Lecture 13: Plotting

Exploring your data with lines and colors!

First, some imports

```
1   import numpy as np
2   import pandas as pd
3   import matplotlib.pyplot as plt
```

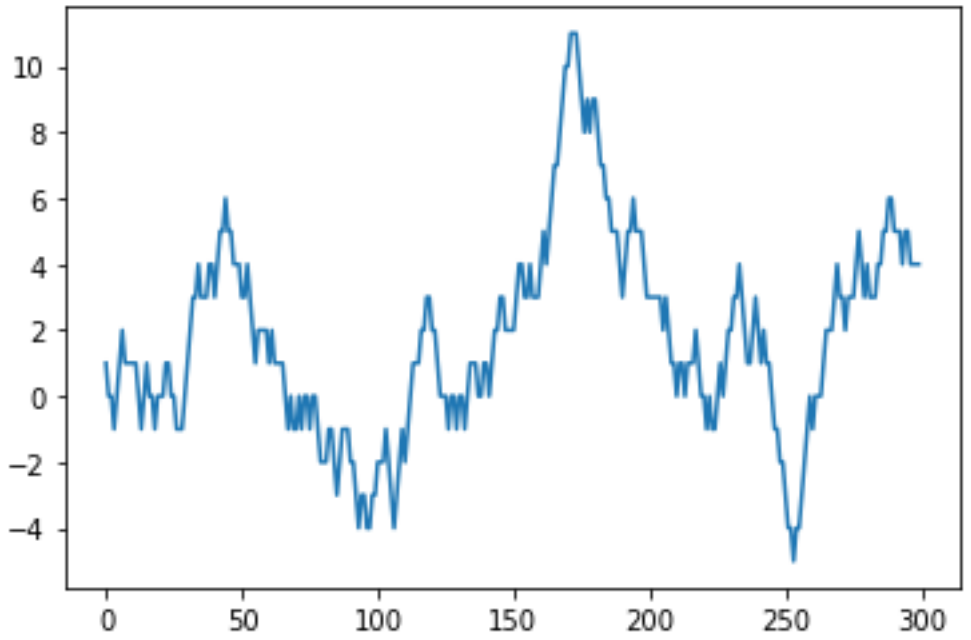
Contents

Random walk

```
5   x = range(300)
6   y = np.random.choice([-1, 0, 1], 300)
7   y = np.cumsum(y) # random walk
```

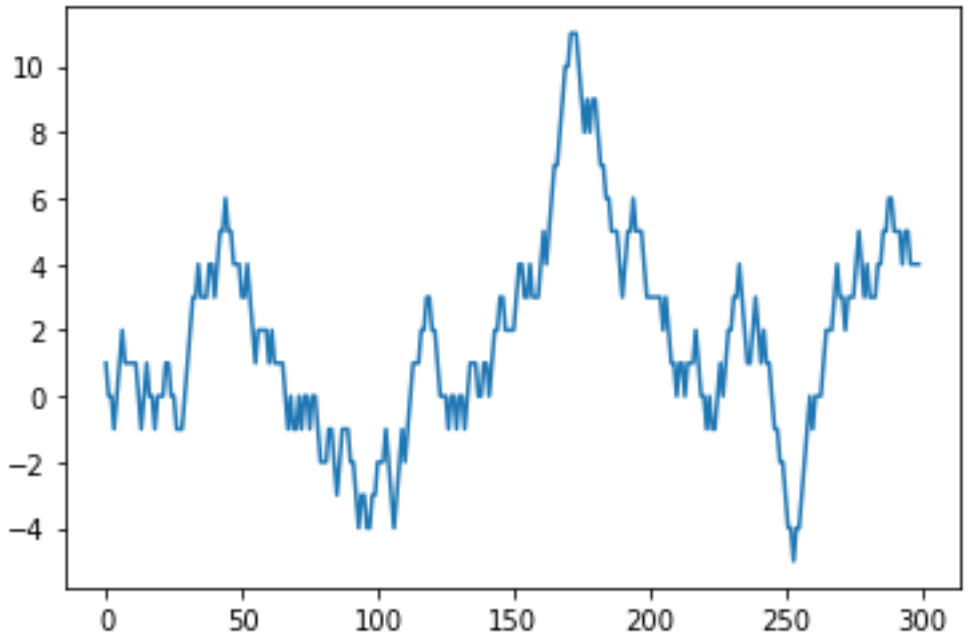
The most basic plot

```
 9   fig, ax = plt.subplots()
10   ax.plot(x, y)
```



The most basic plot

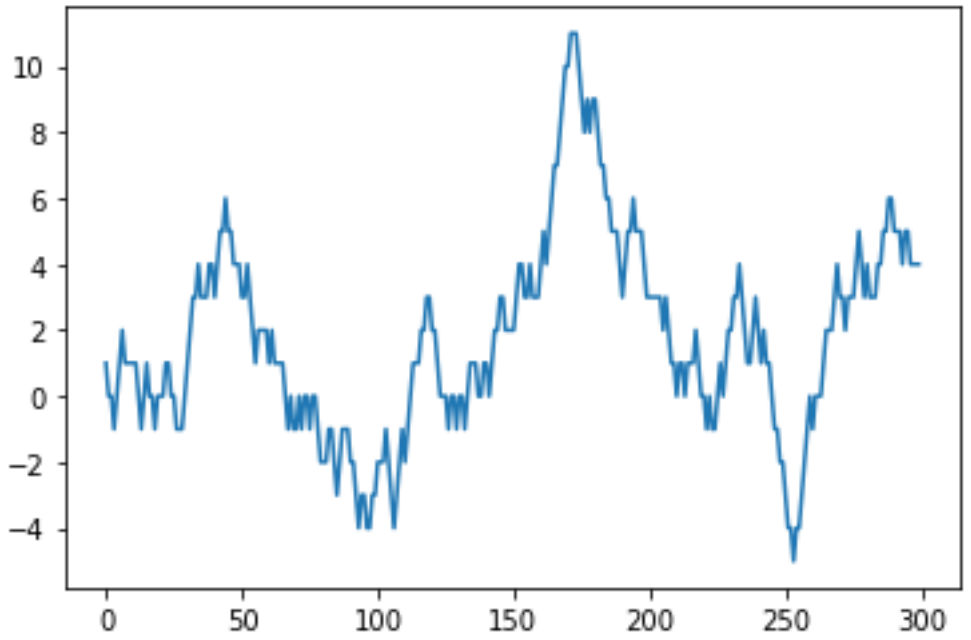
```
 9   fig, ax = plt.subplots()
10   ax.plot(x, y)
```



Displays in the “Plots” tab in the upper right, by default, in Spyder

The most basic plot

```
 9   fig, ax = plt.subplots()
10   ax.plot(x, y)
```



To explicitly render a plot, rather than waiting for the interpreter to assume you want it, call “fig.show()”

Displays in the “Plots” tab in the upper right, by default, in Spyder

The most basic plot

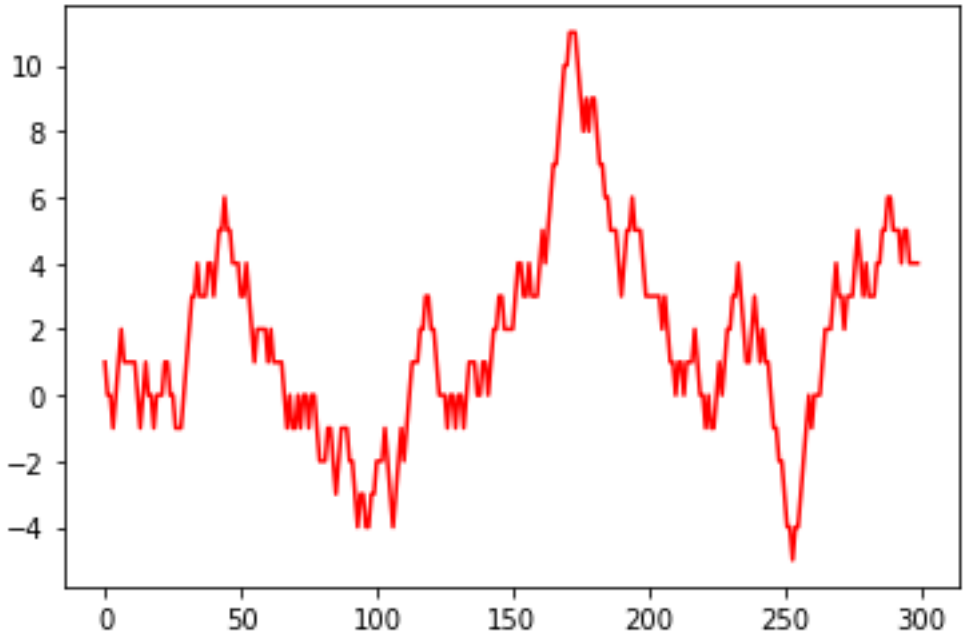
```
 9   fig, ax = plt.subplots()
10   ax.plot(x, y)
```

`fig`: one object representing the canvas that all plots show on

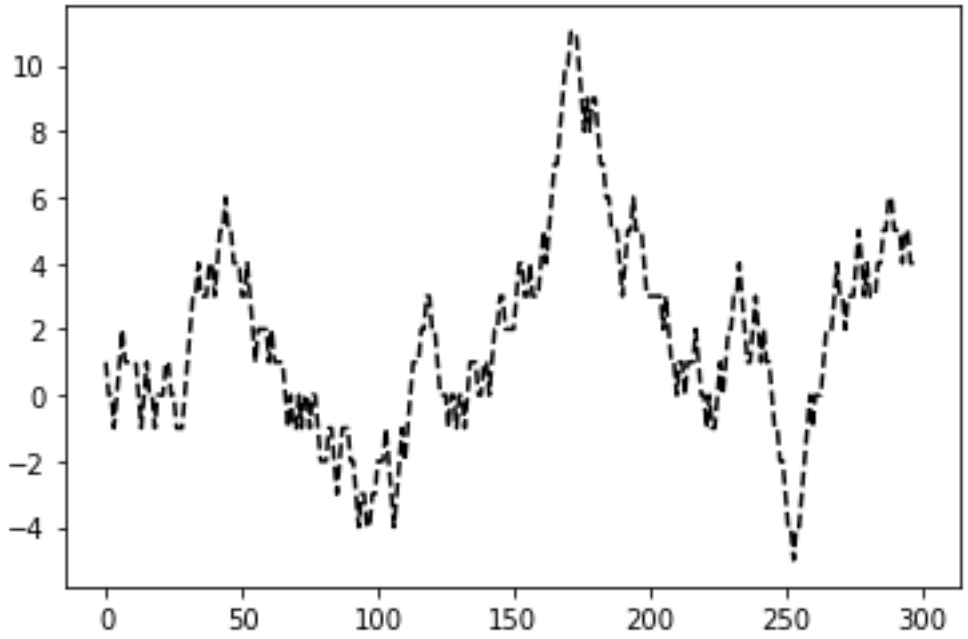
`ax`: one plot (axis) that goes on a figure

Expanding our plot: colors and lines

```
19  fig, ax = plt.subplots()
20  ax.plot(x, y, color='red', linestyle='solid')
```

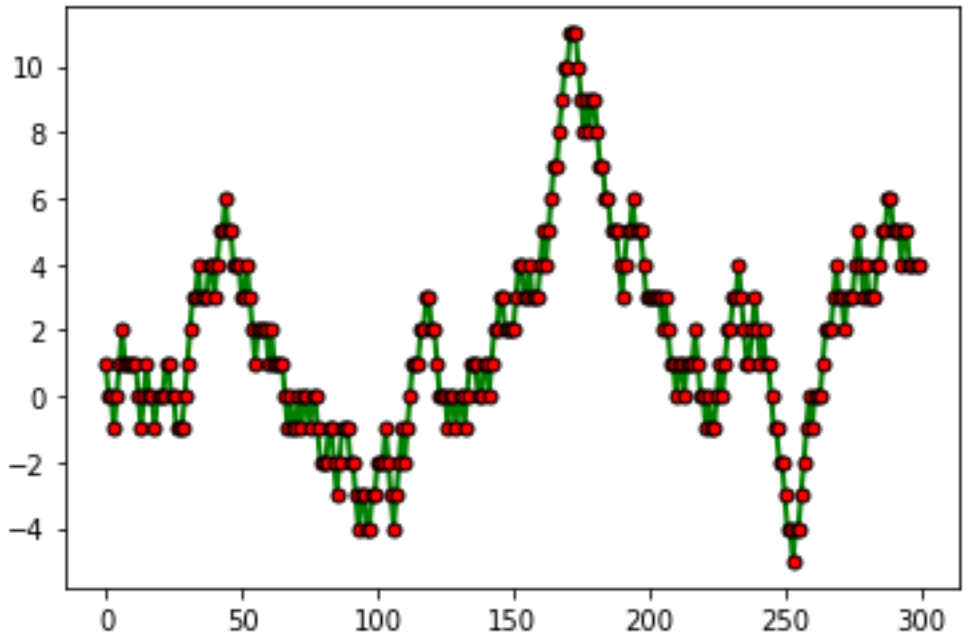



```
22  fig, ax = plt.subplots()
23  ax.plot(x, y, color='black', linestyle='dashed')
```



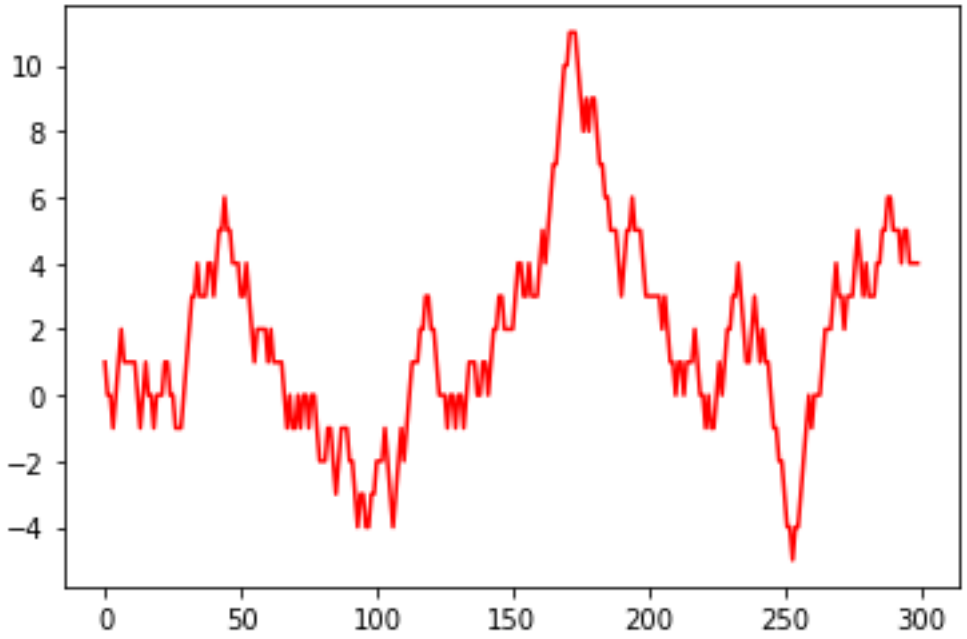
Expanding our plot: we can control everything!

```
25  fig, ax = plt.subplots()
26  ax.plot(x, y, color='green', linestyle='solid', marker='o', linewidth=2,
27          markersize=5, markerfacecolor='red', markeredgecolor='black')
```

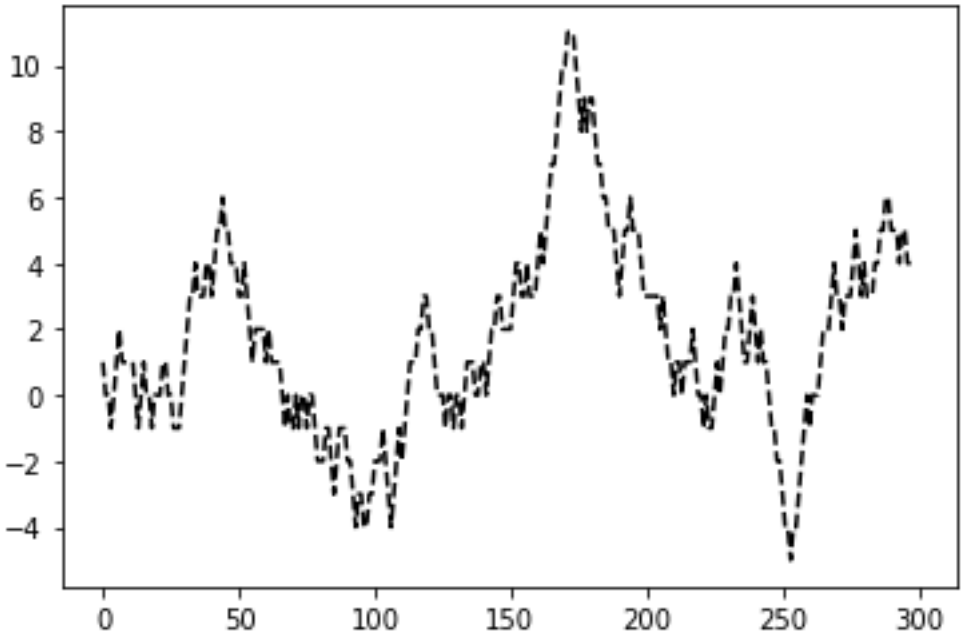


Expanding our plot: keeping it simple

```
13  fig, ax = plt.subplots()
14  ax.plot(x, y, 'r-')
```




```
16  fig, ax = plt.subplots()
17  ax.plot(x, y, 'k--')
```



Expanding our plot: colors and lines

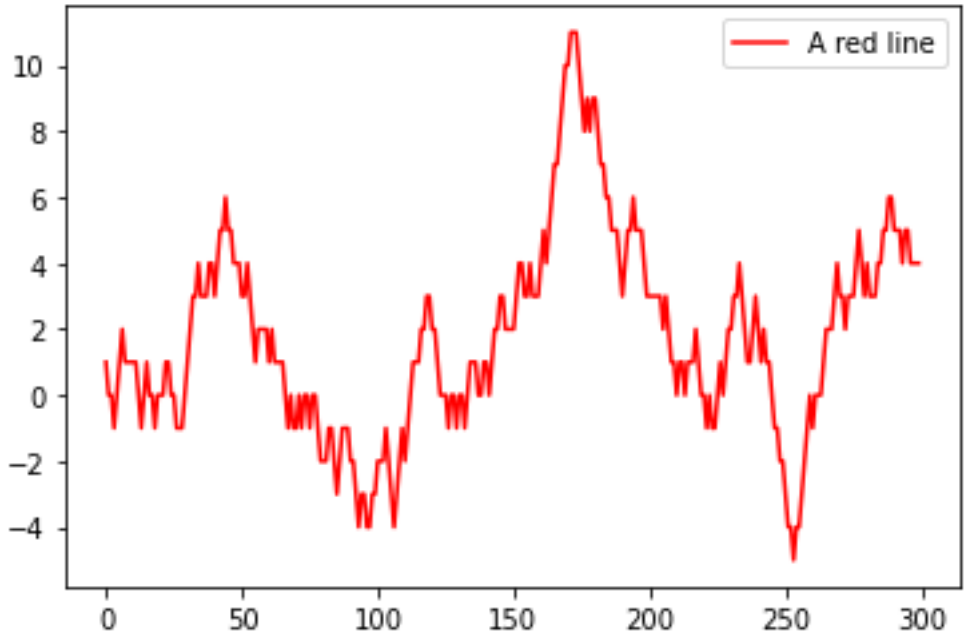
character	description
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
'8'	octagon marker
's'	square marker
'p'	pentagon marker

'P'	plus (filled) marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'X'	x (filled) marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

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

Expanding our plot: legends and labels

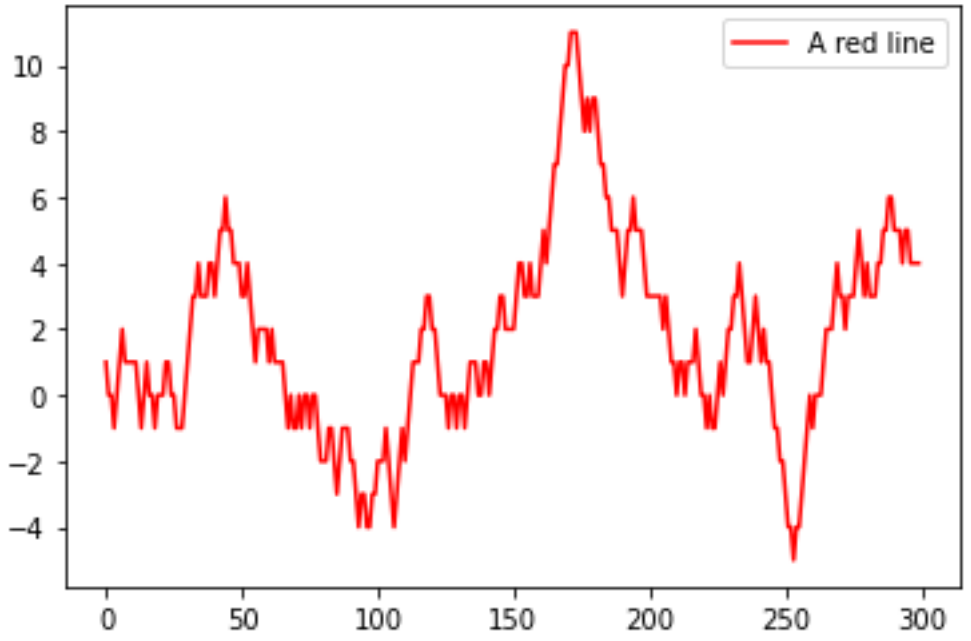
```
30  fig, ax = plt.subplots()
31  ax.plot(x, y, 'r-', label='A red line')
32  ax.legend(loc='best')
```



Expanding our plot: legends and labels

Note that here we perform two separate operations on the axis object

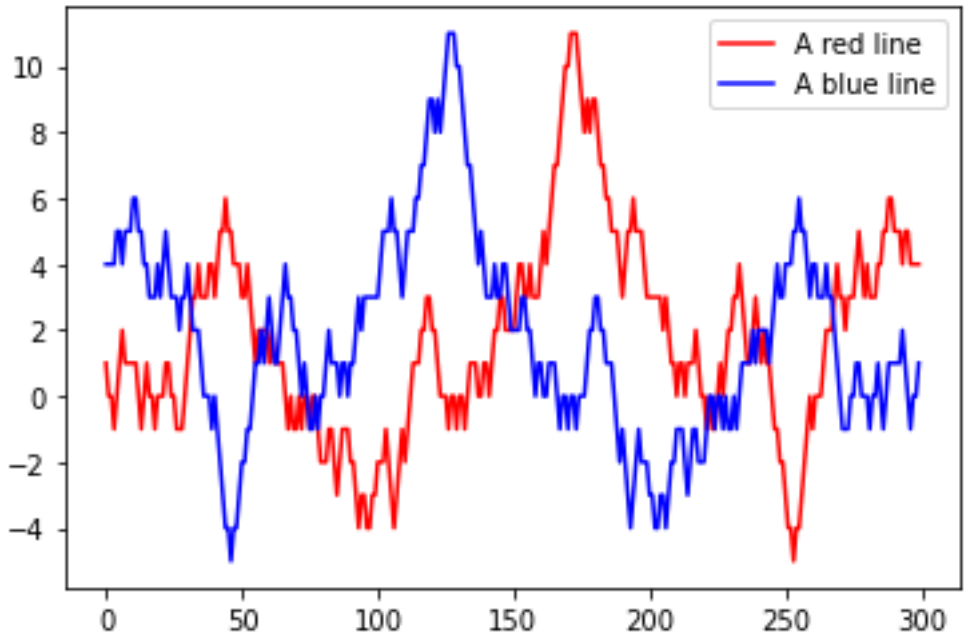
```
30  fig, ax = plt.subplots()
31  ax.plot(x, y, 'r-', label='A red line')
32  ax.legend(loc='best')
```



Expanding our plot: multiple lines

Each plot method creates a new line

```
34  fig, ax = plt.subplots()
35  ax.plot(x, y, 'r-', label='A red line')
36  ax.plot(x, y[::-1], 'b-', label='A blue line')
37  ax.legend(loc='best')
```



Python review

Instances of MatPlotLib
“figure” and “axis” objects

The MatPlotLib module

```
34  fig, ax = plt.subplots()
35  ax.plot(x, y, 'r-', label='A red line')
36  ax.plot(x, y[::-1], 'b-', label='A blue line')
37  ax.legend(loc='best')
```

Python review

Instances of MatPlotLib
“figure” and “axis” objects

The MatPlotLib module

“plot” is a method of an axis

```
34  fig, ax = plt.subplots()
35  ax.plot(x, y, 'r-', label='A red line')
36  ax.plot(x, y[::-1], 'b-', label='A blue line')
37  ax.legend(loc='best')
```

Every time we call the “plot” method, it creates a new line within that same axis

Python review

Instances of MatPlotLib
“figure” and “axis” objects

The MatPlotLib module

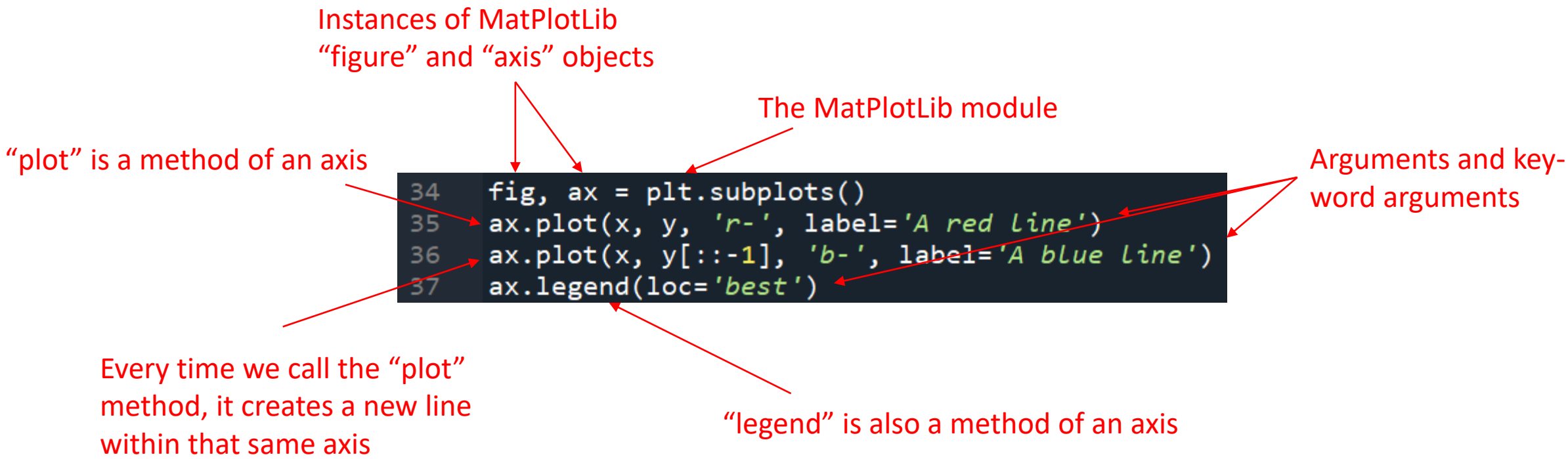
“plot” is a method of an axis

```
34  fig, ax = plt.subplots()
35  ax.plot(x, y, 'r-', label='A red line')
36  ax.plot(x, y[::-1], 'b-', label='A blue line')
37  ax.legend(loc='best')
```

Every time we call the “plot” method, it creates a new line within that same axis

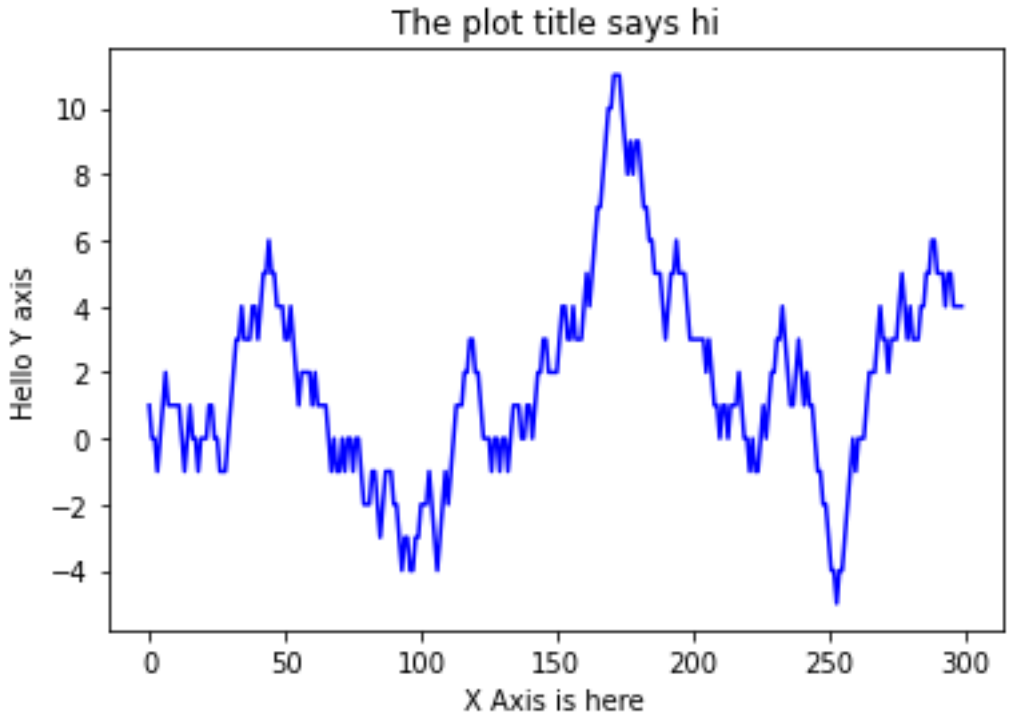
“legend” is also a method of an axis

Python review



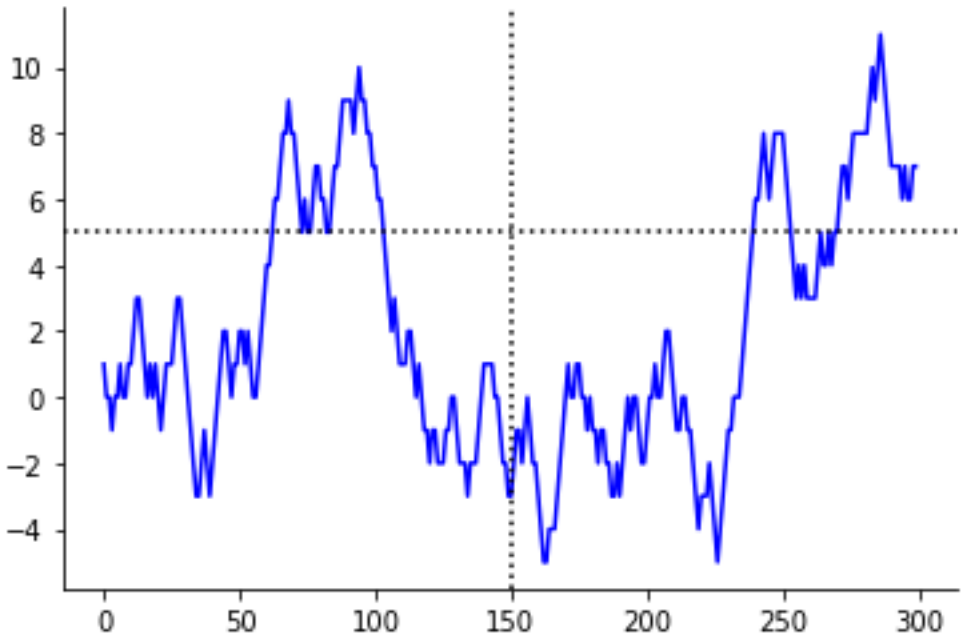
More axis methods: labels

```
40  fig, ax = plt.subplots()
41  ax.plot(x, y, 'b-')
42
43  ax.set_ylabel('Hello Y axis')
44  ax.set_xlabel('X Axis is here')
45  ax.set_title('The plot title says hi')
```



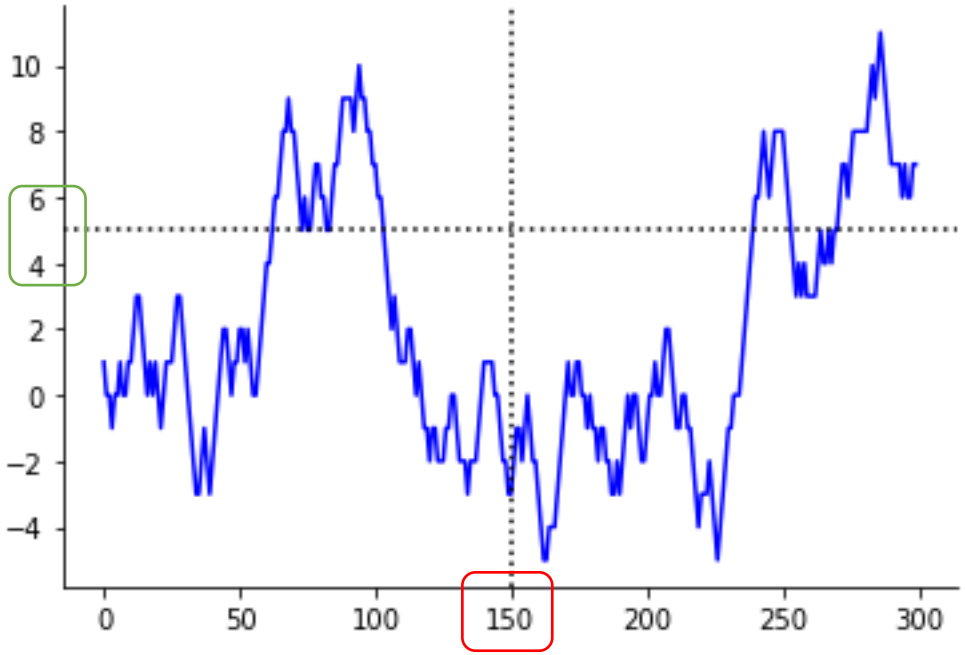
More axis methods: spines and lines

```
48  fig, ax = plt.subplots()
49  ax.plot(x, y, 'b-')
50
51  ax.spines['top'].set_visible(False)
52  ax.spines['right'].set_visible(False)
53
54  ax.axvline(150, color='k', linestyle=':')
55  ax.axhline(5, color='k', linestyle=':')
```



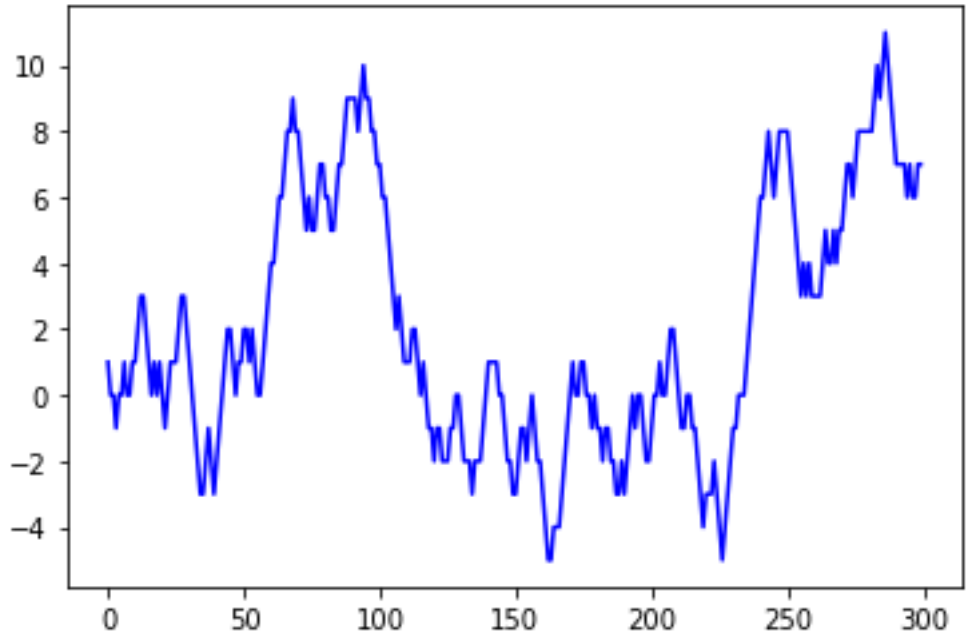
More axis methods: spines and lines

```
48  fig, ax = plt.subplots()
49  ax.plot(x, y, 'b-')
50
51  ax.spines['top'].set_visible(False)
52  ax.spines['right'].set_visible(False)
53
54  ax.axvline(150, color='k', linestyle=':')
55  ax.axhline(5, color='k', linestyle=':')
```

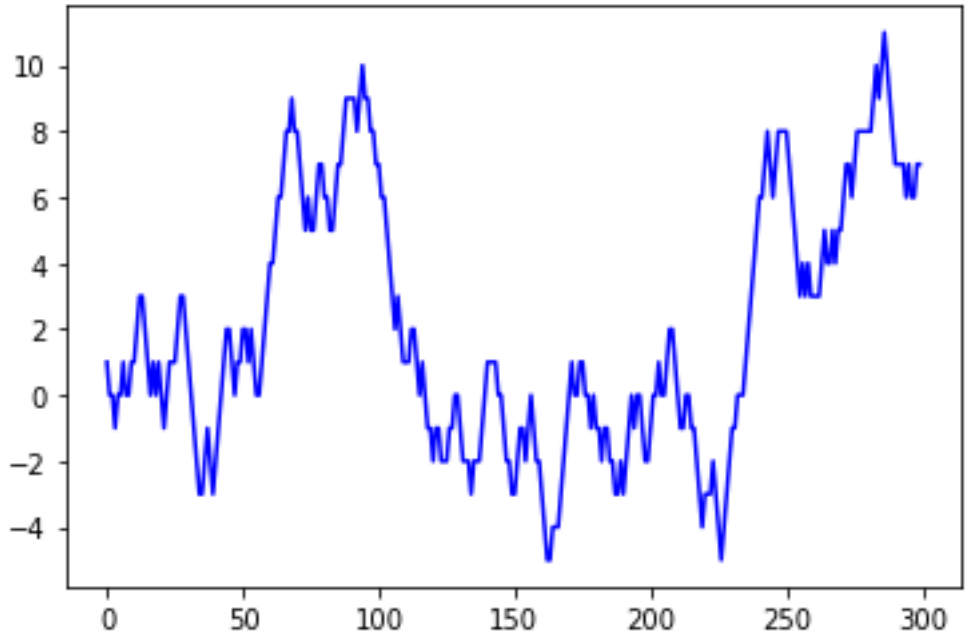


MatPlotLib gotcha: axis vs module objects

```
58    plt.plot(x, y, 'b-')
```

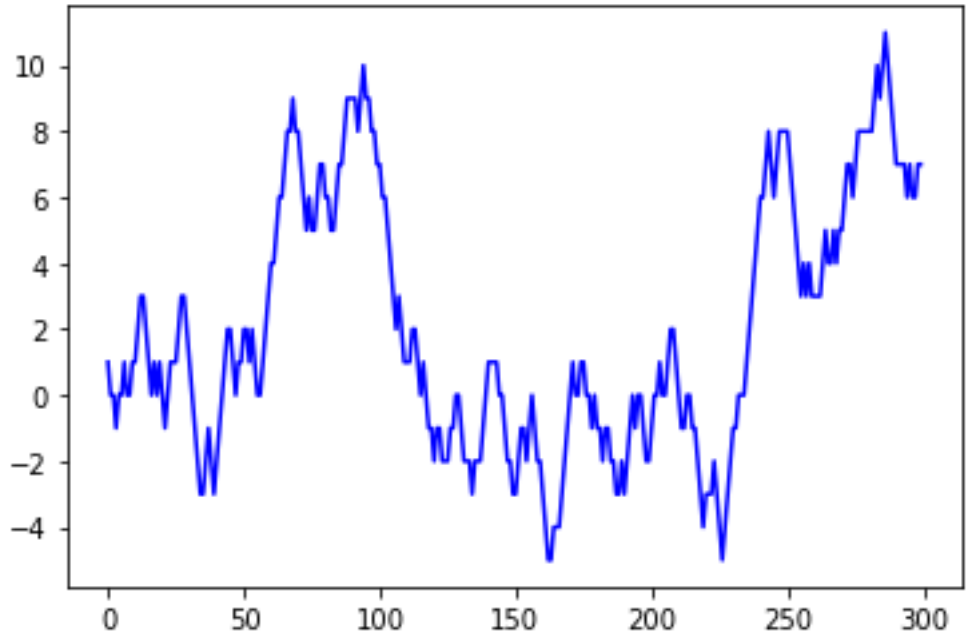



```
60    fig, ax = plt.subplots()
61    ax.plot(x, y, 'b-')
```

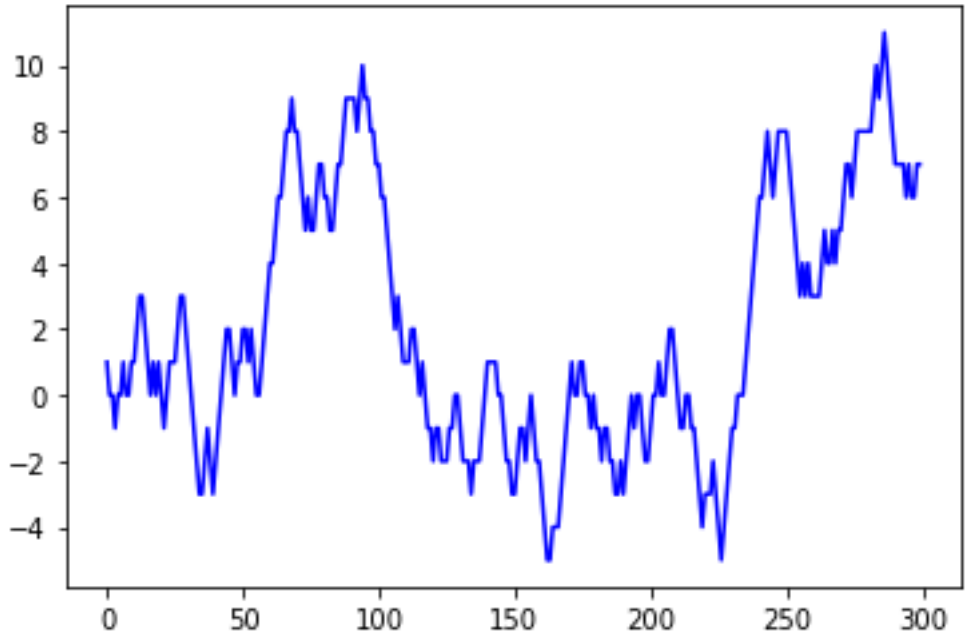


MatPlotLib gotcha: axis vs module objects

```
58    plt.plot(x, y, 'b-')
```




```
60    fig, ax = plt.subplots()
61    ax.plot(x, y, 'b-')
```



MatPlotLib and Pandas

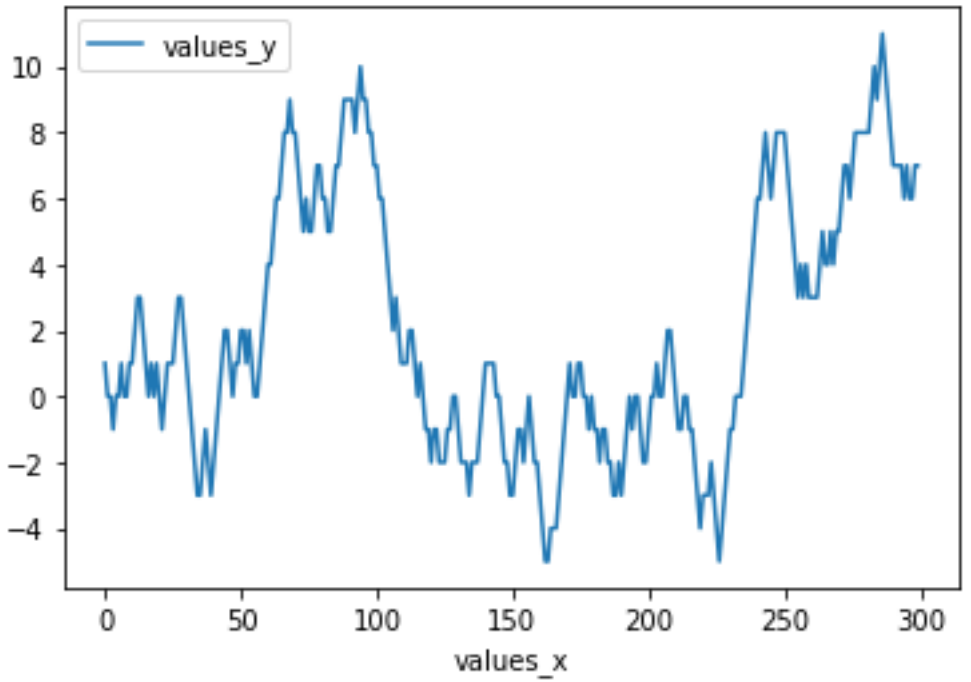
```
64  df = pd.DataFrame({'values_x':x, 'values_y':y})
65  df
```

	values_x	values_y
0	0	1
1	1	0
2	2	0
3	3	-1
4	4	0
..	...	...
295	295	7
296	296	6
297	297	6
298	298	7
299	299	7

[300 rows x 2 columns]

MatPlotLib and Pandas

```
67    df.plot(x='values_x', y='values_y')
```



MatPlotLib and Pandas

Pandas DataFrame method

MatPlotLib axis method

```
70    ax = df.plot(x='values_x', y='values_y')
71    ax.axvline(150, color='k', linestyle=':')
```