

math 모듈 활용

<https://docs.python.org/3/library/math.html>

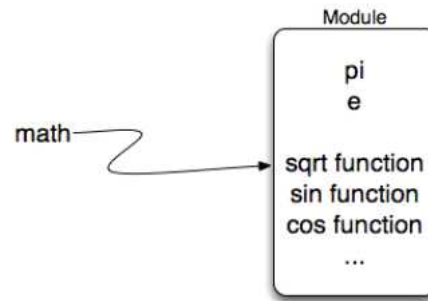
math 모듈

잘 알려진 수학 함수들의 대부분을 제공

(모듈 : 서로 관련있는 함수들을 모아 놓은 파일)

import math

math모듈에 정의된 함수에 접근하기 위해서는 모듈 이름과 함수 이름을 점으로 연결하는 점 표기법 사용



다음 내용을 실행해보세요.	실행결과 및 알게 된 내용
<pre>import math print(math.pi) print(math.e) print(math.sqrt(4)) print(math.radians(90))</pre>	<p>pi 원주율</p> <p>자연 상수(e)</p> <p>제곱근</p> <p>60분법으로 표현된 각도를 라디안으로 변환</p>
<pre>from math import * r = radians(180) print(cos(r))</pre>	
<pre>from math import * for i in range(10): print(pow(2, i))</pre>	
<pre>from math import * for i in range(10): print(factorial(i))</pre>	
<pre>import math print(math.exp(1)) for i in range(16): print(math.log(i+1, 2))</pre>	

㉠ 파이썬으로 삼각함수표 만들기

㉢ 파이썬 레퍼런스를 참고하여 어떤 수식을 파이썬으로 나타내보자.

csv 모듈 활용

<https://docs.python.org/3/library/csv.html>

1) CSV (Comma-Separated Values)

각 항목의 값들이 쉼표(,)로 분리되어 있고, 행은 개행 문자(new line)로 구분하는 파일 포맷

① CSV 파일 읽어오기

csv 파일에서 읽어온 내용은 각 행별로 리스트 형태로 저장

	A	B	C
1	3101	30	40
2	3102	30	40
3	3103	30	32
4	3104	30	34
5	3105	18	19

eval.csv(학번, 과제1, 과제2)

```
1 import csv
2 f=open('eval.csv')
3 f_score=csv.reader(f)
4 for stu in f_score :
5     print(stu)
```

eval.py

```
['3101', '30', '40']
['3102', '30', '40']
['3103', '30', '32']
['3104', '30', '34']
['3105', '18', '19']
```

eval.py 실행결과

```
for stu in f_score :
    print(stu[1])
```

출력 내용 :

② CSV 파일 다루기

```
#csv 모듈에 있는 reader 함수를 통해서 파일(객체, f) 내용을 읽어 f_score에 저장
import csv
f=open('eval.csv')
f_score=csv.reader(f)
```

```
#for문을 이용해 한 줄(리스트)씩 읽어옴
for stu in f_score :
    print(stu)      # 출력내용 : ['3101', '30', '40']
    print(stu[2])   # 출력내용 : 40
```

[실습-1] 파일 데이터의 평균 구하기

① 각 과제의 평균을 출력하는 프로그램 작성(소수점 둘째자리까지)

② 학생의 학번을 입력하면 과제1, 과제2 점수의 평균(정수)를 출력하는 프로그램 작성

인덱스	요소
0	['3101', 35]
1	['3102', 35]
2	['3103', 31]
3	['3104', 32]
4	['3105', 33]

리스트 구조

키	값
'3101'	35
'3102'	35
'3103'	31
'3104'	32
'3105'	33

딕셔너리 구조

[실습-2] 수행평가 프로그램 만들기

학생의 학번을 입력하면 해당 학생의 총점/평균, 학생이 받은 최고점/최하점, 학생의 상중하 평가를 순서대로 출력한다. (5번의 수행 평가 점수 평균이 35점 이상일 경우 '상' 28점 이상일 경우 '중' 28점 미만일 경우 '하' 이다.)

입력	출력
3101	118 23.6 40 30 하

입출력 예시

	A	B	C	D	E	F
1	학번	평가1	평가2	평가3	평가4	평가5
2	3101	30	40	18	30	35
3	3102	30	40	30	30	38
4	3103	30	32	30	18	38
5	3104	30	34	20	18	35
6	3105	18	19	30	20	36
7	3106	18	40	30	18	35

i_eval.xls(수행평가 점수 파일)

① 딕셔너리 설계

키	값
'3101'	[30,40,18,30,35]
'3102'	[30,40,30,30,38]
'3103'	[30,32,30,18,38]
'3104'	[30,34,20,18,35]
'3105'	[18,19,30,20,36]

stu_dict(딕셔너리 구조)

```
import csv
f=open('i_eval.csv')
f_score=csv.reader(f)
stu_dict={}
for stu in f_score :
```

② 함수 설계



함수 기능	총점, 평균 구하기		
함수 이름			
매개 변수			
함수 결과			

③ 함수 만들기

```
def SumAvg(id) :
```

```
print(tot, avg)
```

```
def MaxMin(id) :
```

```
print(high, low)
```

```
def stuEval(id) :
```

```
if avg>=35 :
    print('상')
elif avg>=28 :
    print('중')
else :
    print('하')
```

2) 공공 데이터

① 공공데이터

- “공공데이터”란 데이터베이스, 전자화 된 파일 등 공공기관이 법령 등에서 정하는 목적을 위하여 생성 또는 취득하여 관리하고 있는 광(光) 또는 전자적 방식으로 처리된 자료 또는 정보를 말한다.
- 공공데이터는 정부에서 운영하는 공공데이터포털(www.data.go.kr)에서 제공하고 있으며 정부부처 및 지방자치단체 홈페이지(<http://data.seoul.go.kr>)에서도 찾아볼 수 있다.
- 또한 검색 포털인 네이버에서 운영하는 네이버 데이터랩(datalab.naver.com)에서도 공공데이터를 포함한 다양한 검색 관련 데이터를 제공하고 있다.

② 공공데이터 찾기

③ 데이터 가공하기(전처리 과정)

공공 데이터 포털에서 제공하는 원본 파일을 파이썬에서 활용할 수 있는 형태로 가공(불필요한 데이터 삭제, 오류 수정 등)하는 과정이다.

A	B	C	D	E
Period	자치구	Total	Male	Female
2016.1/4	total	40.9	40.1	41.6
2016.1/4	Jongno-gu	42.8	42.2	43.5
2016.1/4	Jung-gu	43.2	42.6	44.0
2016.1/4	Yongsan-gu	41.9	41.2	42.7
2016.1/4	Seongdong-gu	40.9	40.3	41.4
2016.1/4	Gwangjin-gu	39.6	39.0	40.1
2016.1/4	Dongdaemun-gu	41.4	40.7	42.2
2016.1/4	Jungnang-gu	42.4	41.5	43.4
2016.1/4	Seongbuk-gu	41.1	40.4	41.9
2016.1/4	Gangbuk-gu	43.7	42.4	45.1
2016.1/4	Dobong-gu	42.9	41.7	44.1
2016.1/4	Nowon-gu	41.1	39.7	42.3
2016.1/4	Eunpyeong-gu	41.8	40.9	42.8
2016.1/4	Seodaemun-gu	41.4	40.6	42.2
2016.1/4	Mapo-gu	39.3	39.0	39.7
2016.1/4	Yangcheon-gu	40.8	39.9	41.6
2016.1/4	Gangseo-gu	39.9	39.1	40.7
2016.1/4	Guro-gu	41.2	40.7	41.8
2016.1/4	Geumcheon-gu	43.1	42.1	44.2

원본 엑셀 파일

수정한 CSV 파일

[실습-1] 공공데이터 활용

서울시 산불발생 (원인별) 통계 데이터(영문)를 찾아 저장하고 전처리 과정을 거친 후 csv 파일로 변환

서울 통계포 (stat)						
분류설정 <input checked="" type="radio"/> 조회범위 상세설정하기 <input type="radio"/> 언어 English <input type="button" value="소수점"/>						
기간 년 2009 년 2015 년						
<input type="button" value="자료검색"/>						
산불발생(원인별)						
기간	구별	합계	실화 부주의	방화 방화명확	방화 방화의심	발화요인(미상)
	합계	51	37	1	1	12
	종로구	1	1	-	-	-
	용산구	1	1	-	-	-

① 구 이름을 리스트로 만드는 프로그램 작성

```
[ 'Jongno-gu', 'Jung-gu', 'Seongdong-gu', 'Jungnang-gu', 'Seongbuk-gu', ..... ]
```

② 연도별 산불발생 수를 딕셔너리로 만드는 프로그램 작성

```
{2009: 71, 2010: 49, 2011: 51, 2012: 59, 2013: 38, 2014: 55, 2015: 51}
```

③ 구별 산불발생 수를 딕셔너리로 만드는 프로그램 작성


```
{ 'Seocho-gu': 17, 'Seodaemun-gu': 25, 'Geumcheon-gu': 1, 'Jongno-gu': 18, ..... }
```

matplotlib(pylab) 모듈 활용

http://matplotlib.org/1.3.1/users/pyplot_tutorial.html

matplotlib은 데이터를 차트나 플롯(plot)으로 시각화하는 패키지로 해당 기능과 관련된 여러 모듈의 집합이다. pylab은 matplotlib 패키지에 있는 서브패키지이다. pylab은 matlab의 수치해석 소프트웨어의 시각화 명령을 거의 그대로 사용할 수 있도록 하는 명령어 집합을 제공한다.

- plot() 함수의 첫 번째 인수는 x축 상에 표시하고자 하는 숫자의 리스트, 두 번째 인수는 y축 상에 표시하고자 하는 숫자의 리스트이다. (한 개만 입력시 y값, x값은 0부터 자동 생성)
- plot() 함수가 반환하는 객체는 그래프 정보를 포함한다. 실제로 그래프를 표시하기 위해서는 show() 함수를 호출해야 한다.

다음 내용을 실행해보세요.	실행결과 및 알게 된 내용
<pre>import pylab a=[2, 4, 6] pylab.plot(a) pylab.show()</pre>	
<pre>import pylab as plt x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] y=[1, 4, 9, 16, 25, 36, 49, 64, 81, 100] plt.plot(x, y) #plt.plot(x, y, marker='o') plt.savefig('C:\graph.png') plt.show()</pre>	
<pre>import pylab x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] p=[1, 4, 9, 16, 25, 36, 49, 64, 81, 100] fibo=[1, 1, 2, 3, 5, 8, 13, 21, 34, 55] pylab.title('my graph') pylab.xlabel('x coordinate') pylab.ylabel('y coordinate') pylab.plot(x, p, 'ro-', x, fibo, 'cs--') pylab.legend(['pow', 'fibo']) #pylab.legend(['pow', 'fibo'], loc=0) pylab.show()</pre>	<p>리스트 x의 항목을 1~20까지 자동 생성해보세요. 리스트 p의 항목을 x값에 따라 자동 생성해보세요. 리스트 fibo의 항목을 x값에 따라 자동 생성해보세요.</p>
<pre>import pylab x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] p=[1, 4, 9, 16, 25, 36, 49, 64, 81, 100] fibo=[1, 1, 2, 3, 5, 8, 13, 21, 34, 55] pylab.subplot(2,1,1) pylab.plot(x, p, 'm*-') pylab.subplot(2,1,2) pylab.plot(x, fibo, 'kd--') pylab.show()</pre>	<p>pylab.subplot(1,2,1), pylab.subplot(1,2,2)로 변경 후 비교해보자.</p> 

1) 함수

plot(x, y, marker='o') plot(x, y, 'o')	marker='o' 리스트 내의 각 점을 o와 같은 모양의 작은 점으로 표시한다. 해당 점들을 연결하는 라인을 포함하는 것을 기본설정으로 하고, marker=를 생략할 경우 선으로 연결하지 않고 점만 표시한다.
plot(x, y, 'bo')	plot x and y using blue circle markers
plot(x, y_1, 'ro', x, y_2, 'bs--')	라인 플롯에서 선을 하나가 아니라 여러 개 그릴 경우, x 데이터, y 데이터, 스타일 문자열을 반복하여 인수로 넘긴다.
legend(['graph1', 'graph2']) legend(['g1', 'g2'], loc=0)	여러 개의 라인 플롯을 동시에 그리는 경우에 각 선이 무슨 자료를 표시하는 지를 보여주기 위해 legend 명령으로 범례를 추가한다. 범례의 위치는 자동으로 정해지지만 수동으로 설정할 경우 loc 인수를 사용한다. (0은 best 위치를 의미함)
subplot(2,1,2)	하나의 윈도우(Figure)안에 여러 개의 플롯을 배열 형태로 보여준다. subplot의 인수는 (2,2,1)를 줄여서 221 라는 하나의 숫자로 표시할 수 있다.
axis([0, 6, 0, 20])	[xmin, xmax, ymin, ymax]
xticks(x, xlabel) yticks(y, ylabel)	플롯이나 차트에서 축상의 위치 표시 지점을 틱(tick)이라고 하고 이 틱에 써진 숫자 혹은 글자를 틱 라벨(tick label)이라고 한다. 틱의 위치나 틱 라벨을 수동으로 설정하고 싶다면 xticks 명령이나 yticks 명령을 사용한다.
grid(True) grid(False)	그리드 사용 여부
barh() 함수	막대차트
scatter() 함수	산점도

2) 스타일

색깔, 마커, 선 종류 순서로 지정하고, 이중 일부가 생략되면 디폴트 값이 적용된다.

① 색깔(color) : 색 이름 또는 약자, RGB코드를 사용한다.

색상 문자열	약자	색상 문자열	약자
blue	b	magenta	m
green	g	yellow	y
red	r	black	k
cyan	c	white	w

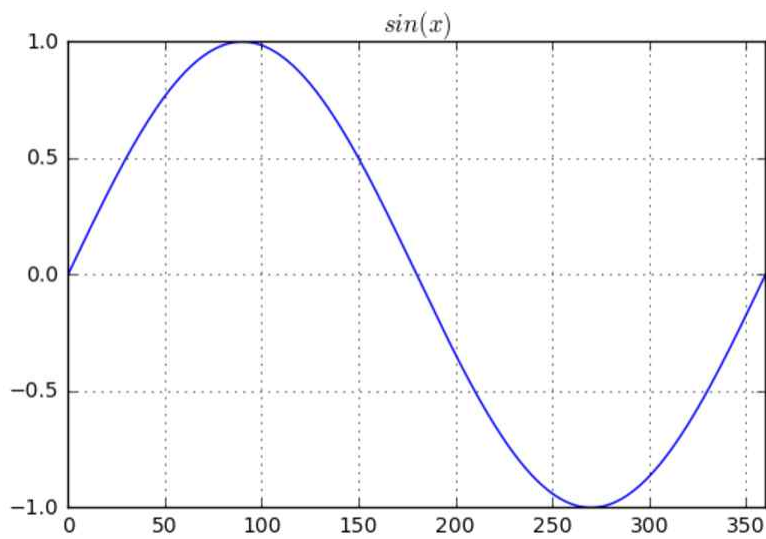
② 마커(marker) : 데이터 위치를 나타내는 기호를 마커라고 한다.

마커 문자열	의미	마커 문자열	의미
o	circle	*	star
v	triangle_down	h	hexagon
^	triangle_up	+	plus
s	squar	x	x
p	pentagon	d	diamond

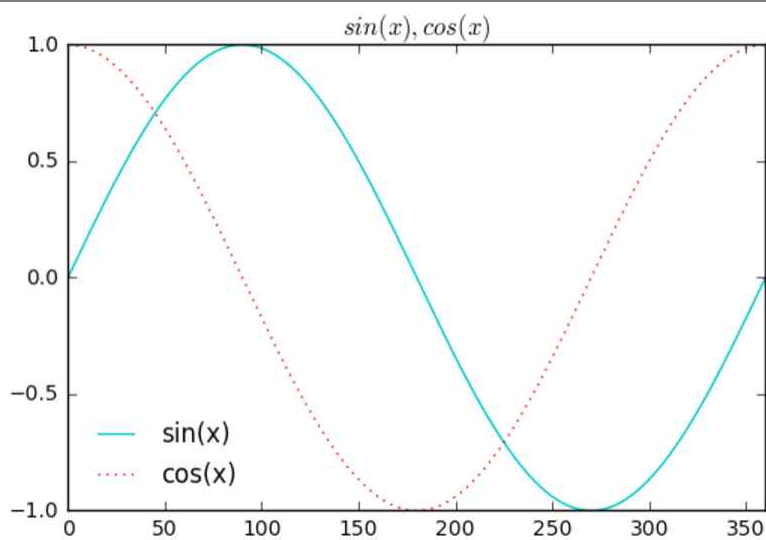
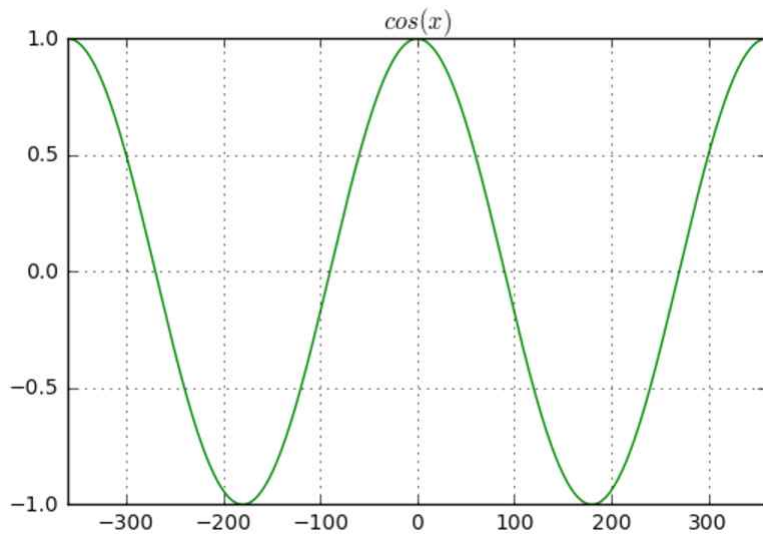
③ 선 스타일(line style) : 실선(solid), 대시선(dashed), 점선(dotted), 대시-점선(dash-dot)이 있다.

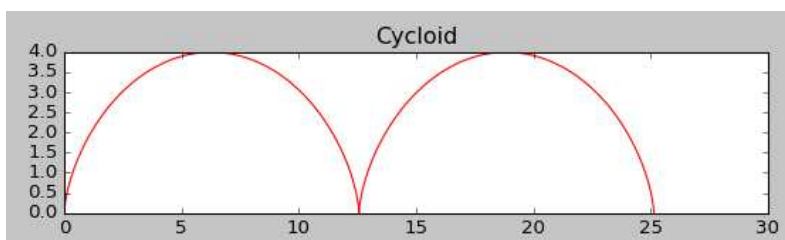
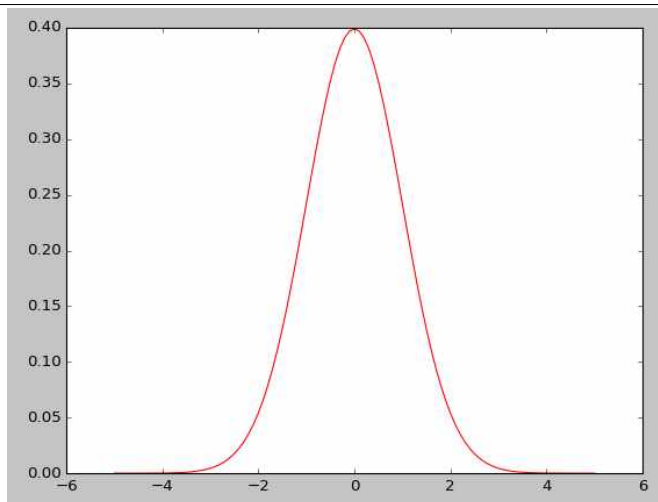
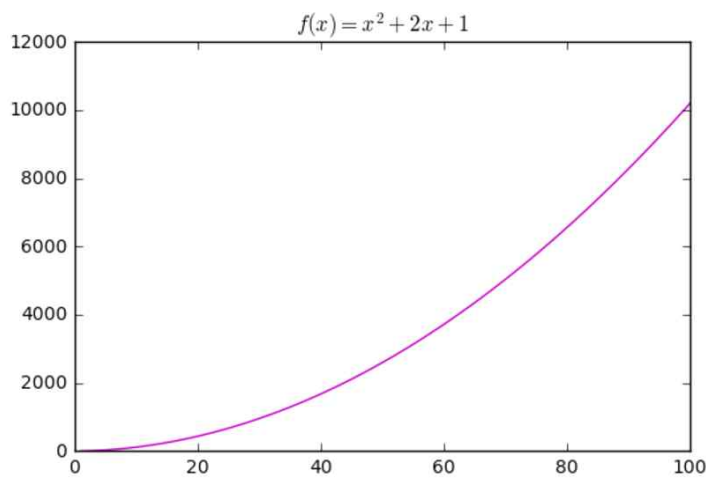
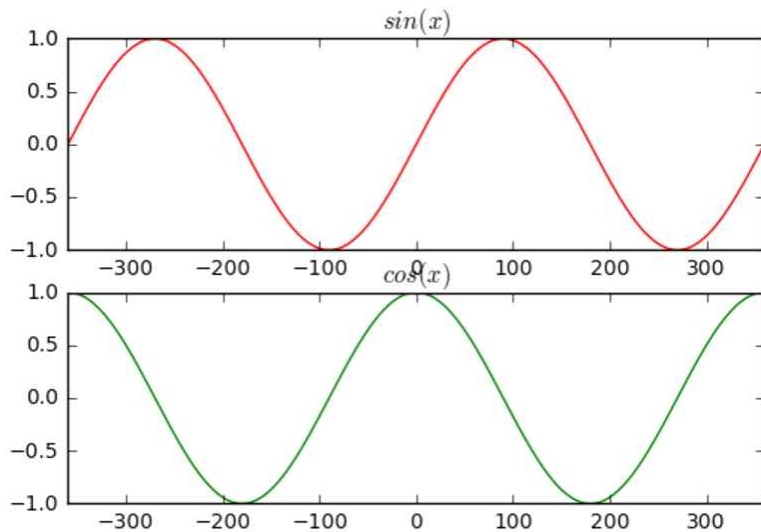
선 스타일 문자열	의미
-	solid line
--	dashed line
-.	dash-dot line
:	dotted

3) python을 활용해 그래프 그리기



```
import math
import pylab
x=range(0,361)
y_s=[]
for i in x :
    r=math.radians(i)
    y_s.append(math.sin(r))
pylab.plot(x,y_s)
pylab.axis([0,360,-1,1])
pylab.title(r'$sin(x)$')
pylab.grid(True)
pylab.show()
```





```
a=[]
for i in range(101):
    a.append(-5+0.1*i)
b=[]
for x in a:
    t=math.exp(-1*(x**2)/2)/math.sqrt(2
    *math.pi)
    b.append(t)
```

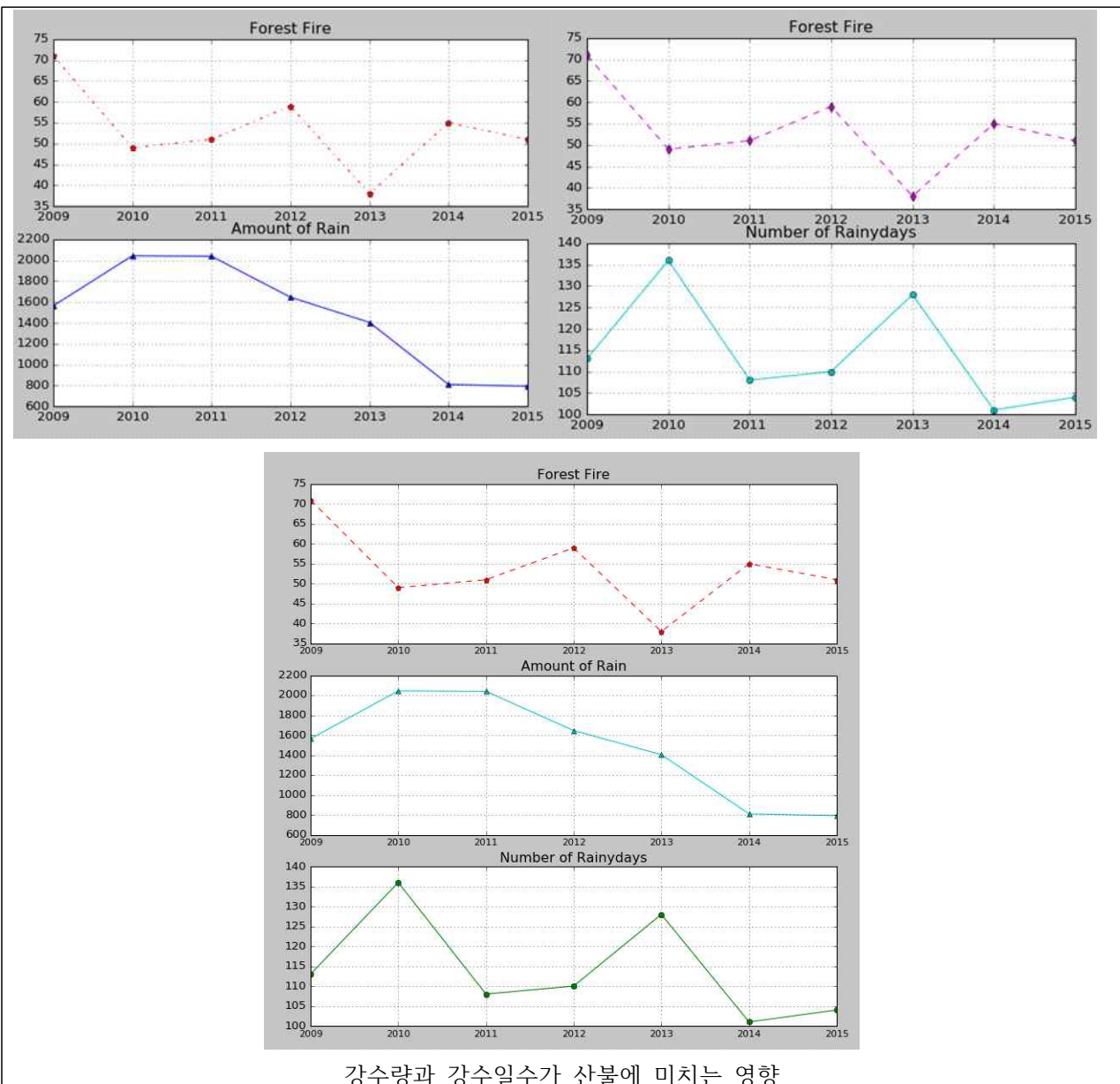
```
def Cyx(t):
    return r*(rad(t)-sin(rad(t)))
def Cyy(t):
    return r*(1-cos(rad(t)))
t=range(0, 721, 1)
x=[]
y=[]
r=float(input('radius:')) #2
for i in t:
    x.append(Cyx(i))
    y.append(Cyy(i))
```


4) python을 활용해 데이터 시각화

① 서울시 산불발생 통계 데이터(4쪽) 와 서울시 강수량 및 강수일수 데이터 분석을 위한 시각화

	A	B	C	D	E	F
1	기간	구분	합계	1월	2월	3월
2	2008	강수량	1356	18	15	54
3	2008	강수일수	110	3	3	10
4	2009	강수량	1564	6	37	64
5	2009	강수일수	113	3	7	8
6	2010	강수량	2044	29	55	83
7	2010	강수일수	136	8	8	12
8	2011	강수량	2039	9	29	15
9	2011	강수일수	108	5	3	4
10	2012	강수량	1646	7	1	47
11	2012	강수일수	110	6	4	8
12	2013	강수량	1404	22	74	27
13	2013	강수일수	128	7	8	8
14	2014	강수량	809	13	16	7
15	2014	강수일수	101	5	5	6

서울시 강수량 강수일수 데이터



② 근무시간과 사회 불평등 상관관계 분석을 위한 시각화

	A	B	C	D	E	F	G
1	LOCATION	INDICATOR	SUBJECT	MEASURE	FREQUENCY	TIME	Value
2	AUS	HRWKD	TOT	HR_WKD	A	1979	1834
3	AUS	HRWKD	TOT	HR_WKD	A	1980	1836
4	AUS	HRWKD	TOT	HR_WKD	A	1981	1820
5	AUS	HRWKD	TOT	HR_WKD	A	1982	1802
6	AUS	HRWKD	TOT	HR_WKD	A	1983	1791
7	AUS	HRWKD	TOT	HR_WKD	A	1984	1802
8	AUS	HRWKD	TOT	HR_WKD	A	1985	1792
9	AUS	HRWKD	TOT	HR_WKD	A	1986	1799
10	AUS	HRWKD	TOT	HR_WKD	A	1987	1797

OECD, 국가별 노동자 1인당 연간 평균 노동시간

	A	B	C	D	E	F	G
1	LOCATION	INDICATOR	SUBJECT	MEASURE	FREQUENCY	TIME	Value
2	AUS	INCOMEIP	P50P10	RT	A	2012	2.2
3	AUS	INCOMEIP	P50P10	RT	A	2014	2.2
4	AUS	INCOMEIP	P90P10	RT	A	2012	4.4
5	AUS	INCOMEIP	P90P10	RT	A	2014	4.3
6	AUS	INCOMEIP	P90P50	RT	A	2012	2
7	AUS	INCOMEIP	P90P50	RT	A	2014	2
8	AUS	INCOMEIS	S90S10	RT	A	2012	8.8
9	AUS	INCOMEIS	S90S10	RT	A	2014	9.4
10	AUS	INCOMEIGINI	INEQ	INEQ	A	2012	0.326

OECD, 국가별 사회 불평등 지수

