a_str = input("Give me a number: ")
while a_str != '':
    a = int(a_str)
    numbers = numbers + [a]
    a_str = input("Give me a number: ")
```

The loop is stopped if the user hits only Enter (returning an empty string).

Recap: Entering data manually

A little fancier (and better): Exception handling

```
numbers = []
a_str = input("Give me a number (or simply 'Enter' to end reading) : ")

while a_str != '':
    try:
        a = int(a_str)
    except ValueError:
        print("Last input was not a number. I stop reading input.")
        break
    numbers = numbers + [a]
    a_str = input("Give me a number (or simply 'Enter' to end reading) : ")
```

More on Exception handling in Python: https://docs.python.org/3/tutorial/errors.html

2) Make up some data

Entering data manually is tedious.

2) Make up some data

```
import random

rN = []
for i in range(10):
    r = random.random()
    rN = rN + [r]
print(rN)
```

This prints something like:

```
[0.9600347071275791, 0.8962529234181172, 0.3497583189612403,
0.1744407926779697 5, 0.8017287245819745, 0.08828001263443297,
0.953900870869054, 0.5539364293176726, 0.1415746654329738,
0.2681635941027083]
```

(random numbers drawn from uniform distribution)

2) Make up some data

```
import random

random.seed(24102019)
rN = []
for i in range(10):
    r = random.random()
    rN = rN + [r]
print(rN)
```

This always prints:

```
[0.3455885141390541, 0.23728051859614996, 0.42269425812458084,
0.005893225371031874, 0.310890120026234 6, 0.6402619888670006,
0.20329670623540808, 0.5741181022063454, 0.16080577229727 8,
0.7714901240178259]
```

2) Make up some data

Sometimes it is better to have integers (e.g., Q5 from Section 1.5 … longest run in a sequence)

```
import random

rN = []
for i in range(20):
    r = random.randint(0,10)
    rN = rN + [r]
print(rN)
```

This prints something like:

```
[7, 10, 9, 5, 3, 9, 9, 5, 3, 6, 6, 3, 8, 6, 1, 2, 4, 5, 0, 8]
```

3) Input from text files

There are various ways how to read and write files in Python.

I show you my preferred one.

It uses Numpy.

3) Input from text files

Put this at the top of your Python code:

```
import numpy as np
import matplotlib.pyplot as plt
```

Using Anaconda (i.e. in Spyder) the above should work.
If not, type this at the OS command prompt first:

```
> python -m pip install --user numpy matplotlib
```

3) Input from text files

Save the random numbers to a file called someNumbers.txt (created in the current working directory):

```
np.savetxt("someNumbers.txt",rN)
```

With a little bit of formatting:

```
np.savetxt("someNumbers.txt",rN,"%15.12f")
```

or

```
np.savetxt(fname="someNumbers.txt",X=rN,fmt="%15.12f")
```

Each number is at most 15 characters wide with exactly 12 after the decimal point.

3) Input from text files

Read the numbers back from someNumbers.txt:

```
Data_arr = np.loadtxt("someNumbers.txt")
```

Data_arr is a numpy array (more later). To turn it into a list:

```
Data = list(Data_arr)
```

Slightly more sophisticated example

```
fileName = "AberdeenDyce.txt"
Data_arr = np.loadtxt(fileName,skiprows=1)
```

To get the header:

```
ColNames_arr = \
np.genfromtxt(fileName,dtype="str",skip_footer=12,delimiter='\t‘)
```

(\ line continuation character)

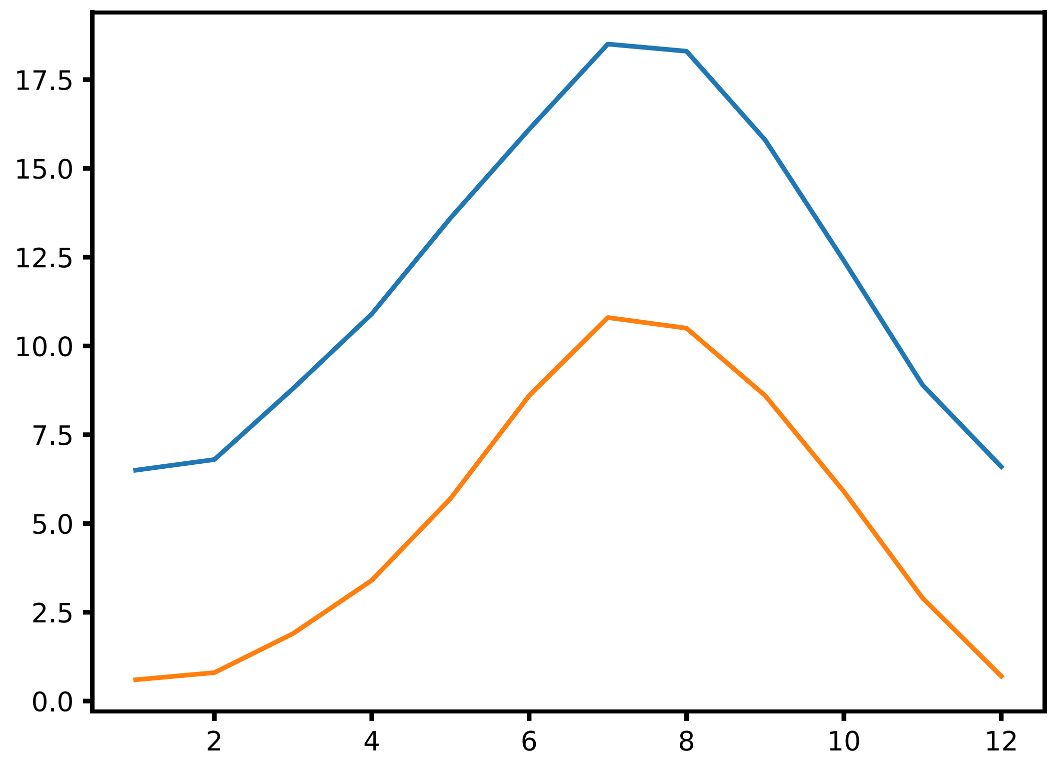
Again, turn it into a list:

```
ColNames = list(ColNames_arr)
```

Slightly more sophisticated example

```
month = list(Data_arr[:,0])
maxTemp = list(Data_arr[:,1])
minTemp = list(Data_arr[:,2])

plt.figure()
plt.plot(month,maxTemp)
plt.plot(month,minTemp)
```



17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12