matplotlib_1

February 11, 2020

1 An introduction to matplotlib

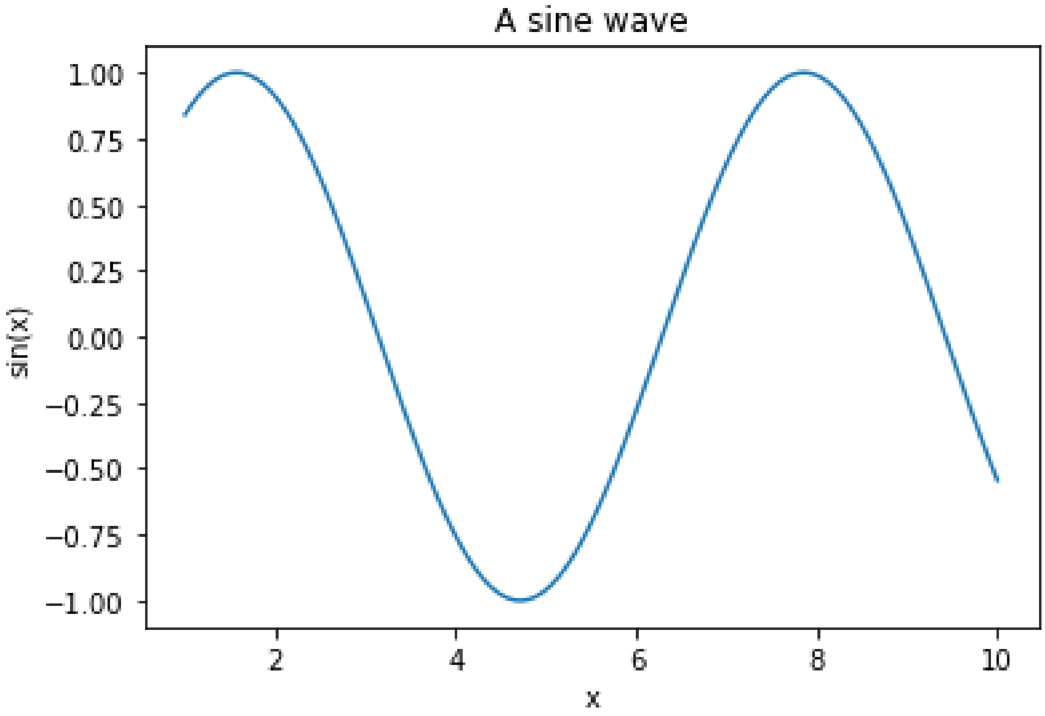
[1]:

```
import numpy as np
import matplotlib.pyplot as plt
# The following line helps charts to be displayed in Jupyter Notebooks
%matplotlib inline
```

1.1 A simple line chart

[2]:

```
x = np.arange(1,10.01,0.01) # array from 1 to 10 in steps of 0.01
y = np.sin(x)
plt.plot(x,y)
plt.xlabel ('x')
plt.ylabel ('sin(x)')
plt.title('A sine wave')
plt.show()
```



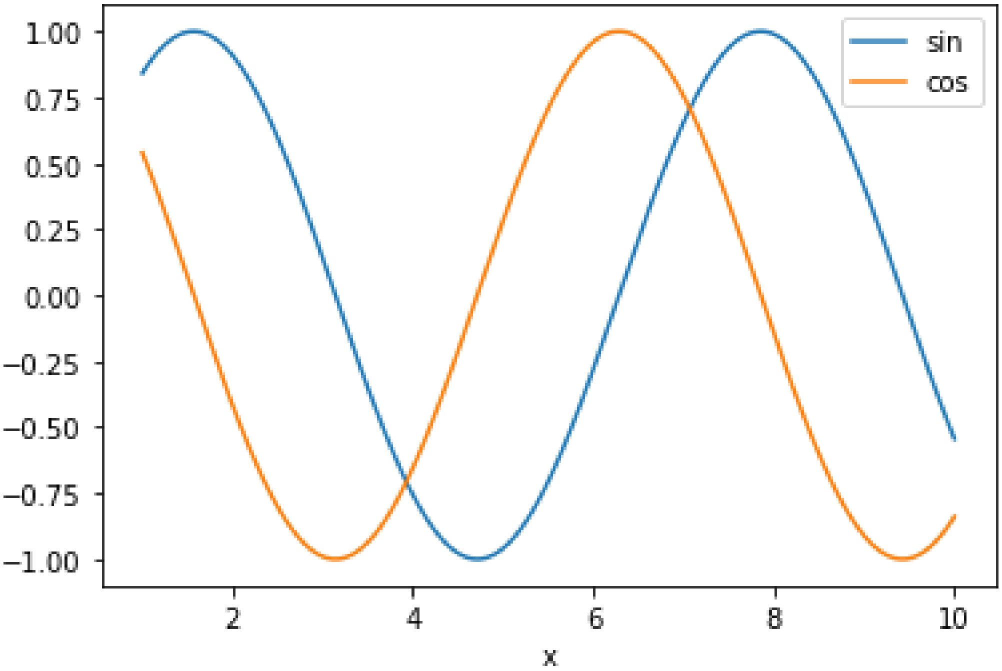
1.2 Two lines with legend

[3]:

```
x = np.arange(1,10.01,0.01) # array from 1 to 10 in steps of 0.01
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot(x,y1,label='sin')
plt.plot(x,y2,label='cos')
plt.legend()
plt.xlabel ('x')

# If you want to save a plot, then save before showing
plt.savefig('line_plot.png')

plt.show()
```

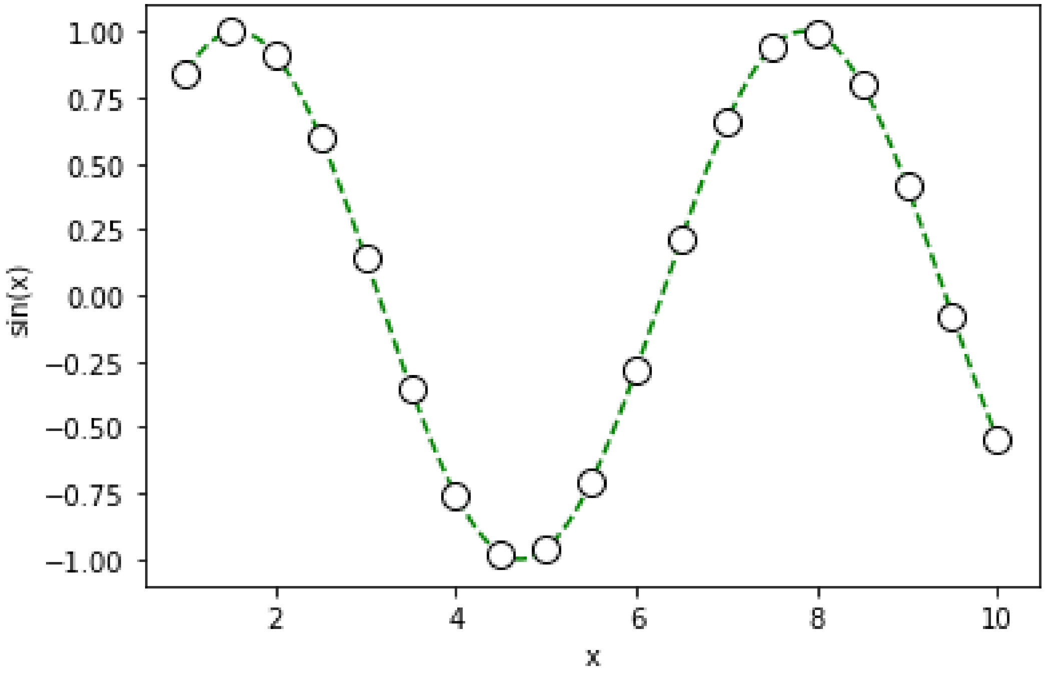


1.3 Customizing a line plot

[4]:

```
x = np.arange(1,10.01,0.01) # array from 1 to 10 in steps of 0.01
y = np.sin(x)
plt.plot(x,y,
         linestyle='--',
         color='g',
         marker='o',
         markeredgecolor='k',
         markerfacecolor='w',
         markersize=10,
         markevery=50)

# Adding options (keywaord arguments. Note the dastardly American spelling)
# Use help (plt.plot to see all options)
plt.xlabel ('x')
plt.ylabel ('sin(x)')
plt.show()
```



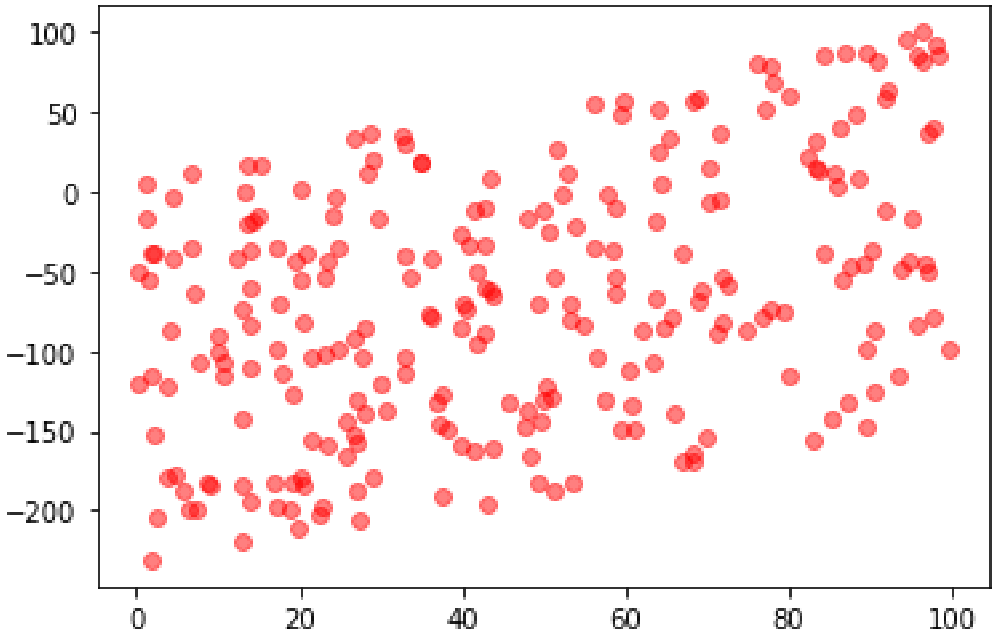
1.4 A scatter plot

[6]:

```
x = np.random.rand(250) * 100 # 250 values between 0 and 100
jitter = 10 - np.random.rand(250) * 250 # 100 values between -10 and 10
y = x + jitter

# Scatter plot with transparency of points (alpha)
plt.scatter(x, y,
            color='r',
            alpha = 0.5)

plt.show()
```

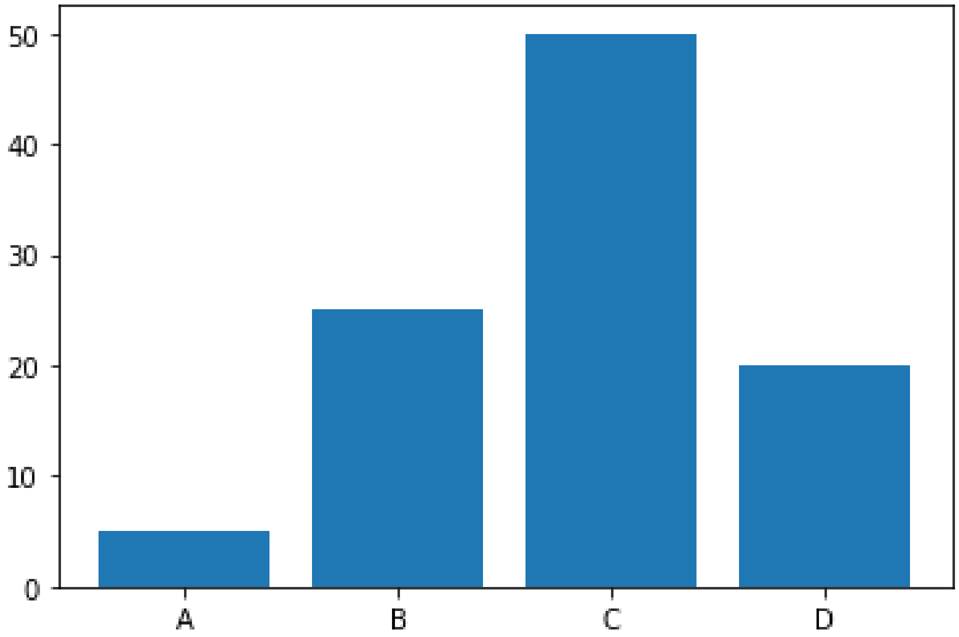


1.5 A bar chart

[7]:

```
x = [1, 2, 3, 4]
y = [5.,25.,50.,20.]

labels = ['A', 'B', 'C', 'D']
plt.bar(x, y)
# To set xlabels we need to pass location (an array starting at 1, and labels)
plt.xticks(np.arange(1,5), labels)
plt.show()
```



1.6 Boxplots

Box plots display a box for the interquartile range, median as organe line, range without outliers, and outliers.

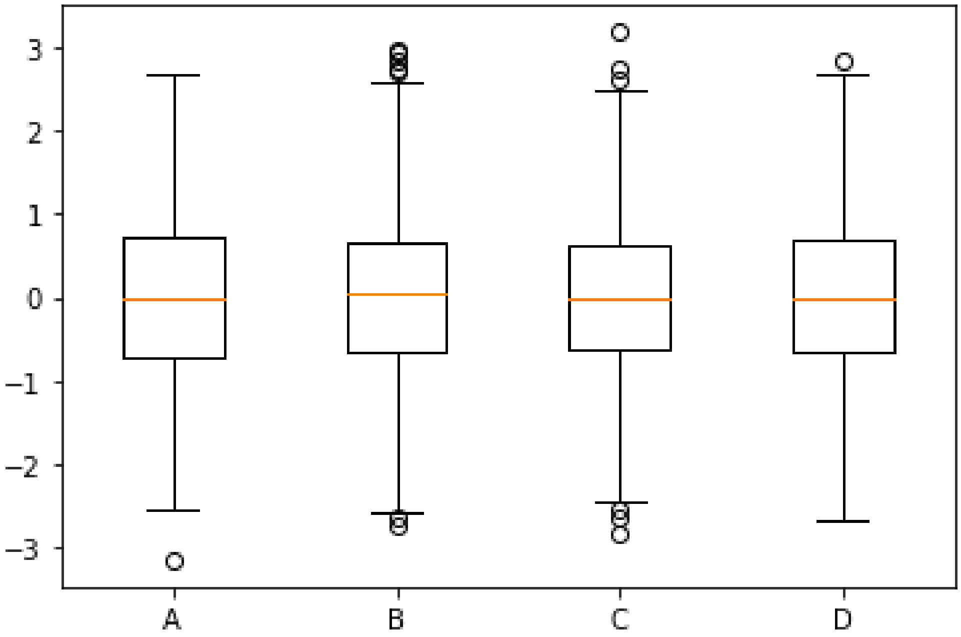
Outliers usually defined as 1.5*IQR (interquartile range) outside IQR.

[8]:

```
x=np.random.randn(1000,4) # samples from a normal distribution

labels = ['A', 'B', 'C', 'D']
plt.boxplot(x)
# To set xlabels we need to pass location (an array starting at 1, and labels)
plt.xticks(np.arange(1,5), labels)

plt.show()
```

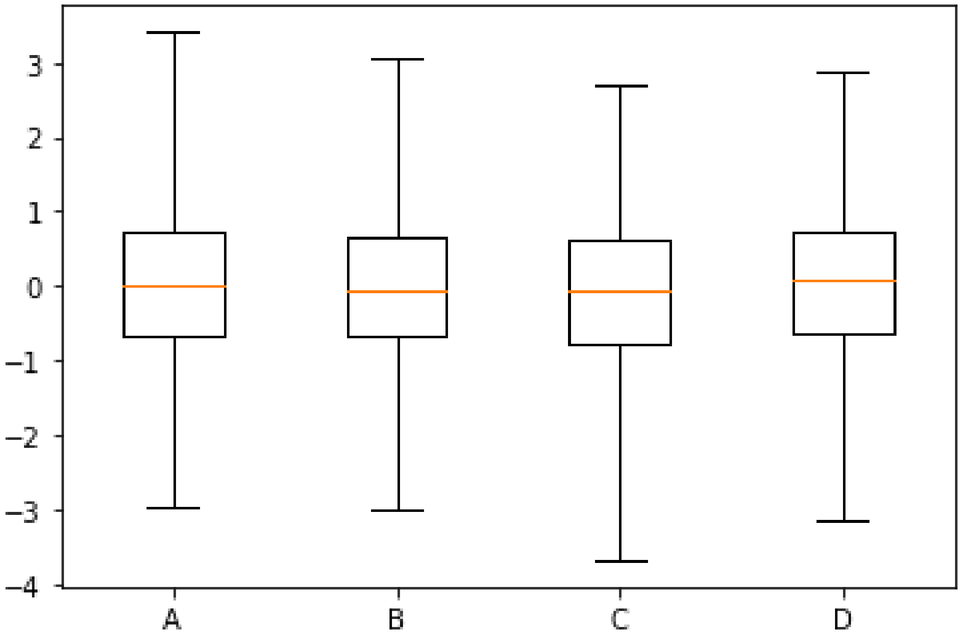


To avoid plotting outliers separately change the `whis` (whiskers) argument from default 1.5 to a high value.

[9]:
```
x=np.random.randn(1000,4) # samples from a normal distribution

labels = ['A', 'B', 'C', 'D']
plt.boxplot(x, whis=1000)
# To set xlabels we need to pass location (an array starting at 1, and labels)
plt.xticks(np.arange(1,5), labels)

plt.show()
```



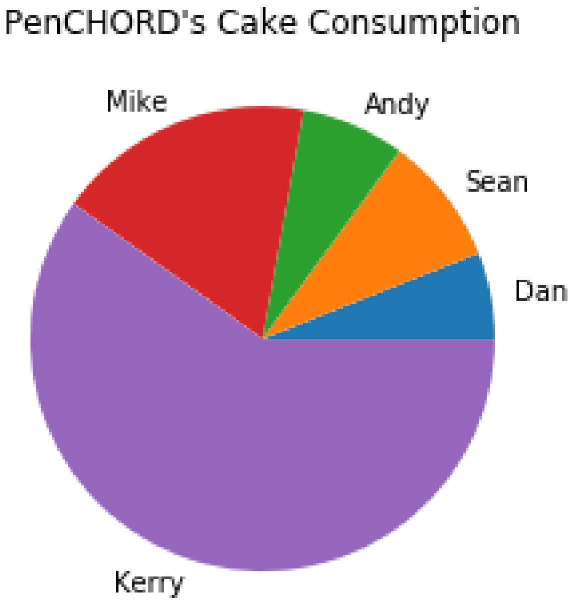
1.7 A pie chart

[10]:

```
cake_consumption = [10, 15, 12, 30, 100]
labels = 'Dan','Sean','Andy','Mike','Kerry'

plt.pie(cake_consumption, labels=labels)
plt.title ("PenCHORD's Cake Consumption")\

plt.show()
```


PenCHORD's Cake Consumption
Mike
Andy
Sean
Dan
Kerry