EVEN MORE PYTHON..

SESSION 3

EVEN MORE PYTHON

- FILES
- FASTA FILES
- PACKAGES

FILES

- When we want to read from or write to a file, we need to open it first, and when we are done, it needs to be closed (so its better to implement the closing step before the operations.)

THERE ARE FOUR DIFFERENT METHODS (MODES) FOR OPENING A FILE:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exists

OPENING A FILE

APPEND MODE

```
f = open("filenew.txt", 'a')
f.close()
```

OPENING A FILE
CREATE MODE

Creates the file

```
f = open("fileOne.txt", 'x')
f.close()
```

Returns an error (the file exists)

```
f = open("filenew.txt", 'x')
f.close()
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4968/696236040.py in <module>
----> 1 f = open("filenew.txt", 'x')
      2 f.close()

FileExistsError: [Errno 17] File exists: 'filenew.txt'
```

READING FROM TEXT FILES:-

Files return strings even if we store numbers, so if we want to use them as numbers not strings we have to use ‘int()’ or ‘float()’ functions.

• .READ()

- RETURN THE WHOLE TEXT IN THE FILE.

```
f = open("file.txt", 'r')
print(f.read())
f.close()
```

```
hello
second
```

.READLINE()

- Return one line then the cursor moves to the next line in the file.

```
f= open("file.txt")
print(f.readline())
f.close()
#is just enough, you dont have to set the mode to 'rt' as its default
```

```
hello
```

- .READLINES()
- Return all lines in a list, where each item represents a line.

```
f= open("file.txt")
print(f.readlines())
f.close()
```

```
['hello\n', 'second']
```

WRITING IN TEXT FILES:-

- ● .write(string)
- Write a given string in the file, if it's on 'a' mode it writes at the
- end of file, if it's on 'w' mode it will overwrite any text found.

```
f= open("file.txt", 'w')
f.write("hello again")
```

- .WRITELINES(LIST OF STRINGS)
  - Write a given list of strings in the file, all on the same line if a line break isn't inserted inside the list

```
f=open("file.txt", 'w')
f.writelines(["hello",'\n',"second"])
```

```
hello
second
```

FASTA FILES:-

- In bioinformatics, FASTA is a writing method used to write files containing nucleotide sequences (DNA or RNA sequences) or

```
from Bio import SeqIO
fasta =SeqIO. parse(open('fastaFile'),'fasta')
```

- You can deal with them like normal files

They involve:

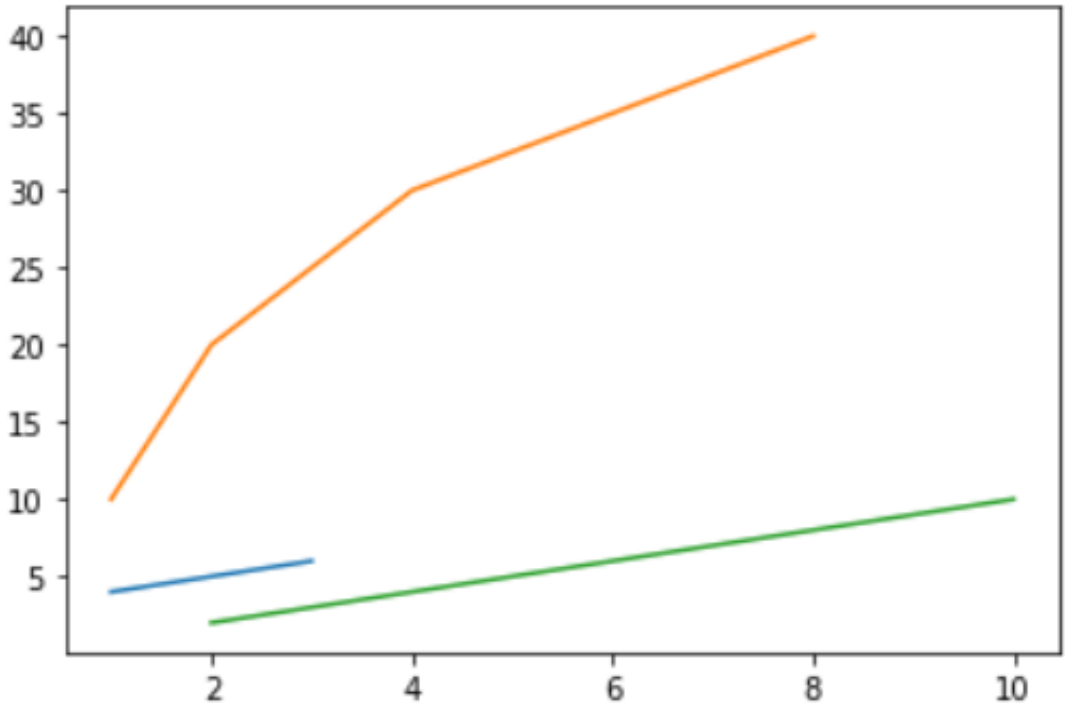
>Sequence ID

Sequence

PACKAGES:

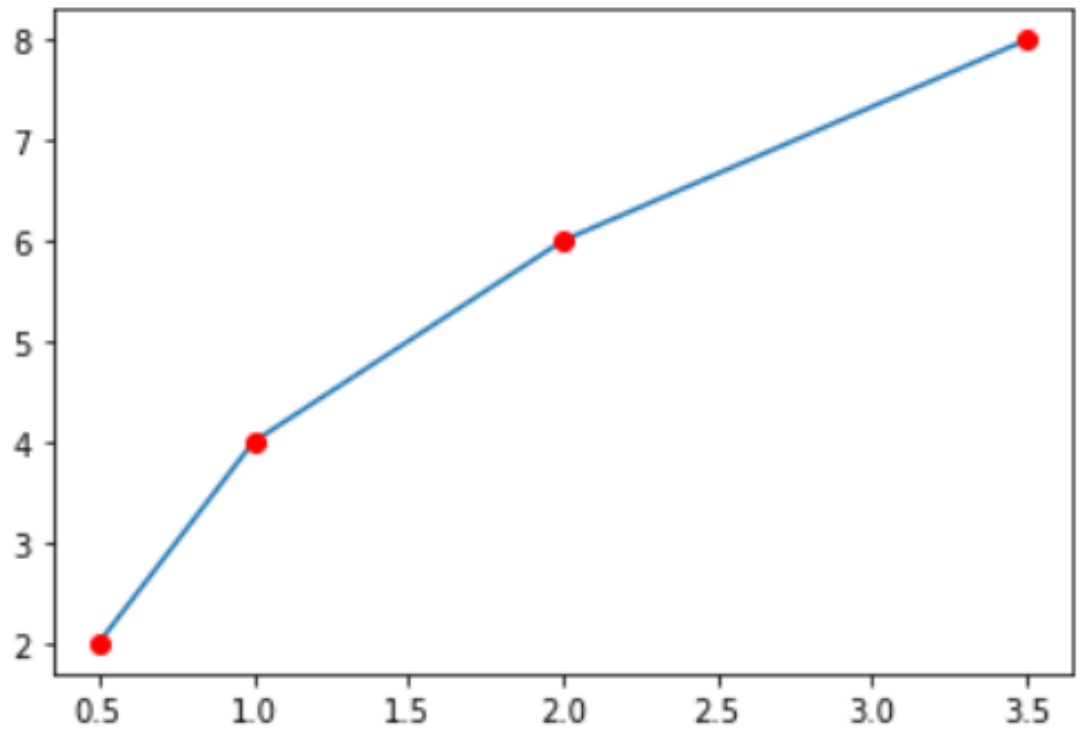
- **matplotlib:-**
- Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
- Matplotlib was created by John D. Hunter.
- Matplotlib is open source and we can use it freely.
- Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

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

plt.plot([1, 2, 3], [4, 5, 6])
plt.plot([1, 2, 4, 8], [10, 20, 30, 40])
plt.plot([2, 4, 6, 8, 10], [2, 4, 6, 8, 10])
```



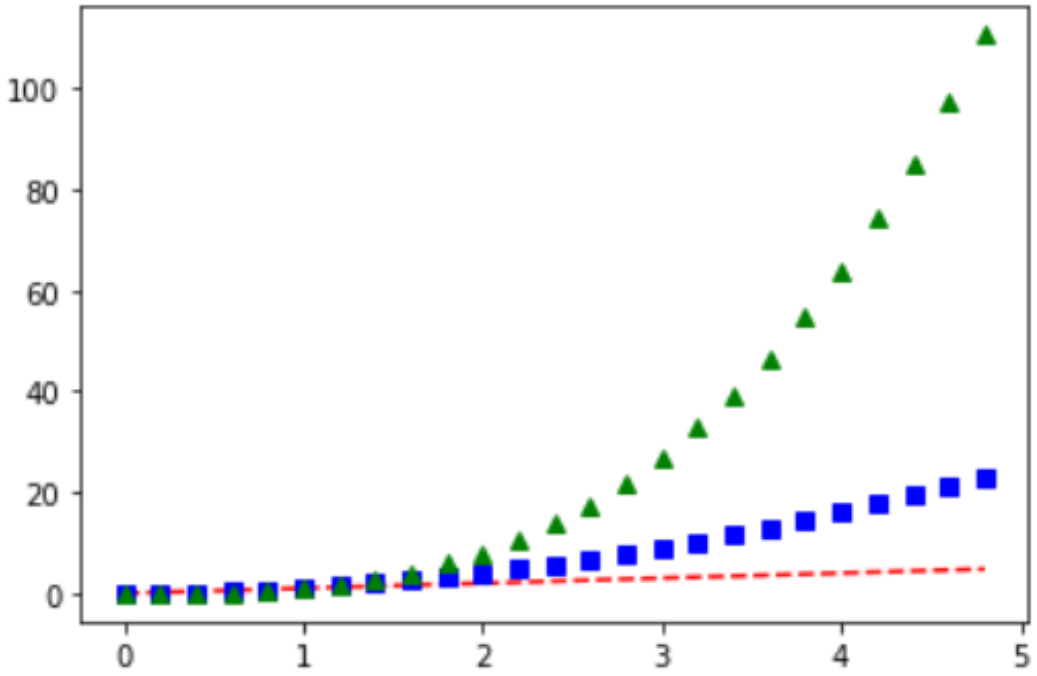
```python
import matplotlib.pyplot as plt
import numpy as np



plt.plot([0.5, 1, 2, 3.5], [2, 4, 6, 8])
plt.plot([0.5, 1, 2, 3.5], [2, 4, 6, 8], 'ro')
```

(ro) selects the point on the graph by points



You can use it to plot functions in time

```
# red dashes, blue squares and green triangles
t = np.arange(0., 5., 0.2) #time at 200ms intervals
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
```



OTHER PLOTS LIKE PIE PLOT

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

y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```

BAR PLOT

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

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```