Live Plot in Matplotlib (Basics)

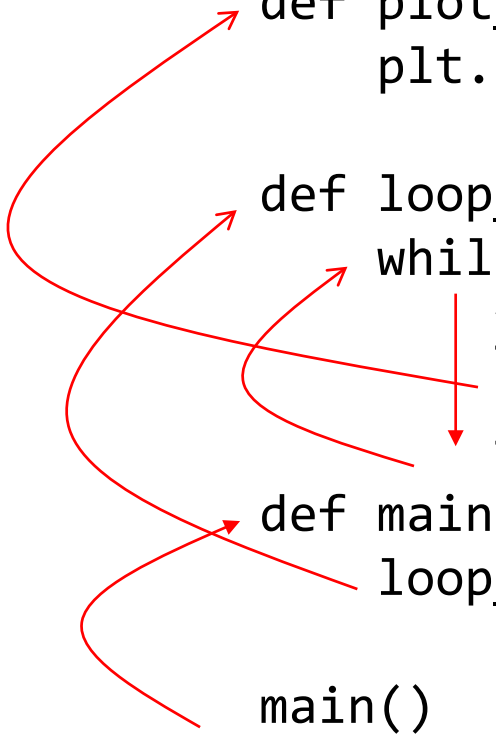
Saptarshi Ghosh

Plotting live graph

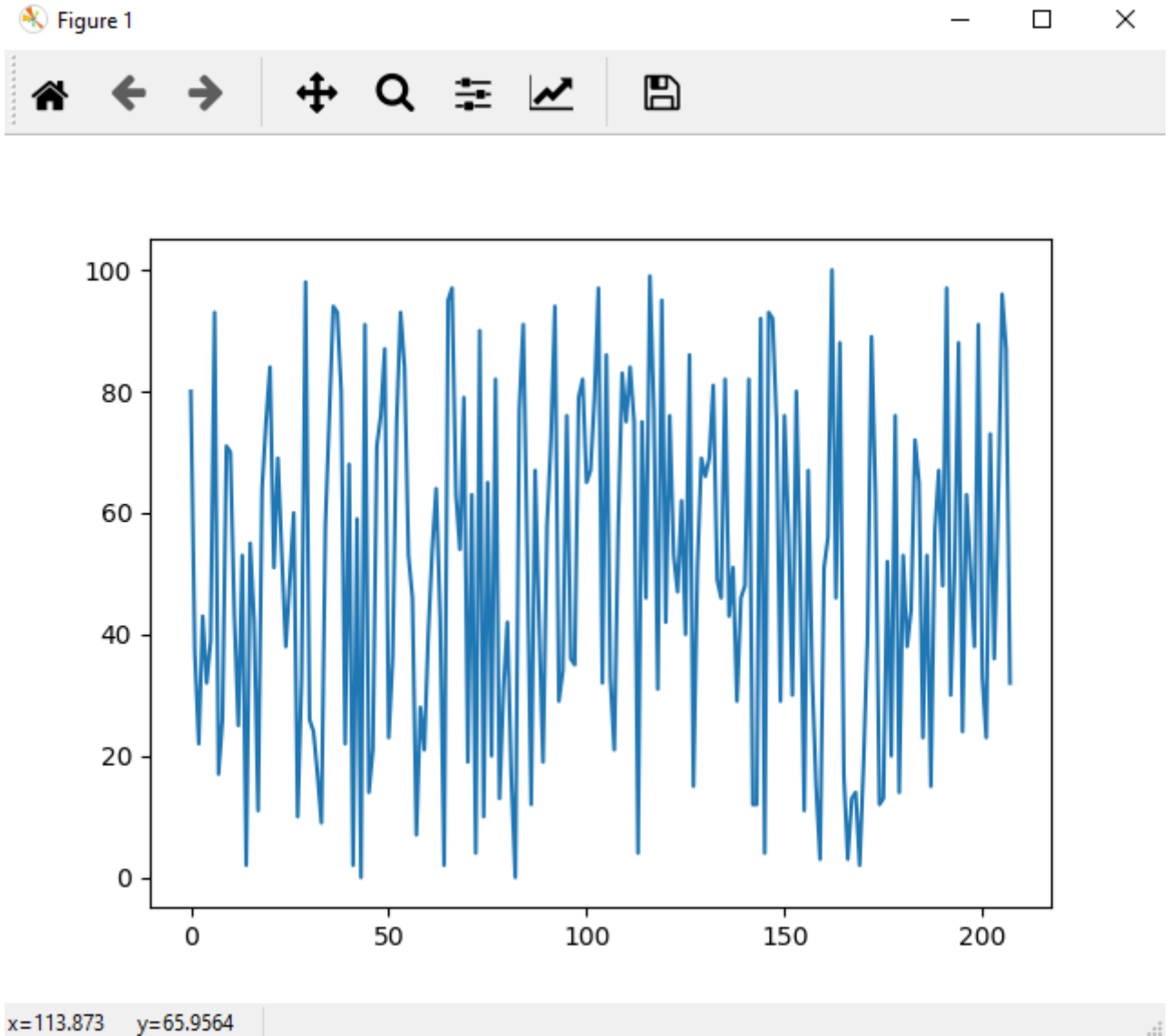
- Install **drawnow** library
- Make sure Matplotib is installed

A Simple Code

```
from drawnow import *
import matplotlib.pyplot as plt
import random
import time

y=[]

def plot_me():      # define your plot function
    plt.plot(y)

def loop_me():      # define a loop function
    while True:
        y.append(random.randint(0,100))
        drawnow(plot_me)   # call plot function
        time.sleep(1)      # put a time delay
def main():
    loop_me()              # call the loop function

main()
```



Output



Make a window

```
from drawnow import *
import matplotlib.pyplot as plt
import random
import time

y=[]
window = 10                    # Fix window size

def plot_me():
    plt.plot(y)

def loop_me():
    while True:
        if len(y) > window: #if window size exceeds
            y.pop(0)        # Remove the first element

        y.append(random.randint(0,100))

        drawnow(plot_me)
        time.sleep(1)
def main():
    loop_me()

main()
```

Make X-axis dynamic

```
from drawnow import *
import matplotlib.pyplot as plt
import random
import time

x=[]                    #Add a new axis
y=[]
base = 0                #base value of x axis
window = 10

def plot_me():
    plt.plot(x,y)
```

```
def loop_me():
    global base
    while True:
        if len(y) > window:
            y.pop(0)
            x.pop(0)            # update x axis
        base += 1               # increase by 1
        x.append(base)          # add new value
        y.append(random.randint(0,100))

        drawnow(plot_me)
        time.sleep(1)
def main():
    loop_me()

main()
```

Calculate Moving Average

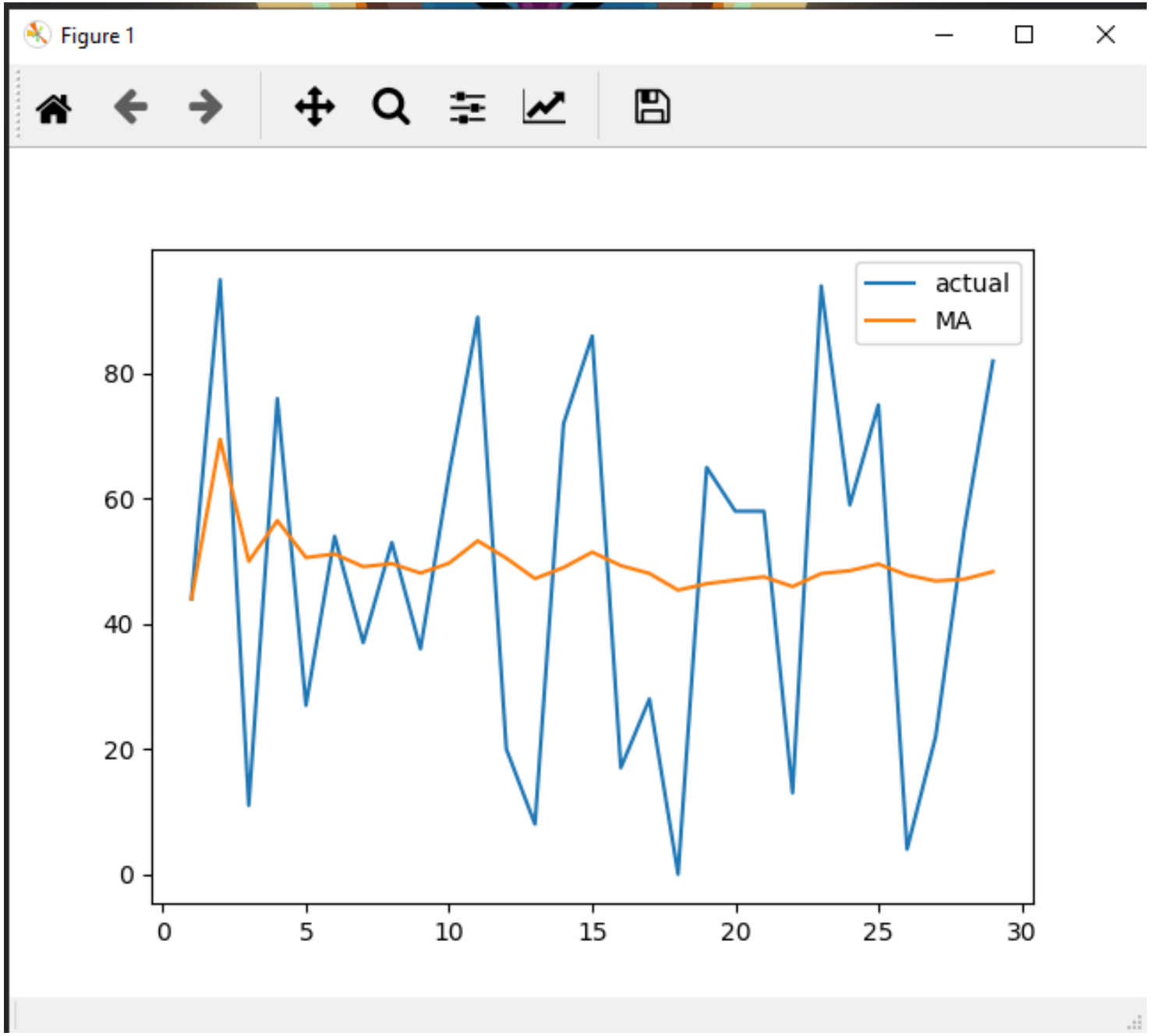
```
from drawnow import *
import matplotlib.pyplot as plt
import random
import time

x=[]
y=[]
avg=[]
base = 0
window = 30

def plot_me():
    plt.plot(x,y)
    plt.plot(x,avg)
    plt.legend()
def get_avg(a):
    return sum(a)/len(a)
```

```
def loop_me():
    global base
    while True:
        if len(y) > window:
            avg.pop(0)
            y.pop(0)
            x.pop(0)

        base += 1
        x.append(base)
        y.append(random.randint(0,100))
        avg.append(get_avg(y))

        drawnow(plot_me)
        time.sleep(1)
def main():
    loop_me()

main()
```

Observation

- Change the window size from 10 to 30 or higher
- What did you observe ?
- Can you reason your observation ?

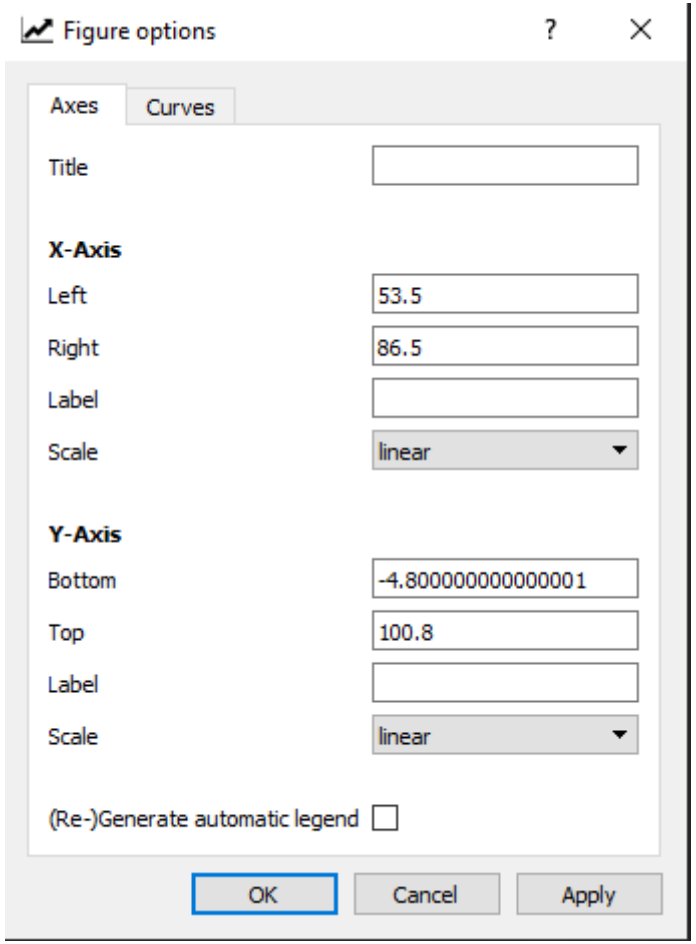
Output

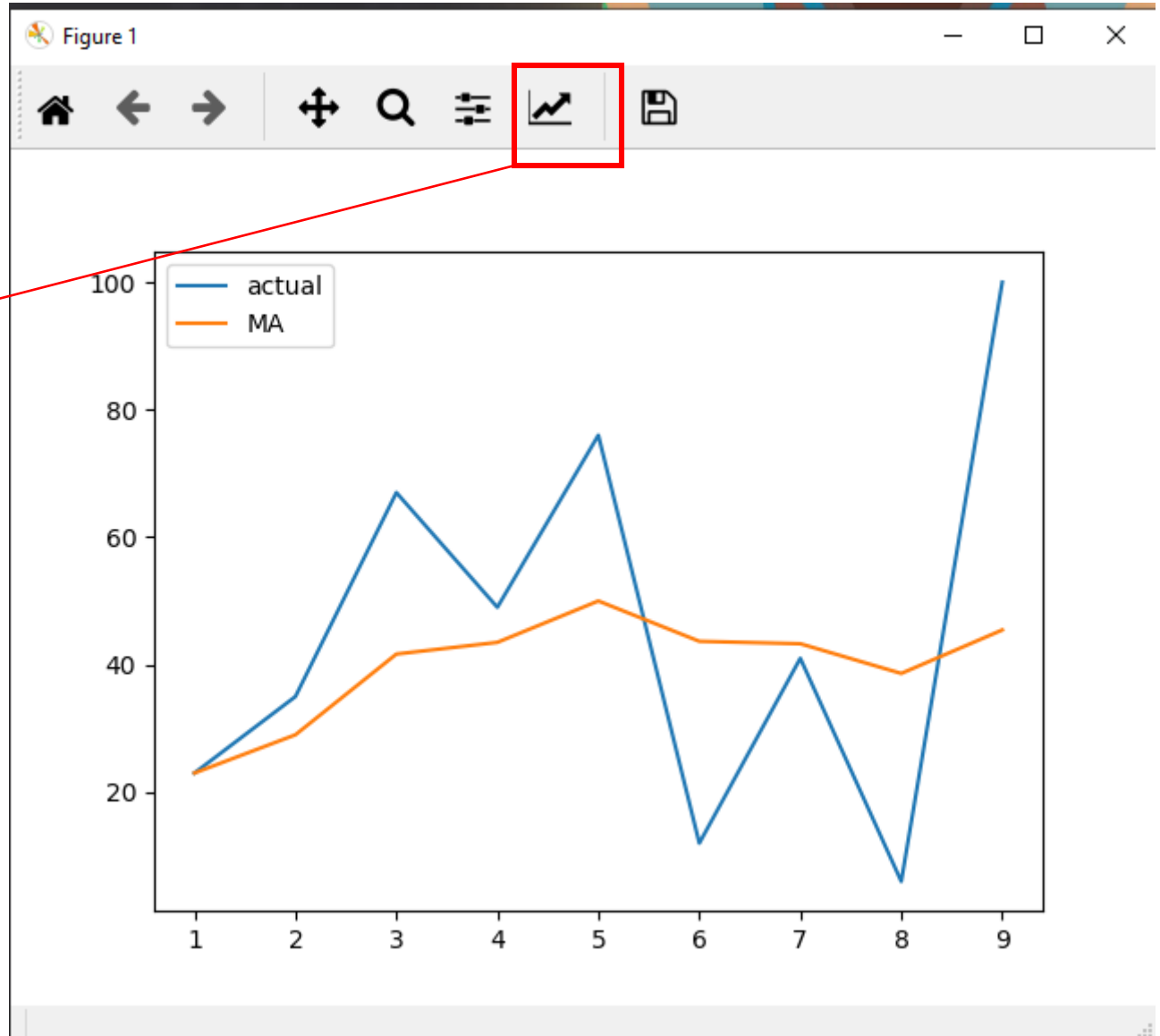


Observation

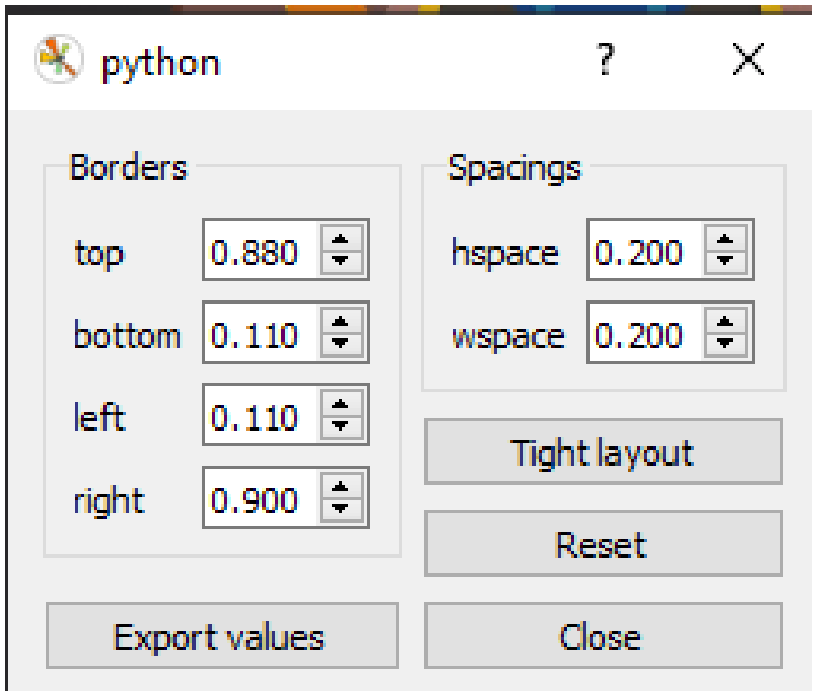
- Change the window size from 10 to 30 or higher
- What did you observe ?
  - Moving average has less fluctuation than the actual
- Can you reason your observation ?
  - Average summarizes the overall set

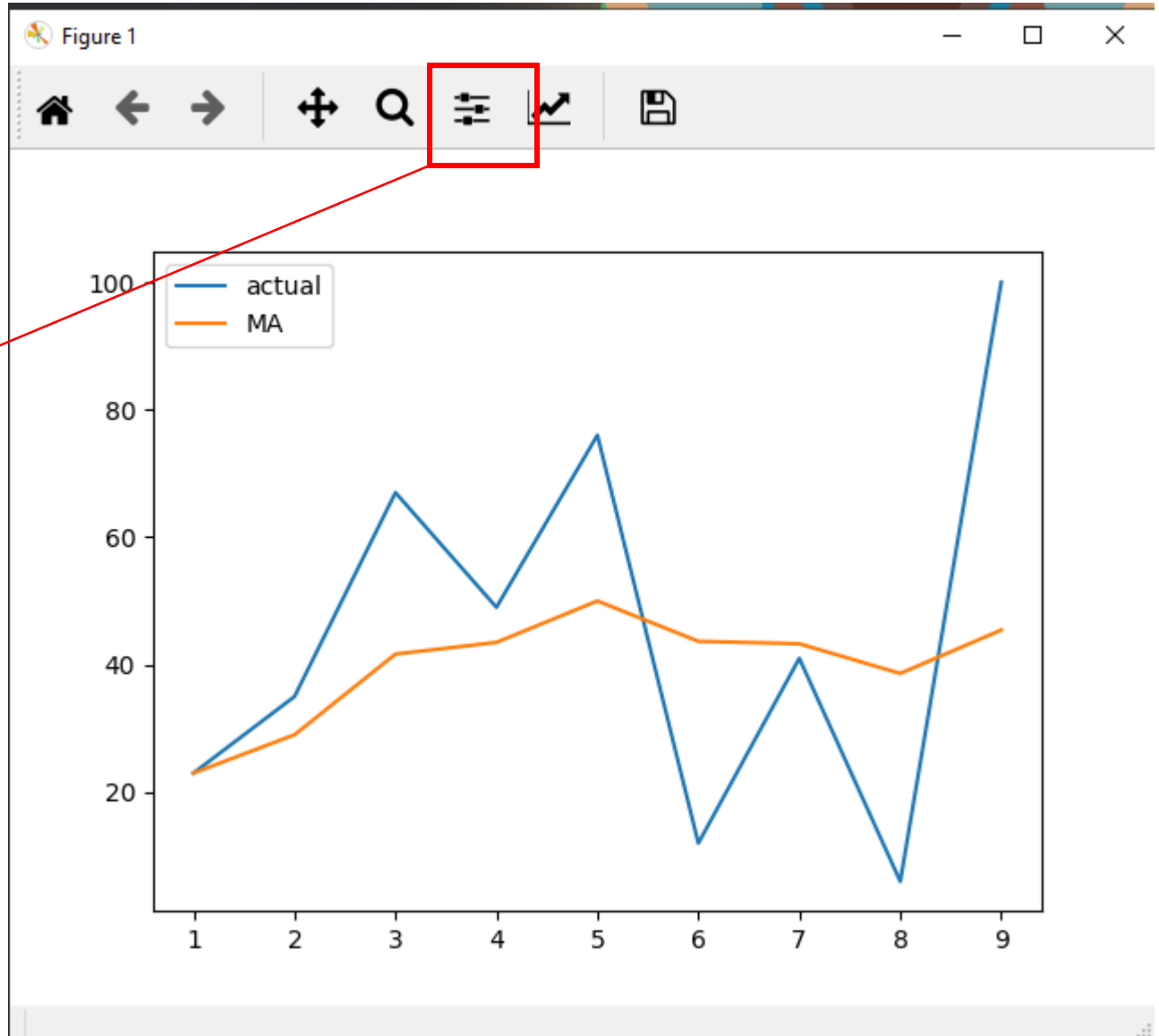
Customizing your graphs





Customizing your graphs





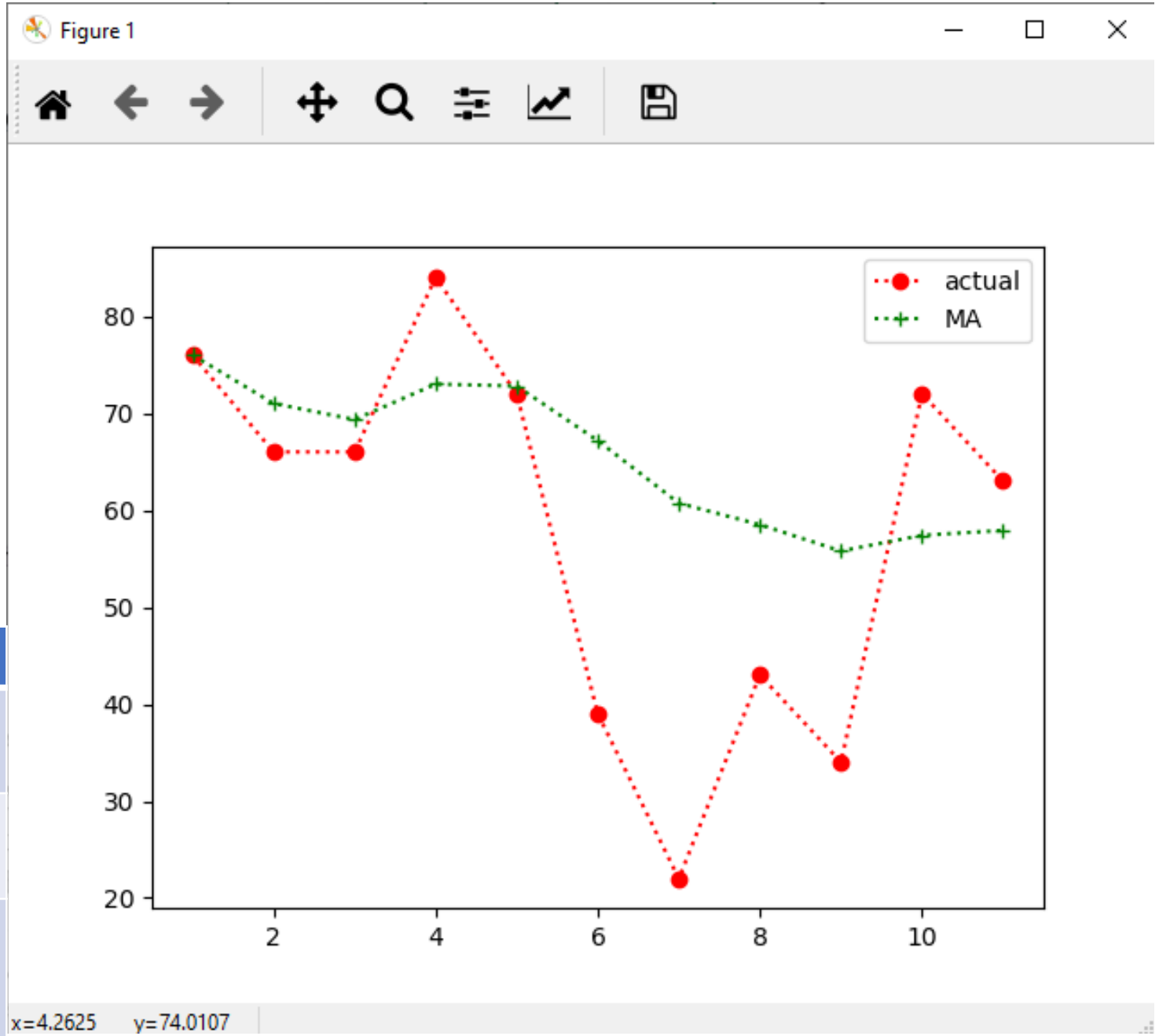
Change Plot styles

```
from drawnow import *
import matplotlib.pyplot as plt
import random
import numpy
import time

x=[]
y=[]
avg=[]
base = 0
window = 30

def plot_me():
    plt.plot(x,y,'ro:',label='actual')
    plt.plot(x,avg,'g+:',label='MA')
    plt.legend()

def get_avg(a):
    return sum(a)/len(a)
```

r – Red
o – Marker o
: - Dotted Line

```
def loop_me():
    global base
    while True:
        if len(y) > window:
            avg.pop(0)
            y.pop(0)
            x.pop(0)

        base += 1
        x.append(base)
        y.append(random.randint(0,100))
        avg.append(get_avg(y))

        drawnow(plot_me)
        time.sleep(1)
def main():
    loop_me()

main()
```

Change Plot styles



Matplotlib Stylesheet	
Colors	https://matplotlib.org/2.0.2/api/colors_api.html
Markers	https://matplotlib.org/3.1.1/api/markers_api.html
Line Styles	https://matplotlib.org/gallery/lines_bars_and_markers/line_styles_reference.html

Add a Grid

```
from drawnow import *
import matplotlib.pyplot as plt
import random
import numpy
import time

x=[]
y=[]
avg=[]
base = 0
window = 30

def plot_me():
    plt.plot(x,y,'ro:',label='actual')
    plt.plot(x,avg,'g+:',label='MA')
    plt.grid(True)
    plt.legend()

def get_avg(a):
    return sum(a)/len(a)
```

```
def loop_me():
    global base
    while True:
        if len(y) > window:
            avg.pop(0)
            y.pop(0)
            x.pop(0)

        base += 1
        x.append(base)
        y.append(random.randint(0,100))
        avg.append(get_avg(y))

        drawnow(plot_me)
        time.sleep(1)
def main():
    loop_me()

main()
```

Add a Grid