syncbio

August 7, 2024

[4]:
```
import numpy as np
```

[3]:
```
data=np.loadtxt(fname="/home/bharath/Downloads/inflammation-01.
↪csv",delimiter=",")
```

[12]:
```
print(np.mean(data))
```

```
6.14875
```

[15]:
```
print(np.amax(data))
```

```
20.0
```

[17]:
```
print(np.amin(data))
```

```
0.0
```

[20]:
```
print(np.std(data))
```

```
4.613833197118566
```

[22]:
```
print(np.mean(data,axis=1))
```

```
[5.45  5.425 6.1   5.9   5.55  6.225 5.975 6.65  6.625 6.525 6.775 5.8
 6.225 5.75  5.225 6.3   6.55  5.7   5.85  6.55  5.775 5.825 6.175 6.1
 5.8   6.425 6.05  6.025 6.175 6.55  6.175 6.35  6.725 6.125 7.075 5.725
 5.925 6.15  6.075 5.75  5.975 5.725 6.3   5.9   6.75  5.925 7.225 6.15
 5.95  6.275 5.7   6.1   6.825 5.975 6.725 5.7   6.25  6.4   7.05  5.9  ]
```

[24]:
```
print(np.std(data,axis=0))
```

```
[0.         0.49749372 0.73238575 1.05869416 1.1308797  1.37628728
 1.71075032 1.95014244 1.9267128  2.26194361 2.74119803 2.50798724
 3.13488437 3.0922843  3.75040739 4.02285139 3.82226663 3.54008788
 3.94701012 4.50120354 4.24116729 4.54960316 4.19907397 4.09945796
 3.7103459  3.60863163 3.21364279 2.77863876 2.68121034 3.19526559
 2.5157283  2.10890967 1.62369606 1.78138523 1.76351921 1.38283123
 1.11790976 1.161895   0.8055364  0.49553562]
```

[26]:
```
data[:3,36:]
```

[26]:
```
array([[2., 3., 0., 0.],
       [1., 1., 0., 1.],
       [2., 2., 1., 1.]])
```

[29]:
```
import time
print(time.ctime())
```

```
Thu Aug  1 21:51:01 2024
```

[32]:
```
patient_0=data[0,:]
maximum=np.amax(patient_0)
maximum
```

[32]:
```
18.0
```

[35]:
```
A= np.array([[1,2,3],[4,5,6],[7,8,9]])
A
```

[35]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[40]:
```
horizontal=np.hstack([A,A])
horizontal
```

[40]:
```
array([[1, 2, 3, 1, 2, 3],
       [4, 5, 6, 4, 5, 6],
       [7, 8, 9, 7, 8, 9]])
```

[44]:
```
vertical=np.vstack([A,A])
vertical
```

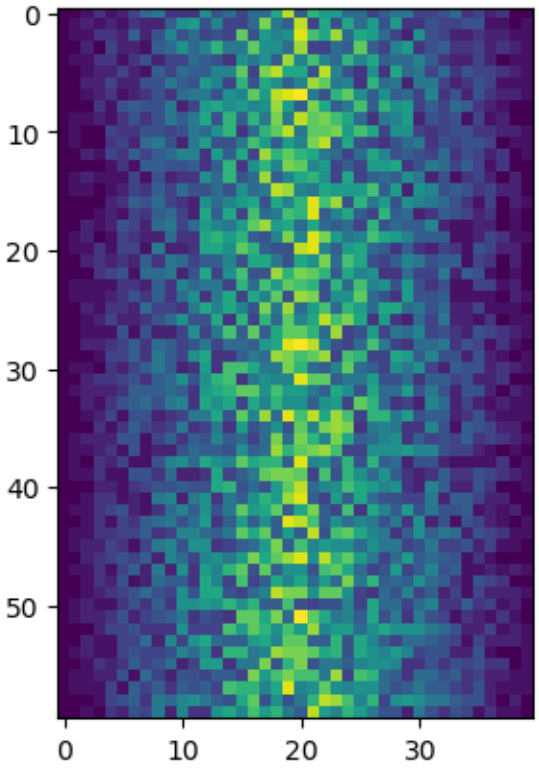
[44]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9],
       [1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[5]:
```
patient3_7=data[3,:7]
np.diff(patient3_7)
```

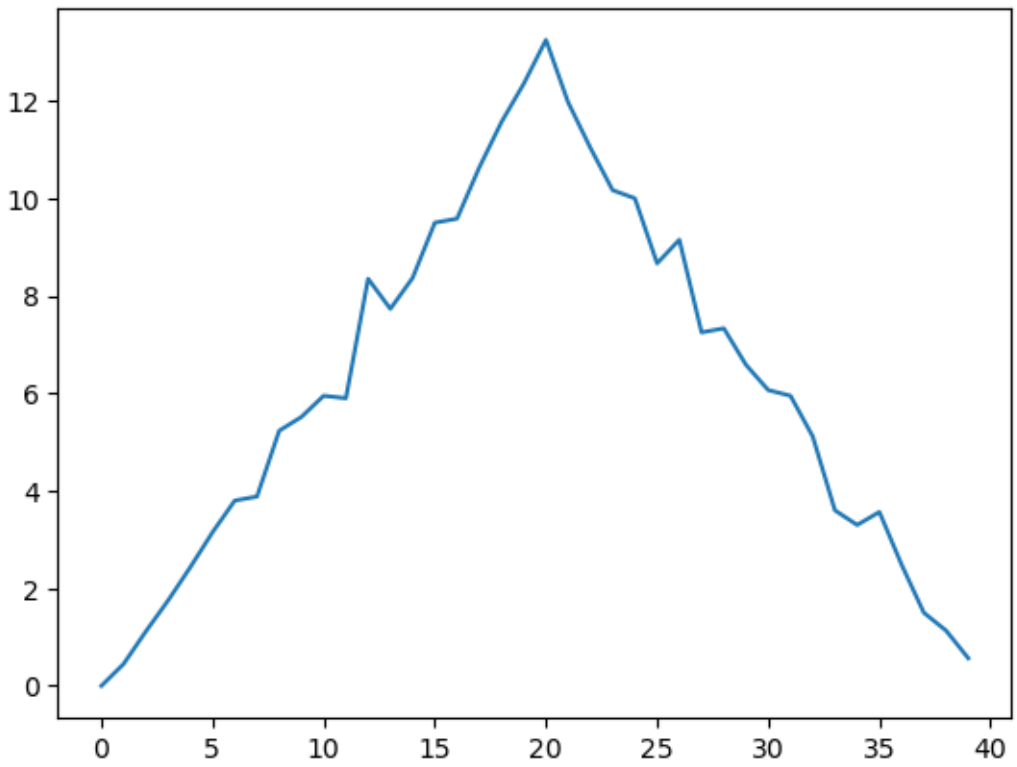
[5]:
```
array([ 0.,  2., -2.,  4., -2.,  0.])
```

[8]:
```
from matplotlib import pyplot as plt
plt.imshow(data)
```

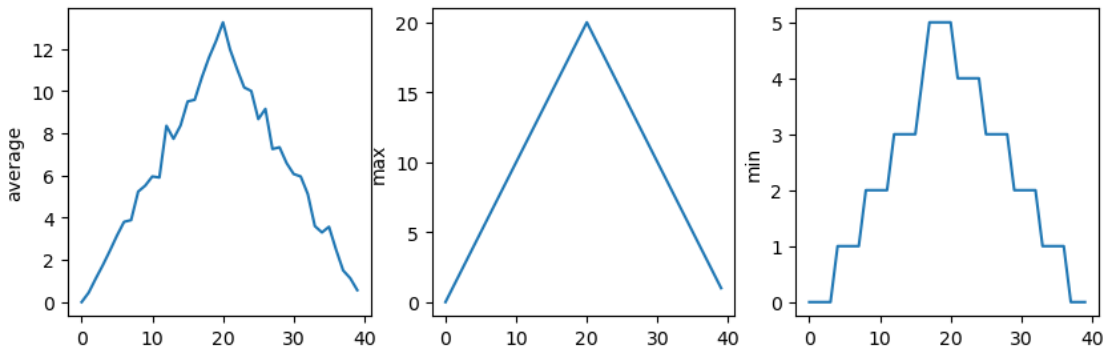
[8]:
```
<matplotlib.image.AxesImage at 0x7ae77eaeff10>
```



[13]:
```
average=np.mean(data,axis=0)
plt.plot(average)
plt.show()
```

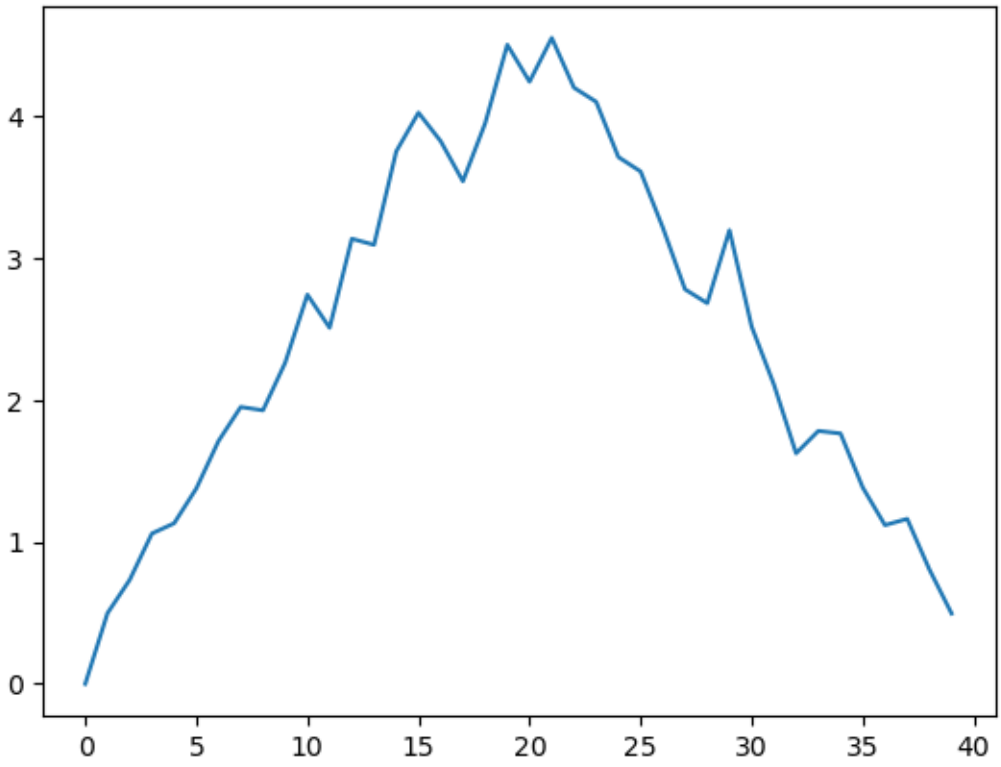


[21]:
```
fig=plt.figure(figsize=(10.0,3.0))

axes1=fig.add_subplot(1,3,1)
axes2=fig.add_subplot(1,3,2)
axes3=fig.add_subplot(1,3,3)

axes1.set_ylabel('average')
axes1.plot(np.mean(data, axis=0))

axes2.set_ylabel('max')
axes2.plot(np.amax(data, axis=0))

axes3.set_ylabel('min')
axes3.plot(np.amin(data, axis=0))


plt.savefig("inflammation.png")
```



[24]:
```
standard_deviation=np.std(data,axis=0)
plt.plot(standard_deviation)
```

[24]:
```
[<matplotlib.lines.Line2D at 0x7ae77d113e50>]
```



[26]:
```
veg = [['lettuce', 'lettuce', 'peppers', 'zucchini'],
     ['lettuce', 'lettuce', 'peppers', 'zucchini'],
     ['lettuce', 'cilantro', 'peppers', 'zucchini']]
```

[30]:
```
veg[2][0]
```

[30]:
```
'lettuce'
```

[33]:
```
odds = [1, 3, 5, 7]
for num in odds:
    print(num)
```

```
1
3
5
7
```

[36]:
```
length = 0
names = ['Curie', 'Darwin', 'Turing','Bharath']
for value in names:
    length = length + 1
print('There are', length, 'names in the list.')
```

```
There are 4 names in the list.
```

[39]:
```
length=0
mylist=[1,2,3,4,5,6]
for x in mylist:
    length=length+1
print(length)
```

```
6
```

[41]:
```
5**3
```

[41]:
```
125
```

[48]:
```
result=1
for x in range(0,3):
    result=result*5
print(result)
```

```
125
```

[49]:
```
result = 1
for number in range(0, 3):
    result = result * 5
print(result)
```

```
125
```

[51]:
```
mylist=[124, 402, 36]
count=0
for x in mylist:
    count=count+x
print(count)
```

562

[54]:
```
cd csv
```

/home/bharath/pythonworkspace/csv

/home/bharath/.local/lib/python3.10/site-packages/IPython/core/magics/osm.py:417: UserWarning: This is now an optional IPython functionality, setting dhist requires you to install the `pickleshare` library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]

[56]:
```
import glob
print(glob.glob('inflammation*.csv'))
```

['inflammation-04.csv', 'inflammation-06.csv', 'inflammation-10.csv', 'inflammation-11.csv', 'inflammation-12.csv', 'inflammation-03.csv', 'inflammation-08.csv', 'inflammation-02.csv', 'inflammation-05.csv', 'inflammation-01.csv', 'inflammation-07.csv', 'inflammation-09.csv']

[58]:
```
ls
```

inflammation-01.csv inflammation-05.csv inflammation-09.csv
inflammation-02.csv inflammation-06.csv inflammation-10.csv
inflammation-03.csv inflammation-07.csv inflammation-11.csv
inflammation-04.csv inflammation-08.csv inflammation-12.csv

[60]:
```
cd ..
```

/home/bharath

[62]:
```
ls
```

anaconda3/ docker/
finalpipelines/   mamba/
Screenshots/
ankur/   dockercourse/   final.tar
mapping/   skype.deb
autodock4* dockerexample/ germline/
miniconda3/   snakemake/
bin/   Documents/   git/
movies/   snakemakenew/

```
Bioinfo/    Downloads/      google-cloud-
sdk/  nextflow/         snap/
check/      fasta/          igv/
pipelines/        somatic/
course/     fastaglob/      interview/
portfolio/        teams.deb
data/       file.txt        Lifecel/
Python-3.11.8/    thesis/
datalist/   final/          Lifecell/
pythonworkspace/  Wallpapers/
Desktop/    final1/         linuxcommands/
rna/              work/
```

[64]:
```
cd pythonworkspace/
```

```
[Errno 2] No such file or directory: 'pythonworkspace/'
/home/bharath/pythonworkspace
```

```
/home/bharath/.local/lib/python3.10/site-
packages/IPython/core/magics/osm.py:393: UserWarning: This is now an optional
IPython functionality, using bookmarks requires you to install the `pickleshare`
library.
  bkms = self.shell.db.get('bookmarks', {})
```

[66]:
```
cd csv
```

```
[Errno 2] No such file or directory: 'csv'
/home/bharath/pythonworkspace/csv
```

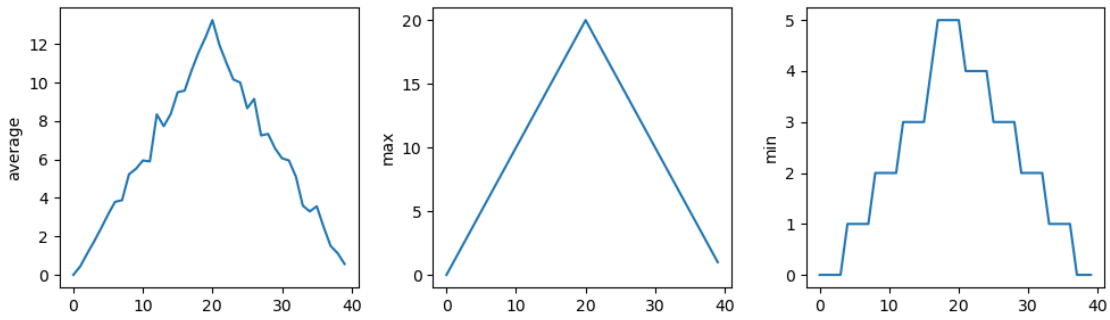
[68]:
```
ls
```

```
inflammation-01.csv  inflammation-05.csv  inflammation-09.csv
inflammation-02.csv  inflammation-06.csv  inflammation-10.csv
inflammation-03.csv  inflammation-07.csv  inflammation-11.csv
inflammation-04.csv  inflammation-08.csv  inflammation-12.csv
```

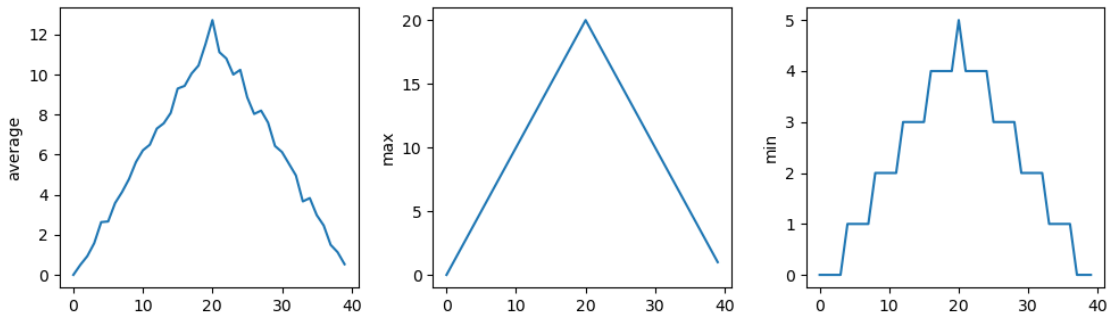
[74]:
```
import glob
import numpy
import matplotlib.pyplot

filenames = sorted(glob.glob('inflammation*.csv'))
filenames = filenames[0:3]
for filename in filenames:
    print(filename)

    data = numpy.loadtxt(fname=filename, delimiter=',')

    fig = matplotlib.pyplot.figure(figsize=(10.0, 3.0))
```

```
    axes1 = fig.add_subplot(1, 3, 1)
    axes2 = fig.add_subplot(1, 3, 2)
    axes3 = fig.add_subplot(1, 3, 3)

    axes1.set_ylabel('average')
    axes1.plot(numpy.mean(data, axis=0))

    axes2.set_ylabel('max')
    axes2.plot(numpy.amax(data, axis=0))

    axes3.set_ylabel('min')
    axes3.plot(numpy.amin(data, axis=0))

    fig.tight_layout()
    matplotlib.pyplot.show()
```

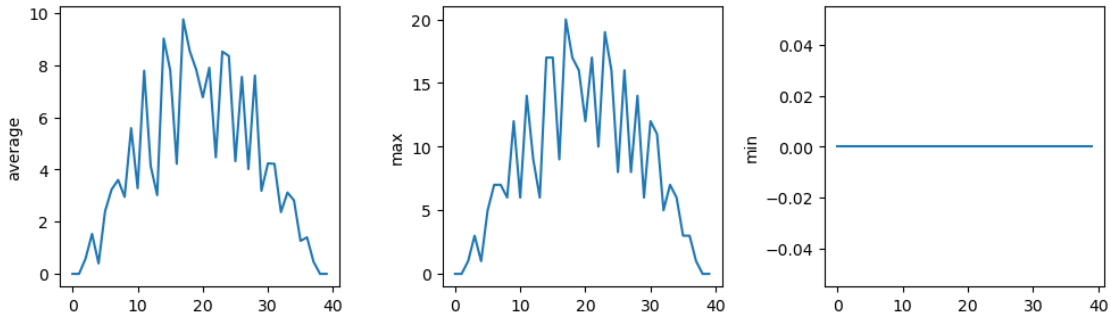
inflammation-01.csv



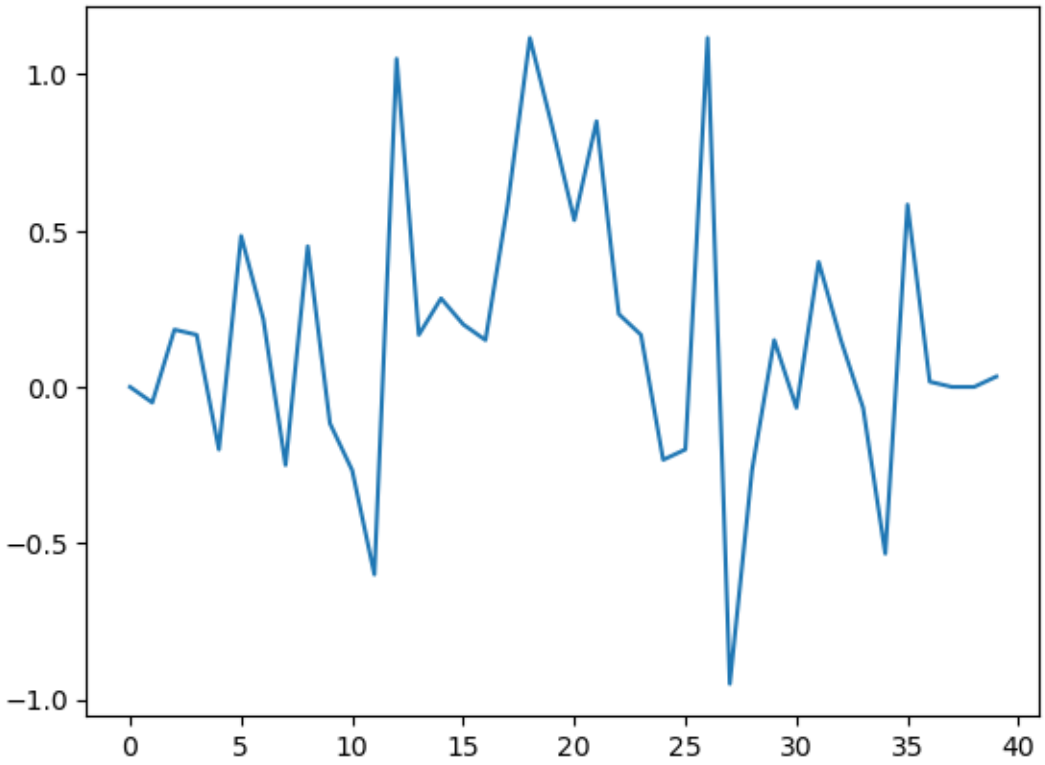
inflammation-02.csv



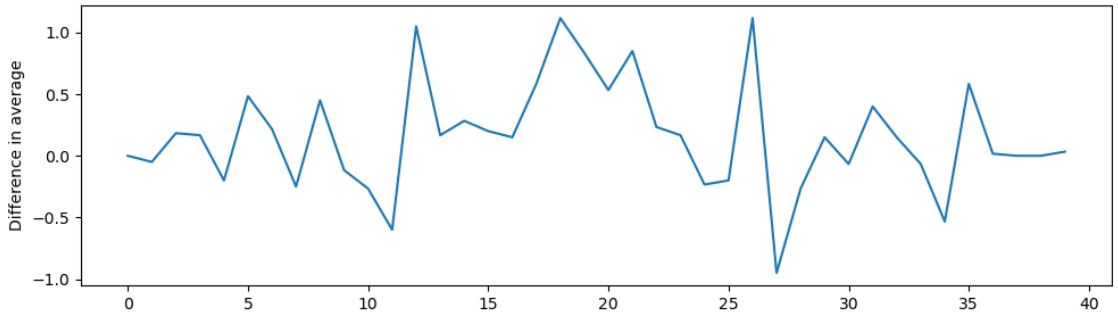
inflammation-03.csv



[83]:
```
import glob
import numpy as np
from matplotlib import pyplot as plt

files=sorted(glob.glob("inflammation*.csv"))

data0=np.loadtxt(fname="inflammation-01.csv",delimiter=",")
data1=np.loadtxt(fname="inflammation-02.csv",delimiter=",")

mean0=np.mean(data0,axis=0)
mean1=np.mean(data1,axis=0)

mean_diff=mean0-mean1

my_plot=plt.plot(mean_diff)
```



[85]:
```
import glob
import numpy
import matplotlib.pyplot

filenames = sorted(glob.glob('inflammation*.csv'))

data0 = numpy.loadtxt(fname=filenames[0], delimiter=',')
data1 = numpy.loadtxt(fname=filenames[1], delimiter=',')

fig = matplotlib.pyplot.figure(figsize=(10.0, 3.0))

matplotlib.pyplot.ylabel('Difference in average')
matplotlib.pyplot.plot(numpy.mean(data0, axis=0) - numpy.mean(data1, axis=0))

fig.tight_layout()
matplotlib.pyplot.show()
```



[2]:
```
num=50

if num>100:
    print("greater than 100")
else:
    print("lesser than 100")
```

```
lesser than 100
```

[8]:
```
def posorneg(num):
    if num < 0:
        return "negative"
    elif num == 0:
        return "0"
    else:
        return "positive"
```

[9]:
```
posorneg(5)
```

[9]: 'positive'

[14]:
```
def input_num(a,b):
    if a==33 and b==33:
        return True
    else:
        return False
```

[15]:
```
input_num(32,33)
```

[15]: False

[22]:
```
def atleast_one(a,b):
    if a<=1 or b<=1:
        return True
    else:
```

```
        return False
```

[24]:
```
atleast_one(1,2)
```

[24]: True

[27]:
```
cd csv
```

```
/home/bharath/pythonworkspace/csv
```

```
/home/bharath/.local/lib/python3.10/site-
packages/IPython/core/magics/osm.py:417: UserWarning: This is now an optional
IPython functionality, setting dhist requires you to install the `pickleshare`
library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]
```

[29]:
```
ls
```

```
inflammation-01.csv  inflammation-05.csv  inflammation-09.csv
inflammation-02.csv  inflammation-06.csv  inflammation-10.csv
inflammation-03.csv  inflammation-07.csv  inflammation-11.csv
inflammation-04.csv  inflammation-08.csv  inflammation-12.csv
```

[38]:
```
import numpy
data = numpy.loadtxt(fname='inflammation-03.csv', delimiter=',')

max_inflammation_0 = numpy.amax(data, axis=0)[0]
max_inflammation_20 = numpy.amax(data, axis=0)[20]

if max_inflammation_0 == 0 and max_inflammation_20 == 20:
    print('Suspicious looking maxima!')
elif numpy.sum(numpy.amin(data, axis=0)) == 0:
    print('Minima add up to zero!')
else:
    print('Seems OK!')
```

```
Minima add up to zero!
```

[41]:
```
def ten_percent(a,b):
    b_value=b/10*100
    if a<b_value:
        return True
    else:
        return False
```

[46]:
```
ten_percent(5,5.1)
```

[46]: True

[55]:
```
a=5
b=3
if abs(a-b) < 0.1* abs(b):
    print("True")
else:
    print("False")
```

False

[58]:
```
#6

a=-3
a+=9
a
```

[58]: 6

[64]:
```
def returnposorneg(mylist):
    positive=0
    negative=0
    for x in mylist:
        if x > 0:
            return positive + x
        else:
            return negative + x
```

[66]:
```
returnposorneg([1,2,3,0,-1,-2,-3])
```

[66]: 1

[72]:
```
filenames = ['inflammation-01.csv',
             'myscript.py',
             'inflammation-02.csv',
             'small-01.csv',
             'small-02.csv']
large_files = []
small_files = []
other_files = []

for filename in filenames:
    if filename.startswith("inflammation"):
        large_files.append(filename)
    elif filename.startswith("s"):
```

```
        small_files.append(filename)
    else:
        pass
```

[75]:
```
print(large_files)
print(small_files)
```

```
['inflammation-01.csv', 'inflammation-02.csv']
['small-01.csv', 'small-02.csv']
```

[79]:
```
def vowels(mystring):
    for x in mystring:
        if x == "a":
            print(x.count("a"))
```

[80]:
```
vowels("bharath")
```

```
1
1
```

[83]:
```
vowels="aeiouAEIOU"
words="It is a small sentence"
count=0
for x in words:
    if x in vowels:
        count+=1
```

[85]:
```
print(count)
```

```
7
```

[89]:
```
fahrenheit=10
celcius=((fahrenheit-32)*(5/9))
celcius
```

[89]:
```
-12.222222222222223
```

[93]:
```
def fah_to_cel(temp):
    celcius=((temp-32)*(5/9))
    return celcius
```

[94]:
```
fah_to_cel(30)
```

[94]:
```
-1.1111111111111112
```

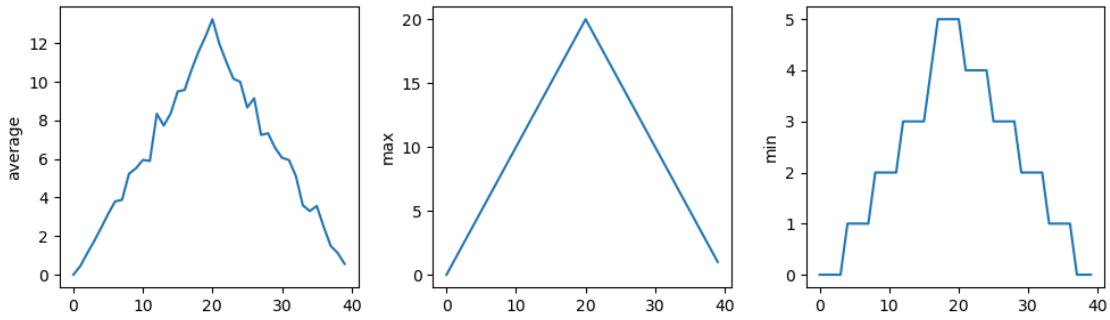
[95]:
```
def cel_kel(value):
    return value + 273.15
```

[103]:
```
import matplotlib.pyplot
import numpy
import glob

def visualize(filename):

    data = numpy.loadtxt(fname=filename, delimiter=',')

    fig = matplotlib.pyplot.figure(figsize=(10.0, 3.0))

    axes1 = fig.add_subplot(1, 3, 1)
    axes2 = fig.add_subplot(1, 3, 2)
    axes3 = fig.add_subplot(1, 3, 3)

    axes1.set_ylabel('average')
    axes1.plot(numpy.mean(data, axis=0))

    axes2.set_ylabel('max')
    axes2.plot(numpy.amax(data, axis=0))

    axes3.set_ylabel('min')
    axes3.plot(numpy.amin(data, axis=0))

    fig.tight_layout()
    matplotlib.pyplot.show()
```

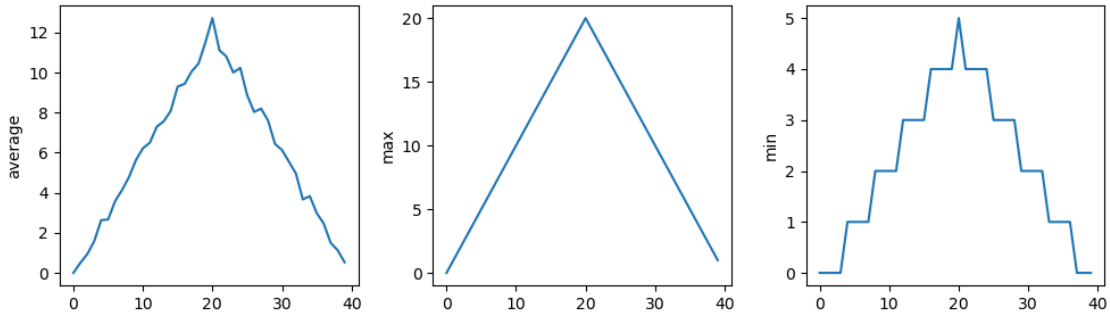
[104]:
```
def detect_problems(filename):

    data = numpy.loadtxt(fname=filename, delimiter=',')

    if numpy.amax(data, axis=0)[0] == 0 and numpy.amax(data, axis=0)[20] == 20:
        print('Suspicious looking maxima!')
    elif numpy.sum(numpy.amin(data, axis=0)) == 0:
        print('Minima add up to zero!')
    else:
        print('Seems OK!')
```

[106]:
```
filenames = sorted(glob.glob('inflammation*.csv'))

for filename in filenames[:3]:
    print(filename)
    visualize(filename)
    detect_problems(filename)
```

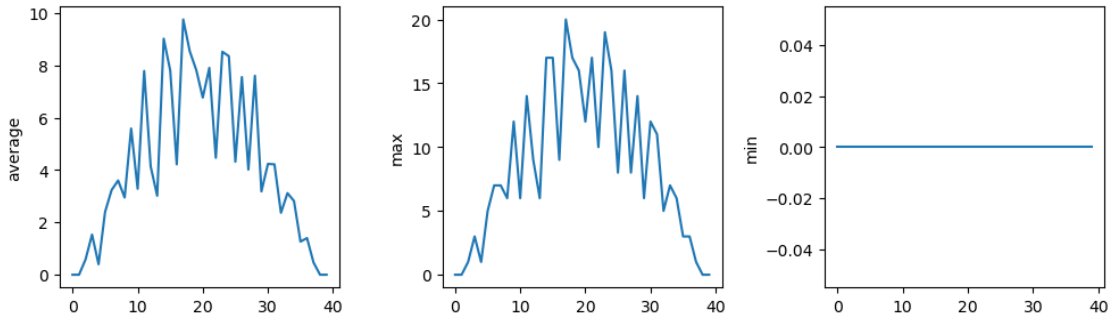
```
inflammation-01.csv
```




```
Suspicious looking maxima!
inflammation-02.csv
```




```
Suspicious looking maxima!
inflammation-03.csv
```




```
Minima add up to zero!
```

[121]:
```
def starring(name):
    return "".join("*" +name+ "*")
```

[123]:
```
starring("Bharath")
```

[123]:
```
'*Bharath*'
```

[125]:
```
def fence(original, wrapper):
    return wrapper + original + wrapper
```

[129]:
```
fence("Bharath","*")
```

[129]:
```
'*Bharath*'
```

[131]:
```
def first_last(mystring):
    first_letter=mystring[0]
    last_letter=mystring[-1]
    return first_letter+last_letter
```

[133]:
```
first_last("muruga")
```

[133]:
```
'ma'
```

[142]:
```
def first_word(mystring):
    word=mystring.split()
    return str([x[0] for x in word])
```

[143]:
```
first_word("The name is Bharath")
```

[143]:
```
"['T', 'n', 'i', 'B']"
```

[150]:
```
def some_function():
    msg = 'hello, world!'
    print(msg)
    return msg
```

[152]:
```
some_function()
```

```
hello, world!
```

[152]:
```
'hello, world!'
```

[157]:
```
def another_function():
    print('Syntax errors are annoying.')
    print('But at least Python tells us about them!')
    print('So they are usually not too hard to fix.')
```

[158]:
```
def another_function():
    print('Syntax errors are annoying.')
    print('But at least Python tells us about them!')
    print('So they are usually not too hard to fix.')
```

[161]:
```
numbers = [1.5, 2.3, 0.7, -0.001, 4.4]
total=0.0
for num in numbers:
    assert num > 0.0
    total+=num
print(total)
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
Cell In[161], line 4
      2 total=0.0
      3 for num in numbers:
----> 4     assert num > 0.0
      5     total+=num
      6 print(total)

AssertionError:
```

[11]:
```
import sys
import numpy as np

def main():
    script=sys.argv[0]
    action=sys.argv[1]
    filenames=sys.argv[2:]
    assert action in ["--min","--max","--mean"]
    if len(filenames)==0:
        process(sys.stdin,action)
    else:
        for file in filenames:
            process(filename,action)
def process(filename,action):
    data=np.loadtxt(file, delimiter=",")

    if action=="--min":
        values=np.amin(data,axis=0)
    elif action=="--max":
        values=np.amax(data,axis=0)
    elif action=="--mean":
        values=np.mean(data,axis=0)
    else:
```

```
        pass

    for val in values:
        print(val)

if __name__ =="__main__":
    main()
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
Cell In[11], line 30
     27         print(val)
     29 if __name__ =="__main__":
---> 30     main()

Cell In[11], line 8, in main()
      6 action=sys.argv[1]
      7 filenames=sys.argv[2:]
----> 8 assert action in ["--min","--max","--mean"]
      9 if len(filenames)==0:
     10     process(sys.stdin,action)

AssertionError:
```

[13]:
```
ls
```

```
analysis.py     Biopython.ipynb  files/            new.txt
test.ipynb
bharath.txt     course.ipynb     inflammation.png  newwrite.txt
Biology.ipynb   csv/             myfile.txt        python.ipynb
Biopython.html  fastq.txt        newfile.txt       syncbio.ipynb
```

[15]:
```
%%writefile syncbio.txt
ashwinth is a good boy
```

```
Overwriting syncbio.txt
```

[17]:
```
ls
```

```
analysis.py     Biopython.ipynb  files/            new.txt
syncbio.txt
bharath.txt     course.ipynb     inflammation.png  newwrite.txt
test.ipynb
Biology.ipynb   csv/             myfile.txt        python.ipynb
```

```
Biopython.html  fastq.txt        newfile.txt       syncbio.ipynb
```

[20]:
```
%%bash

echo "njdiaduqedeq" > syncbio1.txt
```

[22]:
```
ls
```

```
analysis.py     Biopython.ipynb  files/            new.txt
syncbio.ipynb
bharath.txt     course.ipynb     inflammation.png  newwrite.txt
syncbio.txt
Biology.ipynb   csv/             myfile.txt        python.ipynb
test.ipynb
Biopython.html  fastq.txt        newfile.txt       syncbio1.txt
```

ls