CS1028: Programming for Sciences and Engineering

Lecture 12

25/10/2019

Daniel Vogel

University of Aberdeen

Outline

1) More on graphics
2) More on Numpy

Remember:

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

The max temperature from the Dyce data

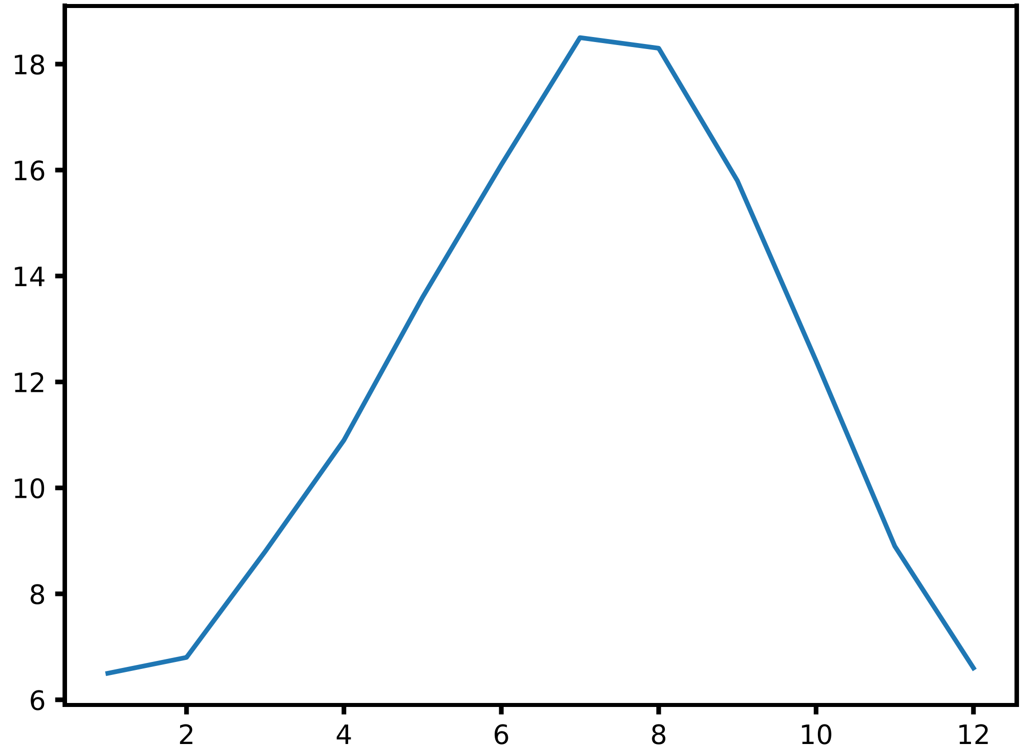
```
Data_arr = np.loadtxt("AberdeenDyce.txt",skiprows=1)
month = list(Data_arr[:,0])
maxTemp = list(Data_arr[:,1])

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

12 points connected by straight lines,

`month` contains their x-coordinates,

`maxTemp` the corresponding y-coordinates



Pimping the plot

```
plt.plot(month,maxTemp, color="green", marker="o", linestyle="dashed")
```

This does the same:

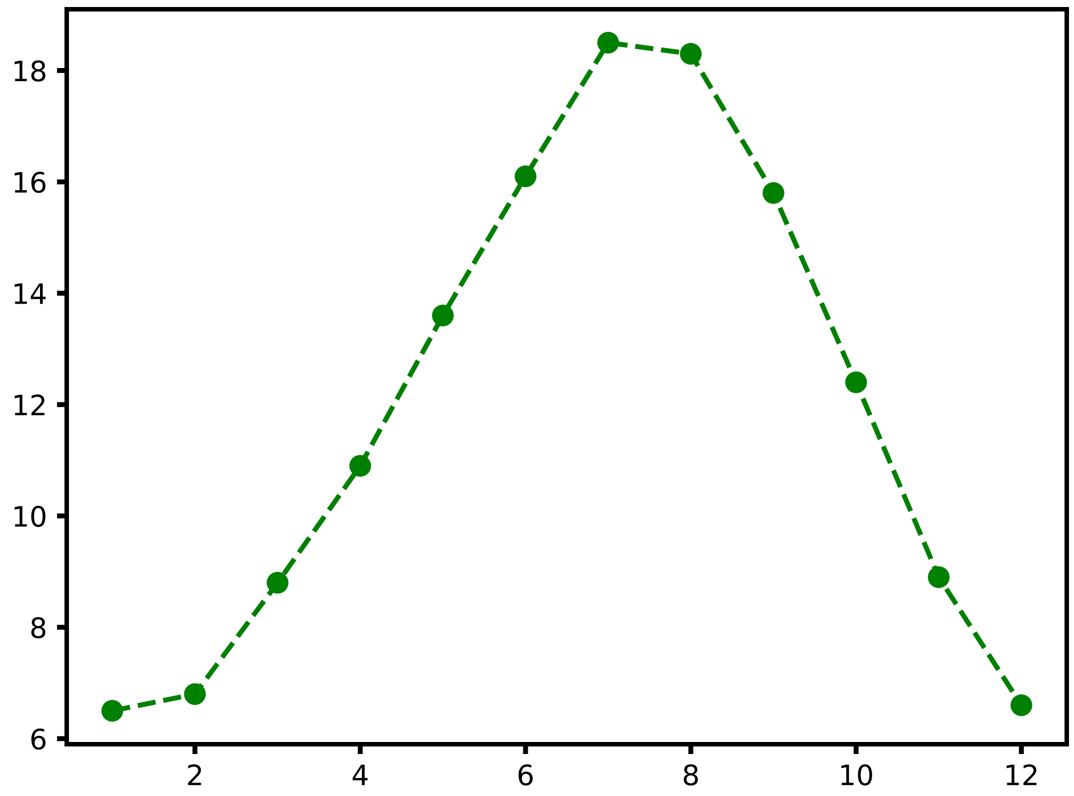
```
plt.plot(month,maxTemp,"g--o")
```

Or magenta diamonds, no lines:

```
plt.plot(month,maxTemp,"md")
```

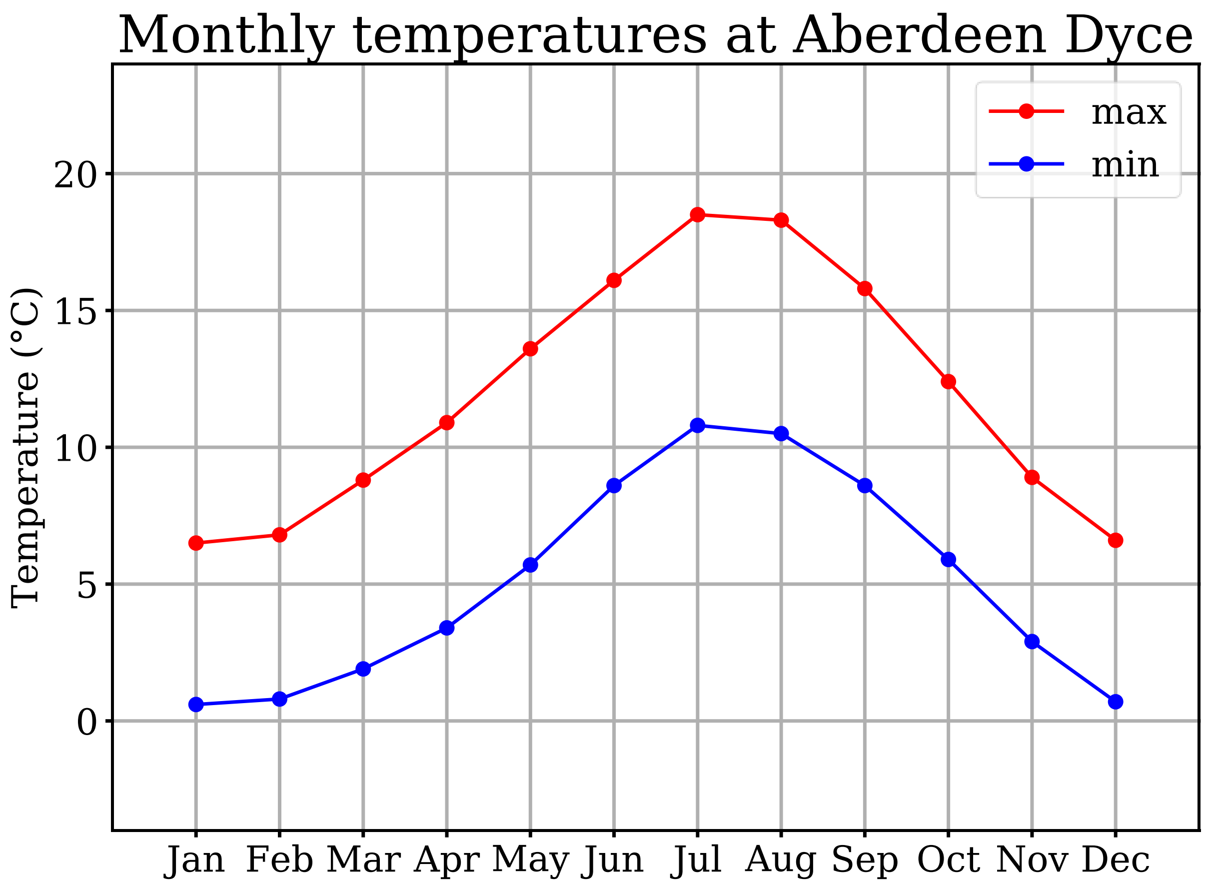
More info:

https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html



Pimping the plot further

```
plt.figure()
fontText = {'family': 'serif', 'size': 18}
fontTitle = {'family': 'serif', 'size': 26}
plt.rc('font', **fontText)
ax = plt.axes()
ax.xaxis.set_major_locator\
(plt.FixedLocator([1,2,3,4,\
5,6,7,8,9,10,11,12]))
ax.xaxis.set_major_formatter\
(plt.FixedFormatter(["Jan","Feb"\
,"Mar","Apr","May","Jun","Jul",\
"Aug","Sep","Oct","Nov","Dec"]))
ax.grid()
plt.plot(month,maxTemp,"r-o",label="max")
plt.plot(month,minTemp,"b-o",label="min")
plt.legend()
plt.xlim(0,13)
plt.ylim(-4,24)
plt.ylabel('Temperature (°C)')
plt.title('Monthly temperatures at Aberdeen Dyce',fontdict=fontTitle)
```



Before we get back to Numpy arrays …let's get back to lists (cf. Lecture 07)

```
L = [3,4,17]
```

Elements can have different types:

```
K = [3,4,"cheese"]
M = [3,4,"cheese", [1,2,17]]
```

Accessing single elements:

```
L[1]
K[0] = "hello"
```

A handy way to create lists: numbers from 10 to 20 (note: these are 11 numbers)

```
N = list(range(10,21))
```

The length

```
len(N)
```

List slicing:

```
N[0:5]
N[0:5:2]
N[3:-2]
N[:5]
N[:]
```

Aliasing:

```
N2 = N
N[0] = -3.1415
N2
```

Numpy arrays

```
import numpy as np
```

The core of numpy is the data type ndarray

```
A = np.array([[1,2,3],[5,6,7]])
A
array([[1, 2, 3],
       [5, 6, 7]])

type(A)
numpy.ndarray

np.shape(A)
(2, 3)
```

accessing individual elements and parts by list slicing syntax:

```
A[1,1]
6
```

this extracts the second column

```
A[0:2,1]
```

this does the same:

```
A[:,1]
```

numpy also contains functions that work on arrays

```
import math
np.sqrt(2) == math.sqrt(2)
True

np.sqrt(A)
array([[1.        , 1.41421356, 1.73205081],
       [2.23606798, 2.44948974, 2.64575131]])
```

Simple arithmetics work element-wise

```
B = A + 1
C = A * B
array([[ 2,  6, 12],
       [30, 42, 56]])
```

Some handy funtions creating arrays:

```
N = np.ones((4,3))
Z = np.zeros((4,3))
L = np.linspace(-2,2,100)
```

Connecting the dots…

Task: create this plot