데이터 분석

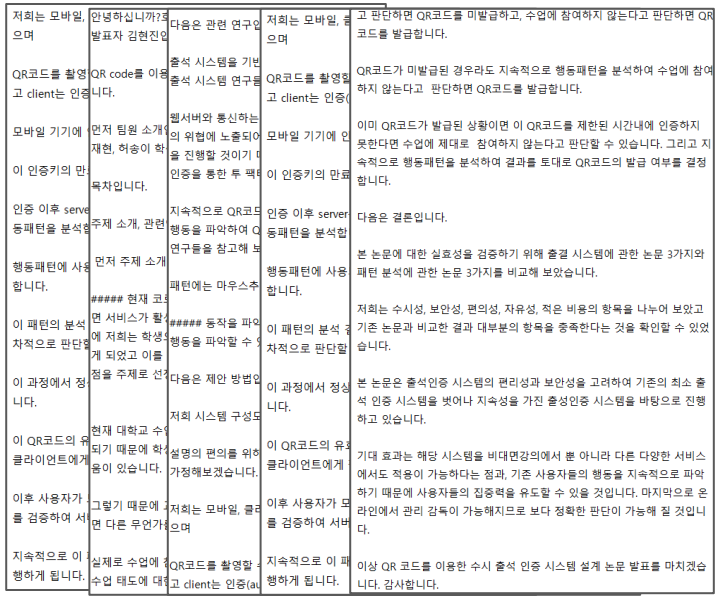
R & Python & 하둡

2021.02.18

텍스트 마이닝

비정형 텍스트 데이터에서 가치와 의미가 있는 정보를 찾아내는 기술

학술 대회



저희는 모바일,
으며

QR코드를 촬영
고 client는 인증

모바일 기기에

이 인증키의 만

인증 이후 serve
동패턴을 분석

행동패턴에 사
합니다.

이 패턴의 분석
차적으로 판단

이 과정에서 정
니다.

이 QR코드의 유
클라이언트에게

이후 사용자가
를 검증하여 서

지속적으로 이
행하게 됩니다.

안녕하십니까?
발표자 김현진

QR code를 이
니다.

먼저 팀원 소개
재현, 허승이 학

목차입니다.

주제 소개, 관련

먼저 주제 소개

현재 코
면 서비스가 활
에 저희는 학생
게 되었고 이를
점을 주제로 선

현재 대학교 수
되기 때문에 학
음이 있습니다.

그렇기 때문에
면 다른 무언가

실제로 수업에
수업 태도에 대

다음은 관련 연구

출석 시스템을 기
출석 시스템 연구

웹서버와 통신하는
의 위협에 노출되
을 진행할 것이기
인증을 통한 투 팩

지속적으로 QR코드
행동을 파악하여 Q
연구들을 참고해 보

패턴에는 마우스추

동작을 파악
행동을 파악할 수

다음은 제안 방법

저희 시스템 구성도

설명의 편의를 위
가정해보겠습니다.

저희는 모바일, 클
으며

QR코드를 촬영할
고 client는 인증(au

저희는 모바일, 클
으며

QR코드를 촬영할
고 client는 인증(

모바일 기기에 인

이 인증키의 만료

인증 이후 server
동패턴을 분석합

행동패턴에 사용
합니다.

이 패턴의 분석 결
차적으로 판단할

이 과정에서 정상
니다.

이 QR코드의 유효
클라이언트에게

이후 사용자가 모
를 검증하여 서버

지속적으로 이 패
행하게 됩니다.

고 판단하면 QR코드를 미발급하고, 수업에 참여하지 않는다고 판단하면 QR코드를 발급합니다.

QR코드가 미발급된 경우라도 지속적으로 행동패턴을 분석하여 수업에 참여하지 않는다고 판단하면 QR코드를 발급합니다.

이미 QR코드가 발급된 상황이면 이 QR코드를 제한된 시간내에 인증하지 못한다면 수업에 제대로 참여하지 않는다고 판단할 수 있습니다. 그리고 지속적으로 행동패턴을 분석하여 결과를 토대로 QR코드의 발급 여부를 결정합니다.

다음은 결론입니다.

본 논문에 대한 실효성을 검증하기 위해 출결 시스템에 관한 논문 3가지와 패턴 분석에 관한 논문 3가지를 비교해 보았습니다.

저희는 수시성, 보안성, 편의성, 자유성, 적은 비용의 항목을 나누어 보았고 기존 논문과 비교한 결과 대부분의 항목을 충족한다는 것을 확인할 수 있었습니다.

본 논문은 출석인증 시스템의 편리성과 보안성을 고려하여 기존의 최소 출석 인증 시스템을 벗어나 지속성을 가진 출성인증 시스템을 바탕으로 진행하고 있습니다.

기대 효과는 해당 시스템을 비대면강의에서 뿐 아니라 다른 다양한 서비스에서도 적용이 가능하다는 점과, 기존 사용자들의 행동을 지속적으로 파악하기 때문에 사용자들의 집중력을 유도할 수 있을 것입니다. 마지막으로 온라인에서 관리 감독이 가능해지므로 보다 정확한 판단이 가능해 질 것입니다.

이상 QR 코드를 이용한 수시 출석 인증 시스템 설계 논문 발표를 마치겠습니다. 감사합니다.

발표 대본

텍스트 마이닝만 보고
전달하고자 한 의미를 파악할 수 있을까?

test.R*

Source on Save

```r
library(KoNLP)

#사전 추가
useNIADic()

#텍스트 파일을 벡터 형식으로 저장
text1 <- readLines("kim.txt")

#명사만 추출
text2 <- extractNoun(text1)

#리스트 형식의 벡터를 벡터로 변환
text3 <- unlist(text2)

#단어의 길이값 설정
text4 <- text3[nchar(text3)>1]

#단어 빈도 순으로 내림차순 설정
text5 <- sort(table(text4), decreasing = T)

#맨 앞부터 300개 단어만 추출
text6 <- head(text5,300)

#wordcloud2 사용
install.packages("wordcloud2")
library(wordcloud2)
wordcloud2(text6)
```

```
> text2 <- extractNoun(text1)
> text2
[[1]]
 [1] "안녕하십니까" "호남"          "지부"          "추계"
 [5] "학술대회"     "김현진"        "조재현"        "허송"
 [9] "논문"         "발표"          "김현진"

[[2]]
[1] ""

[[3]]
 [1] "QR"     "code"   "이용"   "한"     "수시"   "출석"   "인증"   "시스템"
 [9] "설계"   "주제"   "발표"   "시작"

[[4]]
[1] ""

[[5]]
 [1] "팀원"   "소개"   "저희"   "팀원"   "3"      "명"     "김현진" "조재현"
 [9] "허송"   "학생"   "연구"   "참여"

[[6]]
[1] ""

[[7]]
[1] "목차"

[[8]]
[1] ""

[[9]]
 [1] "주제" "소개" "관련" "연구" "제안" "방법" "결론" "순"   "발표" "진행"
```

[1] 대본인 txt 파일을 한 줄 씩 읽어 벡터형태로 저장한 후 명사만 추출

test.R*

Source on Save

```
1  library(KoNLP)
2
3  #사전 추가
4  useNIADic()
5
6  #텍스트 파일을 벡터 형식으로 저장
7  text1 <- readLines("kim.txt")
8
9  #명사만 추출
10 text2 <- extractNoun(text1)
11
12 #리스트 형식의 벡터를 벡터로 변환
13 text3 <- unlist(text2)
14
15 #단어의 길이값 설정
16 text4 <- text3[nchar(text3)>1]
17
18 #단어 빈도 순으로 내림차순 설정
19 text5 <- sort(table(text4), decreasing = T)
20
21 #맨 앞부터 300개 단어만 추출
22 text6 <- head(text5,300)
23
24 #wordcloud2 사용
25 install.packages("wordcloud2")
26 library(wordcloud2)
27 wordcloud2(text6)
```

```
> text3 <- unlist(text2)
> text3
  [1] "안녕하십니까"   "호남"          "지부"
  [4] "주계"           "학술대회"      "김현진"
  [7] "조재현"         "허송"          "논문"
 [10] "발표"           "김현진"        ""
 [13] "QR"             "code"          "이용"
 [16] "한"             "수시"          "출석"
 [19] "인증"           "시스템"        "설계"
 [22] "주제"           "발표"          "시작"
 [25] ""               "팀원"          "소개"
 [28] "저희"           "팀원"          "3"
 [31] "명"             "김현진"        "조재현"
 [34] "허송"           "학생"          "연구"
 [37] "참여"           ""              "목자"
 [40] ""               "주제"          "소개"
 [43] "관련"           "연구"          "제안"
 [46] "방법"           "결론"          "순"
 [49] "발표"           "진행"          ""
 [52] "주제"           "소개"          ""
 [55] "코로나"         "4차산업"       "혁명"
 [58] "가속화"         "시점"          "비대"
 [61] "서비스"         "활성화"        "저희"
 [64] "삶"             "부분"          "변화"
 [67] "중"             "저희"          "학생"
 [70] "학교"           "진행"          "비대"
 [73] "강의"           "문제"          "점"
 [76] "이"             "해결"          "하고"
 [79] "언택트"         "시대"          "비대"
 [82] "강의"           "문제"          "점"
 [85] "주제"           "선정"          "하게"
 [88] ""               ""              "대학교"
 [91] "수업"           "강의실"        "수업"
```

[2] 리스트 형태로 되어 있기 때문에 한 단어 씩 읽기 위하여 unlist 설정

test.R*

Source on Save

```
1  library(KoNLP)
2
3  #사전 추가
4  useNIADic()
5
6  #텍스트 파일을 벡터 형식으로 저장
7  text1 <- readLines("kim.txt")
8
9  #명사만 추출
10 text2 <- extractNoun(text1)
11
12 #리스트 형식의 벡터를 벡터로 변환
13 text3 <- unlist(text2)
14
15 #단어의 길이값 설정
16 text4 <- text3[nchar(text3)>1]
17
18 #단어 빈도 순으로 내림차순 설정
19 text5 <- sort(table(text4), decreasing = T)
20
21 #맨 앞부터 300개 단어만 추출
22 text6 <- head(text5,300)
23
24 #wordcloud2 사용
25 install.packages("wordcloud2")
26 library(wordcloud2)
27 wordcloud2(text6)
```

```
> text4 <- text3[nchar(text3)>1]
> text4
  [1] "안녕하십니까"   "호남"        "지부"
  [4] "추계"          "학술대회"    "김현진"
  [7] "조재현"        "허송"        "논문"
 [10] "발표"          "김현진"      "QR"
 [13] "code"          "이용"        "수시"
 [16] "출석"          "인증"        "시스템"
 [19] "설계"          "주제"        "발표"
 [22] "시작"          "팀원"        "소개"
 [25] "저희"          "팀원"        "김현진"
 [28] "조재현"        "허송"        "학생"
 [31] "연구"          "참여"        "목차"
 [34] "주제"          "소개"        "관련"
 [37] "연구"          "제안"        "방법"
 [40] "결론"          "발표"        "진행"
 [43] "주제"          "소개"        "코로나"
 [46] "4차산업"       "혁명"        "가속화"
 [49] "시점"          "비대"        "서비스"
 [52] "활성화"        "저희"        "부분"
 [55] "변화"          "저희"        "학생"
 [58] "학교"          "진행"        "비대"
 [61] "강의"          "문제"        "해결"
 [64] "하고"          "언택트"      "시대"
 [67] "비대"          "강의"        "문제"
 [70] "주제"          "선정"        "하게"
 [73] "대학교"        "수업"        "강의실"
 [76] "수업"          "온라인"      "진행"
 [79] "때문"          "학생"        "수업"
 [82] "태도"          "수업"        "분위기"
```

[3] 단어가 아닌 것들을 구별하기 위해 길이 값을 설정

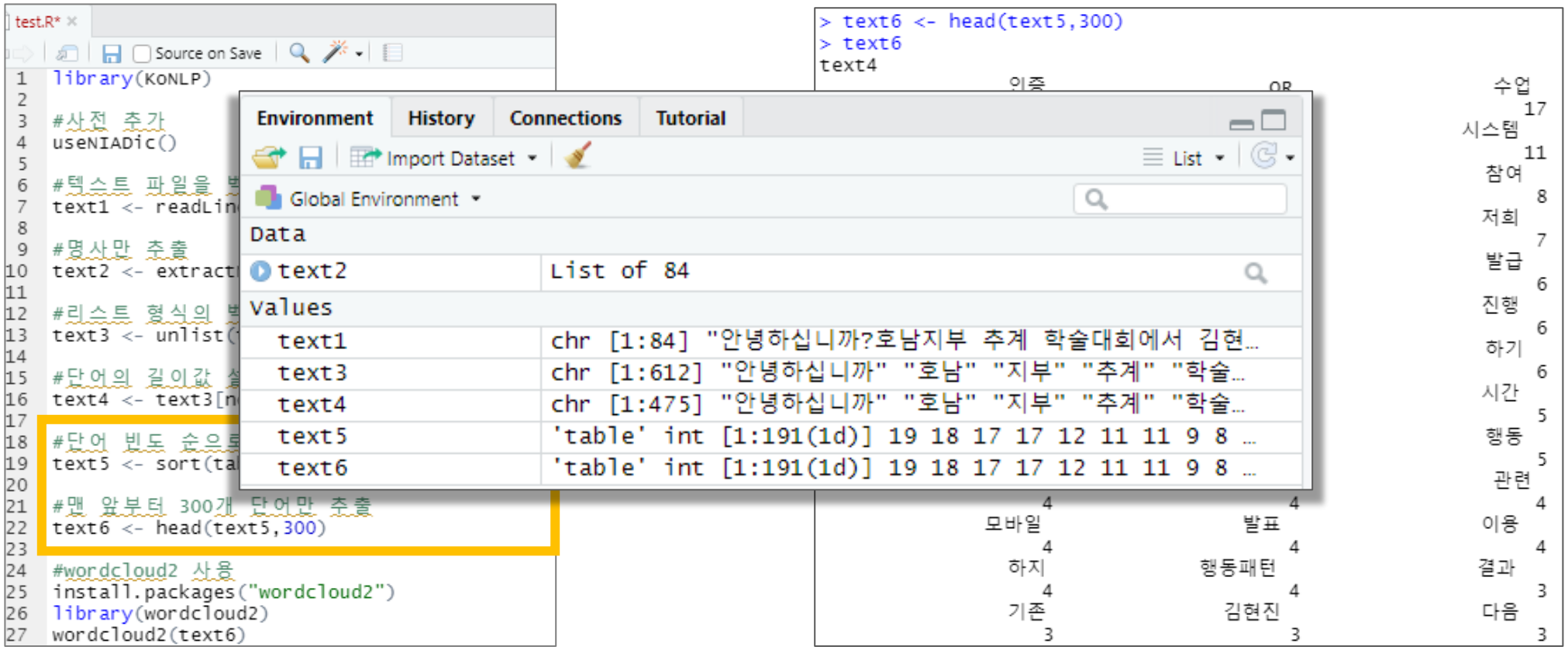
test.R*

Source on Save

```
1  library(KoNLP)
2
3  #사전 추가
4  useNIADic()
5
6  #텍스트 파일을 벡터 형식으로 저장
7  text1 <- readLines("kim.txt")
8
9  #명사만 추출
10 text2 <- extractNoun(text1)
11
12 #리스트 형식의 벡터를 벡터로 변환
13 text3 <- unlist(text2)
14
15 #단어의 길이값 설정
16 text4 <- text3[nchar(text3)>1]
17
18 #단어 빈도 순으로 내림차순 설정
19 text5 <- sort(table(text4), decreasing = T)
20
21 #맨 앞부터 300개 단어만 추출
22 text6 <- head(text5,300)
23
24 #wordcloud2 사용
25 install.packages("wordcloud2")
26 library(wordcloud2)
27 wordcloud2(text6)
```

```
> text6 <- head(text5,300)
> text6
text4
      인증        QR      수업
        19        18        17
      코드      논문  시스템
        17        12        11
      출석      서버  참여
        11         9         8
      학생    사용자  저희
         8         7         7
      패턴      때문  발급
         7         6         6
      분석    지속적  진행
         6         6         6
      파악      판단  하기
         6         6         6
    서비스      수시  시간
         5         5         5
      연구      주제  행동
         5         5         5
    client      강의  관련
         4         4         4
    모바일      발표  이용
         4         4         4
      하지  행동패턴  결과
         4         4         3
      기존    김현진  다음
         3         3         3
```

[4] sort 를 이용하여 단어 빈도수대로 정렬



```
test.R*
Source on Save
1  library(KONLP)
2
3  #사전 추가
4  useNIADic()
5
6  #텍스트 파일을 
7  text1 <- readLin
8
9  #명사만 추출
10 text2 <- extract
11
12 #리스트 형식의 
13 text3 <- unlist(
14
15 #단어의 길이값 
16 text4 <- text3[n
17
18 #단어 빈도 순으로
19 text5 <- sort(ta
20
21 #맨 앞부터 300개 단어만 추출
22 text6 <- head(text5,300)
23
24 #wordcloud2 사용
25 install.packages("wordcloud2")
26 library(wordcloud2)
27 wordcloud2(text6)
```

Environment | History | Connections | Tutorial

Import Dataset | List

Global Environment

Data	
text2	List of 84
Values	
text1	chr [1:84] "안녕하십니까?호남지부 추계 학술대회에서 김현…
text3	chr [1:612] "안녕하십니까" "호남" "지부" "추계" "학술…
text4	chr [1:475] "안녕하십니까" "호남" "지부" "추계" "학술…
text5	'table' int [1:191(1d)] 19 18 17 17 12 11 11 9 8 …
text6	'table' int [1:191(1d)] 19 18 17 17 12 11 11 9 8 …

```
> text6 <- head(text5,300)
> text6
text4
```

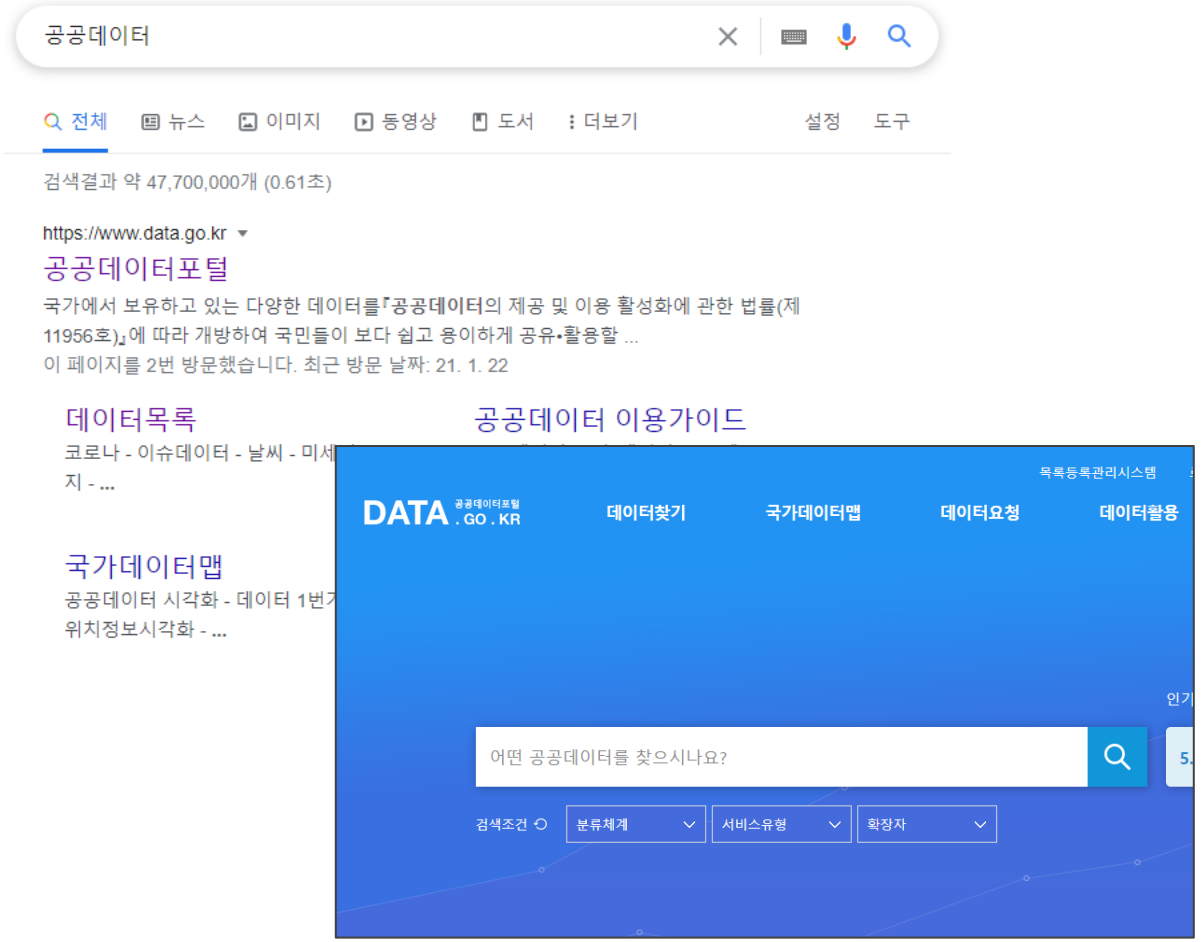
인증	OR	수업 17
		시스템 11
		참여 8
		저희 7
		발급 6
		진행 6
		하기 6
		시간 5
		행동 5
		관련 4
4	4	
모바일	발표	이용 4
4	4	
하지	행동패턴	결과 3
4	4	
기존	김현진	다음 3
3	3	

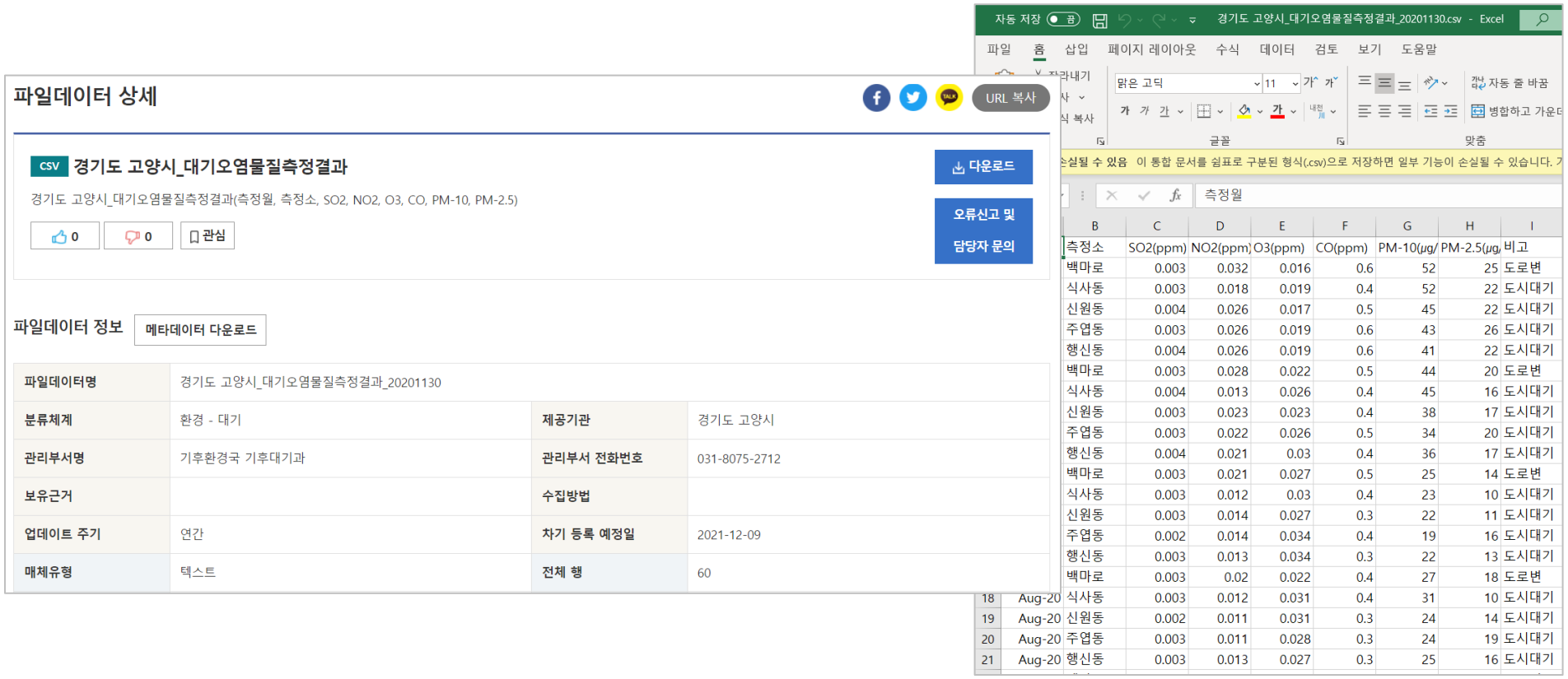
[4] sort 를 이용하여 단어 빈도수대로 정렬 (475개 벡터 중 300개만 확인)

발표대본에서 어떤 것을 말하고자 하는지 확인 할 수 있다!

공공데이터

공공기관이 전자적으로 생성 또는 취득하여 관리하고 있는 모든 데이터 베이스(DB) & 전자화된 파일.





파일데이터 상세

URL 복사

CSV 경기도 고양시_대기오염물질측정결과

경기도 고양시_대기오염물질측정결과(측정월, 측정소, SO2, NO2, O3, CO, PM-10, PM-2.5)

0 0 관심

다운로드

오류신고 및 담당자 문의

파일데이터 정보 메타데이터 다운로드

파일데이터명	경기도 고양시_대기오염물질측정결과_20201130		
분류체계	환경 - 대기	제공기관	경기도 고양시
관리부서명	기후환경국 기후대기과	관리부서 전화번호	031-8075-2712
보유근거		수집방법	
업데이트 주기	연간	차기 등록 예정일	2021-12-09
매체유형	텍스트	전체 행	60

경기도 고양시_대기오염물질측정결과_20201130.csv - Excel

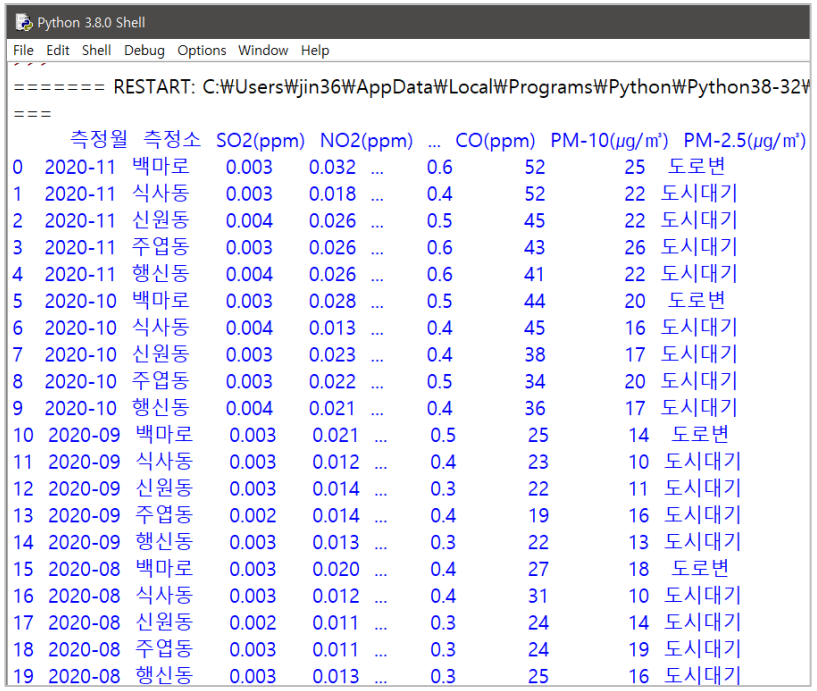
측정월

	B	C	D	E	F	G	H	I
	측정소	SO2(ppm)	NO2(ppm)	O3(ppm)	CO(ppm)	PM-10(㎍/	PM-2.5(㎍/	비고
	백마로	0.003	0.032	0.016	0.6	52	25	도로변
	식사동	0.003	0.018	0.019	0.4	52	22	도시대기
	신원동	0.004	0.026	0.017	0.5	45	22	도시대기
	주엽동	0.003	0.026	0.019	0.6	43	26	도시대기
	행신동	0.004	0.026	0.019	0.6	41	22	도시대기
	백마로	0.003	0.028	0.022	0.5	44	20	도로변
	식사동	0.004	0.013	0.026	0.4	45	16	도시대기
	신원동	0.003	0.023	0.023	0.4	38	17	도시대기
	주엽동	0.003	0.022	0.026	0.5	34	20	도시대기
	행신동	0.004	0.021	0.03	0.4	36	17	도시대기
	백마로	0.003	0.021	0.027	0.5	25	14	도로변
	식사동	0.003	0.012	0.03	0.4	23	10	도시대기
	신원동	0.003	0.014	0.027	0.3	22	11	도시대기
	주엽동	0.002	0.014	0.034	0.4	19	16	도시대기
	행신동	0.003	0.013	0.034	0.3	22	13	도시대기
	백마로	0.003	0.02	0.022	0.4	27	18	도로변
18	Aug-20 식사동	0.003	0.012	0.031	0.4	31	10	도시대기
19	Aug-20 신원동	0.002	0.011	0.031	0.3	24	14	도시대기
20	Aug-20 주엽동	0.003	0.011	0.028	0.3	24	19	도시대기
21	Aug-20 행신동	0.003	0.013	0.027	0.3	25	16	도시대기

CSV, xml, 오픈 API 등의 다양한 형태의 데이터 자료를 받을 수 있음

Python 데이터 시각화

그래프 편 _ 주피터 노트북



Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

======= RESTART: C:₩Users₩jin36₩AppData₩Local₩Programs₩Python₩Python38-32₩
===

	측정월	측정소	SO2(ppm)	NO2(ppm)	...	CO(ppm)	PM-10(㎍/㎥)	PM-2.5(㎍/㎥)	
0	2020-11	백마로	0.003	0.032	...	0.6	52	25	도로변
1	2020-11	식사동	0.003	0.018	...	0.4	52	22	도시대기
2	2020-11	신원동	0.004	0.026	...	0.5	45	22	도시대기
3	2020-11	주엽동	0.003	0.026	...	0.6	43	26	도시대기
4	2020-11	행신동	0.004	0.026	...	0.6	41	22	도시대기
5	2020-10	백마로	0.003	0.028	...	0.5	44	20	도로변
6	2020-10	식사동	0.004	0.013	...	0.4	45	16	도시대기
7	2020-10	신원동	0.003	0.023	...	0.4	38	17	도시대기
8	2020-10	주엽동	0.003	0.022	...	0.5	34	20	도시대기
9	2020-10	행신동	0.004	0.021	...	0.4	36	17	도시대기
10	2020-09	백마로	0.003	0.021	...	0.5	25	14	도로변
11	2020-09	식사동	0.003	0.012	...	0.4	23	10	도시대기
12	2020-09	신원동	0.003	0.014	...	0.3	22	11	도시대기
13	2020-09	주엽동	0.002	0.014	...	0.4	19	16	도시대기
14	2020-09	행신동	0.003	0.013	...	0.3	22	13	도시대기
15	2020-08	백마로	0.003	0.020	...	0.4	27	18	도로변
16	2020-08	식사동	0.003	0.012	...	0.4	31	10	도시대기
17	2020-08	신원동	0.002	0.011	...	0.3	24	14	도시대기
18	2020-08	주엽동	0.003	0.011	...	0.3	24	19	도시대기
19	2020-08	행신동	0.003	0.013	...	0.3	25	16	도시대기

측정월	측정소	SO2(ppm)	NO2(ppm)	O3(ppm)	CO(ppm)	PM-10(㎍/㎥)	PM-2.5(㎍/㎥)	비고
2020-11	백마로	0.003	0.032	0.016	0.6	52	25	도로변
2020-11	식사동	0.003	0.018	0.019	0.4	52	22	도시대기
2020-11	신원동	0.004	0.026	0.017	0.5	45	22	도시대기
2020-11	주엽동	0.003	0.026	0.019	0.6	43	26	도시대기
2020-11	행신동	0.004	0.026	0.019	0.6	41	22	도시대기
2020-10	백마로	0.003	0.028	0.022	0.5	44	20	도로변
2020-10	식사동	0.004	0.013	0.026	0.4	45	16	도시대기
2020-10	신원동	0.003	0.023	0.023	0.4	38	17	도시대기
2020-10	주엽동	0.003	0.022	0.026	0.5	34	20	도시대기
2020-10	행신동	0.004	0.021	0.030	0.4	36	17	도시대기
2020-09	백마로	0.003	0.021	0.027	0.5	25	14	도로변
2020-09	식사동	0.003	0.012	0.030	0.4	23	10	도시대기
2020-09	신원동	0.003	0.014	0.027	0.3	22	11	도시대기
2020-09	주엽동	0.002	0.014	0.034	0.4	19	16	도시대기
2020-09	행신동	0.003	0.013	0.034	0.3	22	13	도시대기
2020-08	백마로	0.003	0.020	0.022	0.4	27	18	도로변

데이터 분석을 위한 필수 패키지 삼대장!

Pandas : 데이터를 처리할 때 사용하는 패키지

Numpy : 수치 데이터를 다루는 패키지 (벡터 및 행렬을 주로 계산)

Matplotlib : 데이터를 차트(char)나 플롯(plot)으로 시각화 하는 패키지

```
%matplotlib inline
# %matplotlib inline 설정 : matplotlib.pyplot 의 show 를 하지 않아도 그래프가 자동적으로 보여짐

# 데이터 처리하기 위한 패키지 : pandas
# 그래프 사용을 위한 패키지 : matplotlib
import pandas as pd
import matplotlib.pyplot as plt

font_name = mpl.font_manager.FontProperties(fname='C:/Windows/Fonts/malgun.ttf').get_name()
mpl.rc('font', family=font_name)

dataframe = pd.read_csv('경기도 고양시_대기오염물질측정결과_20201130.csv', encoding='euc-kr',index_col ="측정월")
dataframe
```

[1] Pandas 와 matplotlib 패키지를 import 시켜준다.

```
%matplotlib inline
# %matplotlib inline 설정 : matplotlib.pyplot 의 show 를 하지 않아도 그래프가 자동적으로 보여짐

# 데이터 처리하기 위한 패키지 : pandas
# 그래프 사용을 위한 패키지 : matplotlib
import pandas as pd
import matplotlib.pyplot as plt
```

```
font_name = mpl.font_manager.FontProperties(fname='C:/Windows/Fonts/malgun.ttf').get_name()
mpl.rc('font', family=font_name)

dataframe = pd.read_csv('경기도 고양시_대기오염물질측정결과_20201130.csv', encoding='euc-kr')
dataframe
```

[2] 공공데이터 포털에서 가져온 csv를 데이터 프레임 형태로 저장한다.

```
%matplotlib inline
# %matplotlib inline 설정

# 데이터 처리하기 위한 패키
# 그래프 사용을 위한 패키지
import pandas as pd
import matplotlib.pyplot a

font_name = mpl.font_manag
mpl.rc('font', family=font

dataframe = pd.read_csv('경
dataframe
```

```
dataframe = pd.read_csv('경기도 고양시_대기오염물질측정결과_20201130.csv', encoding=
dataframe
```

측정월	측정소	SO2(ppm)	NO2(ppm)	O3(ppm)	CO(ppm)	PM-10(㎍/㎥)	PM-2.5(㎍/㎥)	비고
2020-11	백마로	0.003	0.032	0.016	0.6	52	25	도로변
2020-11	식사동	0.003	0.018	0.019	0.4	52	22	도시대기
2020-11	신원동	0.004	0.026	0.017	0.5	45	22	도시대기
2020-11	주엽동	0.003	0.026	0.019	0.6	43	26	도시대기
2020-11	행신동	0.004	0.026	0.019	0.6	41	22	도시대기
2020-10	백마로	0.003	0.028	0.022	0.5	44	20	도로변
2020-10	식사동	0.004	0.013	0.026	0.4	45	16	도시대기
2020-10	신원동	0.003	0.023	0.023	0.4	38	17	도시대기
2020-10	주엽동	0.003	0.022	0.026	0.5	34	20	도시대기
2020-10	행신동	0.004	0.021	0.030	0.4	36	17	도시대기
2020-09	백마로	0.003	0.021	0.027	0.5	25	14	도로변
2020-09	식사동	0.003	0.012	0.030	0.4	23	10	도시대기
2020-09	신원동	0.003	0.014	0.027	0.3	22	11	도시대기
2020-09	주엽동	0.002	0.014	0.034	0.4	19	16	도시대기
2020-09	행신동	0.003	0.013	0.034	0.3	22	13	도시대기

[2] 공공데이터 포털에서 가져온 csv를 데이터 프레임 형태로 저장한다.

측정월	측정소	SO2(ppm)	NO2(ppm)	O3(ppm)	CO(ppm)	PM-10(㎍/㎥)	PM-2.5(㎍/㎥)	비고
2020-11	백마로	0.003	0.032	0.016	0.6	52	25	도로변
2020-11	식사동	0.003	0.018	0.019	0.4	52	22	도시대기
2020-11	신원동	0.004	0.026	0.017	0.5	45	22	도시대기
2020-11	주엽동	0.003	0.026	0.019	0.6	43	26	도시대기
2020-11	행신동	0.004	0.026	0.019	0.6	41	22	도시대기
2020-10	백마로	0.003	0.028	0.022	0.5	44	20	도로변
2020-10	식사동	0.004	0.013	0.026	0.4	45	16	도시대기
2020-10	신원동	0.003	0.023	0.023	0.4	38	17	도시대기
2020-10	주엽동	0.003	0.022	0.026	0.5	34	20	도시대기
2020-10	행신동	0.004	0.021	0.030	0.4	36	17	도시대기
2020-09	백마로	0.003	0.021	0.027	0.5	25	14	도로변
2020-09	식사동	0.003	0.012	0.030	0.4	23	10	도시대기
2020-09	신원동	0.003	0.014	0.027	0.3	22	11	도시대기
2020-09	주엽동	0.002	0.014	0.034	0.4	19	16	도시대기
2020-09	행신동	0.003	0.013	0.034	0.3	22	13	도시대기
2020-08	백마로	0.003	0.020	0.022	0.4	27	18	도로변
2020-08	식사동	0.003	0.012	0.031	0.4	31	10	도시대기
2020-08	신원동	0.002	0.011	0.031	0.3	24	14	도시대기

	A	B	C	D	E	F	G	H	I
1	측정월	측정소	SO2(ppm)	NO2(ppm)	O3(ppm)	CO(ppm)	PM-10(㎍/	PM-2.5(㎍/	비고
2	Nov-20	백마로	0.003	0.032	0.016	0.6	52	25	도로변
3	Nov-20	식사동	0.003	0.018	0.019	0.4	52	22	도시대기
4	Nov-20	신원동	0.004	0.026	0.017	0.5	45	22	도시대기
5	Nov-20	주엽동	0.003	0.026	0.019	0.6	43	26	도시대기
6	Nov-20	행신동	0.004	0.026	0.019	0.6	41	22	도시대기
7	Oct-20	백마로	0.003	0.028	0.022	0.5	44	20	도로변
8	Oct-20	식사동	0.004	0.013	0.026	0.4	45	16	도시대기
9	Oct-20	신원동	0.003	0.023	0.023	0.4	38	17	도시대기
10	Oct-20	주엽동	0.003	0.022	0.026	0.5	34	20	도시대기
11	Oct-20	행신동	0.004	0.021	0.03	0.4	36	17	도시대기
12	Sep-20	백마로	0.003	0.021	0.027	0.5	25	14	도로변
13	Sep-20	식사동	0.003	0.012	0.03	0.4	23	10	도시대기
14	Sep-20	신원동	0.003	0.014	0.027	0.3	22	11	도시대기
15	Sep-20	주엽동	0.002	0.014	0.034	0.4	19	16	도시대기
16	Sep-20	행신동	0.003	0.013	0.034	0.3	22	13	도시대기
17	Aug-20	백마로	0.003	0.02	0.022	0.4	27	18	도로변
18	Aug-20	식사동	0.003	0.012	0.031	0.4	31	10	도시대기
19	Aug-20	신원동	0.002	0.011	0.031	0.3	24	14	도시대기
20	Aug-20	주엽동	0.003	0.011	0.028	0.3	24	19	도시대기
21	Aug-20	행신동	0.003	0.013	0.027	0.3	25	16	도시대기
22	Jul-20	백마로	0.003	0.021	0.033	0.5	27	17	도로변

실제로 다운로드한 csv 형태 그대로 데이터프레임이 생성된 것을 확인할 수 있음!

```
dataframe.shape
```

```
(60, 8)
```

```
dataframe.columns
```

```
Index(['측정소', 'SO2(ppm)', 'NO2(ppm)', 'O3(ppm)', 'CO(ppm)', 'PM-10(㎍/㎥)',
       'PM-2.5(㎍/㎥)', '비고'],
      dtype='object')
```

[3] 실제로 데이터 행과 열이 몇 개인지 파악하고 columns을 확인

(이 과정은 데이터 분석가들이 자신들이 사용할 데이터를 분류하기 위해 꼭 필요한 작업!)

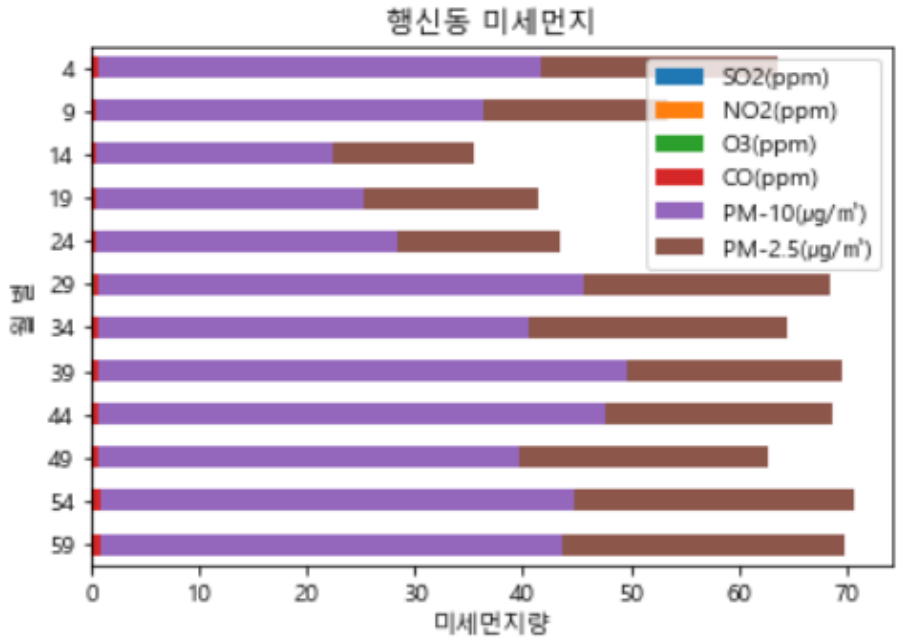
```
data = dataframe.iloc[[4,9,14,19,24,29,34,39,44,49,54,59],:]
data = data.sort_values(by='측정월', ascending=True)
data
```

	측정월	측정소명	이산화질소농도	오존농도	일산화탄소농도	아황산가스농도	미세먼지농도	초미세먼지농도	지역
59	2019-12	행신동	0.004	0.029	0.016	0.7	43	26	도시대기
54	2020-01	행신동	0.003	0.028	0.018	0.7	44	26	도시대기
49	2020-02	행신동	0.003	0.025	0.024	0.6	39	23	도시대기
44	2020-03	행신동	0.004	0.021	0.033	0.6	47	21	도시대기
39	2020-04	행신동	0.003	0.017	0.041	0.5	49	20	도시대기
34	2020-05	행신동	0.003	0.014	0.044	0.5	40	24	도시대기
29	2020-06	행신동	0.004	0.014	0.051	0.5	45	23	도시대기
24	2020-07	행신동	0.003	0.012	0.034	0.4	28	15	도시대기
19	2020-08	행신동	0.003	0.013	0.027	0.3	25	16	도시대기
14	2020-09	행신동	0.003	0.013	0.034	0.3	22	13	도시대기
9	2020-10	행신동	0.004	0.021	0.030	0.4	36	17	도시대기
4	2020-11	행신동	0.004	0.026	0.019	0.6	41	22	도시대기

[4] sort 를 이용하여 측정월을 오름차순으로 정렬 (ascending=True)

```
plot = data.plot(kind='barh',stacked = True)
plt.title("행신동 미세먼지")
plt.ylabel("월 별")
plt.xlabel("미세먼지량")
```

```
Text(0.5, 0, '미세먼지량')
```



[5] plot 사용하여 누적 바 형태로 그래프 제작!

stacked = True : 누적 바

plt (= Matplotlib.pyplot) 를 사용하여

그래프 설명 추가!

Hadoop

대용량의 데이터를 저장, 처리, 분석하는 오픈소스 프레임 워크

대규모 데이터를 여러 대의 컴퓨터에서 병렬로 분산 처리하여 처리속도 향상, 비용 절감

Why Hadoop?

비정형 데이터(동영상, 이미지 등..)의 기하급수적인 증가를 수용하기 위한 기존 방식은 높은 비용을 요구함.

RDBMS

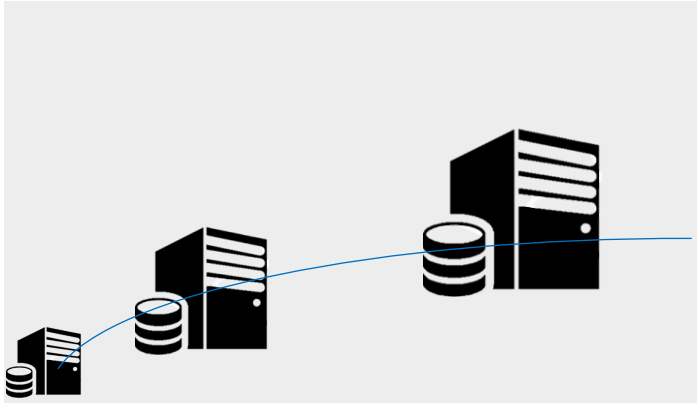
구조화된 데이터

Scale-up 방식

높은 유지보수 비용

데이터 정형화 필요

Giga byte 급 데이터 저장



Hadoop

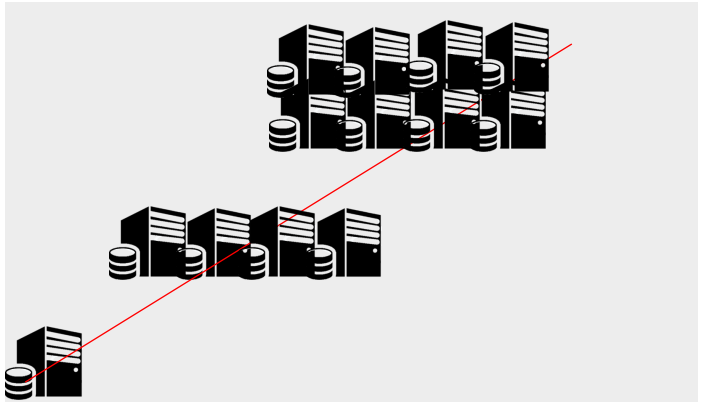
데이터 저장소 확장 가능 (병렬로 분산처리)

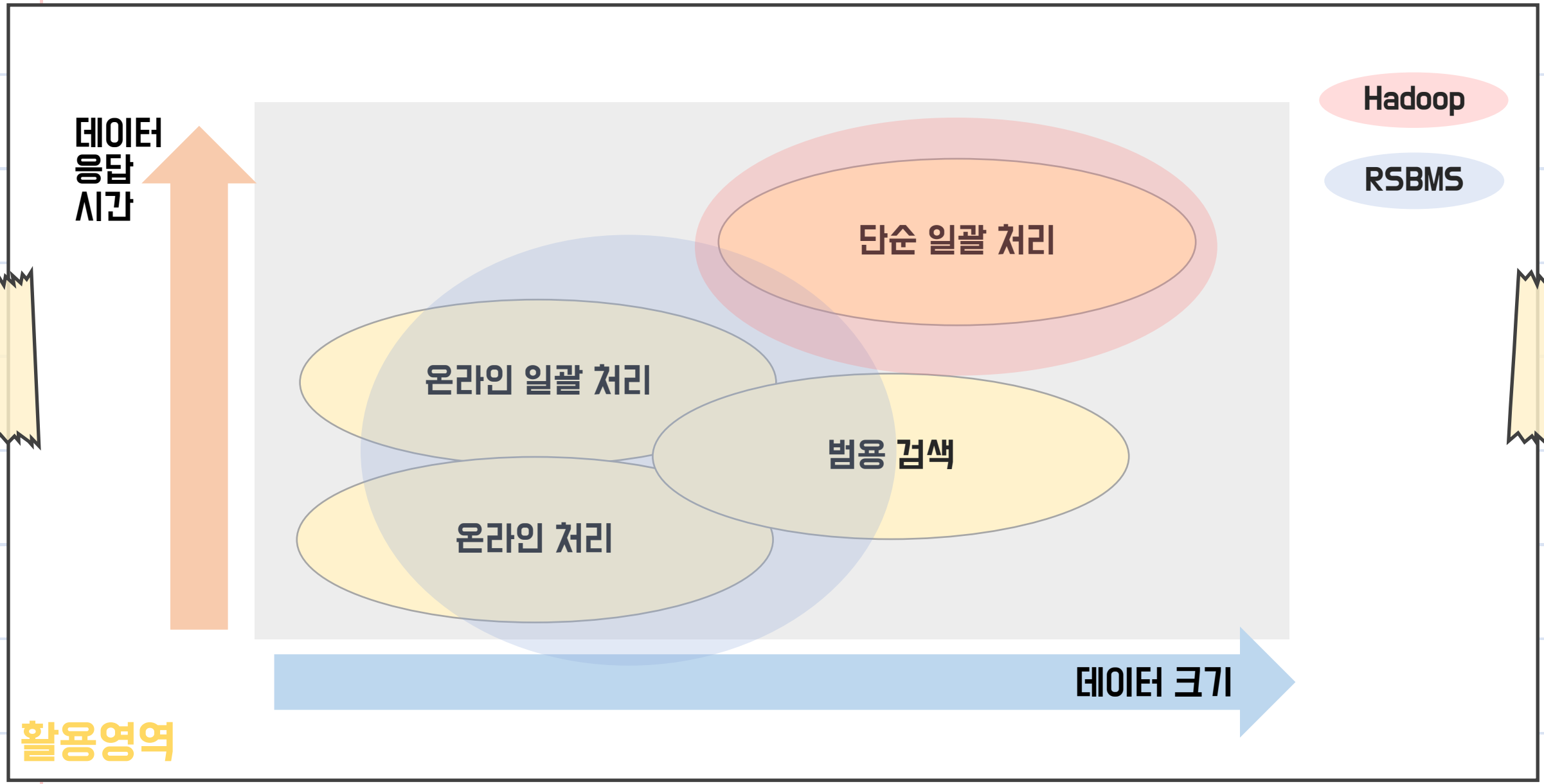
Scale-out 방식

낮은 유지보수 비용

정형/비정형 데이터 적재 가능

Peta byte 급 데이터 저장




데이터
응답
시간
Hadoop
RSBMS
단순 일괄 처리
온라인 일괄 처리
범용 검색
온라인 처리
데이터 크기
활용영역

감사합니다

2021.02.18