

데이터 과학 기반의 파이썬 빅데이터 분석

[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 처벌을 받을 수 있습니다.



데이터 과학 기반의 파이썬 빅데이터 분석

텍스트 빈도 분석

목차

- 01 [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기
- 02 [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

학습목표

- 텍스트 분석을 이해하고 기본 분석 기법인 단어 분석을 수행할 수 있다.
- 영어와 한글의 단어 분석을 수행하여 히스토그램과 워드클라우드 시각화를 수행할 수 있다.

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 분석 미리보기

한글 뉴스 기사의 키워드 분석하기	
목표	'4차 산업혁명'에 관한 한글 기사에서 명사 키워드를 분석한다.
핵심 개념	형태소 분석, 품사 태깅
데이터 수집	4차 산업혁명 기사: 페이스북 전자신문 페이지에서 크롤링하여 저장한 json 파일(예제소스로 제공)
데이터 준비	1. 데이터 추출: json 파일에서 message 항목만 추출, key() 2. 명사 단어 추출: Okt 품사 태깅 패키지로 명사 추출, from konlpy.tag import Okt
데이터 탐색 및 모델링	1. 단어 빈도 탐색 • Counter() 2. 단어 빈도 히스토그램 • font_manager.FontProperties() • matplotlib.pyplot
결과 시각화	
1. 단어 빈도에 대한 히스토그램	2. 단어 빈도에 대한 워드클라우드
	

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 목표설정

- 목표: 'Big data'와 관련된 키워드를 도출하여 분석

■ 핵심 개념 이해

- ① 영문 데이터에서 분석할 특징을 선정
- ② 컴퓨터가 처리할 수 있는 벡터 형태로 변환
- ③ 분석 기법을 적용하여 필요한 정보를 추출

■ 실습 도구

- 아나콘다의 주피터

■ 텍스트 분석

- 자연어 처리와 데이터 마이닝이 결합하여 발전된 분야로 비정형 텍스트 데이터에서 정보를 추출하여 분석하는 방법
- 단어에 대한 분석을 기본으로 함
- 텍스트 분류, 텍스트 군집화, 감성 분석 등

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 핵심 개념 이해

■ 전처리

- 분석 작업의 정확도를 높이기 위해 분석에 사용할 데이터를 먼저 정리하고 변환하는 작업

표 8-1 전처리에서 수행하는 작업

종류	설명
정제 ^{cleaning}	불필요한 기호나 문자를 제거하는 작업으로 주로 정규식을 이용하여 수행한다.
정규화 ^{normalization}	정제와 같은 의미지만 형태가 다른 단어를 하나의 형태로 통합하는 작업으로 대/소 문자 통합, 유사 의미의 단어 통합 등이 있다.
토큰화 ^{tokenization}	데이터를 토큰으로 정한 기본 단위로 분리하는 작업이다. 문장을 기준으로 분리하는 문장 토큰화, 단어를 기준으로 분리하는 단어 토큰화 등이 있다.
불용어 ^{stopword} 제거	의미가 있는 토큰을 선별하기 위해 조사, 관사, 접미사처럼 분석할 의미가 없는 토큰인 불용어 ^{stopword} 를 제거한다.
어간 추출 ^{stemming}	단어에서 시제, 단/복수, 진행형 등을 나타내는 다양한 어간 ^{stem} 을 잘라내어 단어의 형태를 일반화한다.
표제어 추출 ^{lemmatization}	단어에서 시제, 단/복수, 진행형 등을 나타내는 다양한 표제어 ^{lemma} 를 추출하여 단어의 형태를 일반화한다. 품사를 지정하여 표제어를 추출하는 것이 가능하다.

표 8-2 어간 추출과 표제어 추출의 예

사용 단어	어간 추출	표제어 추출
am	am	be
the going	the go	the going
having	hav	have

■ 워드클라우드

- 텍스트 분석에서 많이 사용하는 시각화 기법
- 문서의 핵심 단어를 시각적으로 돋보이게 만들어 키워드를 직관적으로 알 수 있게 하는 것
- 출현 빈도가 높을수록 단어를 크게 나타냄
- 방대한 양의 텍스트 정보를 다루는 빅데이터 분석에서 주요 단어를 시각화하기 위해 사용

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

1. 데이터 검색하기

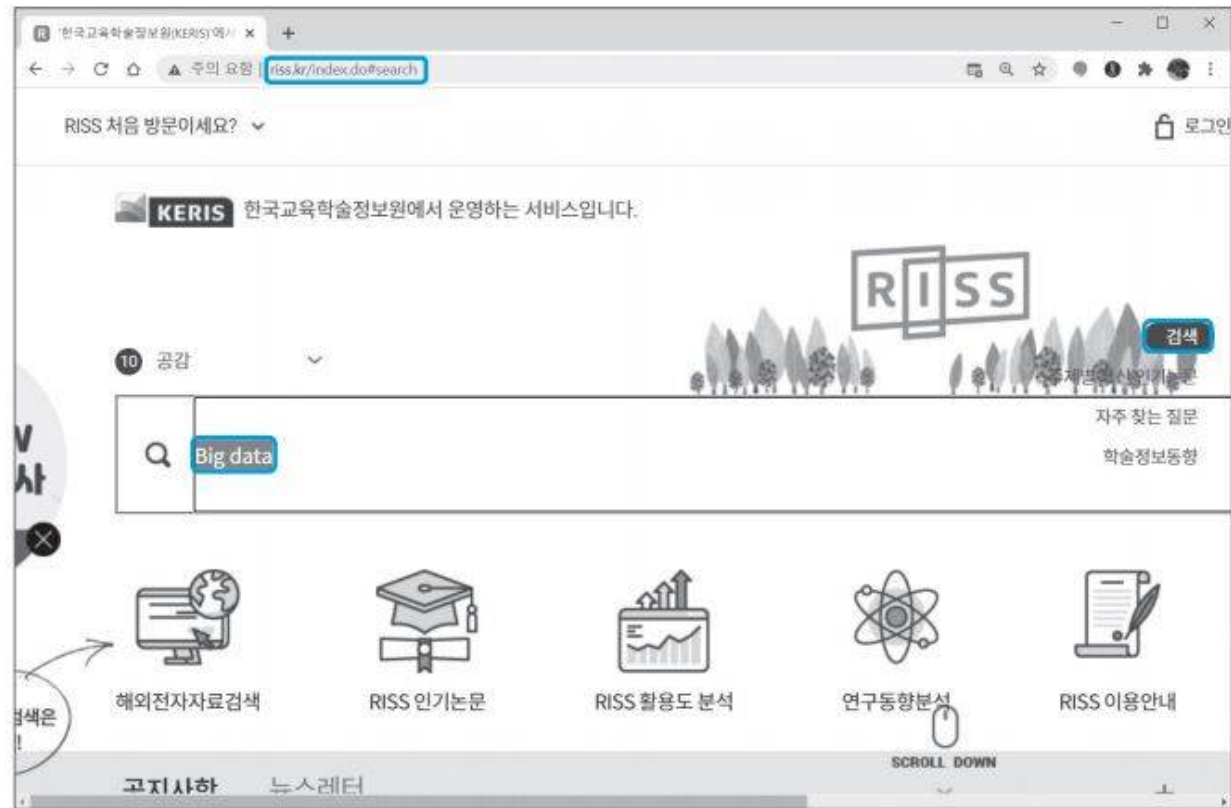


그림 8-1 한국교육학술정보원(KERIS)의 학술정보서비스(RISS) 사이트: www.riss.kr

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

2. 통합검색 결과 페이지에서 [해외학술논문] 메뉴를 클릭



그림 8-2 검색 결과 페이지: 통합검색

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

3. '작성언어'를 [영어]로 선택하고 아래의 버튼을 클릭

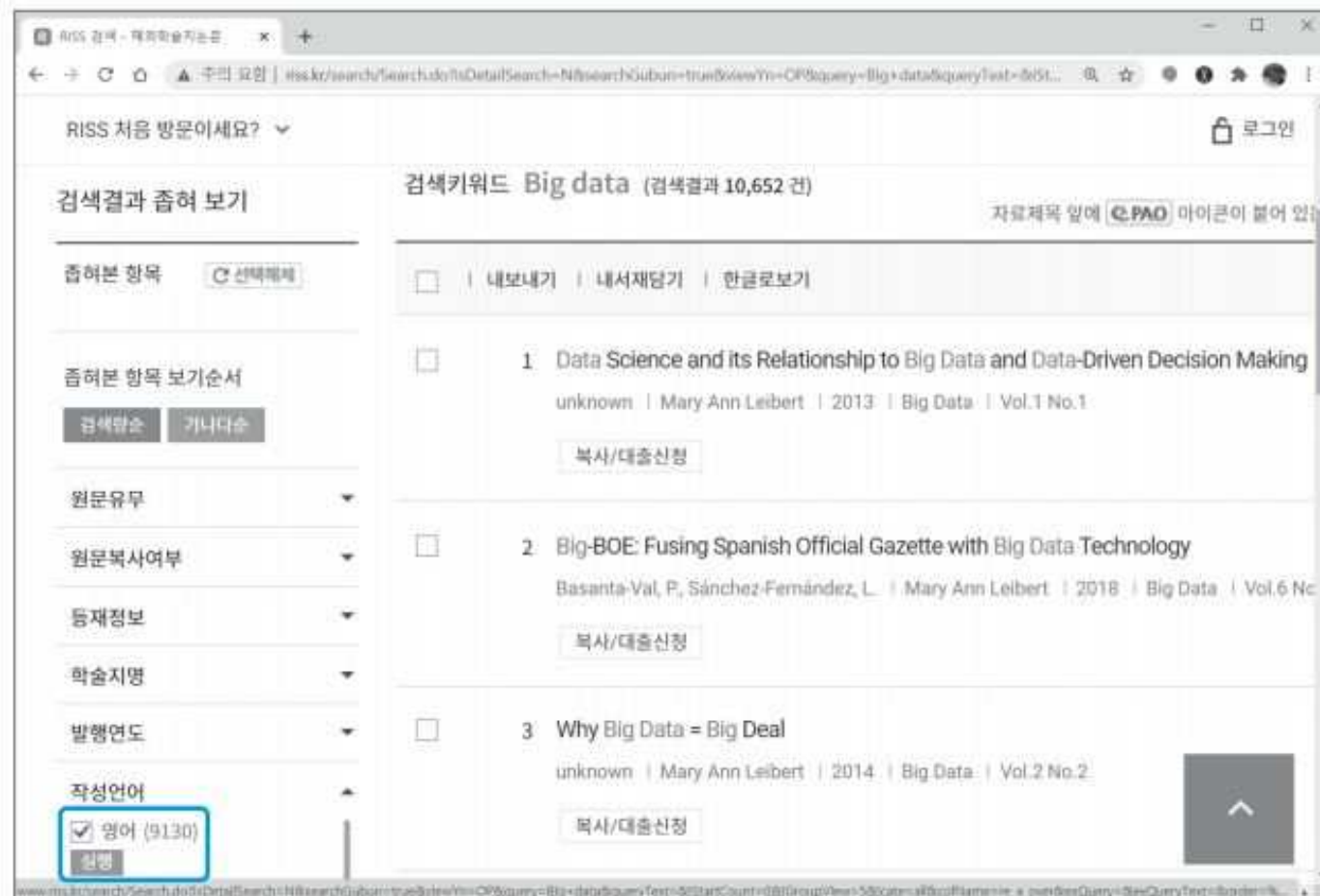


그림 8-3 검색 결과 페이지: 해외학술논문

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

4. 검색 결과 출력 개수 변경하기

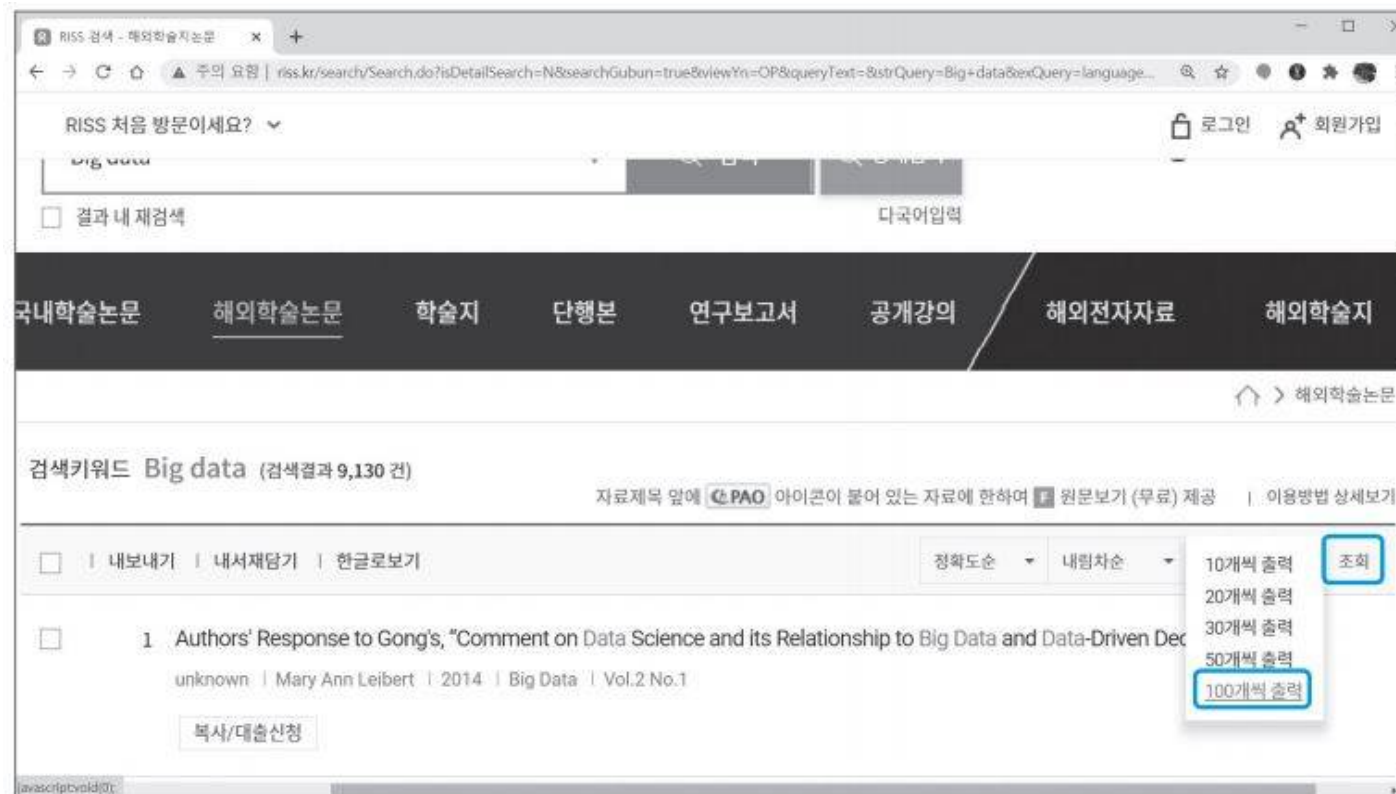


그림 8-4 검색 결과의 출력 개수 변경하기

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

5. 검색 결과 출력 개수 변경하기

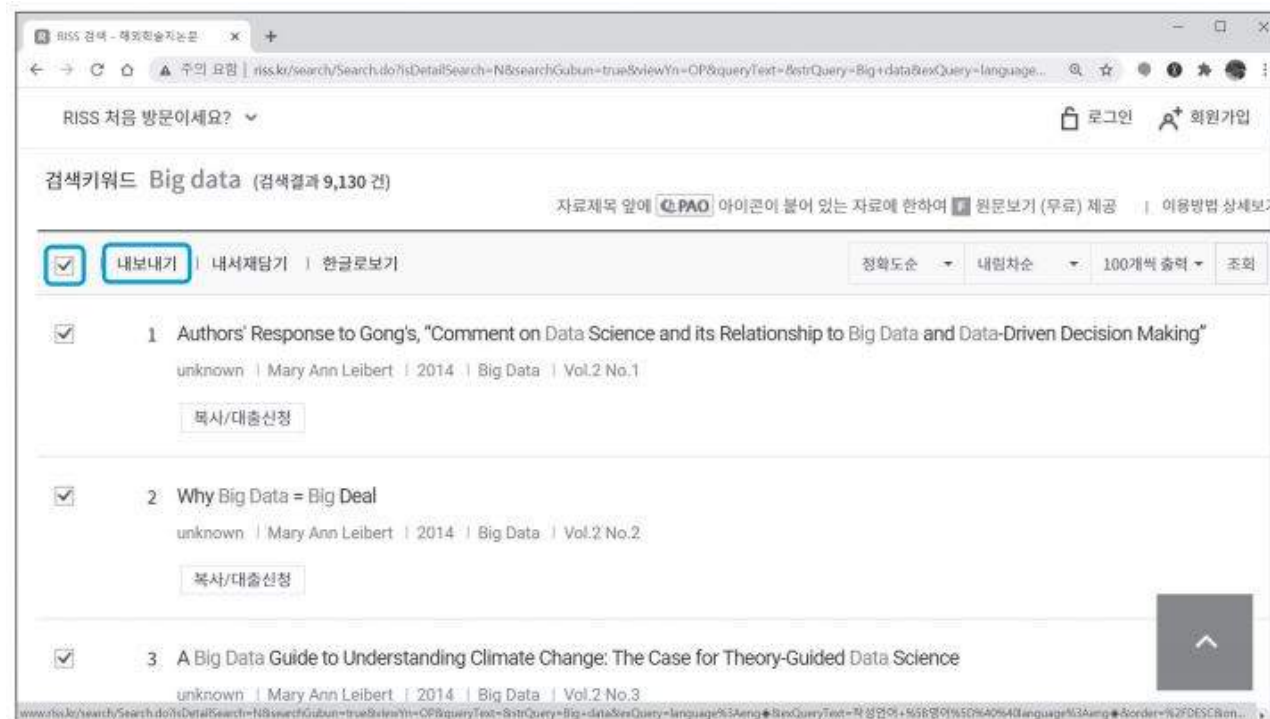


그림 8-5 현재 페이지 100개를 저장하기 위해 [내보내기] 메뉴 선택

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

6. [Excel저장]을 선택 → 버튼을 클릭한 뒤 파일을 저장

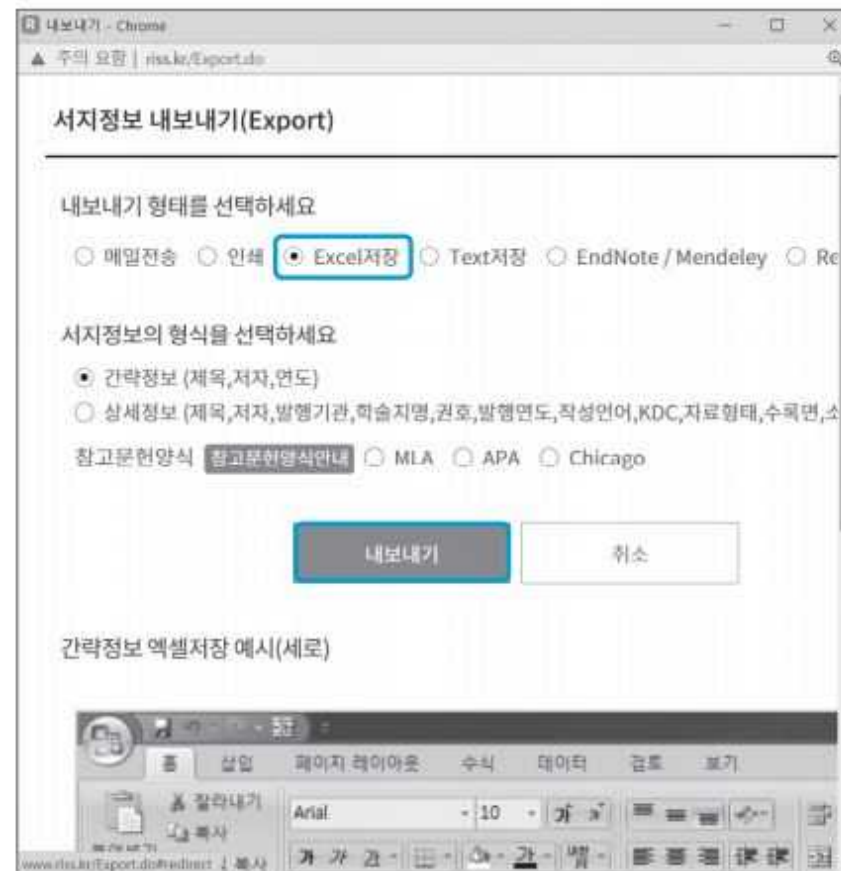


그림 8-6 내보내기 설정하기

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

7. 다음 페이지로 이동하여 이전 과정을 반복

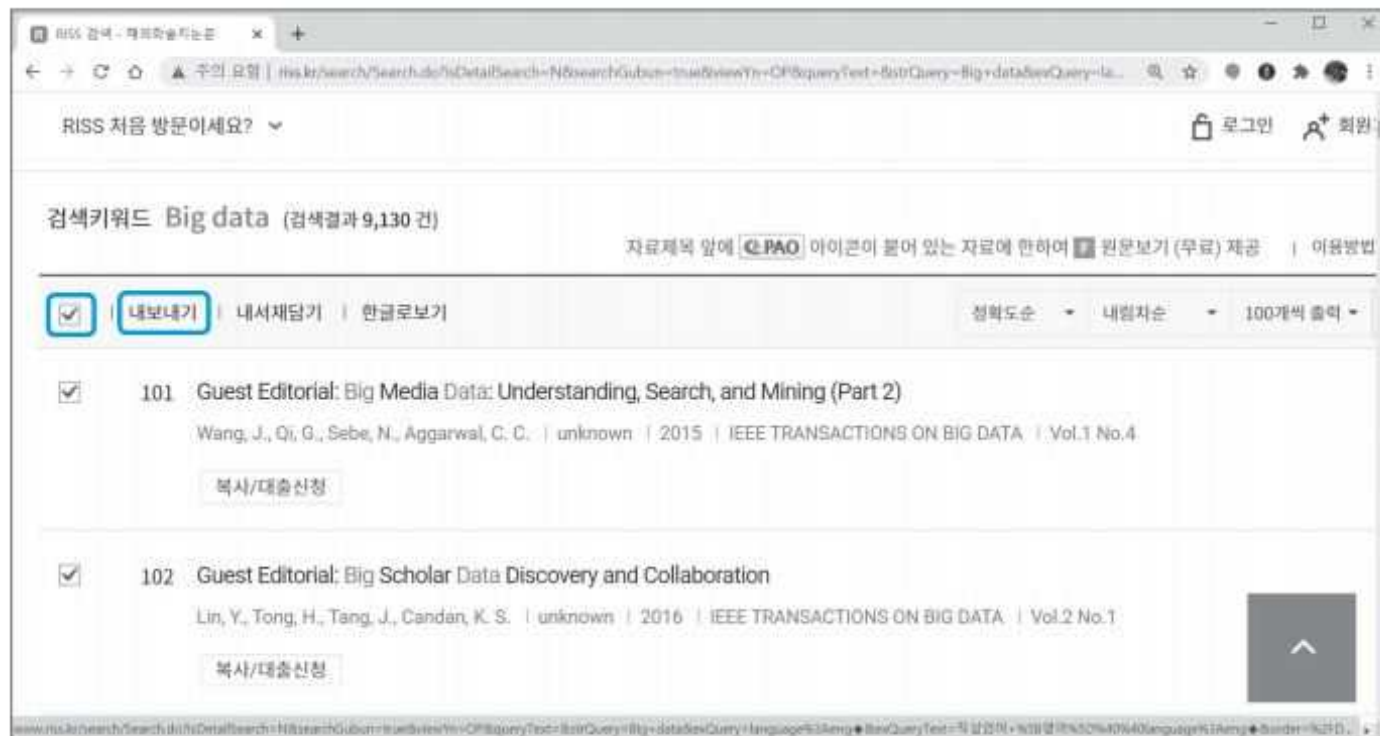


그림 8-7 다음 페이지에서 100개 데이터를 엑셀 파일로 저장하기 위해 [내보내기] 메뉴 선택

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 수집

8. 다운로드한 폴더를 확인

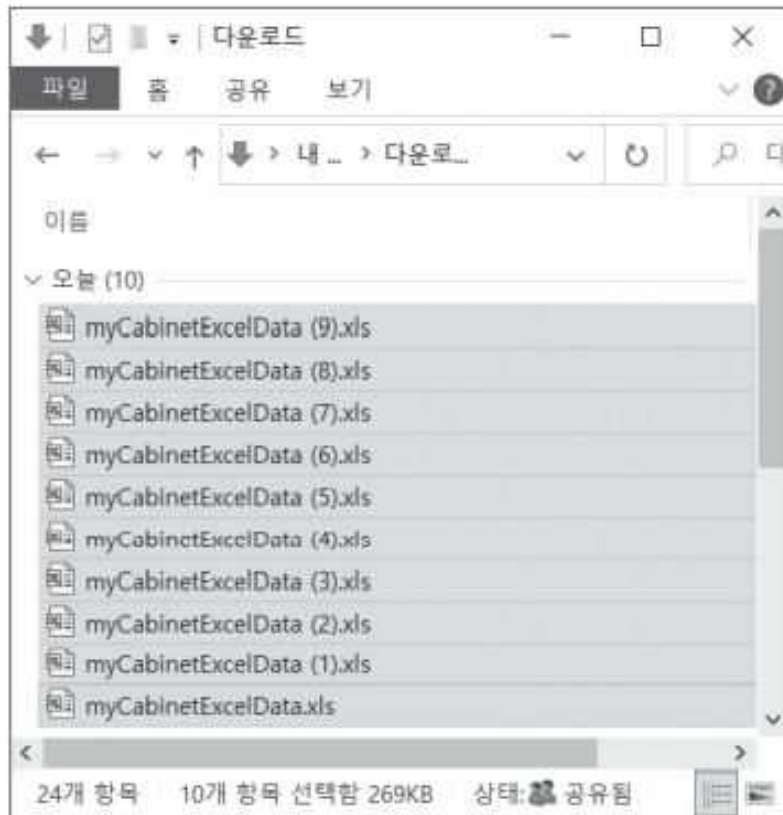


그림 8-8 엑셀 파일로 저장된 검색 결과

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

- 시작 전 저장한 엑셀 파일을 열어 데이터를 확인



번호	제목	저자	출판사	출판년
1	1 Authors' Response to Gong's, "Comment on Data Science and	unknown	Mary Ann Leibert	2014
2	2 Why Big Data = Big Deal	unknown	Mary Ann Leibert	2014
3	3 A Big Data Guide to Understanding Climate Change: The Case	unknown	Mary Ann Leibert	2014
4	4 Correction to: Predictive Modeling with Big Data: Is Bigger Real	unknown	Mary Ann Leibert	2019
5	5 Big-BOE: Fusing Spanish Official Gazette with Big Data Techno	Basanta-V	Mary Ann Leibert	2018
6	6 Data Science and its Relationship to Big Data and Data-Driven	unknown	Mary Ann Leibert	2013
7	7 Thermal-Aware and DVFS-Enabled Big Data Task Scheduling	fc Liu, H.; Li	unknown	2018
8	8 A reference architecture for big data solutions - introducing a	miGeerdink, I	nderscience	2015
9	9 Resting-State fMRI Functional Connectivity: Big Data Preproce	Phinyomai	unknown	2017
10	10 Sports Analytics in the Era of Big Data: Moving Toward the Nex	Assunção, Marv	Ann Leibert	2018

그림 8-9 저장한 파일 확인_myCabinetExcelData.xls

1. 작업 준비하기

- My_Python 폴더에 8장_data 폴더를 생성한 뒤 다운로드한 myCabinetExcelData 파일 10개를 이동
- 아나콘다의 주피터 노트북을 실행하고 My_Python 폴더로 이동한 뒤 [New]-[Python 3]을 선택해 파일을 새로 생성

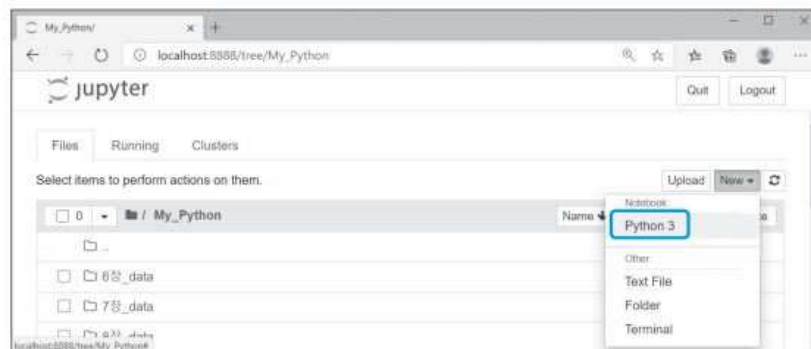


그림 8-10 아나콘다의 주피터 노트북에 새 notebook(Python 3) 추가하기

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

2. 작업 준비하기

- 'Untitled'를 클릭하고 '8장_영어단어분석'을 입력한 뒤 버튼을 클릭하여 파이썬 파일 이름을 변경

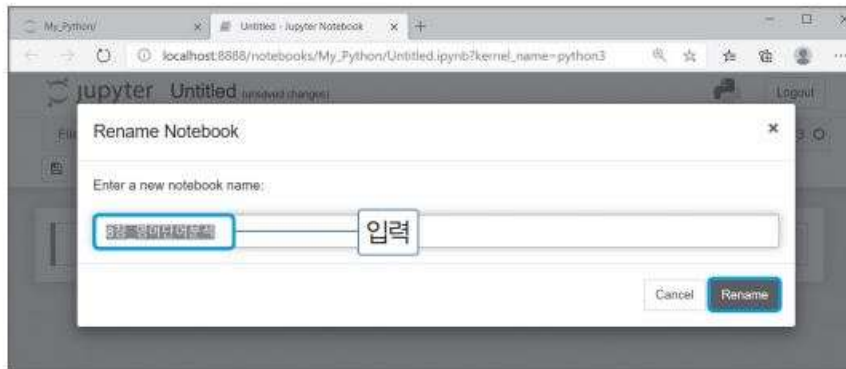


그림 8-11 새로 추가된 노트북 페이지의 이름 변경하기

3. 패키지 설치하기

- matplotlib와 wordcloud를 설치

In []:	!pip install matplotlib
In []:	!pip install matplotlib



01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

4. 프로젝트에 필요한 파이썬 패키지를 импорт

```
In [1]: import pandas as pd
import glob
import re
from functools import reduce
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from collections import Counter
import matplotlib.pyplot as plt
from wordcloud import STOPWORDS, WordCloud
```

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

4. 프로젝트에 필요한 파이썬 패키지를 импорт

표 8-3 파이썬 패키지의 용도

pandas	엑셀, CSV 등의 파일을 읽어서 데이터프레임에 저장하고 작업한 데이터를 데이터 프레임 형태로 구성하여 엑셀, CSV 등의 파일에 저장하는 작업을 수행하는 모듈(이후 pd라는 이름을 사용)
glob	경로와 이름을 지정하여 파일을 다루는 파일 처리 작업을 위한 모듈
re	메타 문자를 이용하여 특정 규칙을 작성하는 정규식을 사용하기 위한 모듈
reduce	2차원 리스트를 1차원 리스트로 차원을 줄이기 위한 모듈
word_tokenize	자연어 처리 패키지(from nltk.tokenize) 중에서 단어 토큰화 작업을 위한 모듈
stopwords	자연어 처리 패키지(from nltk.corpus) 중에서 불용어 정보를 제공하는 모듈
WordNetLemmatizer	자연어 처리 패키지(from nltk.stem) 중에서 단어 형태의 일반화를 위해 표제어 추출을 제공하는 모듈
Counter	데이터 집합에서 개수를 자동으로 계산하기 위한 모듈
matplotlib.pyplot	히스토그램을 그리기 위한 matplotlib 패키지의 내부 모듈(이후 plt라는 이름을 사용)
STOPWORDS, WordCloud	워드클라우드를 그리기 위해 사용할 워드클라우드용 불용어 모듈과 워드클라우드 모듈

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

5. 데이터 조합 – 파일 병합하기

- 병합할 엑셀 파일 이름 10개를 리스트에 저장

In [2]:	<pre>all_files = glob.glob('8장_data/myCabinetExcelData*.xls') all_files #출력하여 내용 확인</pre>
Out[2]:	<pre>['8장_dataWWmyCabinetExcelData (1).xls', '8장_dataWWmyCabinetExcelData (2).xls', '8장_dataWWmyCabinetExcelData (3).xls', '8장_dataWWmyCabinetExcelData (4).xls', '8장_dataWWmyCabinetExcelData (5).xls', '8장_dataWWmyCabinetExcelData (6).xls', '8장_dataWWmyCabinetExcelData (7).xls', '8장_dataWWmyCabinetExcelData (8).xls', '8장_dataWWmyCabinetExcelData (9).xls', '8장_dataWWmyCabinetExcelData.xls']</pre>

In [2]: 10개의 엑셀 파일 이름을 all_files 리스트에 저장

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

5. 데이터 조합 – 파일 병합하기

- 0개의 엑셀 파일 이름을 all_files 리스트에 저장
- 파일을 읽어서 하나의 데이터프레임으로 병합하고 CSV 파일에 저장

In [3]:

```
all_files_data = [] #저장할 리스트  
  
for file in all_files:  
    data_frame = pd.read_excel(file)  
    all_files_data.append(data_frame)  
  
all_files_data[0] #작업 내용 확인
```

Out[3]:

Unnamed: 0	번호	제목	저자	출판사	출판 일	
0	NaN	1	Guest Editorial: Big Media Data: Understanding...	Wang, J.; Qi, G.; Sebe, N.; Aggarwal, C. C.	unknown	2015
1	NaN	2	Guest Editorial: Big Scholar Data Discovery an...	Lin, Y.; Tong, H.; Tang, J.; Candan, K. S.	unknown	2016
2	NaN	3	Guest Editorial: Big Data Analytics and the Web	Sheng, M.; Vasilakos, A. V.; Yu, Q.; You, L.	unknown	2016
3	NaN	4	Parallel computing for preserving privacy usin...	Yaji, Sharath; Neelima, B.	Inderscience	2018
4	NaN	5	NoSQL databases for big data	Oussous, Ahmed; Benjeloun, Fatima-Zahra; Lahc...	Inderscience	2017
96	NaN	97	A watchdog approach - name-matching algorithm ...	Kirubakaran, Anusuya; Aramudhan, M.	Inderscience	2018
97	NaN	98	An intelligent approach to Big Data analytics ...	Verma, Neha; Singh, Jatinder	Emerald Group Publishing Limited	2017
98	NaN	99	How organisations leverage Big Data: a maturit...	Comuzzi, Marco; Patel, Anit	Emerald Group Publishing Limited	2016
99	NaN	100	Effective and efficient distributed management...	Cuzzocrea, Alfredo; Grasso, Giorgio Mario; Nol...	Inderscience	2019

100 rows × 6 columns

100 rows × 6 columns

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

5. 데이터 조합 – 파일 병합하기

In [4]:

all_files_data_concat = pd.concat(all_files_data, axis = 0, ignore_index = True)
all_files_data_concat #출력하여 내용 확인

Out[4]:

Unnamed: 0 번호			제목	저자	출판사	출판일
0	NaN	1	Guest Editorial: Big Media Data: Understanding...	Wang, J.; Qi, G.; Sebe, N.; Aggarwal, C. C.	unknown	2015
1	NaN	2	Guest Editorial: Big Scholar Data Discovery an...	Lin, Y.; Tong, H.; Tang, J.; Candan, K. S.	unknown	2016
2	NaN	3	Guest Editorial: Big Data Analytics and the Web	Sheng, M.; Vasilakos, A. V.; Yu, Q.; You, L.	unknown	2016
3	NaN	4	Parallel computing for preserving privacy usin...	Yaji, Sharath; Neelima, B.	Inderscience	2018
4	NaN	5	SQL database for big data	Oussous, Ahmed; Benjelloun, Fatima-Zahra; Lahc...	Inderscience	2017
994	NaN	95	Guest Editorial: Big Data Infrastructure I	Chen, J.; Wang, R.	unknown	2018
995	NaN	96	Guest Editorial: Big Media Data: Understanding...	Wang, J.; Qi, G.; Sebe, N.; Aggarwal, C.	unknown	2017
996	NaN	97	Guest Editorial: Big Scholar Data Discovery an...	Lin, Y.; Tong, H.; Tang, J.; Candan, K. S.	unknown	2017
997	NaN	98	Guest Editorial: Big Media Data: Understanding...	Wang, J.; Qi, G.; Sebe, N.; Aggarwal, C. C.	unknown	2016
998	NaN	99	Speed Up Big Data Analytics by Unveiling the S...	Wang, J.; Zhang, X.; Yin, J.; Wang, R.; Wu, H...	unknown	2018
999	NaN	100	Architecting Time-Critical Big-Data Systems	Basanta-Val, P.; Audsley, N. C.; Wellings, A. ...	unknown	2016

1000 rows × 6 columns

In [5]:

all_files_data_concat.to_csv('8장_data/riss_bigdata.csv', encoding = 'utf-8', index = False)

In [3]: all_files 리스트에 있는 파일 이름을 이용해 엑셀 파일을 읽어오고 `pd.read_excel()`, 파일 내용을 all_files_data에 추가하는 `append()` 작업을 all_files 리스트의 원소 갯수만큼, 즉 10개 파일에 대해 반복

In [4]: all_files_data를 세로축을 기준으로 `axis=0` 병합하여 all_files_data_concat 리스트 에 저장

In [5]: all_files_data_concat을 CSV 파일로 저장 `to_csv`

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

6. 데이터 전처리

- 수집한 데이터에서 제목을 추출하여 전처리를 수행

In [6]:	<pre>all_title = all_files_data_concat['제목'] all_title #출력하여 내용 확인</pre>
Out[6]:	<pre>0 Guest Editorial: Big Media Data: Understanding... 1 Guest Editorial: Big Scholar Data Discovery an... 2 Guest Editorial: Big Data Analytics and the Web 3 Guest Editorial: Special Issue on Big Scholar ... 4 2016 Index IEEE Transactions on Big Data Vol. 2 ... 995 Architecting Time-Critical Big-Data Systems 996 Guest Editorial: Big Scholar Data Discovery an... 997 Guest Editorial: Big Data Infrastructure I 998 Guest Editorial: Big Media Data: Understanding... 999 Speed Up Big Data Analytics by Unveiling the S... Name: 제목, Length: 1000, dtype: object</pre>
In [7]:	<pre>stopWords = set(stopwords.words("english")) lemma = WordNetLemmatizer(</pre>

In [6]: all_files_data_concat의 컬럼 중에서 분석에 사용할 '제목' 컬럼만 추출해 all_title 에 저장

In [7]: 전처리 작업을 위해 nltk.corpus에서 제공하는 영어 불용어 stopwords.words("english")를 불러와서 저장
그 후, 표제어 추출 작업을 제공하는 WordNetLemmatizer 객체를 생성

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

6. 데이터 전처리

In [8]:	<pre>words = [] for title in all_title: EnWords = re.sub(r"^[a-zA-Z]+", " ", str(title)) EnWordsToken = word_tokenize(EnWords.lower()) EnWordsTokenStop = [w for w in EnWordsToken if w not in stopWords] EnWordsTokenStopLemma = [lemma.lemmatize(w) for w in EnWordsTokenStop] words.append(EnWordsTokenStopLemma)</pre>
In [9]:	<pre>print(words) #출력하여 내용 확인</pre>
Out[9]:	<pre>['guest', 'editorial', 'big', 'medium', 'data', 'understanding', 'search', 'mining'], ['guest', 'editorial', 'big', 'scholar', 'data', 'discovery', 'collaboration'], ..., ['speed', 'big', 'data', 'analytics', 'unveiling', 'storage', 'distribution', 'sub', 'datasets']]</pre>

In [8]: all_title의 제목에 대해 정규식으로 만든 규칙을 적용하여 알파벳 으로 시작하지 않는 단어 "`^[a-zA-Z]+`"는 공백으로 치환하여 `re.sub()` 제거하고, 소문자로 정규화 하고 `lower()`, 단어 토큰화 `word_tokenize()`를 함
그 후, 불용어 `stopWords`를 제거한 후에 표제어 추출 `lemmatize(w)`을 한다.

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 준비

7. 데이터 전처리

- 전처리가 끝난 words는 2차원 리스트이므로 reduce() 함수를 사용하여 1차원 리스트로 변환

In [10]:	<pre>words2 = list(reduce(lambda x, y: x+y, words)) print(words2) #출력하여 내용 확인</pre>
Out[10]:	<pre>[['guest', 'editorial', 'big', 'medium', 'data', 'understanding', 'search', 'mining'], ['guest', 'editorial', 'big', 'scholar', 'data', 'discovery', 'collaboration'], ..., ['speed', 'big', 'data', 'analytics', 'unveiling', 'storage', 'distribution', 'sub', 'datasets']]</pre>

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 탐색 및 분석 모델 구축

1. 데이터 탐색 - 단어 빈도 구하기

In [11]:	<pre>count = Counter(words2) count #출력하여 내용 확인</pre>
Out[11]:	<pre>Counter({'guest': 13, 'editorial': 17, 'big': 1409, 'medium': 11, ...})</pre>
In [12]:	<pre>word_count = dict() for tag, counts in count.most_common(50): if(len(str(tag))>1): word_count[tag] = counts print("%s : %d" % (tag, counts))</pre>
Out[12]:	<pre>data : 1637 big : 1409 analytics : 139 ... network : 18 process : 18</pre>

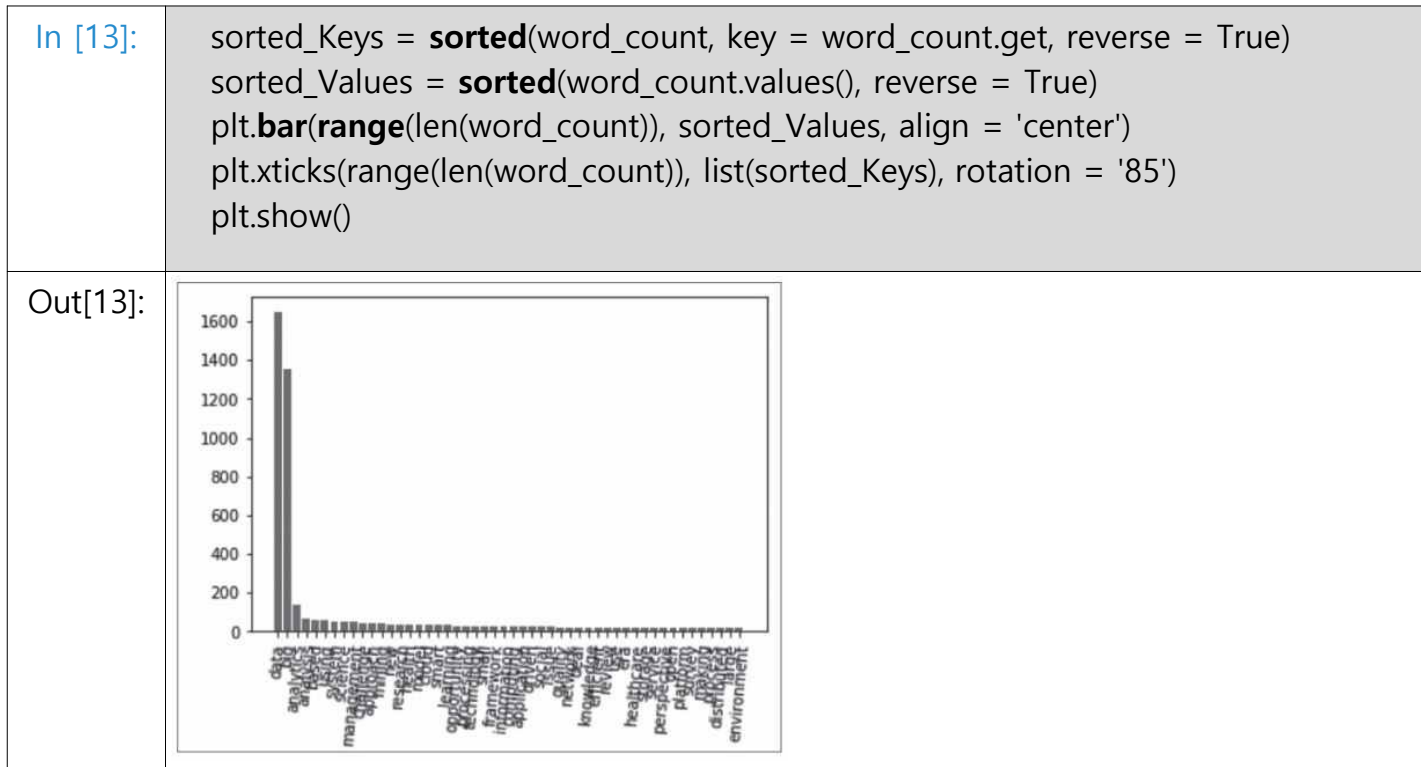
In [11] words2 리스트에 있는 단어별로 출현 횟수를 계산하여 딕셔너리 객체인 count를 생성 Counter()

In [12] 출현 횟수가 많은 상위 50개 단어count.most_common(50) 중에서 단어 길이가 1보다 큰 것만 word_count 딕셔너리에 저장한 후 출력

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 탐색 및 분석 모델 구축

2. 데이터 탐색 - 히스토그램 그리기



In [13]: 히스토그램을 그리기 위해 matplotlib.pyplot을 사용, 히스토그램의 크기 `figure()`를 지정하고
x축 레이블 `xlabel()`과 y축 레이블 `ylabel()`을 지정, 상위 50개만 저장한 word_count 딕셔너리에서
x축 값으로 사용할 sorted_Keys와 y축 값으로 사용할 sorted_Values를 역순으로 정렬하여 `reverse=True` 준비
x축 눈금 `plt.xticks`은 sorted_Keys 리스트의 값(상위 50개 단어)을 순서대로 사용
설정 사항을 적용하여 히스토그램을 그림 `plt.show()`.

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 데이터 탐색 및 분석 모델 구축

- 번외 - 특정 단어를 제거한 뒤 히스토그램 그리기
 - word_count 딕셔너리에서 'data'와 'big' 항목을 제거한 `del` 후 `ln [13]`을 실행

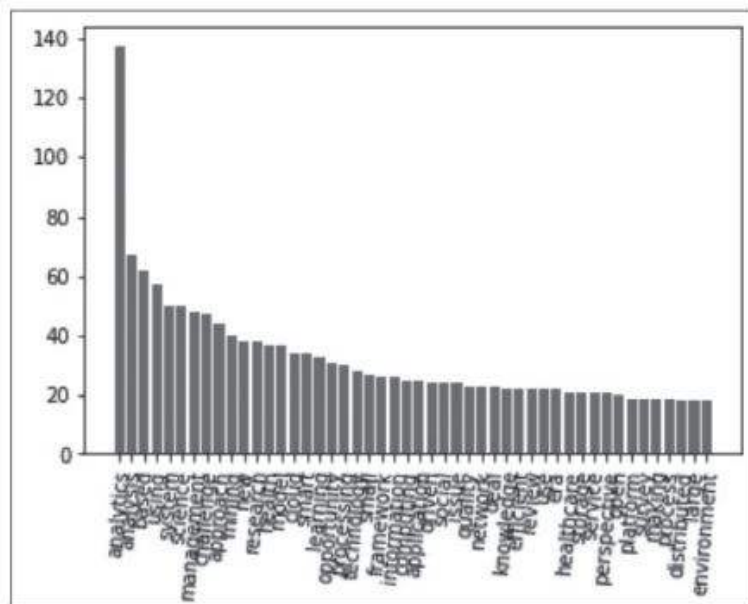


그림 8-12 검색어 'big'과 'data'를 제거한 후 생성한 히스토그램

01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

■ 결과 시각화

1. 그래프 그리기

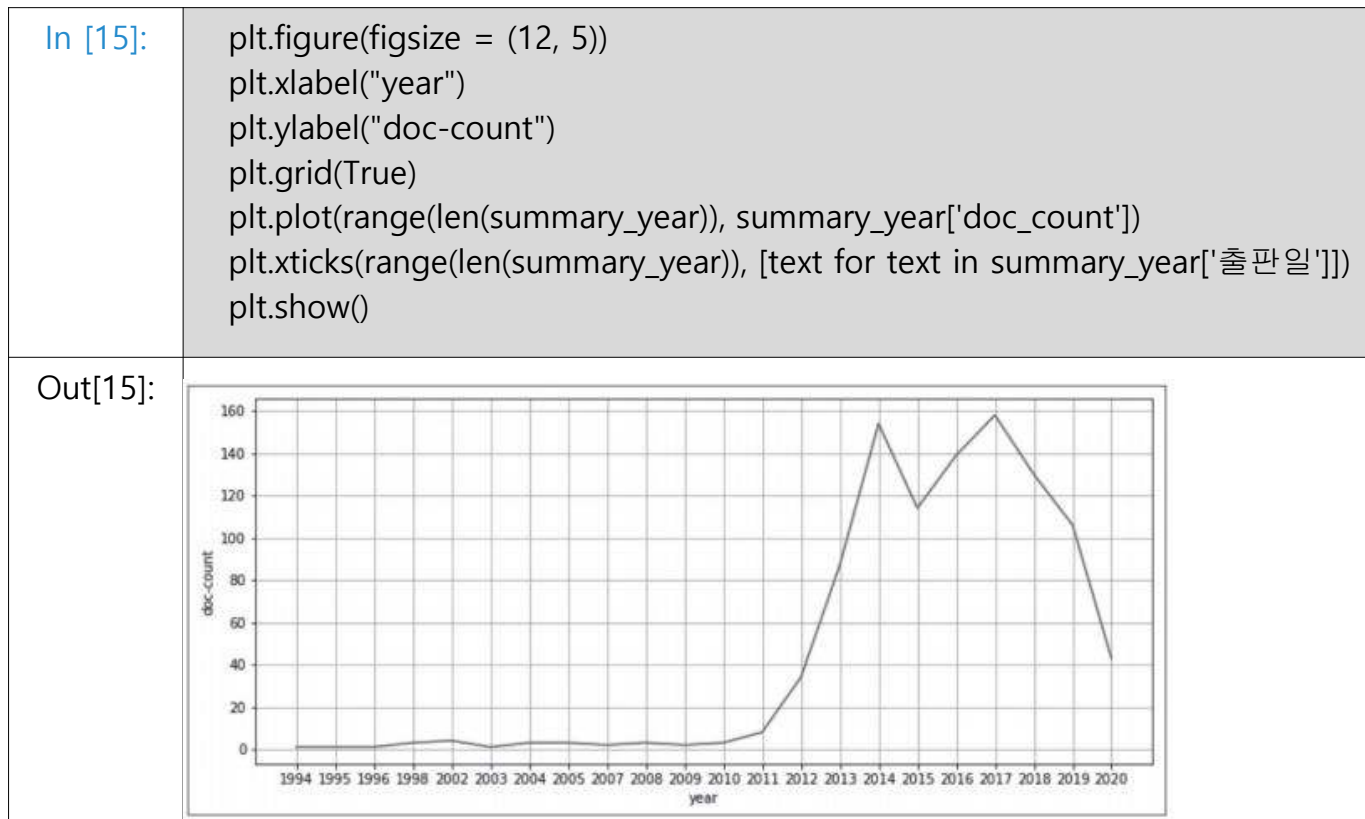
In [14]:	<pre>all_files_data_concat['doc_count'] = 0 summary_year = all_files_data_concat.groupby('출판일', as_index = False)['doc_count'].count() summary_year #출력하여 내용 확인</pre>																											
Out[14]:	<table><thead><tr><th></th><th>출판일</th><th>doc_count</th></tr></thead><tbody><tr><td>0</td><td>1994</td><td>1</td></tr><tr><td>1</td><td>1995</td><td>1</td></tr><tr><td>2</td><td>1996</td><td>1</td></tr><tr><td colspan="3">~~~~~</td></tr><tr><td>18</td><td>2017</td><td>158</td></tr><tr><td>19</td><td>2018</td><td>130</td></tr><tr><td>20</td><td>2019</td><td>106</td></tr><tr><td>21</td><td>2020</td><td>43</td></tr></tbody></table>		출판일	doc_count	0	1994	1	1	1995	1	2	1996	1	~~~~~			18	2017	158	19	2018	130	20	2019	106	21	2020	43
	출판일	doc_count																										
0	1994	1																										
1	1995	1																										
2	1996	1																										
~~~~~																												
18	2017	158																										
19	2018	130																										
20	2019	106																										
21	2020	43																										

In [14]: all_files_data_concat에 doc_count 컬럼을 추가한 뒤 '출판일' 컬럼을 기준으로 그룹을 만들고 `groupby()`, 그룹별 데이터 개수 `count()`를 doc_count 컬럼에 저장하여 summary_year 리스트를 생성

# 01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

## ■ 결과 시각화

### 1. 그래프 그리기



In [15]: summary_year의 doc_count 컬럼을 차트의 y축으로 설정하고 `plt.plot()`, '출판일' 컬럼은 x축으로 설정하여 `plt.xticks()` 차트를 그림

## 01. [영문 분석 + 워드클라우드] 영문 문서 제목의 키워드 분석하기

## ■ 결과 시각화

## 2. 워드클라우드 그리기

In [16]:

```
stopwords = set(STOPWORDS)
wc = WordCloud(background_color = 'ivory', stopwords = stopwords, width = 800, height = 600)
cloud = wc.generate_from_frequencies(word_count)
plt.figure(figsize = (8,8))
plt.imshow(cloud)
plt.axis('off')
plt.show()
```

Out[16]:



In [17]:

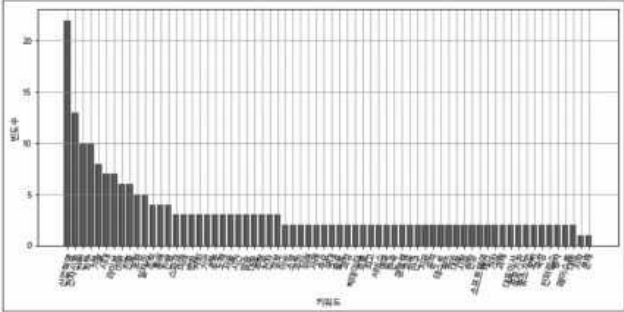
```
cloud.to_file("8장_data/riss_bigdata_wordCloud.jpg")
```

In [16]: 워드클라우드에서 처리할 불용어를 설정하고 `set(STOPWORDS)`, 워드클라우드 객체를 생성 `WordCloud()`  
워드클라우드 객체인 `wc`에 `word_count` 데이터를 담아서 `wc.generate_from_frequencies()` `cloud` 객체를 생성  
생성한 워드클라우드를 `matplotlib.pyplot`을 사용하여 나타냄

In [17]: 워드클라우드를 jpg 파일로 저장 `to_file()`.

# 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

## ■ 분석 미리보기

한글 뉴스 기사의 키워드 분석하기	
목표	'4차 산업혁명'에 관한 한글 기사에서 명사 키워드를 분석한다.
핵심 개념	형태소 분석, 품사 태깅
데이터 수집	4차 산업혁명 기사: 페이스북 전자신문 페이지에서 크롤링하여 저장한 json 파일(예제소스로 제공)
데이터 준비	1. 데이터 추출: json 파일에서 message 항목만 추출, key() 2. 명사 단어 추출: Okt 품사 태깅 패키지로 명사 추출, from konlpy.tag import Okt
데이터 탐색 및 모델링	1. 단어 빈도 탐색 • Counter() 2. 단어 빈도 히스토그램 • font_manager.FontProperties() • matplotlib.pyplot
결과 시각화	
1. 단어 빈도에 대한 히스토그램	2. 단어 빈도에 대한 워드클라우드
	



## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 목표 설정

- 목표: '4차 산업혁명'에 관한 한글 기사에서 명사 키워드를 분석하는 것
- 시각화 기법으로 워드클라우드를 사용
- 한글 텍스트 분석은 영어 텍스트 분석과 같은 절차를 수행

### ■ 핵심 개념 이해

#### ■ 형태소와 형태소 분석

- 언어에서 의미가 있는 가장 작은 단위
- 단어는 의미를 갖는 문장의 가장 작은 단일 요소로, 문장에서 분리될 수 있는 부분
- 독립형 형태소인 단어도 있지만, 대부분의 단어는 형태소와 접사로 구성
- 형태소 분석: 형태소, 어근, 접두사/접미사, 품사 등 다양한 언어학적 속성으로 구조를 파악하는 것

#### ■ 품사 태깅

- 형태소의 뜻과 문맥을 고려하여 품사를 붙이는 것
  - » 예) 가방에 들어가신다 → 가방/NNG + 예/JKM + 들어가/VV + 시/EPH + 니다/EFN

#### ■ 품사 태깅

- KoNLPy에서 사용 가능한 품사 태깅 패키지: Hannanum, Kkma, Komoran, Mecab, Okt(Twitter) 등

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 핵심 개념 이해

#### ■ 품사 태깅

표 8-4 품사 태깅 비교의 예

Hannanum	Kkma	Komoran	Mecab	Okt (Twitter)
아버지가방에들어가 / N	아버지 / NNG	아버지가방에들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시~다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	~다 / EFN		신다 / EP+EC	

### ■ 데이터 수집

- 페이스북 전자신문 페이지에서 '4차 산업혁명' 관련 기사를 크롤링한 'etnews.kr_facebook_2016-01-01_2018-08-01_4차 산업혁명.json' 파일을 사용

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 데이터 준비

1. KoNLPy를 설치 후 주피터 노트북에서 페이지를 추가한 뒤 '8장_한글단어분석'으로 파일 이름을 변경  
프로젝트에 필요한 파이썬 패키지를 임포트하고, 데이터를 준비

In [1]:	<pre>import json import re from konlpy.tag import Okt from collections import Counter import matplotlib import matplotlib.pyplot as plt from matplotlib import font_manager, rc from wordcloud import WordCloud</pre>
In [2]:	<pre>inputFileName = '8장_data/etnews.kr_facebook_2016-01-01_2018-08-01_4차 산업혁명' data = json.loads(open(inputFileName+'.json', 'r', encoding = 'utf8').read()) data #출력하여 내용 확인</pre>
Out[2]:	<pre>[{'created_time': '2018-06-20 18:06:39', 'link': 'https://www.facebook.com/etnews.kr/videos/1981346601899735/', 'message': '6월의 스파크포럼 - "미래 시대, 조직의 변화도 시작됐다!"\n\n스파크포럼은 현 사회의 사회문제 및 이슈를 제기하고, 그 이슈를 혁신적으로 해결하고자 하는 소셜이노베이터를 발굴, 지원하여 우리 사회 따뜻한 변화를 확산시키기 위해 만들 어진 도전과 만남의 자리입니다.\n\n6월의 스파크포럼에서는 4차 산업혁명 시대의 기업조직과 조직문화를 살펴보고, 조직의 변화를 받아들이고 실험해나가는 사례를 통해 미래 시대 조직이 나아가야 할 방향을 함께 생각해보고자 합니다.', 'name': '6월의 스파크포럼 - "미래 시대, 조직의 변화도 시작됐다!"', 'post_id': '407886705912407_1981346601899735', 'total_comments': 3}, {'created_time': '2018-06-14 10:41:16', ...}]</pre>

In [2]: json 파일을 읽어서 `json.loads()` data 객체에 저장, 한글이 깨지지 않도록 utf-8 형식으로 인코딩 `encoding='utf-8'`

- json: json 파일을 다루기 위한 모듈
- Okt: 한글 품사 태깅을 위한 모듈

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 데이터 준비

2. message' 키의 데이터에서 품사가 명사인 단어만 추출

In [3]:	<pre>message = "" for item in data:     if 'message' in item.keys():         message = message + re.sub(r'^\Ww', ' ', item['message']) + " message #출력하여 내용 확인</pre>
Out[3]:	6월의 스파크포럼 미래 시대 조직의 변화도 시작됐다 스파크포럼은 현 사회의 사회문제 및 이슈를 제기하고 그 이슈를 혁신적으로 해결하고자 하는 소셜이노베이터를 발굴 지원하여 우리 사회 따뜻한 변화를 확산시키기 위해 만들어진 도전과 만남의 자리입니다 6월의 스파크포럼에서는 4차 산업혁명 시대의 기업조직과 조직문화를 살펴보고 조직의 변화를 ...
In [4]:	<pre>nlp = Okt() message_N = nlp.nouns(message) message_N #출력하여 내용 확인</pre>
Out[4]:	['스파크', '포럼', '미래', '시대', '조직', '변화', '시작', '스파크', '포럼', '현', '사회', ... '차', '산업혁명', '흐름']

In [3]: 'message' 키의 값(뉴스 본문 내용)에서 문자나 숫자가 아닌 것 `r'^\Ww'`은 공백으로 치환하여 `re.sub()` 제거하면서 연결하여 전체를 하나의 문자열로 구성

In [4]: 품사 태깅 패키지인 Okt를 사용하여 명사만 추출해 `nlp.nouns()` message_N에 저장

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 데이터 탐색 및 분석 모델 구축

#### 1. 명사를 추출하여 저장한 message_N에 있는 단어들을 탐색

In [5]:	<pre>count = Counter(message_N) count #출력하여 내용 확인</pre>
Out[5]:	<pre>Counter({'스파크': 3,         '포럼': 5,         '미래': 3,         '시대': 7,         ...         '앞': 1})</pre>
In [6]:	<pre>word_count = dict() for tag, counts in count.most_common(80):     if(len(str(tag))&gt;1):         word_count[tag] = counts         print("%s : %d" % (tag, counts))</pre>
Out[6]:	<pre>산업혁명 : 22 전자신문 : 13 산업 : 10 ... 시작 : 1 문제 : 1</pre>

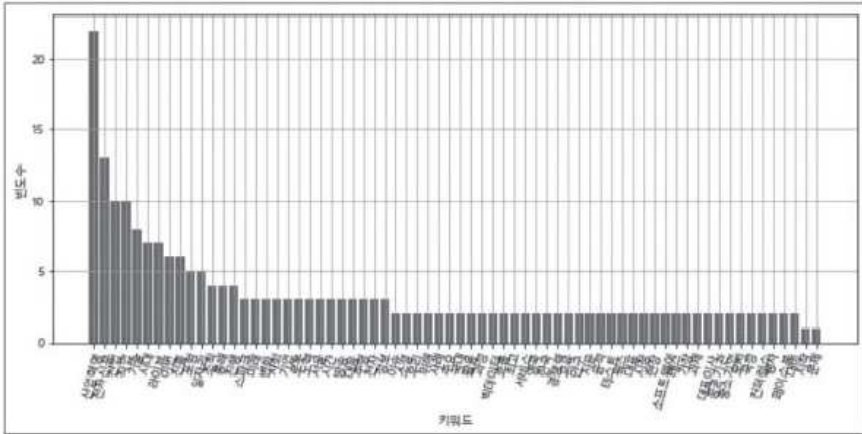
In [5]: Counter() 함수를 사용하여 단어별 출현 횟수를 계산

In [6]: 출현 횟수가 많은 상위 80개의 단어 중에서 길이가 1보다 큰 것만 word_count 딕셔너리에 저장하면서 출력하여 확인

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

### ■ 데이터 탐색 및 분석 모델 구축

#### 1. 히스토그램을 그려 단어 빈도를 시각적으로 탐색

In [7]:	<pre>font_path = "c:/Windows/fonts/malgun.ttf" font_name = font_manager.FontProperties(fname = font_path).get_name() matplotlib.rc('font', family = font_name)</pre>
In [8]:	<pre>plt.figure(figsize = (12, 5)) plt.xlabel('키워드') plt.ylabel('빈도수') plt.grid(True) sorted_Keys = sorted(word_count, key = word_count.get, reverse = True) sorted_Values = sorted(word_count.values(), reverse = True) plt.bar(range(len(word_count)), sorted_Values, align = 'center') plt.xticks(range(len(word_count)), list(sorted_Keys), rotation = '75') plt.show()</pre>
Out[8]:	

In [7]: 히스토그램에 레이블을 한글로 표시하기 위해 한글 폰트인 맑은고딕체 `malgun.ttf`를 설정 `matplotlib.rc()`

In [8]: 히스토그램을 만드는 방법은 8장 01절의 프로젝트와 같음

## 02. [한글 분석 + 워드클라우드] 한글 뉴스 기사의 키워드 분석하기

## ■ 결과 시각화

In [9]:	<pre> wc = WordCloud(font_path, background_color = 'ivory', width = 800, height = 600) cloud = wc.generate_from_frequencies(word_count) plt.figure(figsize = (8, 8)) plt.imshow(cloud) plt.axis('off') plt.show() </pre>
Out[9]:	
In [10]:	<pre> cloud.to_file(inputFileName + '_cloud.jpg') </pre>

In [9]: 워드클라우드 객체를 생성하고 `WordCloud()`, `word_count`에서 단어별 빈도수를 계산해서 `wc.generate_from_frequencies()`

cloud 객체에 저장하고, 워드클라우드를 생성 `plt.imshow()`

In [10]: 워드클라우드를 jpg 파일로 저장 `to_file()`.



# 데이터 과학 기반의 파이썬 빅데이터 분석

감사합니다.

---