

고급통계프로그래밍 # 9

2017580034 통계학과 이유민

#1

$y_1 = \sin(5x)$, $y_2 = \cos(4x)$, $0 \leq x \leq 2\pi$ 일 때, y_1 과 y_2 를 동일한 그래프에 겹쳐서 그리시오.

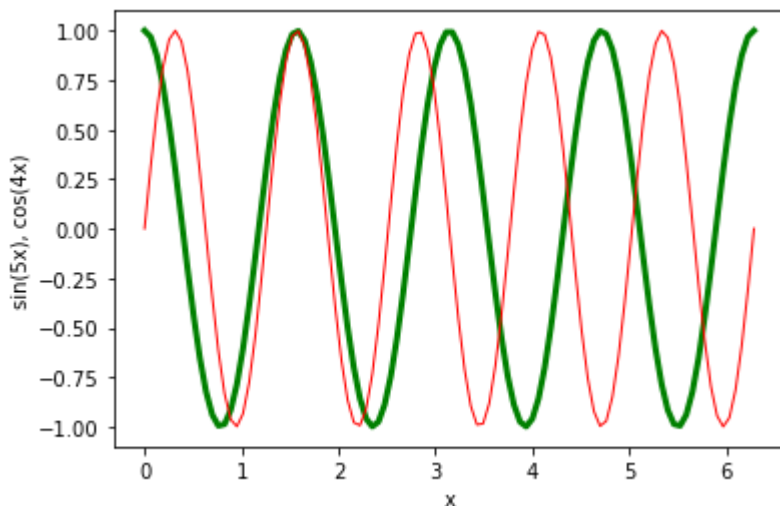
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

x=np.linspace(0, np.pi*2, 100) #x 범위
c,s=np.cos(4*x),np.sin(5*x) # y1, y2

#plot
plt.plot(x,c, color='green', linewidth=3.0) #y2
plt.plot(x,s, color='red', linewidth=1.0) #y1
plt.xlabel('x')
plt.ylabel('sin(5x), cos(4x)')

plt.show()
```



#2

어느 강좌에서 학생들의 몸무게가 65, 78, 80, 140, 100, 75, 78 킬로그램이다. 주어진 데이터를 이용하여 평균, 중위수, 최빈값, 분산, 표준편차 등을 계산하시오

In [2]:

```
# -*- coding:utf-8 -*-
from collections import Counter as cc
import numpy as np
a=[65, 78, 80, 140, 100, 75, 78]

print('최소값', np.amin(a))
print('최대값', np.amax(a))
print('평균:', np.mean(a))
print('중위수', np.median(a))
cnt=cc(a)
print('최빈값:', cnt.most_common()[0][0]) #최빈값&횟수
print('분산:', np.var(a))
print('표준편차:', np.std(a))
```

최소값 65
최대값 140
평균: 88.0
중위수 78
최빈값: [(78, 2)]
분산: 544.2857142857143
표준편차: 23.329931724840396

#3

$U(0,1)$ 에서 난수를 1000개 발생하여 x축에는 표본수 n , y축에는 누적평균을 선그래프로 그리시오. 누적평균값은 n 이 커짐에 따라 어떤 값을 수렴하는가?

In [3]:

```
import numpy as np
import random
from matplotlib import pyplot as plt

a=[]
b=[]

#난수 발생
def ur(a,b):
    for i in range(1000):
        a.append(random.uniform(0,1))
        b.append(averagess(a))
    return a,b

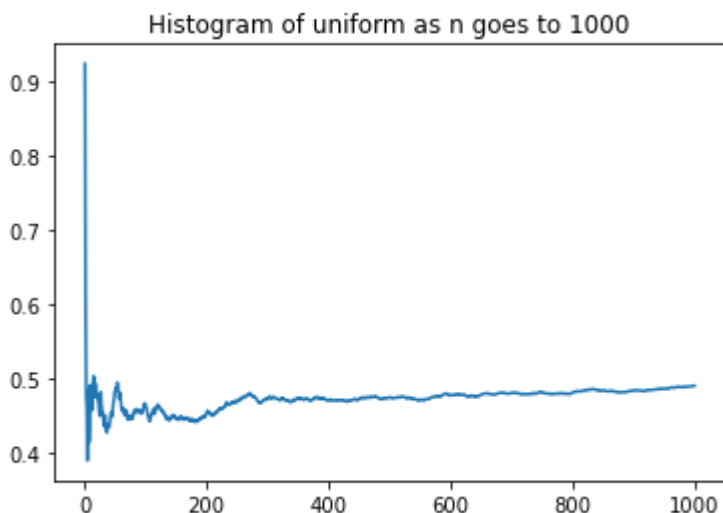
# 누적평균
def averagess(values):
    # 0인 경우는 평균=0 return
    if len(values) == 0:
        return 0
    # 평균 구함
    else:
        return sum(values) / len(values)

# 누적평균 plot
def ss(b):
    xs=np.linspace(0,1000,1000)
    plt.plot(xs,b)
    plt.title("Histogram of uniform as n goes to 1000")

    plt.show()
```

In [4]:

```
ur(a,b)
ss(b)
#As n goes to 1000, cumulative mean goes to 0.5
```



두 3×3 행렬을 생성하는 코드를 작성하시오. 각 원소는 1과 100사이의 정수중 랜덤하게 선택된다. 또한 생성된 두 행렬 및 두 행렬들의 원소별 곱을 출력하시오.

In [5]:

```
import numpy as np
#1~100 random
a=np.matrix(np.random.randint(1,100,size=(3,3)))
b=np.matrix(np.random.randint(1,100,size=(3,3)))
```

In [6]:

a

Out[6]:

```
matrix([[17,  3, 50],
        [74, 67, 80],
        [19, 60, 83]])
```

In [7]:

b

Out[7]:

```
matrix([[18, 18, 33],
        [49, 17, 94],
        [92, 35, 95]])
```

In [8]:

```
#multiply as matrix
a*b
```

Out[8]:

```
matrix([[ 5053,  2107,  5593],
        [11975,  5271, 16340],
        [10918,  4267, 14152]])
```

In [9]:

```
np.dot(a,b)
```

Out[9]:

```
matrix([[ 5053,  2107,  5593],
        [11975,  5271, 16340],
        [10918,  4267, 14152]])
```

In [10]:

```
#multiply as array  
np.array(a)*np.array(b)
```

Out[10]:

```
array([[ 306,   54, 1650],  
       [3626, 1139, 7520],  
       [1748, 2100, 7885]])
```