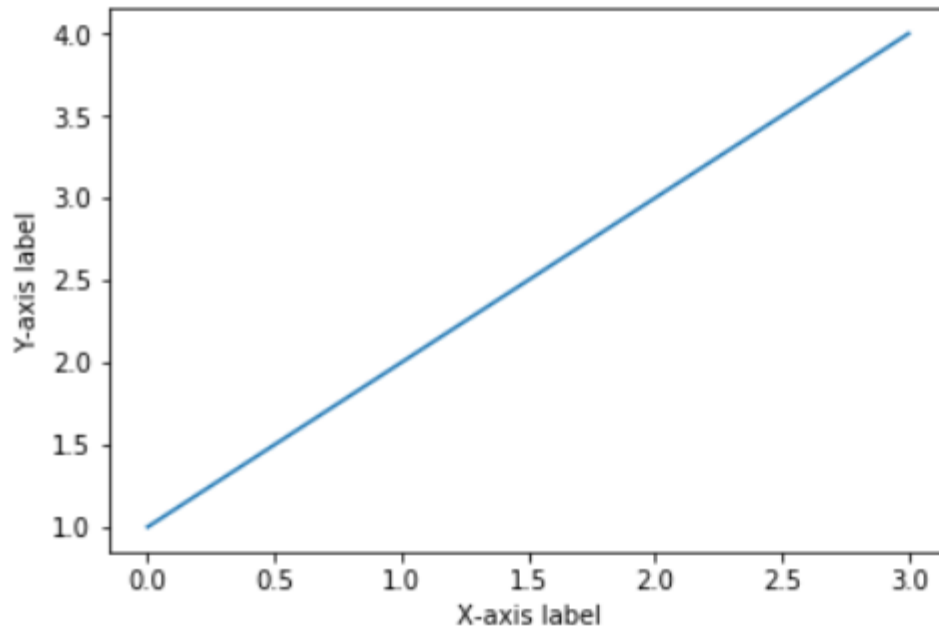

Python

matplotlib와 빈도분석

참고문헌 : 파이썬을 이용한 빅데이터 수집, 분석과 시각화 - 비팬북스, 이원화

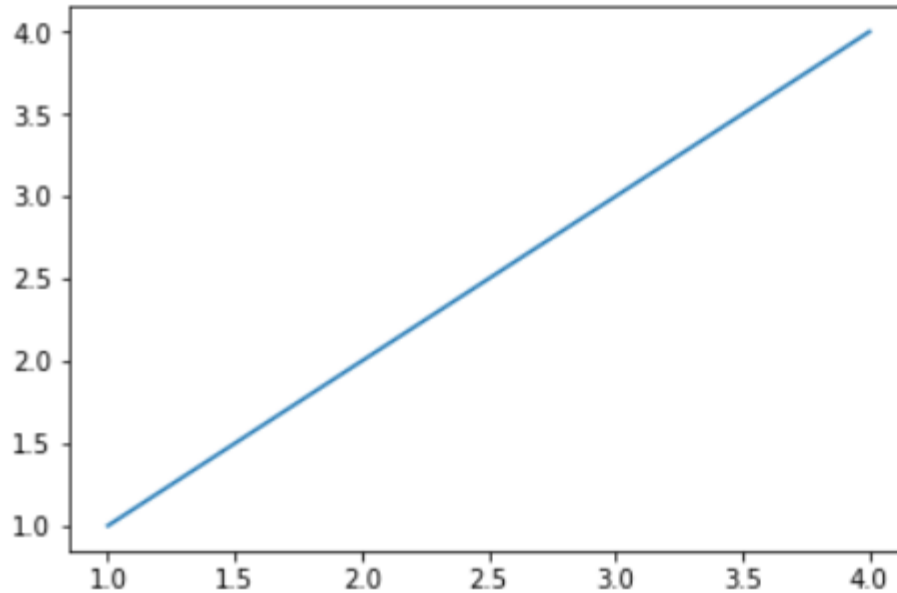
기본 그래프 그리기

```
In [2]: 1 import matplotlib.pyplot as plt
        2 plt.plot([1,2,3,4])
        3 plt.xlabel("X-axis label")
        4 plt.ylabel("Y-axis label")
        5 plt.show()
```



기본 그래프 그리기

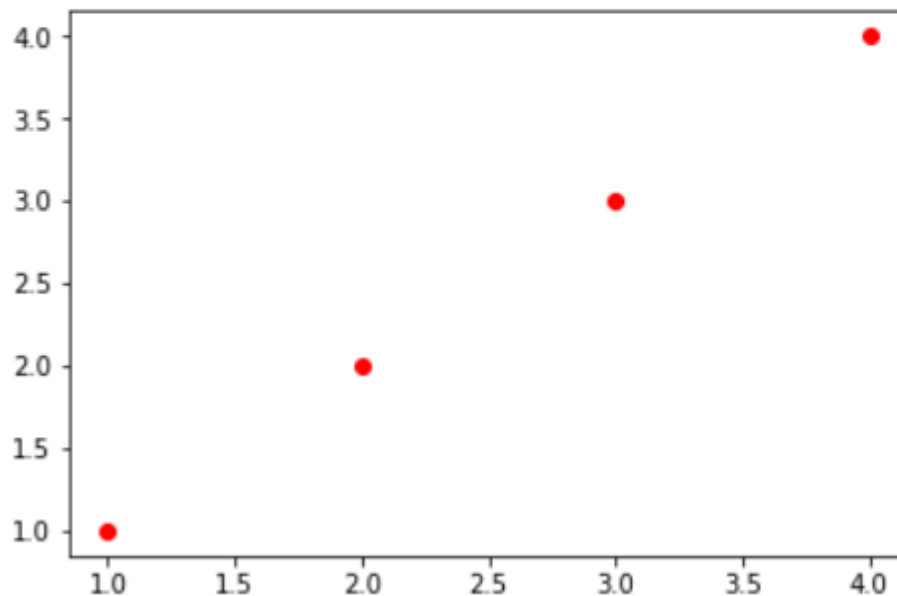
```
In [3]: 1 plt.plot([1,2,3,4],[1,2,3,4])  
        2 plt.show()
```



기본 그래프 그리기

In [6]:

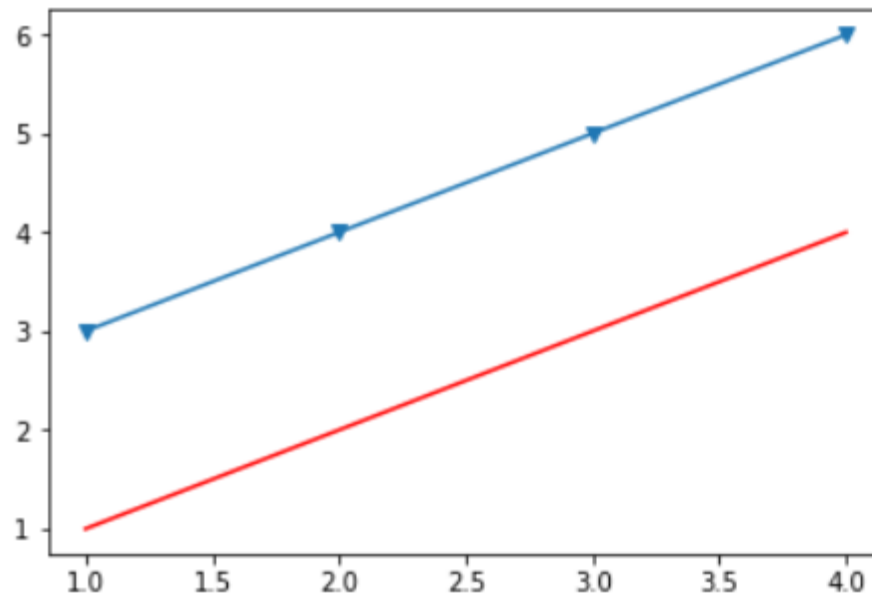
```
1 # 기본값 : 파란색(b) 라인(-)
2 # ro : 적색 o, bv : 파란색 v 마크, 그외 matplotlib 공식사이트에서 확인
3 plt.plot([1,2,3,4],[1,2,3,4],'ro')
4 plt.show()
```



다수의 그래프 그리기

```
In [7]: 1 # 다수의 그래프 그리기
        2 plt.plot([1,2,3,4],[1,2,3,4], 'r-', [1,2,3,4],[3,4,5,6], 'v-')
```

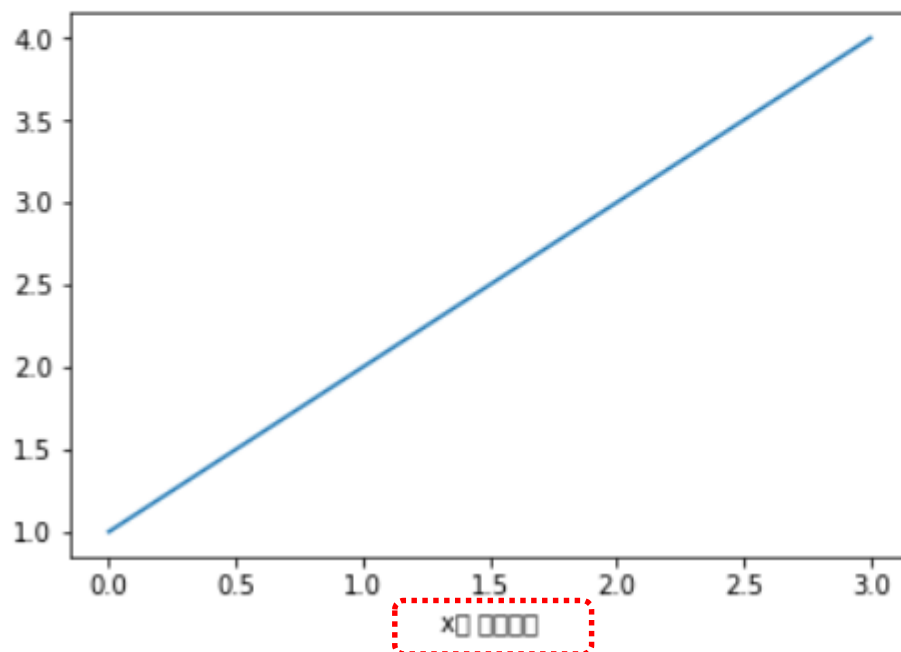
```
Out[7]: [<matplotlib.lines.Line2D at 0x22df8386828>,
        <matplotlib.lines.Line2D at 0x22df83869e8>]
```



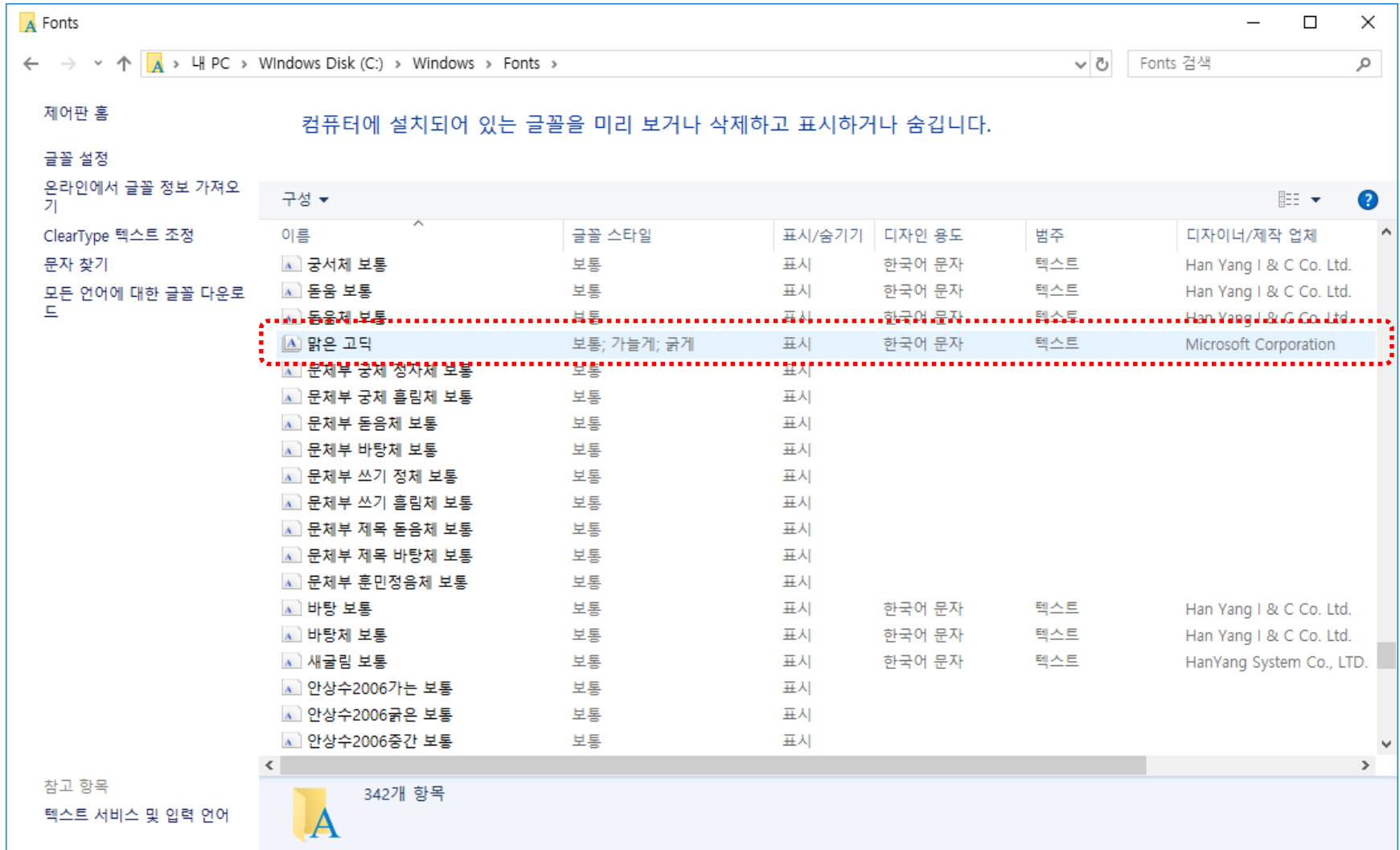
한글처리

```
In [8]: 1 plt.plot([1,2,3,4])  
        2 plt.xlabel('x축 한글표시')  
        3 plt.show
```

Out[8]: <function matplotlib.pyplot.show(*args, **kw)>



한글처리



한글처리

맑은 고딕

← → ↕ ↗

내 PC > Windows Disk (C:) > Windows > Fonts > 맑은 고딕

맑은 고딕 검색

제어판 홈

글꼴 설정

온라인에서 글꼴 정보 가져오기

ClearType 텍스트 조정

문자 찾기

모든 언어에 대한 글꼴 다운로드

글꼴 패밀리 세부 정보

이 글꼴 패밀리의 개별 글꼴을 미리 보기

구성 ▾ 미리 보기 삭제 숨기기

한글

한글

맑은 고딕 가늘게

맑은 고딕 굵게

참고 항목

텍스트 서비스 및 입력 언어

malgunbd.ttf 속성

일반

보안

자세히

이전 버전

malgunbd.ttf

파일 형식:

트루타입 글꼴 파일(.ttf)

연결 프로그램:

Windows 글꼴 뷰어

변경(C)...

위치:

C:\Windows\Fonts

크기:

12.0MB (12,598,360 바이트)

디스크 할당 크기:

12.0MB (12,599,296 바이트)

만든 날짜:

2018년 4월 12일 목요일, 오전 8:34:15

수정한 날짜:

2018년 4월 12일 목요일, 오전 8:34:15

액세스한 날짜:

2018년 4월 12일 목요일, 오전 8:34:15

특성:

☐ 읽기 전용(R)
 ☐ 숨김(H)

고급(D)...

확인

취소

적용(A)

한글

맑은 고딕 굵게

글꼴 스타일: 굵게

표시/숨기기: 표시

디자인 용도: 한국어 문자

범주: 텍스트

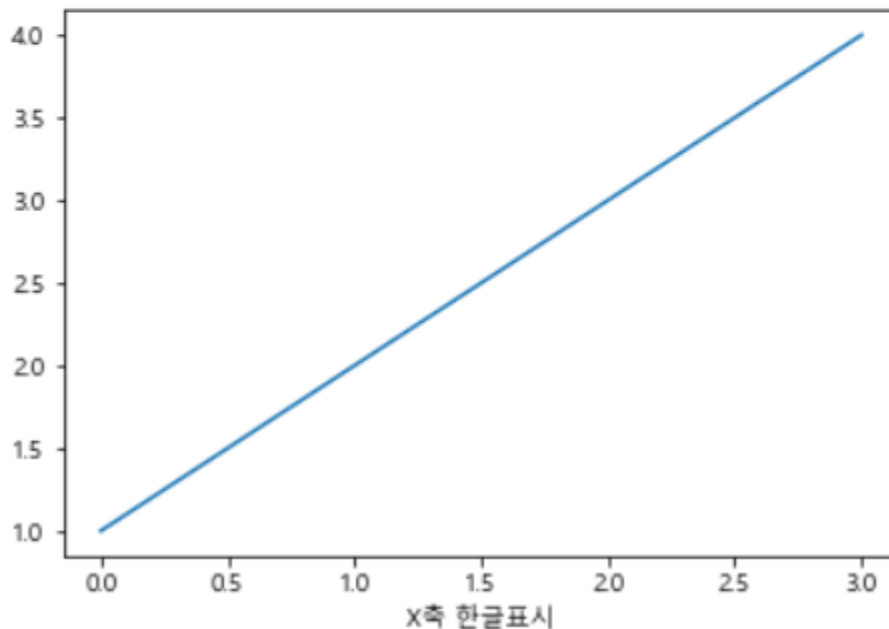
디자이너/제작 업체: Microsoft Corporation

글꼴 포함 가능성: 편집 가능

한글처리

In [10]:

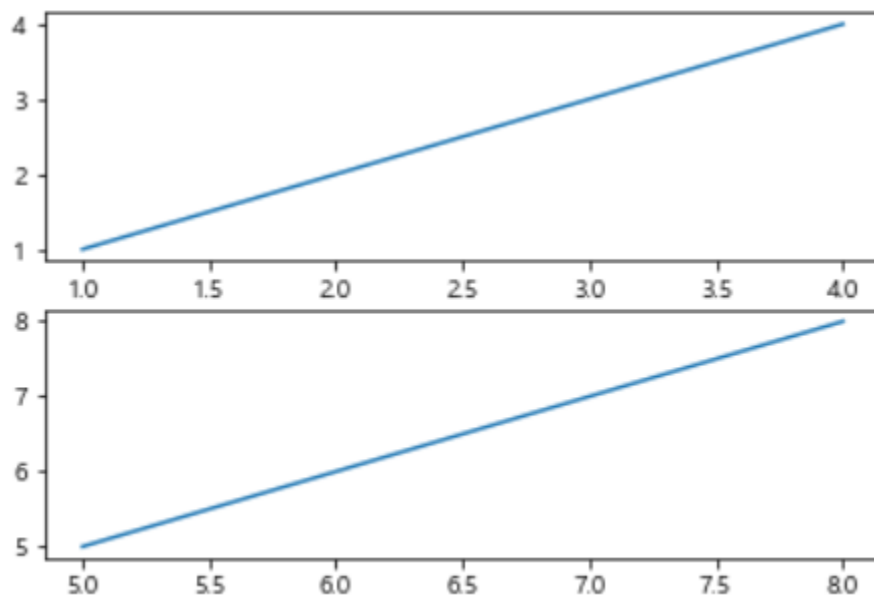
```
1 from matplotlib import font_manager, rc
2 import matplotlib
3 font_location="c:/Windows/fonts/malgunbd.ttf"
4 font_name = font_manager.FontProperties(fname=font_location).get_name()
5 matplotlib.rc('font', family=font_name)
6 plt.plot([1,2,3,4])
7 plt.xlabel("X축 한글표시")
8 plt.show()
```



여러 개의 그래프 그리기

In [15]:

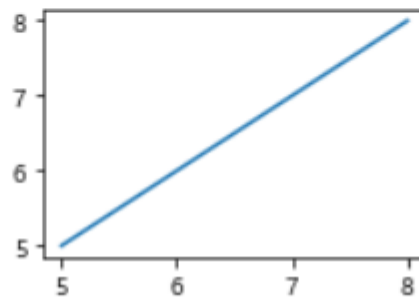
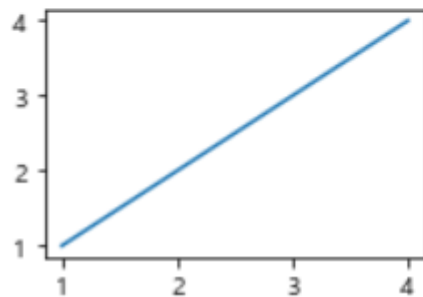
```
1 plt.figure() # 하나의 캔버스를 생성
2 # subplot(m,n,idx)
3 # 매트릭스 형태로 행2 열1개의 창을 의미. idx는 mn형태의 idx번째
4 plt.subplot(2,1,1)
5 plt.plot([1,2,3,4],[1,2,3,4])
6 plt.subplot(2,1,2)
7 plt.plot([5,6,7,8],[5,6,7,8])
8 plt.show()
```



여러 개의 그래프 그리기

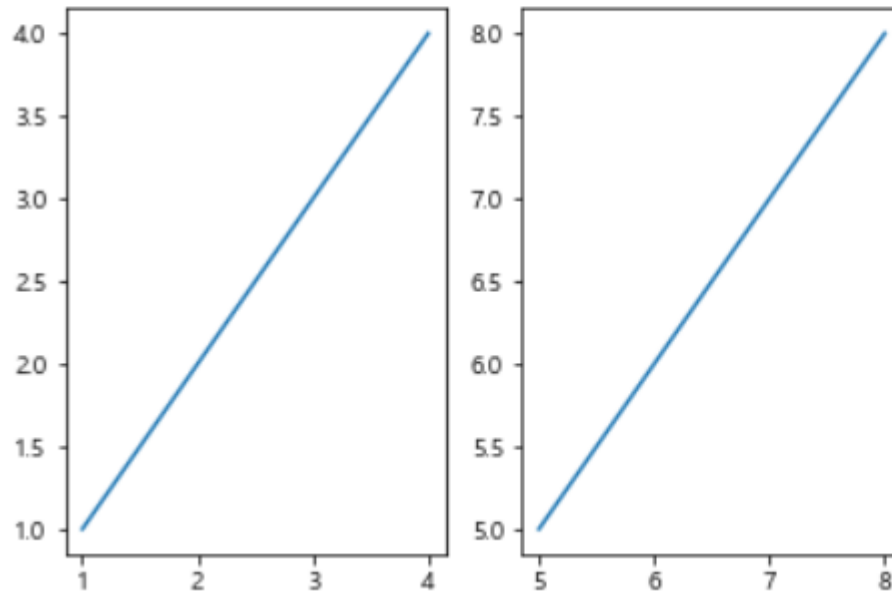
In [18]:

```
1 plt.figure()
2 plt.subplot(2,2,1)
3 plt.plot([1,2,3,4],[1,2,3,4])
4 plt.subplot(2,2,2)
5 plt.plot([5,6,7,8],[5,6,7,8])
6 plt.show()
```



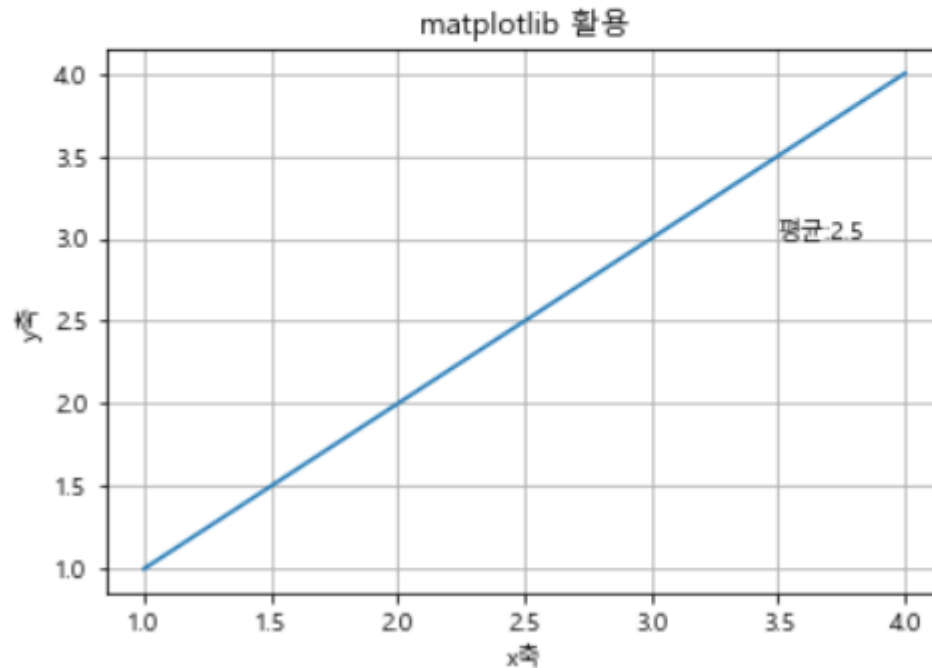
여러 개의 그래프 그리기

```
In [19]: 1 plt.figure()
          2 plt.subplot(1,2,1) # 1행의 첫 번째 컬럼
          3 plt.plot([1,2,3,4],[1,2,3,4])
          4 plt.subplot(1,2,2) # 1행의 두 번째 컬럼
          5 plt.plot([5,6,7,8],[5,6,7,8])
          6 plt.show()
```



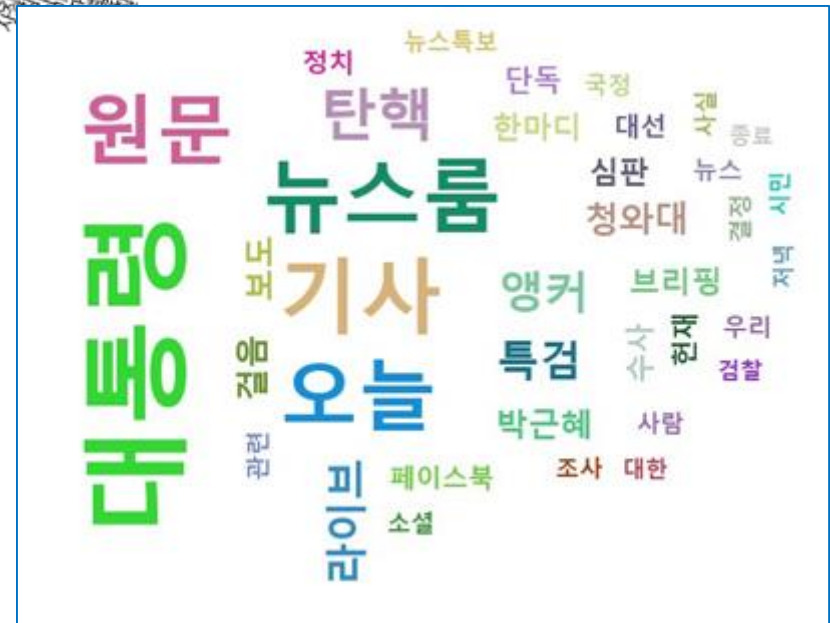
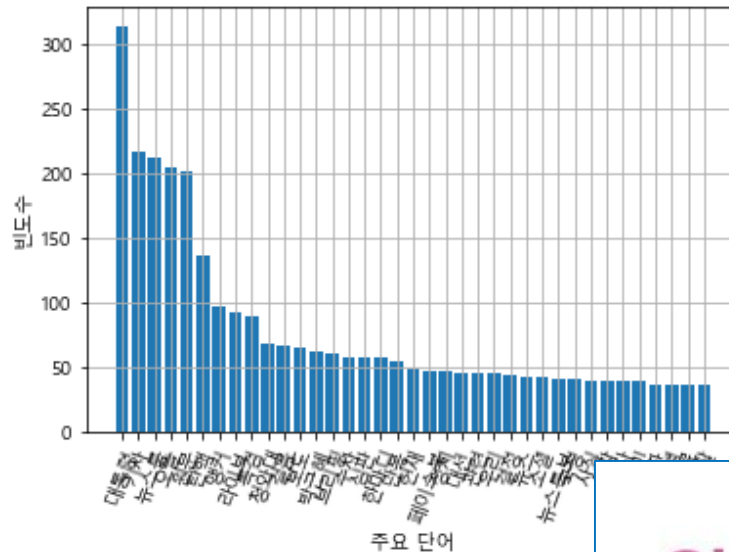
그래프에 문자 삽입

```
In [22]: 1 plt.plot([1,2,3,4],[1,2,3,4])
          2 plt.xlabel('x축')
          3 plt.ylabel('y축')
          4 plt.title('matplotlib 활용')
          5 plt.text(3.5, 3.0, '평균:2.5')
          6 plt.grid(True)
          7 plt.show()
```



명사 추출 및 빈도 분석

대통령 : 314
 기사 : 217
 뉴스룸 : 212
 오늘 : 204
 원문 : 202
 탄핵 : 137
 앵커 : 97
 라이브 : 92
 특검 : 90
 청와대 : 69
 결음 : 67
 보도 : 65
 박근혜 : 62
 브리핑 : 61
 수사 : 58
 심판 : 58
 한마디 : 57
 단독 : 54
 현재 : 49
 페이스북 : 47
 정치 : 47
 대선 : 46
 관련 : 45
 우리 : 45
 결정 : 44
 뉴스 : 43
 소셜 : 42
 뉴스특보 : 41
 국정 : 41
 사실 : 40
 사람 : 40
 조사 : 40
 시민 : 39



명사 추출 및 빈도 분석

- 페이스북의 JTBC뉴스와 조선일보데이터 분석
 - 기간 : 2016-10-01~2017-03-12
- 설치 패키지
 - `pip install JPyep1`
 - `pip install KoNLPy`
 - `pip install pytagcloud`
 - `pip install pygame`
 - `pip install simplejson`

명사 추출 및 빈도 분석

```
In [3]: 1 import json
        2 import re
        3
        4 from konlpy.tag import Okt
        5 from collections import Counter
        6
        7 import matplotlib.pyplot as plt
        8 import matplotlib
        9 from matplotlib import font_manager, rc
       10
       11 import pytagcloud
       12 import webbrowser
       13
       14 #[CODE 1]
       15 def showGraph(wordInfo):
       16
       17     font_location = "c:/Windows/fonts/malgunbd.ttf"
       18     font_name = font_manager.FontProperties(fname=font_location).get_name()
       19     matplotlib.rc('font', family=font_name)
       20
       21     plt.xlabel('주요 단어')
       22     plt.ylabel('빈도수')
       23     plt.grid(True)
       24     # 최대빈도수값과 최대빈도수 단어가 저장
       25     Sorted_Dict_Values = sorted(wordInfo.values(), reverse=True)
       26     Sorted_Dict_Keys = sorted(wordInfo, key=wordInfo.get, reverse=True)
       27     # 막대그래프 그리는 함수
       28     plt.bar(range(len(wordInfo)), Sorted_Dict_Values, align='center')
       29     # x축의 각 데이터 별 문자열(tick)을 지정
       30     plt.xticks(range(len(wordInfo)), list(Sorted_Dict_Keys), rotation='70')
       31
       32     plt.show()
```

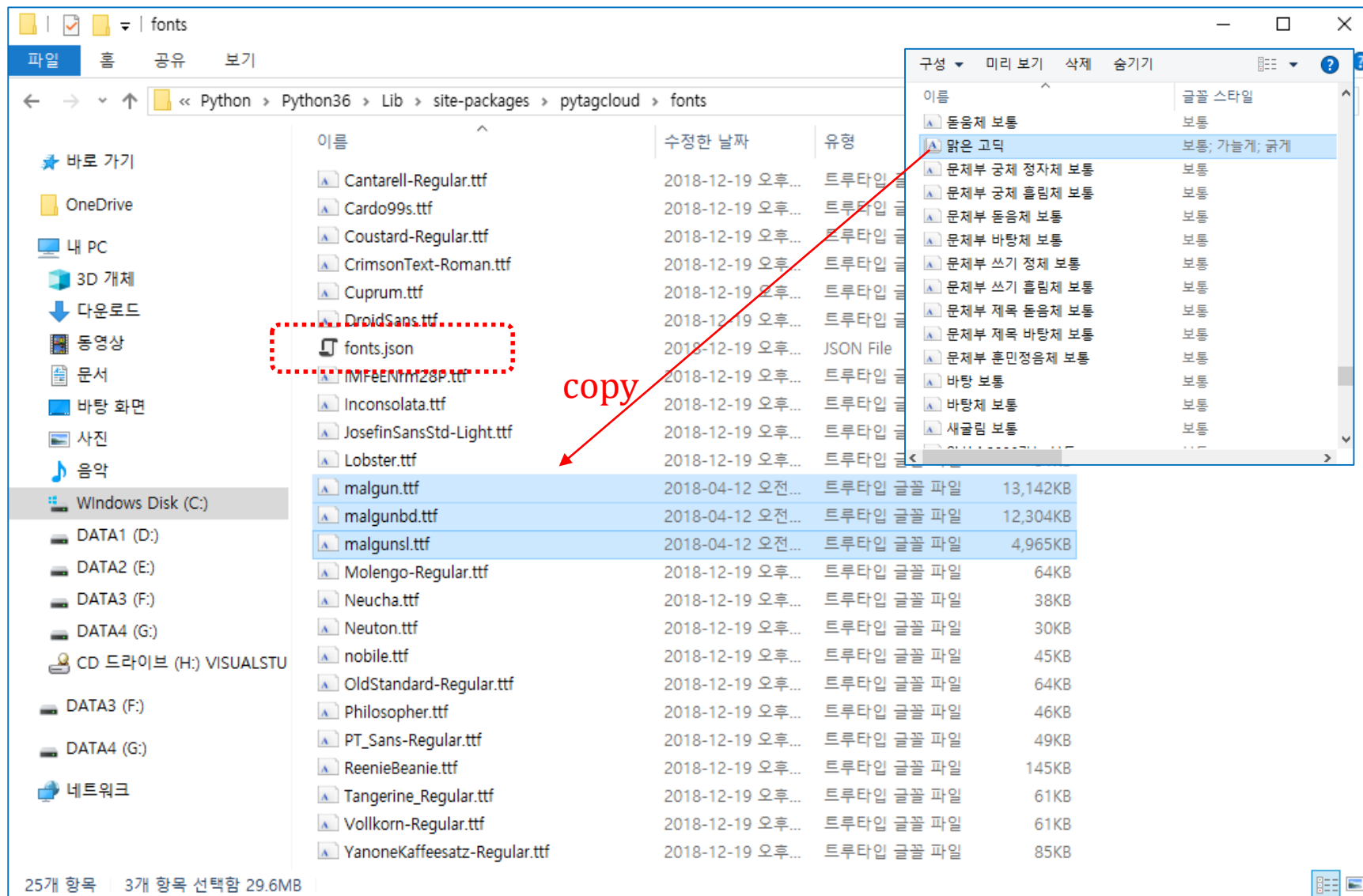

명사 추출 및 빈도 분석

```
In [4]: 1  #[CODE 2]
        2  def saveWordCloud(wordInfo, filename):
        3
        4      taglist = pytagcloud.make_tags(dict(wordInfo).items(), maxsize=80)
        5      pytagcloud.create_tag_image(taglist, filename, size=(640, 480), fontname='korean', rectangular=False)
        6      webbrowser.open(filename)
        7
```

명사 추출 및 빈도 분석

```
In [11]: 1 def main():
2
3     openFileName = 'G:\python_workspace\jupyter\BigData\jtbnews_facebook_2016-10-01_2017-03-12.json'
4     #openFileName = 'G:/python_workspace/jupyter/BigData/chosun_facebook_2016-10-01_2017-03-12.json'
5
6     cloudImagePath = openFileName + '.jpg'
7
8     rfile = open(openFileName, 'r', encoding='utf-8').read()
9
10    jsonData = json.loads(rfile)
11    message = ''
12
13    # jsonData의 개별 message를 합쳐서 하나의 문자열로 구성. 불필요한 \t, \n등의 문자 제거
14    for item in jsonData:
15        if 'message' in item.keys():
16            message = message + re.sub(r'[\t\n]', ' ', item['message']) + ' '
17
18    # 품사 클래스 - 명사만 추출하여 갯수를 세어 상위 50개만 가지고 온다.
19    nlp = Okt()
20    nouns = nlp.nouns(message)
21    count = Counter(nouns)
22
23    wordInfo = dict()
24    for tags, counts in count.most_common(50):
25        if (len(str(tags)) > 1):
26            wordInfo[tags] = counts
27            print("%s : %d" % (tags, counts))
28
29    showGraph(wordInfo)
30    saveWordCloud(wordInfo, cloudImagePath)
31
32    if __name__ == "__main__":
33        main()
```

한글폰트 지정



한글폰트 지정

```
*C:\Python\Python36\Lib\site-packages\pytagcloud\fonts\fonts.json - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?
fonts.json
1 [
2   {
3     "name": "korean",
4     "ttf": "malgunbd.ttf",
5     "web": "http://fonts.googleapis.com/css?family=Nobile"
6   },
7   {
8     "name": "Old Standard TT",
9     "ttf": "OldStandard-Regular.ttf",
10    "web": "http://fonts.googleapis.com/css?family=Old+Standard+TT"
11  },
12  {
13    "name": "Cantarell",
14    "ttf": "Cantarell-Regular.ttf",
15    "web": "http://fonts.googleapis.com/css?family=Cantarell"
16  },
17  {
18    "name": "Reenie Beanie",
19    "ttf": "ReenieBeanie.ttf",
20    "web": "http://fonts.googleapis.com/css?family=Reenie+Beanie"
21  },
22  {
23    "name": "Cuprum",
24    "ttf": "Cuprum.ttf",
25    "web": "http://fonts.googleapis.com/css?family=Cuprum"
26  },
27  {
28    "name": "Molengo",
29    "ttf": "Molengo-Regular.ttf",
```

JSON file | length : 2,589 | lines : 107 | Ln : 4 | Col : 1 | Sel : 0 | 0 | Unix (LF) | UTF-8 | INS