CS 38003 PYTHON PROGRAMMING

Ruby Tahboub

GENERATING RANDOM NUMBERS

PSEUDO-RANDOM NUMBER GENERATOR

- All generated random numbers are called *Pseudo-random* numbers.
- Pseudo-random number generators generate random numbers in a sequences, i.e., each generated random number depends on the previous generated random number.
- The `random` library is a Pseudo-random number generator.

RANDOM SEED

- There is no previous value when a random number is generated for the first time.
- Seeding a pseudo-random number generator gives it its first "previous" value.
- Using the same seed twice, would generate the same sequence of numbers twice.
- If no seed is provided, a default mechanism to pick up a seed would be used.
    - The current time is a frequently used seed, as it continuously changes leading to generating different sequences of numbers.

The `random` Library

`randrange(start, end, step)`

Returns a random integer value within the specified range.

`random()`

Returns a random float value from zero to one (exclusive).

`choice(seq)`

Returns a random element from the sequence seq.

`shuffle(seq)`

Shuffles the sequence in place.

The `randrange` Function

Returns a random integer value within the specified range.

- `randrange(n)` returns an integer from 0 to n-1
- `randrange(m,n)` returns an integer from m to n-1
- `randrange(m, n, k)` returns an integer from m (inclusive) to n (exclusive) with steps of k.
  - Step can be negative.

randrange(n)

```python
from random import randrange

for i in range(5):
    print(randrange(10))
```

```
first run     second run

    8             8
    3             1
    8             5
    9             8
    5             8
```

randrange(m,n)

```python
from random import randrange

for i in range(5):
    print(randrange(0,5))

print()

for i in range(5):
    print(randrange(5,10))
```

```
first run     second run

   4             2
   4             2
   2             3
   2             2
   3             2

   6             9
   8             5
   9             8
   9             7
   5             7
```

randrange(m,n,k)

```
from random import randrange

for i in range(5):
    print(randrange(0,10,2))

print()

for i in range(5):
    print(randrange(20,10,-3))
```

```
first run          second run

   6                   2
   8                   6
   6                   6
   6                   4
   0                   6


   17                  17
   14                  14
   17                  11
   11                  17
   20                  20
```

The `randrange` Function with a `seed`

```python
from random import randrange, seed

seed(22)
for i in range(5):
    print(randrange(0,5))

print()

for i in range(5):
    print(randrange(5,10))
```

```
first run      second run

    1              1
    1              1
    0              0
    4              4
    3              3

    6              6
    5              5
    7              7
    5              5
    6              6
```

The `random()` Function

Returns a random float value between 0 (inclusive) and 1 (exclusive)

- `random()` returns a uniformly distributed random value between 0 and 1.

The `random()` Function

```python
from random import random

for i in range(5):
    print(random())
```

```
first run                  second run

0.19906663630160715        0.882867718929644
0.8054359163103013         0.03205603257735179
0.31200284723952587        0.3260438439284473
0.8143181418331964         0.7698073994812211
0.18713080543555582        0.7160519442896809
```

choice and shuffle

```
myList = ["Python", ("A", 1), 22, [1, 2, 3], "X", "Y", "Z"]
```

- `choice(seq)` returns a random element from the sequence seq.

```
for i in range(5):
    print(choice(myList))
```

```
X
[1, 2, 3]
X
22
Y
```

- `shuffle(seq)` returns a randomly shuffled sequence seq.

```
for i in range(5):
    shuffle(myList)
    print(myList)
```

```
['X', 'Python', [1, 2, 3], 'Z', 'Y', ('A', 1), 22]
['X', ('A', 1), 'Python', 'Y', 22, [1, 2, 3], 'Z']
['Y', ('A', 1), 22, 'Python', 'Z', 'X', [1, 2, 3]]
['X', ('A', 1), 'Y', 'Z', 22, [1, 2, 3], 'Python']
[('A', 1), 'Y', 'Z', 22, 'X', 'Python', [1, 2, 3]]
```

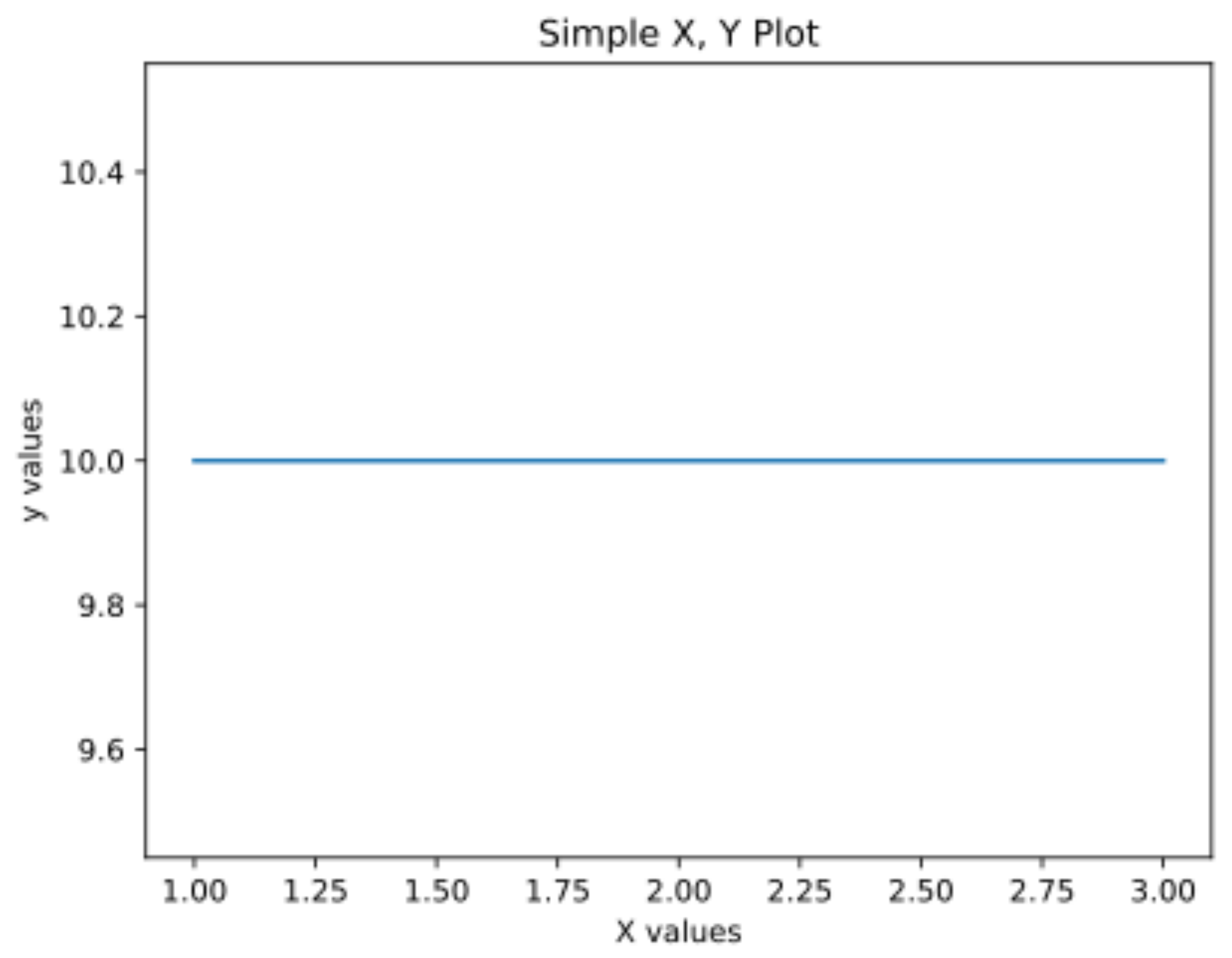
matplotlib LIBRARY

PLOTTING WITH matplotlib

- matplotlib a popular plotting library for Python.
- Typically used to create plots, e.g.,
  - Line plot.
  - Histogram.
  - Bar charts.
  - Pie charts.
  - Scatter plots.
  - …

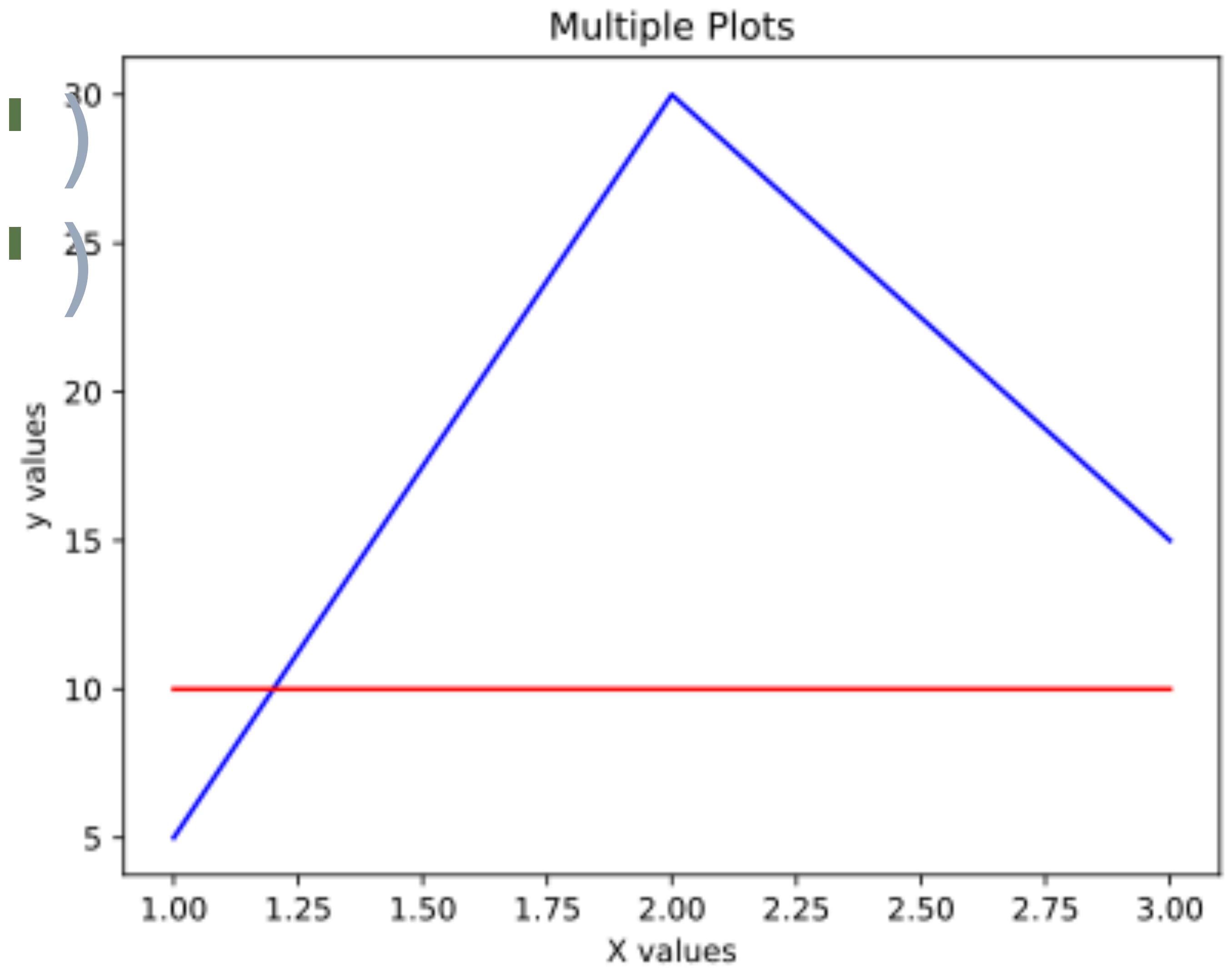
PLOTTING X and Y data

```
import matplotlib.pyplot as pyplot

pyplot.plot([1, 2, 3], [10, 10, 10])
pyplot.title('Simple X, Y Plot' ) # title
pyplot.xlabel('X values') # x-axis label
pyplot.ylabel('y values') # y-axis label

pyplot.show()
```



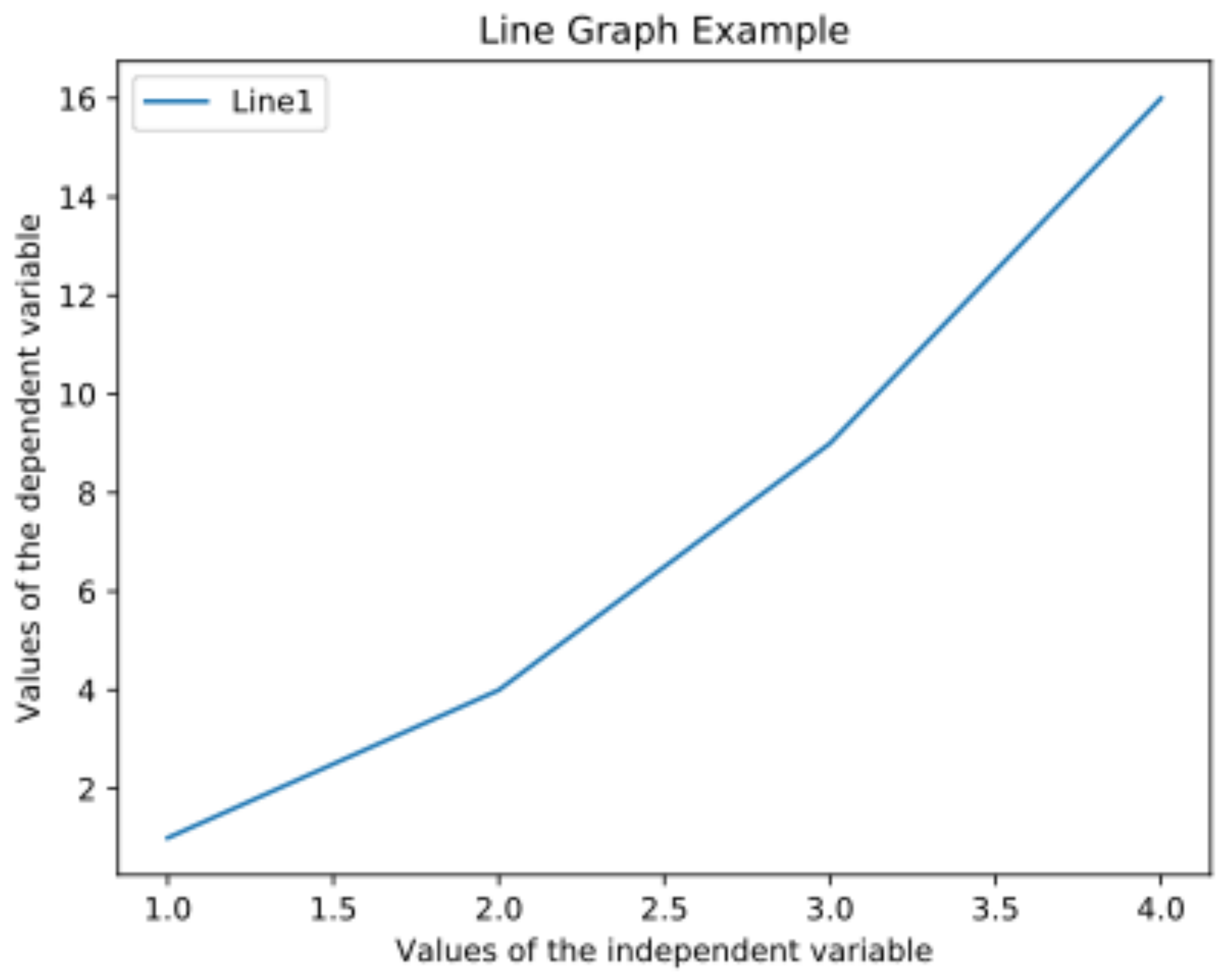
MULTIPLE PLOTS

```
import matplotlib.pyplot as pyplot

pyplot.plot([1, 2, 3], [5, 30, 15], color = 'blue')
pyplot.plot([1, 2, 3], [10, 10, 10], color = 'red')

pyplot.title('Multiple Plots' ) # title
pyplot.xlabel('X values') # x-axis label
pyplot.ylabel('y values') # y-axis label

pyplot.show()
```



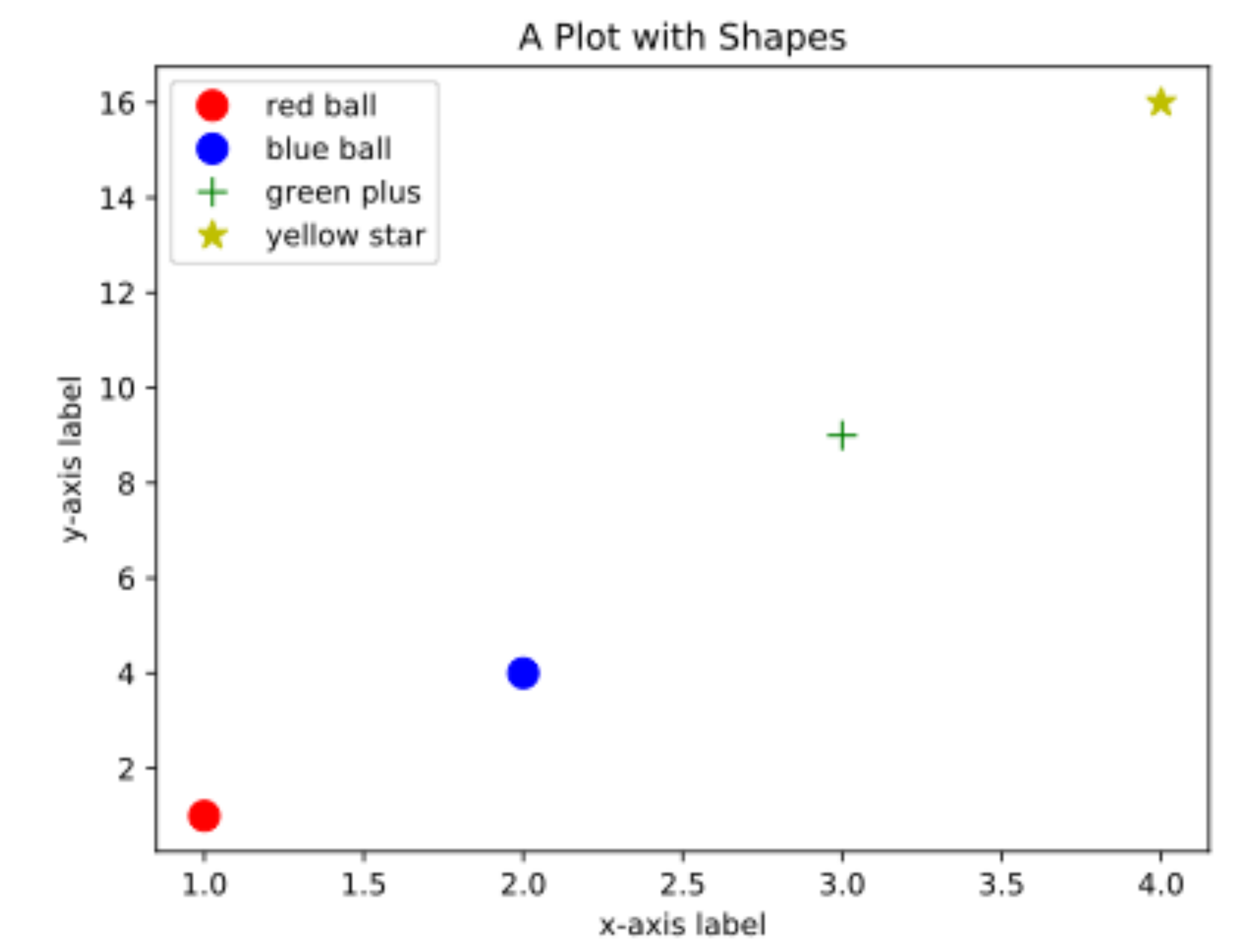
ADDING LABELS AND LEGEND

```
import matplotlib.pyplot as pyplot

pyplot.plot([1,2,3,4], [1,4,9,16], label = 'Line1')
pyplot.title('Line Graph Example')
pyplot.ylabel('Values of the dependent variable')
pyplot.xlabel('Values of the independent variable')
pyplot.legend()


pyplot.show()
```



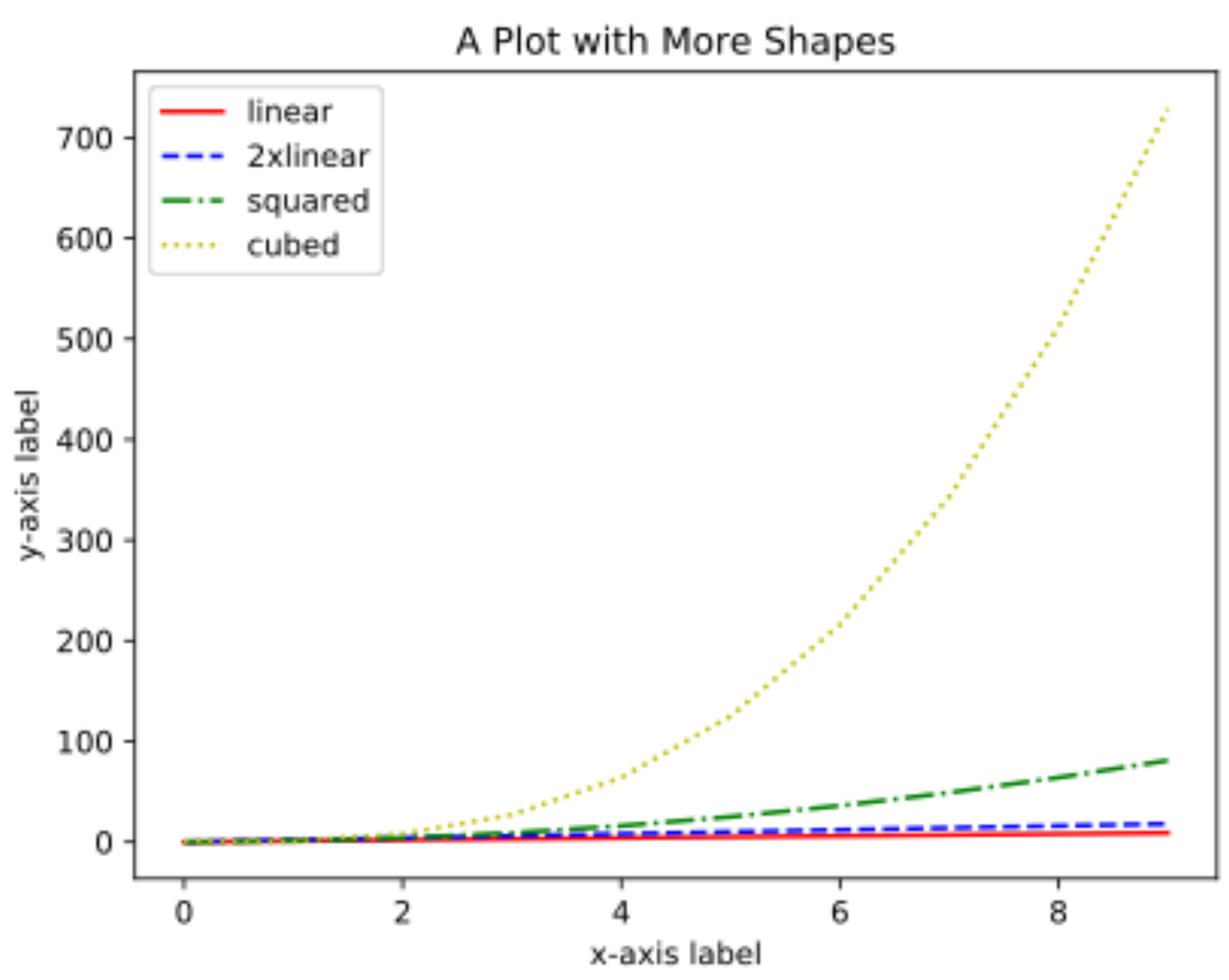
SHAPES and LABELS

```
import matplotlib.pyplot as pyplot

pyplot.plot([1], [1], 'ro', label = 'red ball', markersize = 10)
pyplot.plot([2], [4], 'bo', label = 'blue ball', markersize = 10)
pyplot.plot([3], [9], 'g+', label = 'green plus', markersize = 10)
pyplot.plot([4], [16], 'y*', label = 'yellow star', markersize = 10)
pyplot.title('A Plot with Shapes')
pyplot.xlabel('x-axis label')
pyplot.ylabel('y-axis label')

pyplot.legend()


pyplot.show()
```



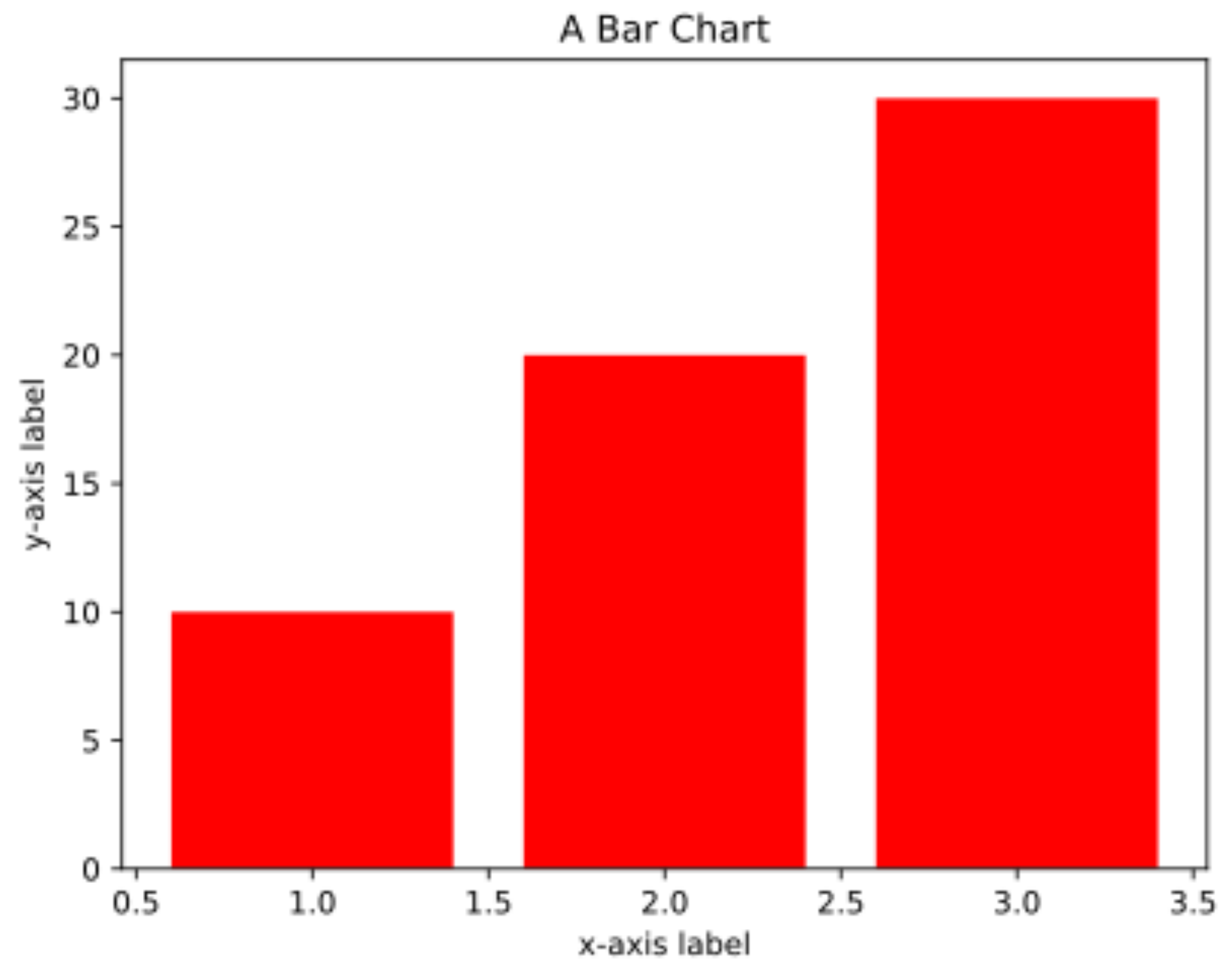
MORE SHAPES

```
import matplotlib.pyplot as pyplot

pyplot.plot(list(range(10)), [x for x in range(10)], 'r', label = 'linear')
pyplot.plot(list(range(10)), [x*2 for x in range(10)], '--b', label = '2xlinear')
pyplot.plot(list(range(10)), [x**2 for x in range(10)], '-.g', label = 'squared')
pyplot.plot(list(range(10)), [x**3 for x in range(10)], ':y', label = 'cubed')
pyplot.title('A Plot with More Shapes')
pyplot.xlabel('x-axis label')
pyplot.ylabel('y-axis label')
pyplot.legend()

pyplot.show()
```



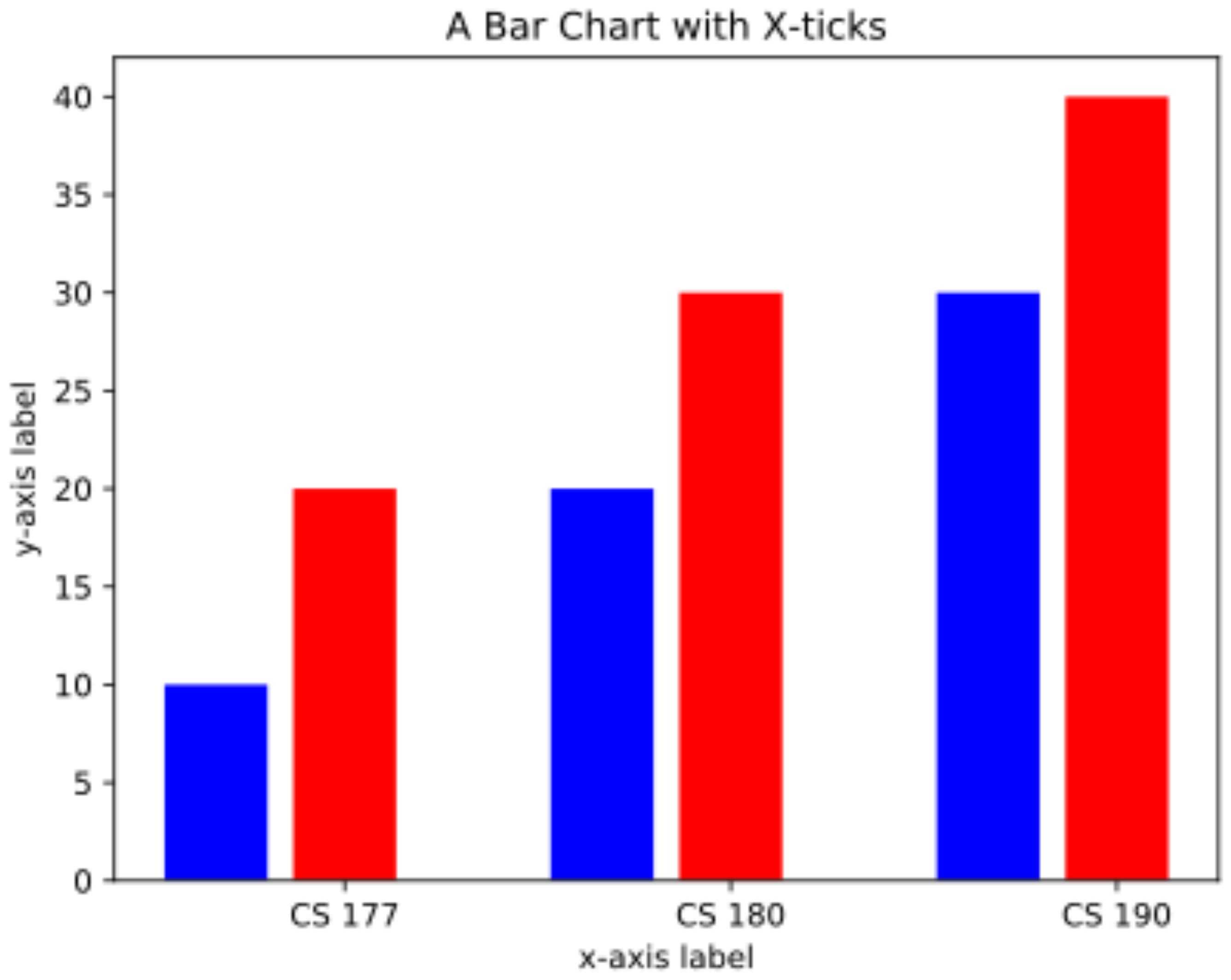
BAR CHARTS

```
import matplotlib.pyplot as pyplot

pyplot.bar([1, 2, 3], [10, 20, 30], color = 'red')
pyplot.title('A Bar Chart')
pyplot.xlabel('x-axis label')
pyplot.ylabel('y-axis label')

pyplot.show()
```



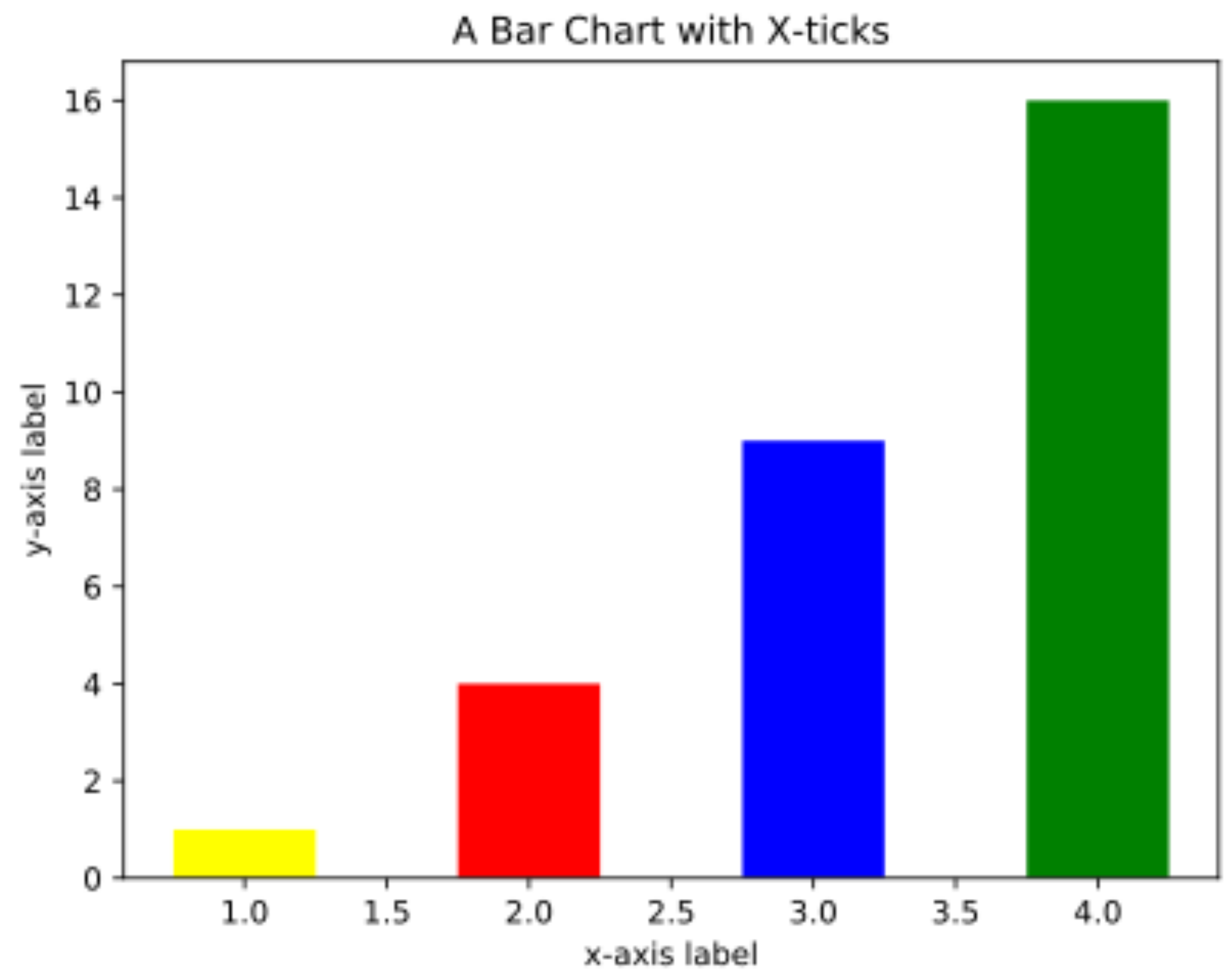
XTICKS

```
import matplotlib.pyplot as pyplot

pyplot.bar([1, 4, 7], [10, 20, 30], color = 'blue')
pyplot.bar([2, 5, 8], [20, 30, 40], color = 'red')
pyplot.xticks([2, 5, 8], ['CS 177', 'CS 180', 'CS 190'])
pyplot.title('A Bar Chart with X-ticks')
pyplot.xlabel('x-axis label')
pyplot.ylabel('y-axis label')

pyplot.show()
```



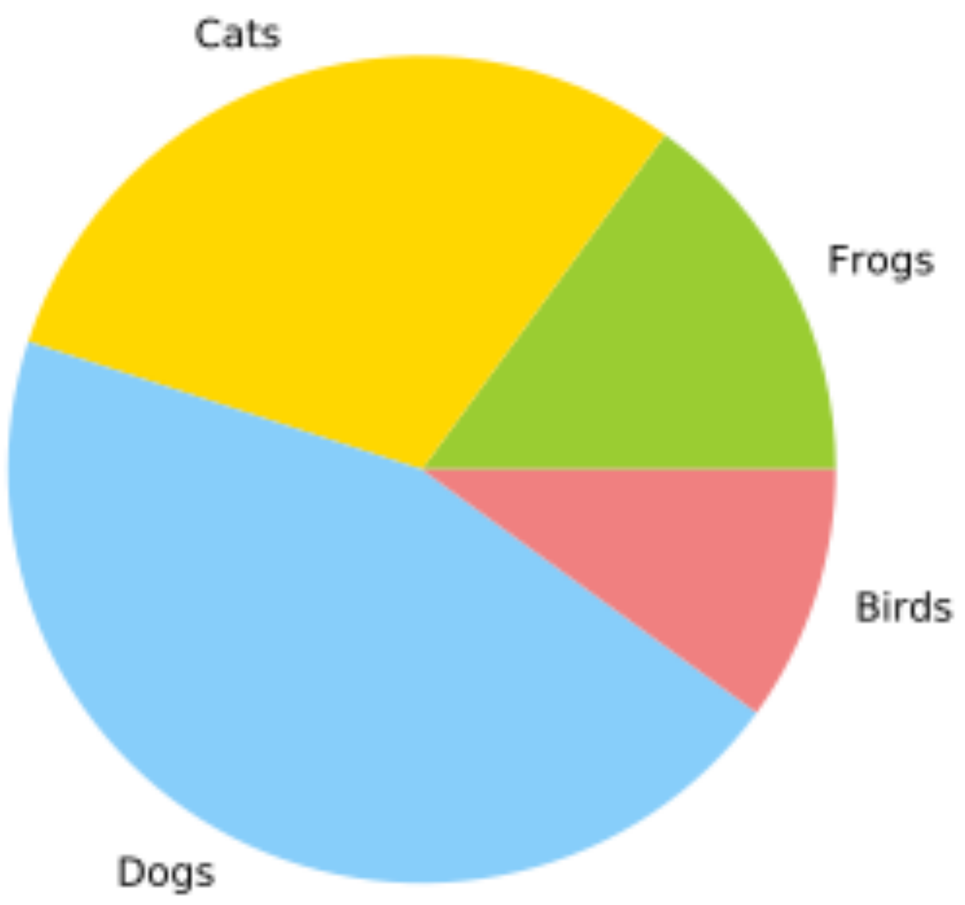
BAR COLORS

```
import matplotlib.pyplot as pyplot

x = [1,2,3,4]
y = [1,4,9,16]
color = ['yellow', 'red', 'blue', 'green']
for i in range(len(x)):
    pyplot.bar(x[i], y[i], width = 0.5, color= color[i])

pyplot.title('A Bar Chart with X-ticks')
pyplot.xlabel('x-axis label')
pyplot.ylabel('y-axis label')

pyplot.show()
```



PIE CHARTS

```
import matplotlib.pyplot as pyplot

labels = 'Frogs', 'Cats', 'Dogs', 'Birds'
sizes = [15, 30, 45, 10]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
pyplot.pie(sizes, labels = labels, colors = colors)
pyplot.show()
```



SAVING YOUR FIGURE

- You can save your figure to a file using savefig

```
import matplotlib.pyplot as pyplot
#
# plot your figure
#
pyplot.savefig('figureName.pdf') # you may change file extension e.g., png, jpg, etc.
```

THANK YOU!