

[Lab 3]

- 파이썬 프로그래밍에 대한 자료는 인터넷에서 쉽게 찾을 수 있다.
 - 예제로 배우는 파이썬 프로그래밍
<http://pythonstudy.xyz/>
python 기초: <http://pythonstudy.xyz/Python/Basics>
python 활용: <http://pythonstudy.xyz/Python/Applications>
 - numpy 사용하기
 - Matplotlib 차트/플롯 그리기

- 과학계산을 위한 라이브러리
- 다차원 배열 연산에 유용
- 아나콘다 사용 시 기본적으로 Numpy 설치 되어있음
- Pip을 이용하여 설치
 - >> pip install numpy

numpy 배열 생성 (1)

- rank: 배열의 차원
- shape: 각 차원의 크기를 튜플로 표시

[1, 2, 3, 4]

[[1, 2, 3]

[4, 5, 6]]

rank: 1

rank: 2

shape: (4,)

shape: (3,2)

```
import numpy as np
```

```
list1 = [1, 2, 3, 4]
```

```
a = np.array(list1)
```

```
print(a.shape) # (4, )
```

```
b = np.array([[1,2,3],[4,5,6]])
```

```
print(b.shape) # (2, 3)
```

```
print(b[0,0]) # 1
```

numpy 배열 생성 (2)

```
import numpy as np
```

```
a = np.zeros((2,2))
```

```
print(a)
```

```
# 출력:
```

```
# [[ 0.  0.]
```

```
# [ 0.  0.]]
```

```
a = np.ones((2,3))
```

```
print(a)
```

```
# 출력:
```

```
# [[ 1.  1.  1.]
```

```
# [ 1.  1.  1.]]
```

```
a = np.full((2,3), 5)
```

```
print(a)
```

```
# 출력:
```

```
# [[5 5 5]
```

```
# [5 5 5]]
```

#shape (2,3) 배열에 5를 채운다

```
a = np.eye(3)
```

```
print(a)
```

```
# 출력:
```

```
# [[ 1.  0.  0.]
```

```
# [ 0.  1.  0.]
```

```
# [ 0.  0.  1.]]
```

```
a = np.array(range(20)).reshape((4,5))
```

```
print(a)
```

```
# 출력:
```

```
# [[ 0  1  2  3  4]
```

```
# [ 5  6  7  8  9]
```

```
# [10 11 12 13 14]
```

```
# [15 16 17 18 19]]
```

range(20): 0부터 19까지의 정수들의 리스트 생성
.reshape((4,5): shape (4,5)의 2차원 배열로 변형

- 각 차원별로 슬라이싱 범위를 지정한다

```
import numpy as np
```

```
lst = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
arr = np.array(lst)
```

```
# 슬라이스  
a = arr[0:2, 0:2]  
print(a)  
# 출력:  
# [[1 2]  
#  [4 5]]
```

```
a = arr[1:, 1:]  
print(a)  
# 출력:  
# [[5 6]  
#  [8 9]]
```

numpy 연산

```
import numpy as np

a = np.array([1,2,3])
b = np.array([4,5,6])

# 각 요소 더하기
c = a + b
# c = np.add(a, b)
print(c) # [5 7 9]

# 각 요소 빼기
c = a - b
# c = np.subtract(a, b)
print(c) # [-3 -3 -3]

# 각 요소 곱하기
# c = a * b
c = np.multiply(a, b)
print(c) # [4 10 18]

# 각 요소 나누기
# c = a / b
c = np.divide(a, b)
print(c) # [0.25 0.4 0.5]
```

```
import numpy as np

lst1 = [
    [1,2],
    [3,4]
]

lst2 = [
    [5,6],
    [7,8]
]

a = np.array(lst1)
b = np.array(lst2)

c = np.dot(a, b)
print(c)
# 출력:
# [[19 22]
#  [43 50]]
```

```
import numpy as np

a = np.array([[1,2],[3,4]])

s = np.sum(a)
print(s) # 10

# axis=0 이면, 컬럼끼리 더함
# axis=1 이면, 행끼리 더함
s = np.sum(a, axis=0)
print(s) # [4 6]

s = np.sum(a, axis=1)
print(s) # [3 7]

s = np.prod(a)
print(s) # 24
```

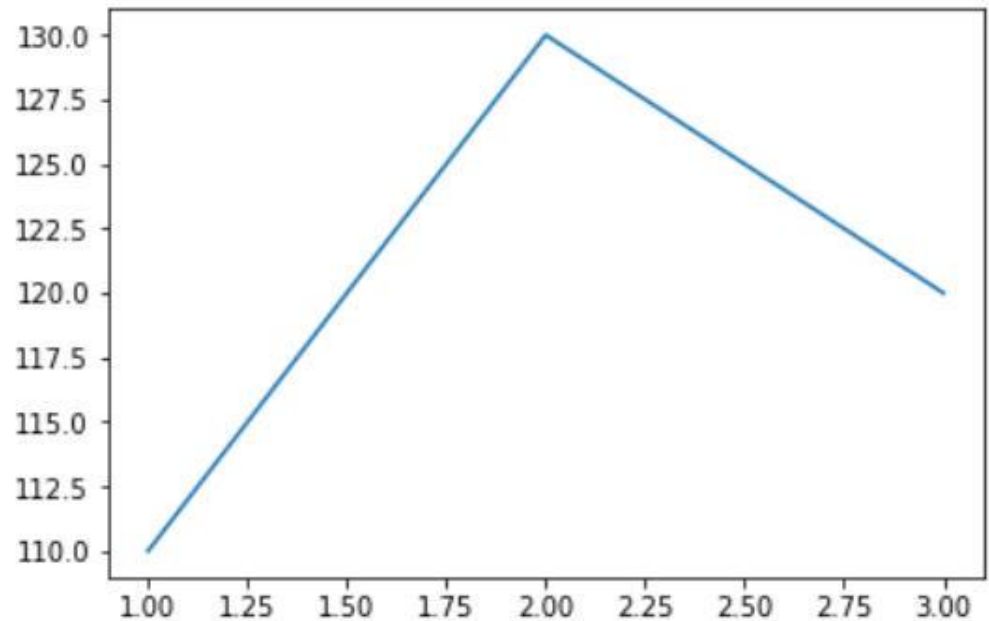
- 데이터를 차트나 플롯으로 그려주는 라이브러리 패키지
- 데이터 시각화 패키지 중 가장 많이 사용된다.
- <https://matplotlib.org/index.html>
- 아나콘다 사용 시 기본적으로 matplotlib 설치
- Pip을 이용하여 설치
 - >> pip install matplotlib

matplotlib 사용하여 그래프 그리기

- matplotlib로부터 matplotlib.pyplot을 import한다
from matplotlib import pyplot as plt
- plt.plot() 는 라인플롯을 그리는 함수이다

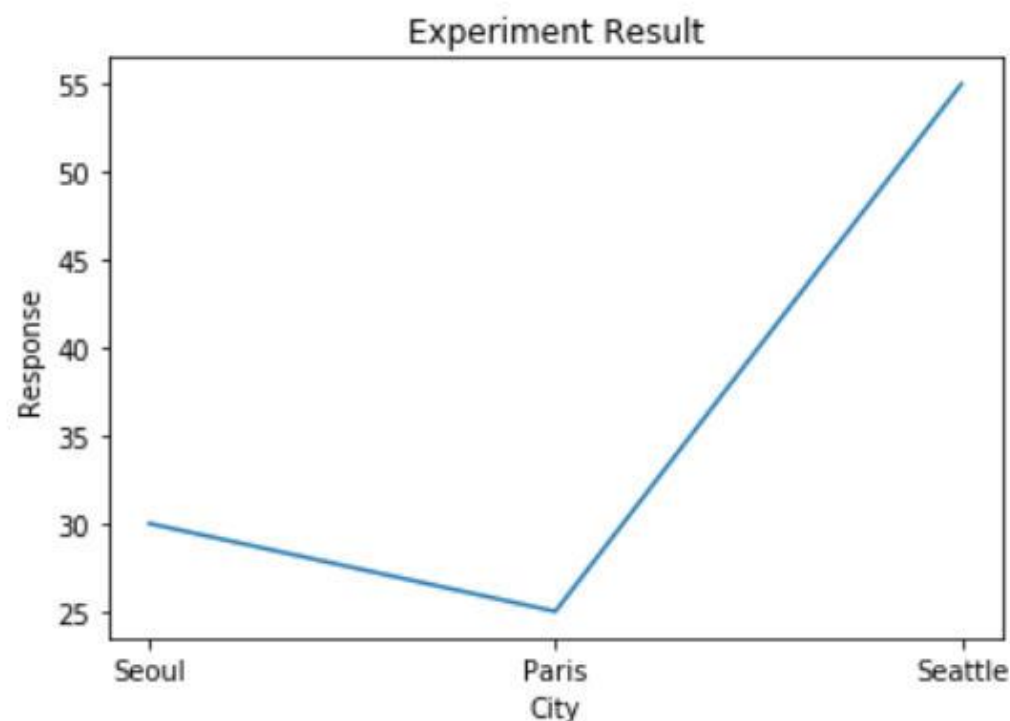
```
from matplotlib import pyplot as plt  
  
plt.plot([1,2,3], [110,130,120])  
plt.show()
```

- Jupyter Notebook에서 위 코드 셀에서 Shift+Enter를 치면 출력란에 차트가 그려진다.
- 첫 번째 라인에
%matplotlib inline
을 넣으면 run 버튼을 눌러 실행하여 그래프를 얻을 수 있다.



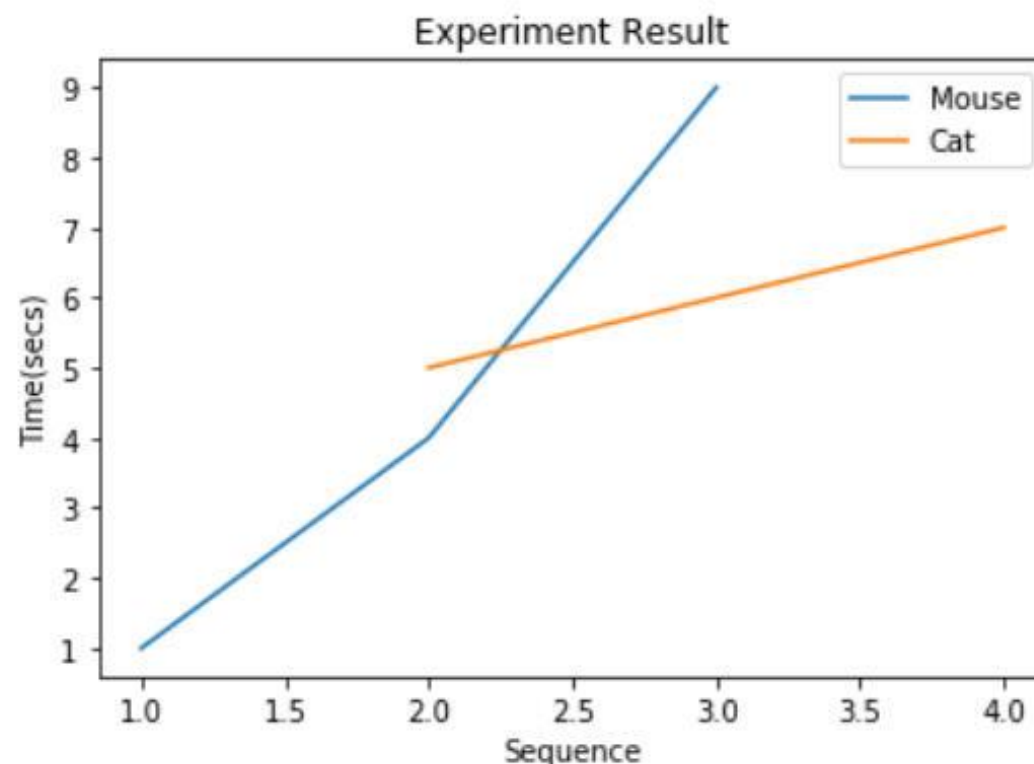
```
from matplotlib import pyplot as plt

plt.plot(["Seoul", "Paris", "Seattle"], [30, 25, 55])
plt.xlabel('City')
plt.ylabel('Response')
plt.title('Experiment Result')
plt.show()
```



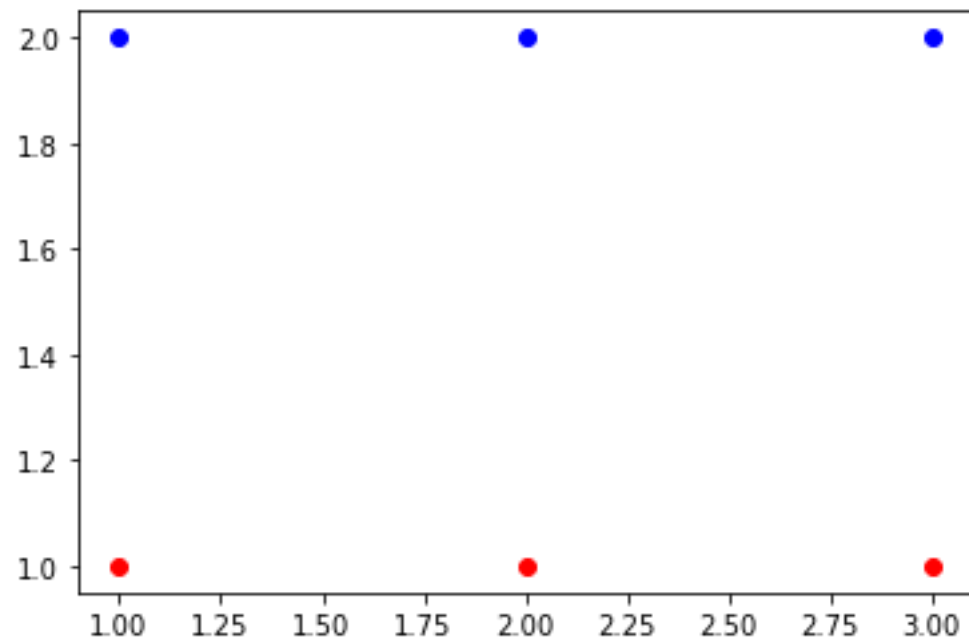
```
from matplotlib import pyplot as plt

plt.plot([1, 2, 3], [1, 4, 9])
plt.plot([2, 3, 4], [5, 6, 7])
plt.xlabel('Sequence')
plt.ylabel('Time(secs)')
plt.title('Experiment Result')
plt.legend(['Mouse', 'Cat'])
plt.show()
```



scatter plot 그리기

```
from matplotlib import pyplot as plt  
  
plt.scatter([1,2,3],[1,1,1],color='r')  
plt.scatter([1,2,3],[2,2,2],color='b')  
plt.show()
```





`np.loadtxt(filename, delimiter, dtype, skiprows)`

Example

```
import numpy as np  
data= np.loadtxt("../data.csv", delimiter=',', dtype=np.float32)  
# skiprows = n 인 경우, 위에서 n번째 줄까지 제외하고 불러온다.
```

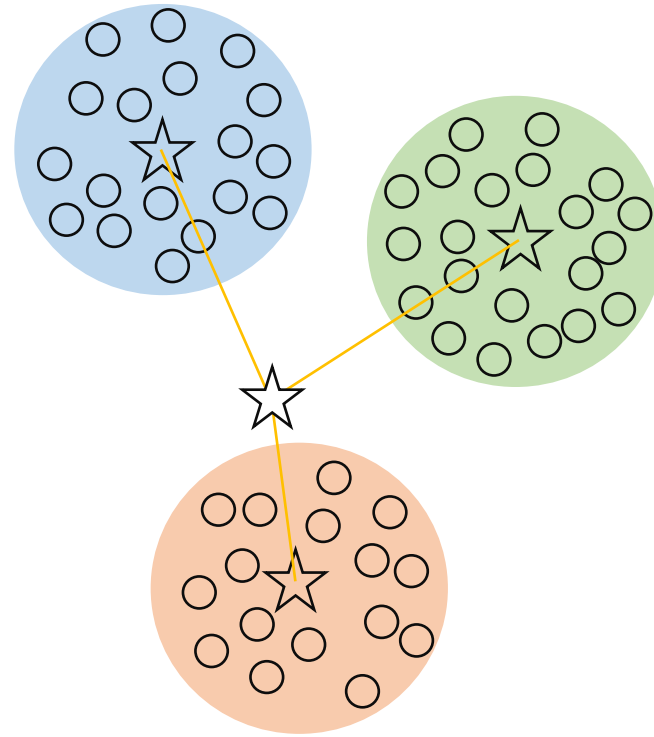
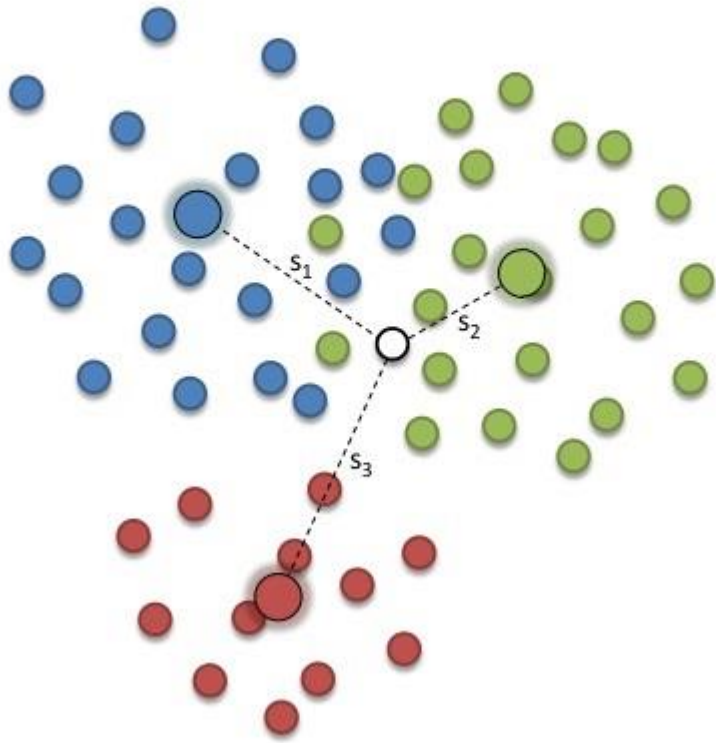
`np.savetxt(filename, X, delimiter)`

Example

```
import numpy as np  
a = np.array([[1,2,3],[4,5,6],[7,8,9]])  
np.savetxt("../data.csv", a, delimiter=',')
```

- UCI machine learning repository 사이트 (<https://archive.ics.uci.edu/ml/index.php>)에서 다운받은 Iris 데이터를 훈련집합과 테스트집합으로 나누어 저장한 파일을 사이버캠퍼스 자료실에서 다운 받아 다음을 수행하는 코드를 작성하시오.
 - Training set을 이용하여 각 클래스의 특징벡터들의 평균 벡터를 구한다.
 - Test set에 있는 각 데이터 샘플에 대해 다음의 방법으로 클래스를 예측한다.
 - 세 개의 클래스 중에서 가장 거리가 가까운 평균 벡터를 가지는 클래스에 속하는 것으로 예측한다.
 - Test set의 데이터들에 대한 예측 정확도(전체 중에 예측이 맞은 데이터의 비율)를 프린트한다.
- [이러한 classifier를 centroid-based classifier라고 합니다]

[centroid-based classifier]



[centroid-based classifier]

아래의 데이터는 어떤 꽃인가?

