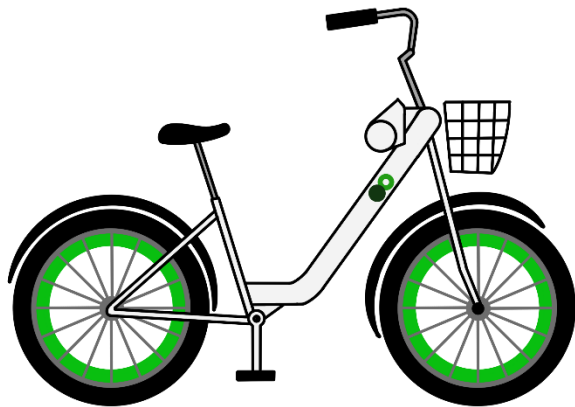


따릉이 DATABASE 구축 및 분석



1829008 컴퓨터공학과 김민영
데이터베이스 과제2



C O N T E N T

- 1. 데이터 소개
- 2. 전처리
- 3. 데이터베이스 구축
- 4. 데이터 분석
- 5. 결론

1 데이터 소개

1. 자전거 대여소 data



7630 x 8

2. 날씨 data



7457 x 11

3. 미세먼지 data



7625 x 11

4. 자전거 대여기록 data

○ 이용시간	□ 거리(km)
179'	10.78km
㎥ 칼로리(kcal)	㎍ 탄소발감효과(kg)
277.48kcal	2.51kg

자전거	대여일시	대여소	반납일시	반납대여소
SPB-14251	2019-02-21 18:32	201. 천미과학 관 앞	2019-02-21 19:02	274. 영등포역 지하소형센터 5 반출구
SPB-14251	2019-02-21 17:52	203. 국회도서관 영역 3반출구 앞	2019-02-21 18:31	201. 천미과학 관 앞

6447020 x 13

main data
6447020 raws
(약 600만개)

1 DATA 소개

2. 날씨 (WEATHER)

DATE	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
TEMPERATURE	FLOAT
HUMID	FLOAT
WIND	FLOAT
PERCIPITATION	FLOAT

3. 미세먼지 (DUST)

DATE	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
FINEDUST	VARCHAR(50)
ULTRAFINEDUST	INT
OZONE	FLOAT
NO2	FLOAT
CO	FLOAT
SO2	FLOAT

1. 자전거 대여소(PLACE)

PLACE_ID	VARCHAR(50)
PLACE_NAME	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
BIKE HOLDER_COUNT	INT
LATITUDE	FLOAT
LONGTITUDE	FLOAT

4. 자전거 대여기록(RENTAL)

DATE	VARCHAR(50)
TIME	VARCHAR(50)
PLACE_ID	VARCHAR(50)
MEMBER	FLOAT
SEX	FLOAT
AGE	VARCHAR(50)
USE_COUNT	INT
EXCERCISE	FLOAT
DISTANCE	FLOAT
DURING_MINUTES	FLOAT
USE_MINUTES	FLOAT

2 전처리

RAW 데이터의 문제들

- 1) 이름 불일치 문제
- 2) 따옴표 등 특수문자 문제
- 3) 파일이 분리 문제

➡ 일관성을 위해 데이터 전처리 후, CSV 파일로 저장



2 데이터 전처리

1. PLACE : 자전거 대여소 정보

연번	구분	대여소ID	대여소번호	대여소명	거치대수	위도	경도	
0	1	강남구	ST-799	2315	2315. 불은사역 5번출구 옆	10	37.514248	127.061119
1	2	강남구	ST-1174	2365	2365. K+ 타워 앞	15	37.523300	127.038475
2	3	강남구	ST-796	2313	2313. 금원빌딩 앞	10	37.525116	127.052467
3	4	강남구	ST-961	2347	2347. 두산건설 본사	15	37.518639	127.035400
4	5	강남구	ST-1171	2362	2362. 신사동 가로수길 입구	10	37.517635	127.022453
...
1448	1449	중랑구	ST-585	1420	1420. 신내어울공원 앞	15	37.614231	127.096764
1449	1450	중랑구	ST-942	1440	1440. 하나은행 면목지점	10	37.586372	127.080070
1450	1451	중랑구	ST-1346	1454	1454. 한국전력공사(동대문 중랑지사)	10	37.607349	127.078590
1451	1452	중랑구	ST-1452	1459	1459. 용마한신아파트사거리	10	37.580349	127.092651
1452	1453	중랑구	ST-1380	1456	1456. 상아빌딩(우림시장 교차로)	10	37.595112	127.100327

1453 rows × 8 columns



	LOCATION_GU	PLACE_ID	PLACE_NAME	BIKE HOLDER_COUNT	LATITUDE	LONGITUDE
0	강남구	2315	불은사역 5번출구 옆	10	37.514248	127.061119
1	강남구	2365	K+ 타워 앞	15	37.523300	127.038475
2	강남구	2313	금원빌딩 앞	10	37.525116	127.052467
3	강남구	2347	두산건설 본사	15	37.518639	127.035400
4	강남구	2362	신사동 가로수길 입구	10	37.517635	127.022453
...
1448	중랑구	1420	신내어울공원 앞	15	37.614231	127.096764
1449	중랑구	1440	하나은행 면목지점	10	37.586372	127.080070
1450	중랑구	1454	한국전력공사(동대문 중랑지사)	10	37.607349	127.078590
1451	중랑구	1459	용마한신아파트사거리	10	37.580349	127.092651
1452	중랑구	1456	상아빌딩(우림시장 교차로)	10	37.595112	127.100327

1453 rows × 6 columns

place.csv

구, 대여소번호, 대여소명, 거치대수, 위도, 경도



2 데이터 전처리

2. WEATHER : 기상정보

	관측일자	지점명	평균기온(°C)	최저기온(°C)	최고기온(°C)	평균습도(%)	최저습도(%)	최고습도(%)	평균풍속(m/s)	최대풍속(m/s)	강수량(mm)
0	20180101	남산	-0.5	-3.6	3.2	29.200001	22.500000	46.000000	1.3	3.2	0.0
1	20180101	동대문	-0.1	-4.1	3.8	44.500000	27.799999	61.099998	1.5	4.8	0.0
2	20180101	성북	-0.3	-3.5	3.5	35.599998	19.299999	54.900002	0.9	4.2	0.0
3	20180101	도봉	-1.6	-6.3	3.9	50.000000	25.900000	73.800003	0.5	2.8	0.0
4	20180101	마포	-1.0	-4.8	3.8	52.000000	28.000000	73.900002	0.8	3.6	0.0
...
7452	20181231	용산	-3.6	-7.7	0.1	35.700001	14.200000	54.500000	1.1	4.0	0.0
7453	20181231	종구	-5.4	-10.2	-1.1	41.099998	18.299999	57.400002	1.1	4.5	0.0
7454	20181231	은평	-5.3	-9.3	-0.9	40.900002	21.299999	52.599998	1.1	3.9	0.0
7455	20181231	동작	-3.8	-7.2	-0.5	42.400002	20.700001	60.200001	1.3	4.3	0.0
...
7452	20181231	용산	-3.6	-7.7	0.1	35.700001	14.200000	54.500000	1.1	4.0	0.0
7453	20181231	종구	-5.4	-10.2	-1.1	41.099998	18.299999	57.400002	1.1	4.5	0.0
7454	20181231	은평	-5.3	-9.3	-0.9	40.900002	21.299999	52.599998	1.1	3.9	0.0
7455	20181231	동작	-3.8	-7.2	-0.5	42.400002	20.700001	60.200001	1.3	4.3	0.0
7456	20181231	강동	-4.2	-9.0	0.2	26.000000	0.000000	48.900002	1.1	3.9	0.0

7457 rows × 11 columns



	DATE	LOCATION_GU	TEMPERATURE	HUMID	WIND	PRECIPITATION
0	20180101	남산	-0.5	29.200001	1.3	0.0
1	20180101	동대문구	-0.1	44.500000	1.5	0.0
2	20180101	성북구	-0.3	35.599998	0.9	0.0
3	20180101	도봉구	-1.6	50.000000	0.5	0.0
4	20180101	마포구	-1.0	52.000000	0.8	0.0
...
7452	20181231	용산구	-3.6	35.700001	1.1	0.0
7453	20181231	종구	-5.4	41.099998	1.1	0.0
7454	20181231	은평구	-5.3	40.900002	1.1	0.0
7455	20181231	동작구	-3.8	42.400002	1.3	0.0

weather.csv

관측일자, 구, 기온, 습도, 풍속, 강수량



2 DATA 전처리

3. DUST : 미세먼지 정보

측정일자	권역코드	권역명	측정소코드	측정소명	미세먼지(μg/㎡)	초미세먼지(μg/㎡)	오존(ppm)	이산화질소농도(ppm)	일산화탄소농도(ppm)	아황산가스농도(ppm)	
0	20180101	100	도심권	111121	중구	32	18	0.015	0.028	0.5	0.005
1	20180101	104	동남권	111273	송파구	52	23	0.013	0.028	0.7	0.004
2	20180101	104	동남권	111282	서초구	49	24	0.011	0.031	0.6	0.005
3	20180101	104	동남권	111281	강남구	34	22	0.010	0.033	0.6	0.006
4	20180101	103	서남권	111301	양천구	37	21	0.008	0.037	0.7	0.005
...
7620	20181231	102	동북권	111152	동대문구	50	42	0.007	0.042	0.9	0.007
7621	20181231	102	동북권	111151	중랑구	45	31	0.007	0.036	0.7	0.007
7622	20181231	102	동북권	111142	성동구	54	34	0.005	0.043	0.7	0.004
7623	20181231	102	동북권	111141	광진구	41	26	0.004	0.042	1.1	0.004
7624	20181231	101	서북권	111201	마포구	49	19	0.008	0.030	0.7	0.003

7625 rows × 11 columns



	DATE	LOCATION_GU	FINEDUST	ULTRAFINEDUST	OZONE	NO2	CO	SO2
0	20180101	중구	32	18	0.015	0.028	0.5	0.005
1	20180101	송파구	52	23	0.013	0.028	0.7	0.004
2	20180101	서초구	49	24	0.011	0.031	0.6	0.005
3	20180101	강남구	34	22	0.010	0.033	0.6	0.006
4	20180101	양천구	37	21	0.008	0.037	0.7	0.005
...
7620	20181231	동대문구	50	42	0.007	0.042	0.9	0.007
7621	20181231	중랑구	45	31	0.007	0.036	0.7	0.007
7622	20181231	성동구	54	34	0.005	0.043	0.7	0.004
7623	20181231	광진구	41	26	0.004	0.042	1.1	0.004
7624	20181231	마포구	49	19	0.008	0.030	0.7	0.003

7625 rows × 8 columns

dust.csv

날짜, 구, 미세먼지, 초미세먼지



2 DATA 전처리

4. RENTAL : 자전거 대여기록 정보

약 677만 rows 의 Big Data

	대여일자	대여시간	대여소번호	대여소명	대여구분코드	성별	연령대코드	이용권수	운동량	탄소량	이동거리	이동시간	사용시간
0	2018-01-01 00:00:00	0	300	정동사거리	정기권	F	~10대	1	37.32	0.34	1450.0	23.0	NaN
1	2018-01-01 00:00:00	0	505	자양사거리 광진아크로텔 앞	정기권	F	20대	1	54.83	0.49	2130.0	15.0	NaN
2	2018-01-01 00:00:00	0	1531	미아사거리 1번 출구	정기권	F	20대	1	78.34	0.73	3140.0	17.0	NaN
3	2018-01-01 00:00:00	0	516	광진메디칼 앞	정기권	F	20대	1	29.95	0.29	1240.0	5.0	NaN
4	2018-01-01 00:00:00	0	3508	화양사거리	정기권	F	20대	1	46.09	0.45	1940.0	43.0	NaN
...
460031	2018-12-31	23	1623	1623. 노원 구민회관	일일권	M	30대	1	38.00	0.23	1010.0	NaN	6.0
460032	2018-12-31	23	394	394. 경희궁 자이 3단지	일일권	M	40대	1	64.86	0.54	2340.0	NaN	26.0
460033	2018-12-31	23	207	207. 여의나루역 1번출구 앞	단채권	WN	~10대	2	786.85	4.61	19870.0	NaN	103.0
460034	2018-12-31	23	338	338. 세운스퀘어 앞	단채권	F	30대	2	30.60	0.32	1380.0	NaN	14.0
460035	2018-12-31	23	1308	1308. 안암로터리 버스정류장 앞	단채권	M	20대	2	283.09	1.96	8410.0	NaN	46.0

6447020 rows × 13 columns

6447020 rows × 13 columns



	DATE	TIME	PLACE_ID	PLACE_NAME	MEMBER	SEX	AGE	USE_COUNT	EXERCISE	CARBON	DISTANCE	DURING_MINUTES	USE_MINUTES
0	20180101	0	300	정동사거리	정기권	F	~10대	1	37.32	0.34	1450.0	23.0	NaN
1	20180101	0	505	자양사거리 광진아크로텔 앞	정기권	F	20대	1	54.83	0.49	2130.0	15.0	NaN
2	20180101	0	1531	미아사거리 1번 출구	정기권	F	20대	1	78.34	0.73	3140.0	17.0	NaN
3	20180101	0	516	광진메디칼 앞	정기권	F	20대	1	29.95	0.29	1240.0	5.0	NaN
4	20180101	0	3508	화양사거리	정기권	F	20대	1	46.09	0.45	1940.0	43.0	NaN
...
460031	20181231	23	1623	1623. 노원 구민회관	일일권	M	30대	1	38.00	0.23	1010.0	NaN	6.0
460032	20181231	23	394	394. 경희궁 자이 3단지	일일권	M	40대	1	64.86	0.54	2340.0	NaN	26.0
460033	20181231	23	207	207. 여의나루역 1번출구 앞	단채권	W	~10대	2	786.85	4.61	19870.0	NaN	103.0
460034	20181231	23	338	338. 세운스퀘어 앞	단채권	F	30대	2	30.60	0.32	1380.0	NaN	14.0
460035	20181231	23	1308	1308. 안암로터리 버스정류장 앞	단채권	M	20대	2	283.09	1.96	8410.0	NaN	46.0

6447020 rows × 13 columns

1월 ~12월까지의 분리된 자료 합치기

RENTAL.pkl
피클파일로 저장후 db에 저장

대여 날짜, 시간, 대여소, 정기권 여부, 성별, 나이, 이동시간



3 DB 구축

Jupyter notebook에서
PyMysql을 사용하여 Database 구축



3 DB 구축

PLACE TABLE 생성

PLACE TABLE		
PLACE_ID	VARCHAR(50)	PRIMARY KEY
PLACE_NAME	VARCHAR(50)	
LOCATION_GU	VARCHAR(50)	
BIKE HOLDER_COUNT	INT	
LATITUDE	FLOAT	
LONGTITUDE	FLOAT	

CREATE PLACE

```
In [29]: def create_table_PLACE():
        conn = pymysql.connect(host=HOST,
                                user=USER,
                                password=PW,
                                db=DB,
                                charset='utf8mb4')

        sql = '''
                CREATE TABLE PLACE
                (
                    PLACE_ID VARCHAR(50) NOT NULL PRIMARY KEY,
                    PLACE_NAME VARCHAR(50),
                    LOCATION_GU VARCHAR(50),
                    BIKE HOLDER_COUNT VARCHAR(50),
                    LATITUDE FLOAT,
                    LONGTITUDE FLOAT
                );

                '''

        try:
            with conn.cursor() as cursor:
                cursor.execute(sql)
                conn.commit()
        finally:
            conn.close()
```

```
In [30]: create_table_PLACE()
```



3 DB 구축

WEATHER TABLE 생성

WEATHER TABLE		
DATE	VARCHAR(50)	PRIMARY KEY
LOCATION_GU	VARCHAR(50)	
TEMPERATURE	FLOAT	
HUMID	FLOAT	
WIND	FLOAT	
PERCIPITATION	FLOAT	

CREATE_WEATHER

```
In [31]: def create_table_WEATHER():  
  
    conn = pymysql.connect(host=HOST,  
                           user=USER,  
                           password=PW,  
                           db=DB,  
                           charset='utf8mb4')  
  
    sql = '''  
        CREATE TABLE WEATHER  
        (  
            DATE VARCHAR(50) NOT NULL,  
            LOCATION_GU VARCHAR(50) NOT NULL,  
            TEMPERATURE FLOAT,  
            HUMID FLOAT,  
            WIND FLOAT,  
            PERCIPITATION INT,  
            PRIMARY KEY(DATE, LOCATION_GU)  
        );  
  
        ...  
  
    try:  
        with conn.cursor() as cursor:  
            cursor.execute(sql)  
            conn.commit()  
    finally:  
        conn.close()
```

```
In [32]: create_table_WEATHER()
```

3 DB 구축

DUST TABLE 생성

DUST TABLE		
DATE	VARCHAR(50)	PRIMARY KEY
LOCATION_GU	VARCHAR(50)	
FINEDUST	VARCHAR(50)	
ULTRAFINEDUST	INT	
OZONE	FLOAT	
NO2	FLOAT	
CO	FLOAT	
SO2	FLOAT	

CREATE DUST

```
In [33]: def create_table_DUST():  
  
    conn = pymysql.connect(host=HOST,  
                           user=USER,  
                           password=PW,  
                           db=DB,  
                           charset='utf8mb4')  
  
    sql = '''  
        CREATE TABLE DUST  
        (  
            DATE VARCHAR(50) NOT NULL,  
            LOCATION_GU VARCHAR(50) NOT NULL,  
            FINEDUST INT,  
            ULTRAFINEDUST INT,  
            OZONE FLOAT,  
            NO2 FLOAT,  
            CO FLOAT,  
            SO2 FLOAT,  
            PRIMARY KEY(DATE, LOCATION_GU)  
        );  
    '''  
  
    try:  
        with conn.cursor() as cursor:  
            cursor.execute(sql)  
            conn.commit()  
    finally:  
        conn.close()
```

```
In [34]: create_table_DUST()
```

3 DB 구축

RENTAL TABLE 생성

RENTAL TABLE		
DATE	VARCHAR(50)	
TIME	VARCHAR(50)	
PLACE_ID	VARCHAR(50)	FOREINGN KEY
MEMBER	FLOAT	
SEX	FLOAT	
AGE	VARCHAR(50)	
USE_COUNT	INT	
EXCERCISE	FLOAT	
DISTANCE	FLOAT	
DURING_MINUTES	FLOAT	
USE_MINUTES	FLOAT	

CREATE RENTAL

```
In [105]: def create_table_RENTAL():  
  
    conn = pymysql.connect(host=HOST,  
        user=USER,  
        password=PW,  
        db=DB,  
        charset='utf8mb4')  
  
    sql = '''  
        CREATE TABLE RENTAL  
        (  
            DATE VARCHAR(50),  
            TIME VARCHAR(50),  
            PLACE_ID VARCHAR(50),  
            PLACE_NAME VARCHAR(50),  
            MEMBER FLOAT,  
            SEX FLOAT,  
            AGE VARCHAR(50),  
            USE_COUNT INT,  
            EXERCISE FLOAT,  
            DISTANCE FLOAT,  
            DURING_MINUTES FLOAT,  
            FOREIGN KEY (PLACE_ID) REFERENCES PLACE(PLACE_ID),  
            FOREIGN KEY (PLACE_NAME) REFERENCES PLACE(PLACE_NAME)  
        );  
    '''  
  
    try:  
        with conn.cursor() as cursor:  
            cursor.execute(sql)  
            conn.commit()  
    finally:  
        conn.close()  
  
In [106]: create_table_RENTAL()
```

단, 사용자를 구분하는 정보가 사생활 보호 차원에서 비공개이기 때문에, RENTAL TABLE에 프라이머리를 지정하기 어려움

3 DB 구축

CSV 파일을 해당 테이블에 INSERT

PLACE INSERT

```
import sys, io
import csv
import pymysql

conn = pymysql.connect(host=HOST, user=USER, password=PW,
                        db=DB, charset='utf8')
curs = conn.cursor()
conn.commit()

f = open('PLACE.csv', 'r', encoding='UTF8')
csvReader = csv.reader(f)
type(csvReader)

_csvv_reader

for row in csvReader:
    PLACE_ID = (row[2])
    LOCATION_BU = (row[1])
    PLACE_NAME = (row[3])
    BIKE HOLDER COUNT = (row[4])
    LATITUDE = (row[5])
    LONGITUDE = (row[6])
    sql = """insert into place (PLACE_ID, LOCATION_BU, PLACE_NAME, BIKE HOLDER COUNT, LATITUDE, LONGITUDE) values (%s, %s, %s, %s, %s, %s)"""
    curs.execute(sql, (PLACE_ID, LOCATION_BU, PLACE_NAME, BIKE HOLDER COUNT, LATITUDE, LONGITUDE))

conn.commit()
f.close()
conn.close()
```

DUST INSERT

```
conn = pymysql.connect(host=HOST, user=USER, password=PW,
                        db=DB, charset='utf8')
curs = conn.cursor()
conn.commit()

f = open('DUST.csv', 'r', encoding='UTF8')
csvReader = csv.reader(f)
type(csvReader)

for row in csvReader:
    DATE = (row[1])
    LOCATION_BU = (row[2])
    FINEDUST = (row[3])
    ULTRAFINEDUST = (row[4])
    OZONE = (row[5])
    NO2 = (row[6])
    CO = (row[7])
    SO2 = (row[8])
    sql = """insert into DUST (DATE, LOCATION_BU, FINEDUST, ULTRAFINEDUST, OZONE, NO2, CO, SO2) values (%s, %s, %s, %s, %s, %s, %s, %s)"""
    curs.execute(sql, (DATE, LOCATION_BU, FINEDUST, ULTRAFINEDUST, OZONE, NO2, CO, SO2))

conn.commit()
f.close()
conn.close()
```

WEATHER INSERT

```
conn = pymysql.connect(host=HOST, user=USER, password=PW,
                        db=DB, charset='utf8')
curs = conn.cursor()
conn.commit()

f = open('WEATHER.csv', 'r', encoding='UTF8')
csvReader = csv.reader(f)
type(csvReader)

for row in csvReader:
    DATE = (row[1])
    LOCATION_BU = (row[2])
    TEMPERATURE = (row[3])
    HUMID = (row[4])
    WIND = (row[5])
    PERCIPITATION = (row[6])
    sql = """insert into WEATHER (DATE, LOCATION_BU, TEMPERATURE, HUMID, WIND, PERCIPITATION) values (%s, %s, %s, %s, %s, %s)"""
    curs.execute(sql, (DATE, LOCATION_BU, TEMPERATURE, HUMID, WIND, PERCIPITATION))

conn.commit()
f.close()
conn.close()
```

RENTAL INSERT

```
conn = pymysql.connect(host=HOST, user=USER, password=PW,
                        db=DB, charset='utf8')
curs = conn.cursor()
conn.commit()

for row in range(5447000):
    DATE = (rental.iloc[row]['DATE'])
    TIME = (rental.iloc[row]['TIME'])
    PLACE_ID = (rental.iloc[row]['PLACE_ID'])
    MEMBER = (rental.iloc[row]['MEMBER'])
    SEX = (rental.iloc[row]['SEX'])
    AGE = (rental.iloc[row]['AGE'])
    USE_COUNT = (rental.iloc[row]['USE_COUNT'])
    EXERCISE = (rental.iloc[row]['EXERCISE'])
    DISTANCE = (rental.iloc[row]['DISTANCE'])
    DURING_MINUTES = (rental.iloc[row]['DURING_MINUTES'])
    USE_MINUTES = (rental.iloc[row]['USE_MINUTES'])
    sql = """insert into DUST (DATE, LOCATION_BU, FINEDUST, ULTRAFINEDUST, OZONE, NO2, CO, SO2) values (%s, %s, %s, %s, %s, %s, %s, %s)"""
    curs.execute(sql, (DATE, LOCATION_BU, FINEDUST, ULTRAFINEDUST, OZONE, NO2, CO, SO2))

conn.commit()
f.close()
conn.close()

# 5/22
from sqlalchemy import create_engine
pymysql.install_as_MySQLdb()
import MySQLdb

engine = create_engine("mysql+mysql://user01:*@localhost/mydb", encoding='utf8')
conn = engine.connect()
rental.to_sql(name='rental', con=engine, if_exists='append')
```

3 DB 구축

최종 data 형태

select * from place

	LOCATION_GU	PLACE_ID	PLACE_NAME	BIKE HOLDER COUNT	LATITUDE	LONGITUDE
0	강남구	2315	봉은사역 번출구 옆	10	37.514248	127.061119
1	강남구	2365	K+ 타워 앞	15	37.523300	127.038475
2	강남구	2313	금원빌딩 앞	10	37.525116	127.052467
3	강남구	2347	두산건설 본사	15	37.518639	127.035400
4	강남구	2362	신사동 가로수길 입구	10	37.517635	127.022453
...
1448	중랑구	1420	신내어울공원 앞	15	37.614231	127.096764
1449	중랑구	1440	하나은행 면목지점	10	37.586372	127.080070
1450	중랑구	1454	한국전력공사(동대문 중랑지사)	10	37.607349	127.078590
1451	중랑구	1459	올마한신아파트사거리	10	37.580349	127.092651
1452	중랑구	1456	상아빌딩(우림시장 교차로)	10	37.595112	127.100327

1453 rows × 6 columns

select * from dust

	DATE	LOCATION_GU	FINE DUST	ULTRAFINE DUST	OZONE	NO2	CO	SO2
0	20180101	중구	32	18	0.015	0.028	0.5	0.005
1	20180101	송파구	52	23	0.013	0.028	0.7	0.004
2	20180101	서초구	49	24	0.011	0.031	0.6	0.005
3	20180101	강남구	34	22	0.010	0.033	0.6	0.006
4	20180101	양천구	37	21	0.008	0.037	0.7	0.005
...
7620	20181231	동대문구	50	42	0.007	0.042	0.9	0.007
7621	20181231	중랑구	45	31	0.007	0.036	0.7	0.007
7622	20181231	성동구	54	34	0.005	0.043	0.7	0.004
7623	20181231	광진구	41	26	0.004	0.042	1.1	0.004
7624	20181231	마포구	49	19	0.008	0.030	0.7	0.003

7625 rows × 8 columns

select * from weather

	DATE	LOCATION_GU	TEMPERATURE	HUMID	WIND	PRECIPITATION
0	20180101	남산	-0.5	29.200001	1.3	0.0
1	20180101	동대문구	-0.1	44.500000	1.5	0.0
2	20180101	성북구	-0.3	35.599998	0.9	0.0
3	20180101	도봉구	-1.6	50.000000	0.5	0.0
4	20180101	마포구	-1.0	52.000000	0.8	0.0
...
7452	20181231	용산구	-3.6	35.700001	1.1	0.0
7453	20181231	중구	-5.4	41.099998	1.1	0.0
7454	20181231	은평구	-5.3	40.900002	1.1	0.0
7455	20181231	동작구	-3.8	42.400002	1.3	0.0

select * from rental

	DATE	TIME	PLACE_ID	PLACE_NAME	MEMBER	SEX	AGE	USE_COUNT	EXERCISE	CARBON	DISTANCE	DURING_MINUTES	USE_MINUTES
0	20180101	0	300	정동사거리	정기권	F	~10대	1	37.32	0.34	1450.0	23.0	NaN
1	20180101	0	505	자양사거리 광진아파트 앞	정기권	F	20대	1	54.83	0.49	2130.0	15.0	NaN
2	20180101	0	1531	미아사거리 1번 출구	정기권	F	20대	1	78.34	0.73	3140.0	17.0	NaN
3	20180101	0	516	광진메디칼 앞	정기권	F	20대	1	29.95	0.29	1240.0	5.0	NaN
4	20180101	0	3508	화양사거리	정기권	F	20대	1	46.09	0.45	1940.0	43.0	NaN
...
460031	20181231	23	1623	1623. 노원구 민회관	월말권	M	30대	1	38.00	0.23	1010.0	NaN	6.0
460032	20181231	23	394	394. 경희궁 자이 3단지	월말권	M	40대	1	64.86	0.54	2340.0	NaN	26.0
460033	20181231	23	207	207. 여의나루역 1번출구 앞	단체권	W	~10대	2	786.85	4.61	19870.0	NaN	103.0
460034	20181231	23	338	338. 세운스퀘어 앞	단체권	F	30대	2	30.60	0.32	1380.0	NaN	14.0
460035	20181231	23	1308	1308. 안암로터리 버스정류장 앞	단체권	M	20대	2	283.09	1.96	8410.0	NaN	46.0

6447020 rows × 13 columns

3 DB 구축

최종 data 형태

2. 날씨 (WEATHER)

DATE	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
TEMPERATURE	FLOAT
HUMID	FLOAT
WIND	FLOAT
PERCIPITATION	FLOAT

3. 미세먼지 (DUST)

DATE	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
FINEDUST	VARCHAR(50)
ULTRAFINEDUST	INT
OZONE	FLOAT
NO2	FLOAT
CO	FLOAT
SO2	FLOAT

1. 자전거 대여소(PLACE)

PLACE_ID	VARCHAR(50)
PLACE_NAME	VARCHAR(50)
LOCATION_GU	VARCHAR(50)
BIKE HOLDER_COUNT	INT
LATITUDE	FLOAT
LONGITUDE	FLOAT

4. 자전거 대여기록(RENTAL)

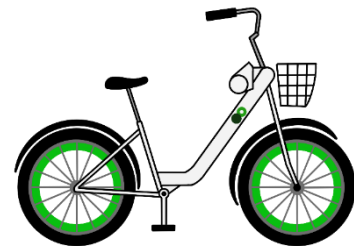
DATE	VARCHAR(50)
TIME	VARCHAR(50)
PLACE_ID	VARCHAR(50)
MEMBER	FLOAT
SEX	FLOAT
AGE	VARCHAR(50)
USE_COUNT	INT
EXCERCISE	FLOAT
DISTANCE	FLOAT
DURING_MINUTES	FLOAT
USE_MINUTES	FLOAT

4

Data Analysis

SELECT 등을 이용한 Data Analysis - 코드 참조

1. 이화여대 따릉이 대여소 이용자 분석
2. 지역별 자전거 대여소 거치대 수 분석
3. 따릉이 이용량 시계열 분석
4. 미세먼지와 연관성 분석
5. 날씨와의 연관성 분석



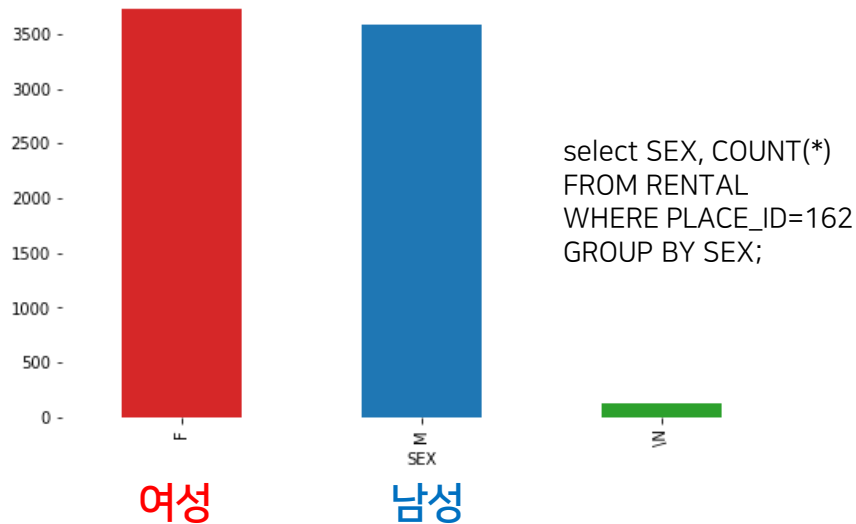
4 DATA 분석



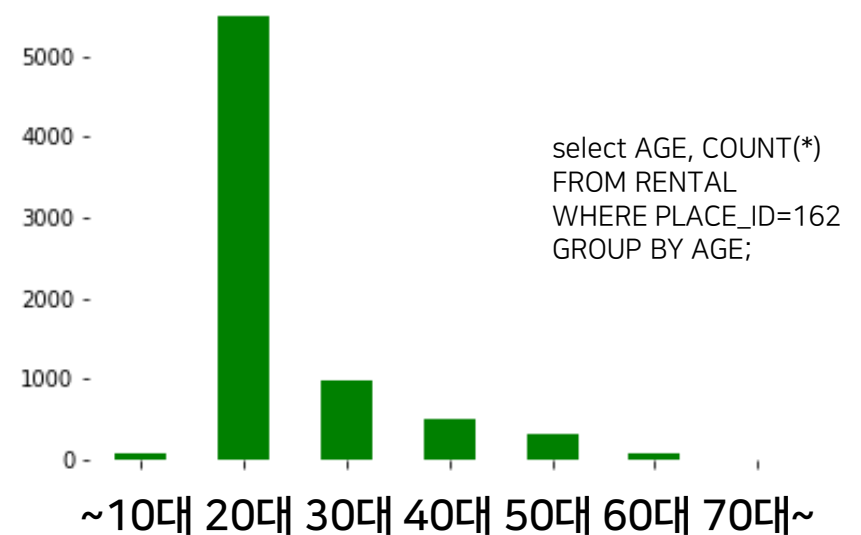
이화여대 따릉이 대여소 사용자 분석



<봉원고가차도 밑 따릉이 대여소 이용자 분석> 봉원고가차도 밑 따릉이대여소
PLACE_ID = 162



여성 사용자가 더 많음



20대 이용자가 압도적으로 많음

4 DATA 분석

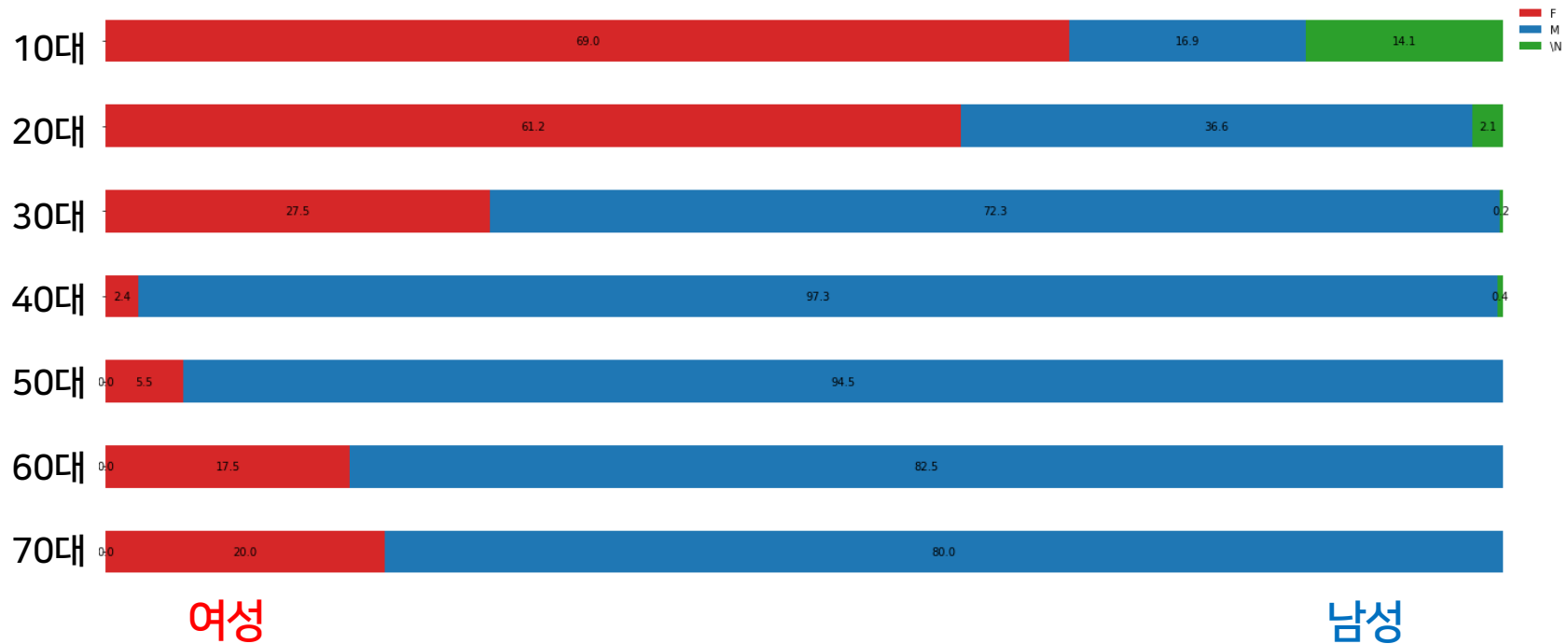


이화여대 따릉이 대여소 사용자 분석

10,20대는 여성 사용자가 많고, 30대 이상은 남성 사용자가 많음

<봉원고가차도 및 따릉이 대여소 이용자 분석>

```
SELECT AGE, SEX, COUNT(*)  
FROM RENTAL  
WHERE PLACE_ID=162  
GROUP BY AGE, SEX;
```

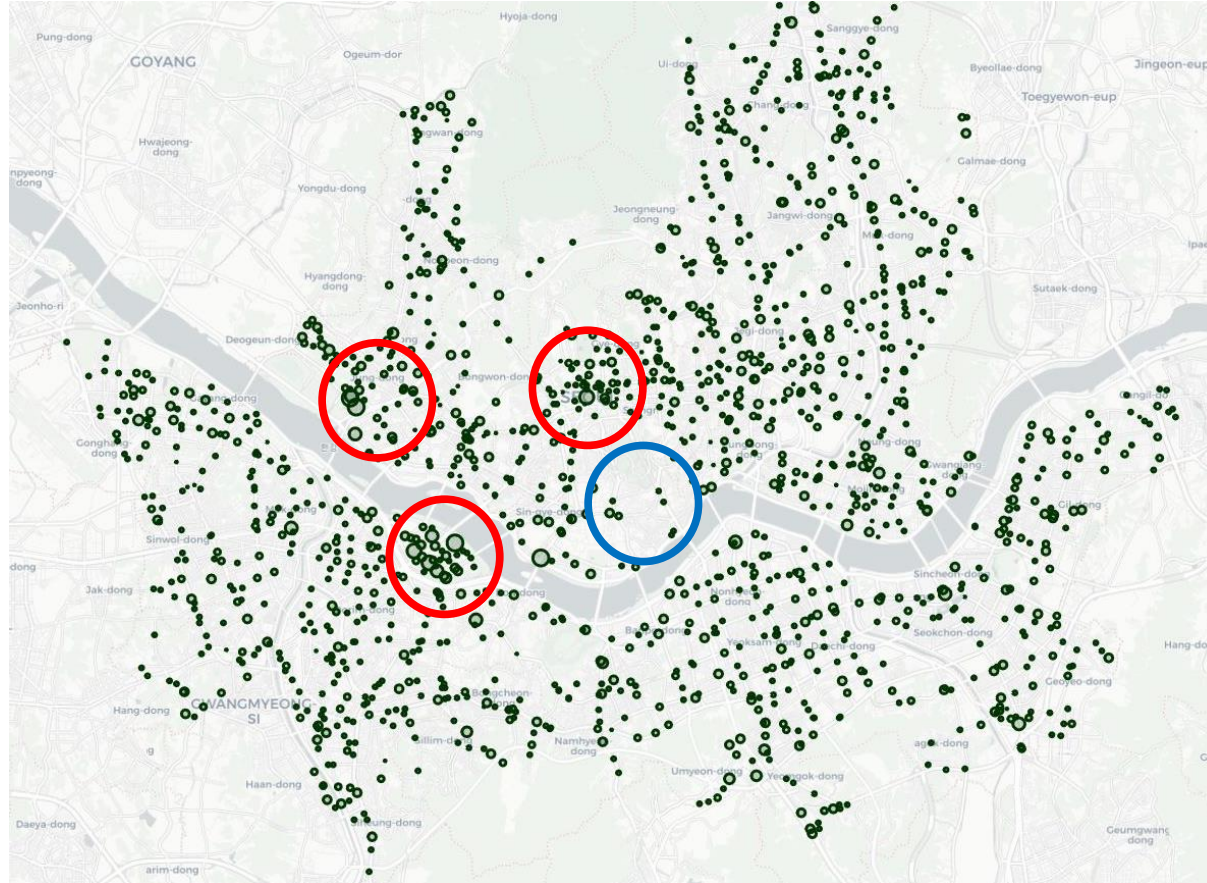


4 DATA 분석



자전거 대여소 거치대수 분석

```
SELECT LATITUDE, LONGITUDE, BIKE HOLDER_COUNT FROM PLACE;
```



한강공원 근처, 종로구 회사 밀집지역 근처
자전거 거치대수 많음

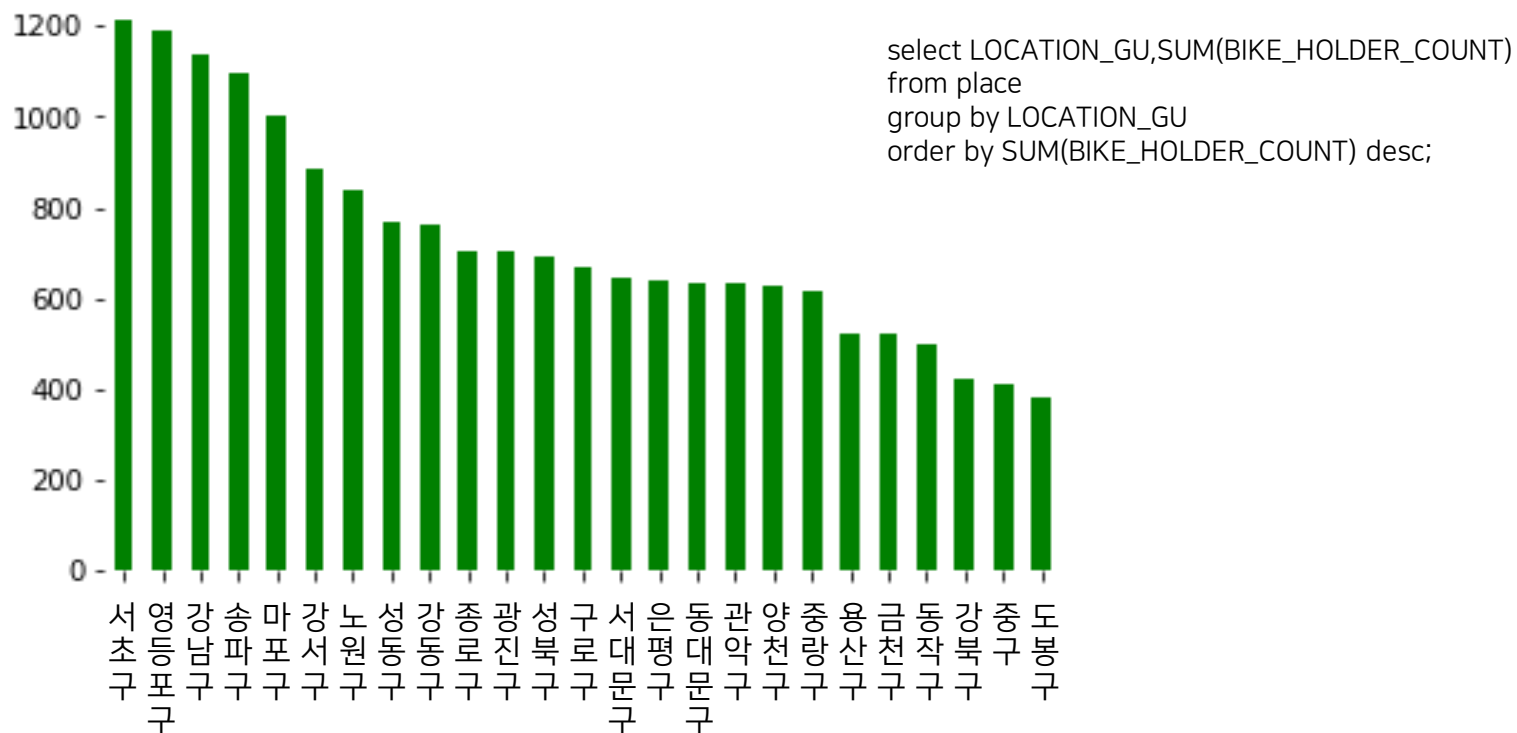
오르막, 내리막길이 많은 이태원 근처
자전거 거치대수 적음

4 DATA 분석



행정구역 별 자전거 대여소 거치대수 분석

행정구역 별 자전거 거치대수



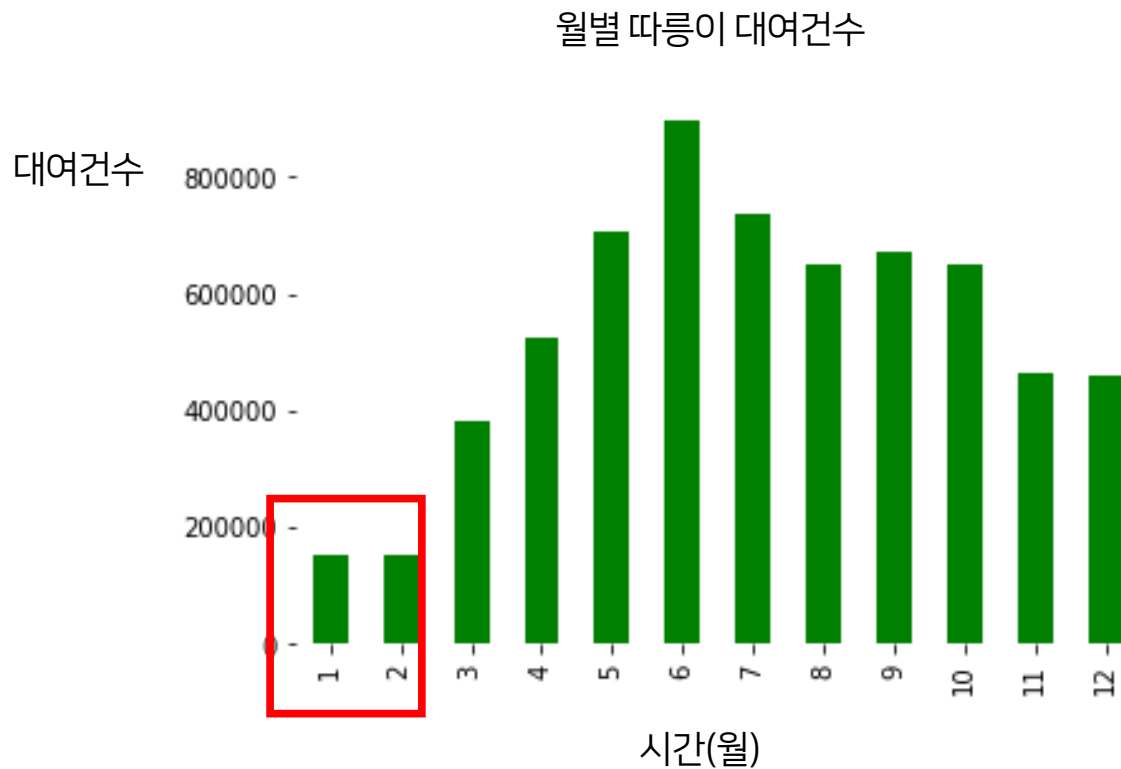
많은 지역 : 서초, 영등포, 강남, 송파, 마포

적은 지역 : 중구, 도봉구

4 DATA 분석



월별 따릉이 자전거 대여건수 분석



```
SELECT MONTH, COUNT(*)  
FROM RENTAL  
group by MONTH;
```

추운 겨울에는 따릉이 자전거 대여 건수가 현저히 낮음

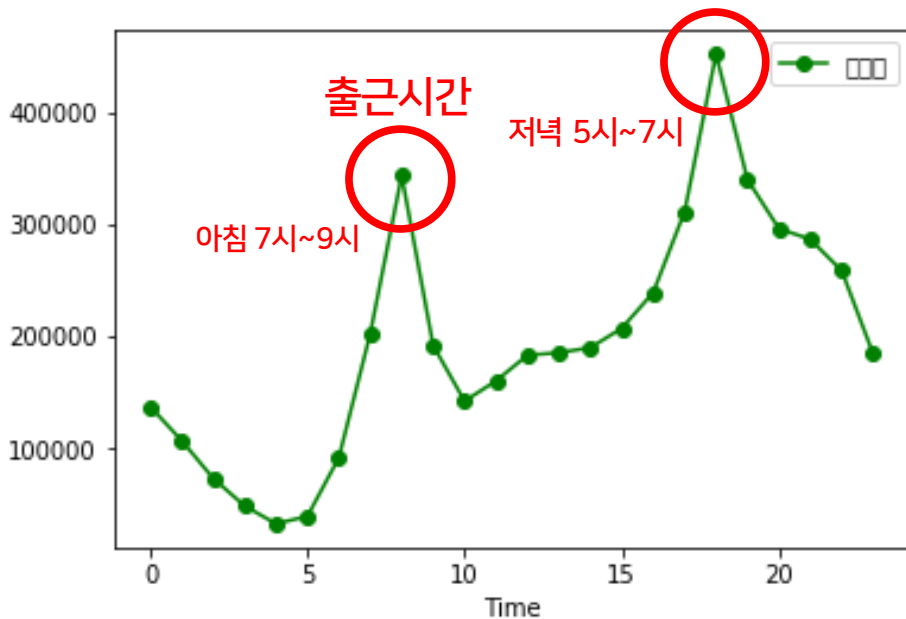
4 DATA 분석



시간에 따른 따릉이 이용량 분석

평일

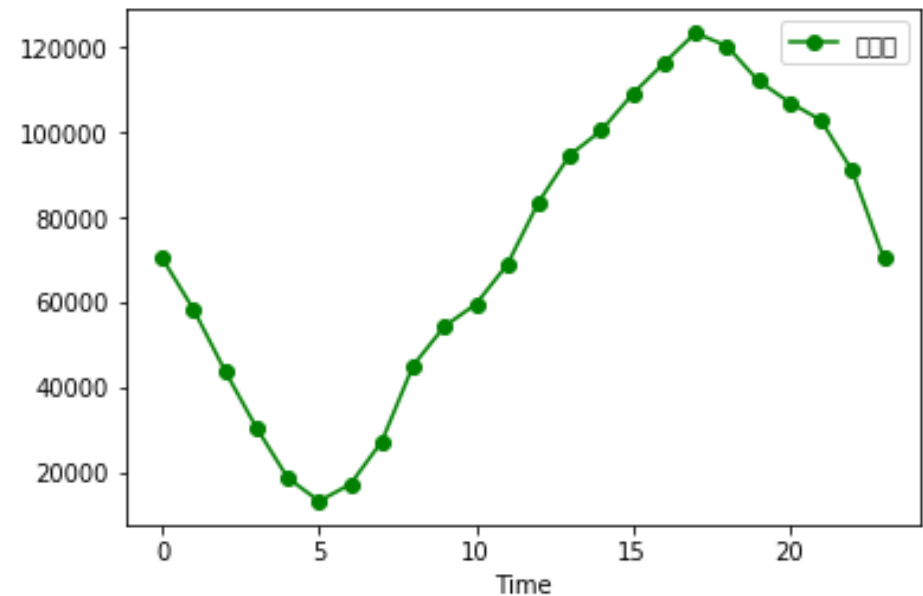
```
SELECT TIME, COUNT(*)  
FROM RENTAL  
WHERE WEEK=1  
group by TIME;
```



출퇴근 시간 이용량 도드라짐

주말

```
SELECT TIME, COUNT(*)  
FROM RENTAL  
WHERE WEEK=2  
group by TIME;
```



낮~저녁시간에 자전거를 이용하는 사람이 많음

4 DATA 분석

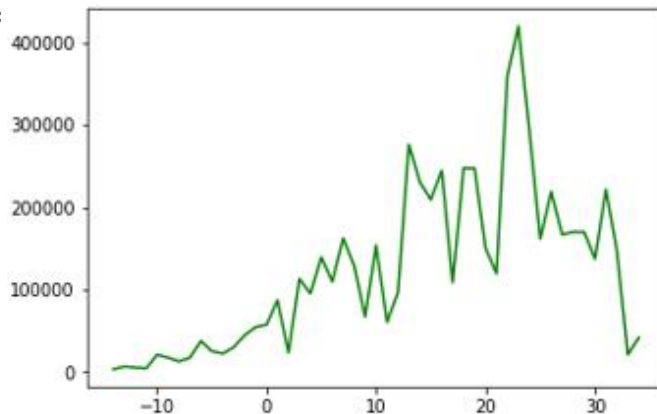


날씨와의 연관성

```
SELECT WEATHER.TEMPERATURE, COUNT(*)  
FROM WEATHER, RENTAL  
GROUP BY WEATHER.TEMPERATURE ;
```

< 기온에 따른 따릉이 이용량 >

따릉이 이용량



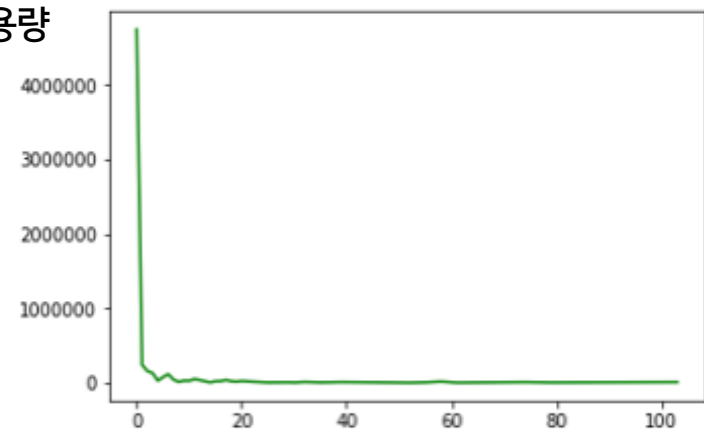
기온(온도 섭씨)

너무 춥거나, 너무 더운 경우
따릉이 이용량이 적음

```
SELECT WEATHER.PERCIPITANT, COUNT(*)  
FROM WEATHER, RENTAL  
GROUP BY WEATHER.PERCIPITANT;
```

< 강수량 에 따른 따릉이 이용량 >

따릉이 이용량



강수량

비오는 날의 경우
따릉이 이용량이 적음

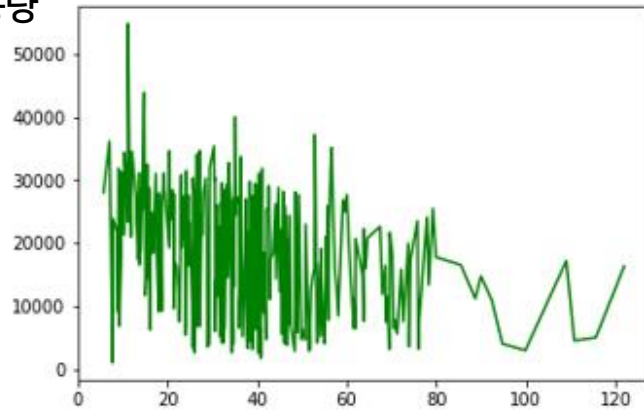
4 DATA 분석



미세먼지와의 연관성

```
SELECT DUST.FINEDUST, COUNT(*)  
FROM DUST, RENTAL  
GROUP BY DUST.FINEDUST;
```

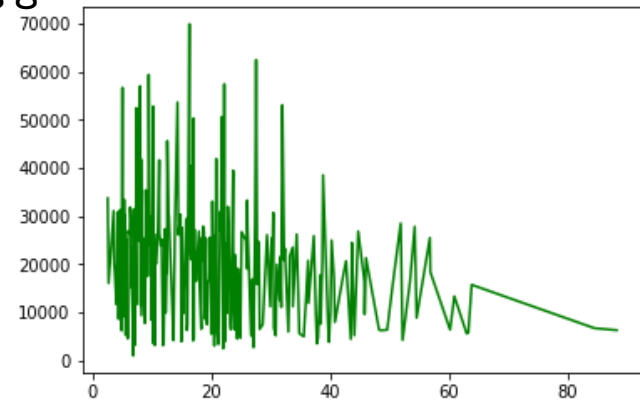
따릉이 이용량



미세먼지

```
SELECT DUST.ULTRAFINEDUST, COUNT(*)  
FROM DUST, RENTAL  
GROUP BY DUST.ULTRAFINEDUST;
```

따릉이 이용량



초미세먼지

미세먼지와 따릉이 이용량과는 직접적인 연관성은 거의 없는 것으로 보임

결론



이화여대 주변 따릉이 대여소 이용자 분석

- 이화여대 주변 따릉이 대여소 이용자는 여성, 20대가 많다.
- 그 중에서도 10-20대는 여성이 많았고, 30대이상은 남성사용자가 많음



지역별 자전거 거치대수 분석

- 한강 주변에 따릉이 거치대수가 많음
- 행정구역 별로 비교해본 결과 서초,영등포, 강남, 마포구가 가장 많았다.



따릉이 이용량 시계열 분석

- 추운 겨울에는 따릉이 이용자수가 적음
- 평일에는 출퇴근 시간에 이용량이 많았음
- 주말에는 낮~저녁 시간에 이용량 많음



날씨와의 연관성

- 비오는 날에 따릉이 이용자수가 거의 없었으며
- 기온이 너무 덥거나 추운 날의 경우 따릉이 이용자수가 적었음



미세먼지와의 연관성

- 미세먼지, 초미세먼지 농도에 상관없이 사람들이 따릉이를 이용하는 것으로 나타남

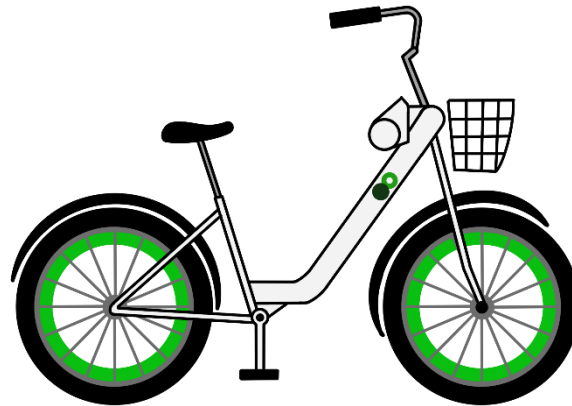
데이터 출처

서울 열린데이터광장

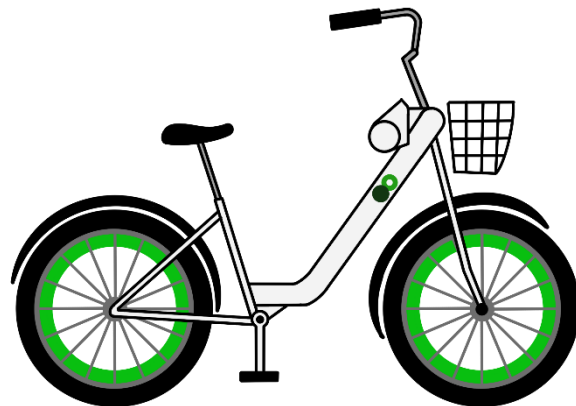
<https://data.seoul.go.kr/dataList/OA-15182/F/1/datasetView.do#>

<https://data.seoul.go.kr/dataList/OA-13252/F/1/datasetView.do>

<https://data.seoul.go.kr/dataList/OA-2220/S/1/datasetView.do#>



감사합니다



* (참고용) 그래프에 사용된 결과창들 : 자세히 - 주피터 노트북 코드 참조)

```
select LATITUDE, LONGTITUDE, BIKE HOLDER_COUNT
FROM PLACE;
```

```
mysql> select LATITUDE, LONGTITUDE, BIKE HOLDER_COUNT FROM PLACE;
```

LATITUDE	LONGTITUDE	BIKE HOLDER_COUNT
37.541805	127.12471799999999	15
37.545219	127.12591599999999	10
37.543915000000005	127.12545800000001	20
37.553349	127.12886	10
37.55492	127.142799	15
37.559063	127.130646	16
37.549549	127.127151	10
37.538658	127.124214	20
37.549561	126.905754	5
37.534481	127.13834399999999	20
37.534771	127.13523899999998	20
37.536026	127.122726	15
37.530772999999996	127.120926	15
37.5508	127.142624	15
37.550079	127.14691200000001	13
37.559399	127.171158	10
37.535472999999996	127.145172	20
37.524681	127.135406	20
37.526825	127.130028	10
37.555649	126.910629	20
37.528290000000005	127.126801	15
37.564301	127.17395	15
37.533764	127.13803100000001	20
37.549999	127.17468999999998	15

```
select LOCATION_GU,SUM(BIKE HOLDER_COUNT)
from place
group by LOCATION_GU
order by SUM(BIKE HOLDER_COUNT) desc;
```

0	서초구	1214.0
1	영등포구	1191.0
2	강남구	1135.0
3	송파구	1095.0
4	마포구	1000.0
5	강서구	884.0
6	노원구	840.0
7	성동구	771.0
8	강동구	763.0
9	종로구	706.0
10	광진구	705.0
11	성북구	692.0
12	구로구	671.0
13	서대문구	643.0
14	은평구	637.0
15	동대문구	633.0
16	관악구	632.0
17	양천구	628.0
18	중랑구	616.0
19	울산구	523.0
20	금천구	520.0
21	동작구	501.0
22	강북구	421.0
23	중구	413.0
24	도봉구	383.0

```
mysql> select LOCATION_GU,SUM(BIKE HOLDER_COUNT)
-> from place
-> group by LOCATION_GU
-> order by SUM(BIKE HOLDER_COUNT)DESC;
```

LOCATION_GU	SUM(BIKE HOLDER_COUNT)
서초구	1214
영등포구	1191
강남구	1135
송파구	1095
마포구	1000
강서구	884
노원구	840
성동구	771
강동구	763
종로구	706
광진구	705
성북구	692
구로구	671
서대문구	643
은평구	637
동대문구	633
관악구	632
양천구	628
중랑구	616
울산구	523
금천구	520
동작구	501
강북구	421
중구	413
도봉구	383

... 이하 생략...

* (참고용) 그래프에 사용된 결과창들 : 자세히 - 주피터 노트북 코드 참조)

```
select SEX, COUNT(*)
FROM RENTAL
WHERE PLACE_ID=162
GROUP BY SEX;
```

SEX	
F	3728
M	3591
NaN	131

```
select AGE, COUNT(*)
FROM RENTAL
WHERE PLACE_ID=162
GROUP BY AGE;
```

AGE	
~10대	71
20대	5498
30대	974
40대	510
50대	307
60대	80
70대~	10

```
SELECT AGE, SEX, COUNT(*)
FROM RENTAL
WHERE PLACE_ID=162
GROUP BY AGE, SEX;
```

AGE	SEX	F	M	NaN
70대~		0.200000	0.800000	NaN
60대		0.175000	0.825000	NaN
50대		0.055375	0.944625	NaN
40대		0.023529	0.972549	0.003922
30대		0.275154	0.722793	0.002053
20대		0.612223	0.366497	0.021280
~10대		0.690141	0.169014	0.140845

```
SELECT WEATHER.TEMPERATURE, COUNT(*)
FROM WEATHER, RENTAL
GROUP BY WEATHER.TEMPERATURE ;
```

TEMPERATURE	count
0	2931
1	3154
2	3113
3	3981
4	4017
...	...
317	24890
318	18563
319	20794
320	20693
321	20712

```
SELECT WEATHER.PERCIPITANT, COUNT(*)
FROM WEATHER, RENTAL
GROUP BY WEATHER.PERCIPITANT;
```

PRECIPITATION	count
0	4498629
1	10495
2	45162
3	15856
4	5377
...	...
84	1721
85	3231
86	5200
87	3521
88	9092

* (참고용) 그래프에 사용된 결과창들 : 자세히 - 주피터 노트북 코드 참조)

SELECT MONTH, COUNT(*)
FROM RENTAL
group by MONTH;

MONTH	
1	150006
2	151833
3	380594
4	524227
5	707088
6	896887
7	734460
8	651887
9	673612
10	650675
11	465715
12	460036

SELECT TIME, COUNT(*)
FROM RENTAL
WHERE WEEK=1
group by TIME;

0	137386
1	107067
2	73491
3	48827
4	32833
5	39528
6	91764
7	202620
8	343690
9	192026
10	142300
11	160358
12	183250
13	185499
14	189979
15	207285
16	239105
17	311402
18	452008
19	340042
20	296291
21	286697
22	258634
23	184348

SELECT TIME, COUNT(*)
FROM RENTAL
WHERE WEEK=2
group by TIME;

TIME	
0	70485
1	58400
2	44006
3	30663
4	18818
5	13334
6	17311
7	27366
8	45219
9	54443
10	59665
11	68935
12	83680
13	94759
14	100485
15	109091
16	116360
17	123436
18	120384
19	112201
20	107110
21	102923
22	90997
23	70519

SELECT DUST.FINEDUST, COUNT(*)
FROM DUST, RENTAL
GROUP BY DUST.FINEDUST;

TEMPERATURE		count
0	-13.98	2931
1	-13.47	3154
2	-13.31	3113
3	-10.53	3981
4	-10.22	4017
...
317	31.93	24890
318	32.08	18563
319	33.30	20794
320	33.79	20693
321	33.92	20712

SELECT DUST.ULTRAFINEDUST, COUNT(*)
FROM DUST, RENTAL
GROUP BY DUST.ULTRAFINEDUST;

PRECIPITATION		count
0	0.00	4498629
1	0.02	10495
2	0.05	45162
3	0.07	15856
4	0.08	5377
...
84	60.86	1721
85	73.69	3231
86	74.48	5200
87	78.91	3521
88	102.73	9092